Artificial Intelligence and Applications

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 9th International Conference on Artificial Intelligence and Applications (AIAPP 2022), May 28~29, 2022, Vancouver, Canada
- 3rd International Conference on Natural Language Processing and Machine Learning (NLPML 2022)
- 8th International Conference on Data Mining and Applications (DMA 2022)
- 8th International Conference on Cryptography and Information Security (CRIS 2022)
- 8th International Conference on Software Engineering (SEC 2022)
- 9th International Conference on Computer Science and Information Technology (CoSIT 2022)
- 9th International Conference on Signal and Image Processing (SIGL 2022)
- 9th International Conference on Cybernetics & Informatics (CYBI 2022)

Published By

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds),
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-68-8
DOI: 10.5121/csit.2022.120901 - 10.5121/csit.2022.120923

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

9th International Conference on Artificial Intelligence and Applications (AIAPP 2022), May 28~29, 2022, Vancouver, Canada, 3rd International Conference on Natural Language Processing and Machine Learning (NLPML 2022), 8th International Conference on Data Mining and Applications (DMA 2022), 8th International Conference on Cryptography and Information Security (CRIS 2022), 8th International Conference on Software Engineering (SEC 2022), 9th International Conference on Computer Science and Information Technology (CoSIT 2022), 9th International Conference on Signal and Image Processing (SIGL 2022), 9th International Conference on Cybernetics & Informatics (CYBI 2022) was collocated with 9th International Conference on Artificial Intelligence and Applications (AIAPP 2022). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The AIAPP 2022, NLPML 2022, DMA 2022, CRIS 2022, SEC 2022, CoSIT 2022, SIGL 2022 and CYBI 2022. Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, AIAPP 2022, NLPML 2022, DMA 2022, CRIS 2022, SEC 2022, CoSIT 2022, SIGL 2022 and CYBI 2022 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the AIAPP 2022, NLPML 2022, DMA 2022, CRIS 2022, SEC 2022, CoSIT 2022, SIGL 2022 and CYBI 2022.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds) Wireilla Net Solutions, Australia

Program Committee Members

A K Daniel, M M M University of Technology, India
Abdalhossein Rezai, University of Science and Culture, Iran
Abdelhadi Assir, Hassan 1st University, Morocco
Abdelhak Merizig, Mohamed Khider University of Biskra, Algeria
Abderrahmane EZ-Zahout, Mohammed V University, Morocco
Abdullah, Chandigarh University, India
Ablah AlAmri, King Abdulaziz University, Saudi Arabia
Addisson Salazar, Universitat Politècnica de València, Spain
Afaq Ahmad, Sultan Qaboos University, Oman
Ajay Anil Gurjar, Sipna College of Engineering and Technology, India
Akhil Gupta, Lovely Professional University, India
Alexander Gelbukh, Instituto Politécnico Nacional, Mexico
Ali Asghar Rahmani Hosseinabadi, Islamic Azad University, Iran
Ali Okatan, Istanbul Aydin University, Turkey
Alireza Valipour Baboli, University Technical and Vocational, Iran
Alper Ugur, Pamukkale University, Turkey
Altynbek Sharipbay, Gumilyov Eurasian National University, Kazakhstan
Amal Azeroual, Center of Guidance and Planning, Morocco
Amel Ourici, Badji Mokhtar University of Annaba, Algeria
Anirban Banik, National Institute of Technology Agartala, India
Ankur Singh Bist, Signy Advanced Technology, India
Arnaud Soulet, University of Tours, France
Ashraf Elnagar, College of Computing and Informatics, UAE
Assem Abdel Hamied Moussa, chief Eng, Egypt
Assia Djenouhat, University of Algiers, Algeria
Atanu Nag, IFTM University, India
Atika Archid, Applied mathematics, Morocco
Ayad Ghany Ismaeel, Rector of Al-Kitab University, Iraq
Azeddine Wahbi, Hassan II University, Morocco
Belkacem Bekhiti, University of Boumerdes, Algeria
Ben Lahmar El Habib, Hassan 2 University, Morocco
Beshair Alsiddiq, Prince Sultan University, Saudi Arabia
Bo Li, Harbin Institute of Technology, Weihai, China
Boukari Nassim, Skikda Univrsity, Algeria
Brahami Menaouer, National Polytechnic School of Oran, Algeria
Brahim Lejdel, University of El-Oued, Algeria
Charles Abiodun Robert, Campbellsville University, USA
Christian Mancas, DATASIS ProSoft srl, Romania
Christos Politis, Kingston University, United Kingdom
Claude Frasson, University of Montreal, Canada
Dadmehr Rahbari, Tallinn University of Technology, Estonia
Daniel Gomes de Oliveira, Estácio University, Brazil
Daniel Rosa Canêdo, Federal Institute of Goiás(FIG), Brazil
Daniela Lopez De Luise, CI2S Labs, Argentina

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

9th International Conference on Artificial Intelligence and Applications (AIAPP 2022)

3rd International Conference on Natural Language Processing and Machine Learning (NLPML 2022)

8th International Conference on Data Mining and Applications (DMA 2022)

8th International Conference on Cryptography and Information Security (CRIS 2022)

8th International Conference on Software Engineering (SEC 2022)

9th International Conference on Computer Science and Information Technology (CoSIT 2022)

9th International Conference on Signal and Image Processing (SIGL 2022)

9th International Conference on Cybernetics & Informatics (CYBI 2022)

Deep learning pipeline for image classification on mobile phones

Muhammad Muneeb, Samuel F. Feng, and Andreas Henschel

Department of Mathematics and Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology,
Abu Dhabi, UAE

Abstract. This article proposes and documents a machine-learning framework and tutorial for classifying images using mobile phones. Compared to computers, the performance of deep learning model performance degrades when deployed on a mobile phone and requires a systematic approach to find a model that performs optimally on both computers and mobile phones. By following the proposed pipeline, which consists of various computational tools, simple procedural recipes, and technical considerations, one can bring the power of deep learning medical image classification to mobile devices, potentially unlocking new domains of applications. The pipeline is demonstrated on four different publicly available datasets: COVID X-rays, COVID CT scans, leaves, and colorectal cancer. We used two application development frameworks: TensorFlow Lite (real-time testing) and Flutter (digital image testing) to test the proposed pipeline. We found that transferring deep learning models to a mobile phone is limited by hardware and classification accuracy drops. To address this issue, we proposed this pipeline to find an optimized model for mobile phones. Finally, we discuss additional applications and computational concerns related to deploying deep-learning models on phones, including real-time analysis and image preprocessing. We believe the associated documentation and code can help physicians and medical experts develop medical image classification applications for distribution.

Keywords: Image classification, machine learning, medical image classification, mobile phone application, cancer

1 Introduction

Disease diagnosis plays a crucial role in healthcare, and for many conditions, medical image data can aid in diagnosing diseases [1,2,3,4,5,6,7,8]. Recent research has made these diagnostic tools more efficient, primarily through deep-learning methods [9]. However, these methods typically require relatively powerful and expensive computational hardware (e.g. modern GPUs), which may not be available in remote or poor areas of the world that lack modern infrastructure. This study proposes a pipeline for classifying images using mobile phones with a remote computer for model training. Even though a central server is required for training, there are free options [10,11] which are sufficient for training the model.

Due to the widespread availability of mobile phones and applications, tasks such as image classification can now be completed at the point-of-care. The deployment of medical image classification models on mobile phones can lead to several technical issues. For example, the model's performance when trained or tested on a computer degrades significantly when the same model is deployed on a mobile phone. Neural network models easily become large enough for phones' memory, and images captured in real time can degrade classification performance. Furthermore, it is unclear how to build a proper workflow connecting training data, often from different sources with varying image sizes and quality, with a model deployed on a smartphone to aid diagnosis. We highlight these issues (and other) and provide feasible solutions to tackle them. The result is an amalgamation of best practices taken from the modern deep learning and data science landscape, including implementations for parameter reduction (Section 2.3) and data augmentation (Section 2.3), resulting in a cost-effective diagnosis pipeline that can be used for medical image classification aiding expert in the diagnosis of the diseases.

The following section explains the differences and similarities between similar projects and the pipeline proposed in this study.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 01-20, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120901

In this study, [12], researchers proposed a mobile-based deep learning application for image classification. However, they used Unity and MATLAB for implementation, whereas we used the Android Studio TensorFlow application development template and Python for model training. This study [13] investigated the usability of mobile phones for medical image classification. In this project [14,15], researchers designed an application for skin diseases, with attached hardware for accurate detection. In this paper [16], an artificial intelligence diagnostic system on mobile Android terminals for cholelithiasis disease is proposed. This article [17] discusses the various challenges that arise when deploying a machine learning model on mobile phones. This work [18] employs edge computing on a mobile device with an integrated web server to diagnose and forecast metastasis in histopathology pictures.

Among the existing studies [19,14,20,13,21,22,15,12,23,24], researchers have developed a machine-learning pipeline and shed light on the medical and general images on mobile phone applications. However, they did not provide the source code and applications that can be used for result replication.

Many research papers explain image classification on mobile phones, but the following are the reasons for reproducing existing work.

- The existing papers do not shed light on real-time testing and the concerns that may arise when deploying a machine-learning model on mobile phones. For example, some android mobile phones require models having weights in specific data types (Float32 and Unsigned Int8), and if the model is trained on different data types, then the application crashes.
- The existing papers do not include source code and documentation, which are essential for reproducing the results and developing an application for some other dataset.
- The existing papers developed different pipelines for various images like plants, tissues, and X-ray/CT-Scan classification. However, we proposed a general application that works for any classification problem by including and excluding substeps in the pipeline.
- We compared the model's performance on the computer, mobile phone when pictures were loaded from the camera, and mobile phone in real-time.

We believe there must be a generalized application that can capture images in real time and from mobile galleries and classify them into various categories; for that purpose, we used existing templates. Using such a template assists in classification problems involving birds, flowers, plants, objects, tissues, and lung cancer classification. We also analyzed the performance of the same model on real-time and digital images, which showed that the performance of the model was highly degraded in real-time analysis. Finally, we present a method for improving the model's performance on a mobile phone.

Section 2 provides the technical context and describes the entire seven-step pipeline process. Section 3 demonstrates the implementation of the pipeline on covid-19, plants, and cancer classification as a use case and discusses the model performance and technical considerations. Sections 4 and 5 contain a discussion and conclusion, including a link to all the data and codes to reproduce results.

2 A pipeline for image classification on mobile phones

There are already several contexts in which deep learning or other classification models are deployed on mobile phones, including small-scale applications such as emoji selection from text [25] to larger-scale recommendation systems [26,27] and face detection from camera images [28,29]. Implementing image classification presents many challenges, such as ensuring that the image dimensions are the same for training and testing data. Furthermore, the image background/environment in which the model trained should be the same as in the field; otherwise, the model might be trained

to see spurious details in the image background, leading to incorrect results. One must also ensure that the model is implemented efficiently to fit inside mobile memory, often forcing reductions in the model size that can sacrifice accuracy. Finally, if one wants to leverage publicly available medical image data in a mobile context, the details of implementing best practices are unclear. Our answer to all these considerations is a machine-learning pipeline, illustrated in Figure 1 and described in this section.

A pipeline is only as good as its data, and starting, one must obtain data suitable for model training and identify the end-user's mobile devices.

As is standard in supervised learning, the training data must be labeled; for medical images, this is typically performed by a domain expert [30]. Many such medical image datasets are available in the public domain [31,32], and one should verify labels with the help of a local physician.

2.1 Step 1: Choose different images sizes and generate sub-datasets

The first step is preprocessing, which consists of image rescaling, normalization, and image resizing [33], and organizing the data appropriately for later analysis. The user selects a small number of different image sizes for testing. Extra testing in this first step will help avoid excess model parameters, which might crash the mobile application in a later stage.

In practical applications, we recommend choosing 3-5 different sizes ranging from 30 x 30 to 500 x 500 as sufficient, and these choices may depend on the expected aspect ratios of the training data. During the model validation in step 3 2.3, one of these image sizes will be selected automatically depending on the model performance. Figure 2 shows the subdatasets having different image sizes generated from the original dataset.

2.2 Step 2: Data splitting for validation

It is essential to use stratified-`k`-fold validation for each image size to avoid over-and under-fitting during the training [34]. This ensures that the folds are chosen such that the mean response value is equal across all folds, ultimately decreasing model bias.

Our recommendation is to start with `k=5`, which repeatedly trains models using 80 percent of the original data and uses the other 20 percent to evaluate model performance.

As is typical with cross-validation, one then systematically trains the model on 4 of the 5 folds and uses the held out to assess the model performance, and the results are averaged to estimate overall model performance.

2.3 Step 3: Model Architecture and Training

Image analysis using deep learning methods is a rapidly growing field with many algorithms competing over a wide variety of applications (e.g. LSTM and RCNN) [35]. For medical image classification, Convolutional Neural Networks (CNN) are the most popular [36].

Convolution is the process of multiplying pixel values by weights and summing them. The first layer of the CNN frequently detects essential characteristics such as horizontal, vertical, and diagonal edges. The first layer's output is then sent to the second layer, which extracts more complex features like corners and edge combinations. Subsequent layers recognize higher-level characteristics such as objects and faces [37]. Based on the activation map of the last convolution layer, the terminal layer outputs a series of confidence ratings (numbers ranging from 0 to 1) that indicate how probable the image belongs to a specific class.

Models are ultimately fit in Keras using `model.fit`. But before training the model, we must address a few key considerations. First, choose the number of parameters or neurons in each layer and the number of layers. If there are too many layers, then there is a possibility that the model

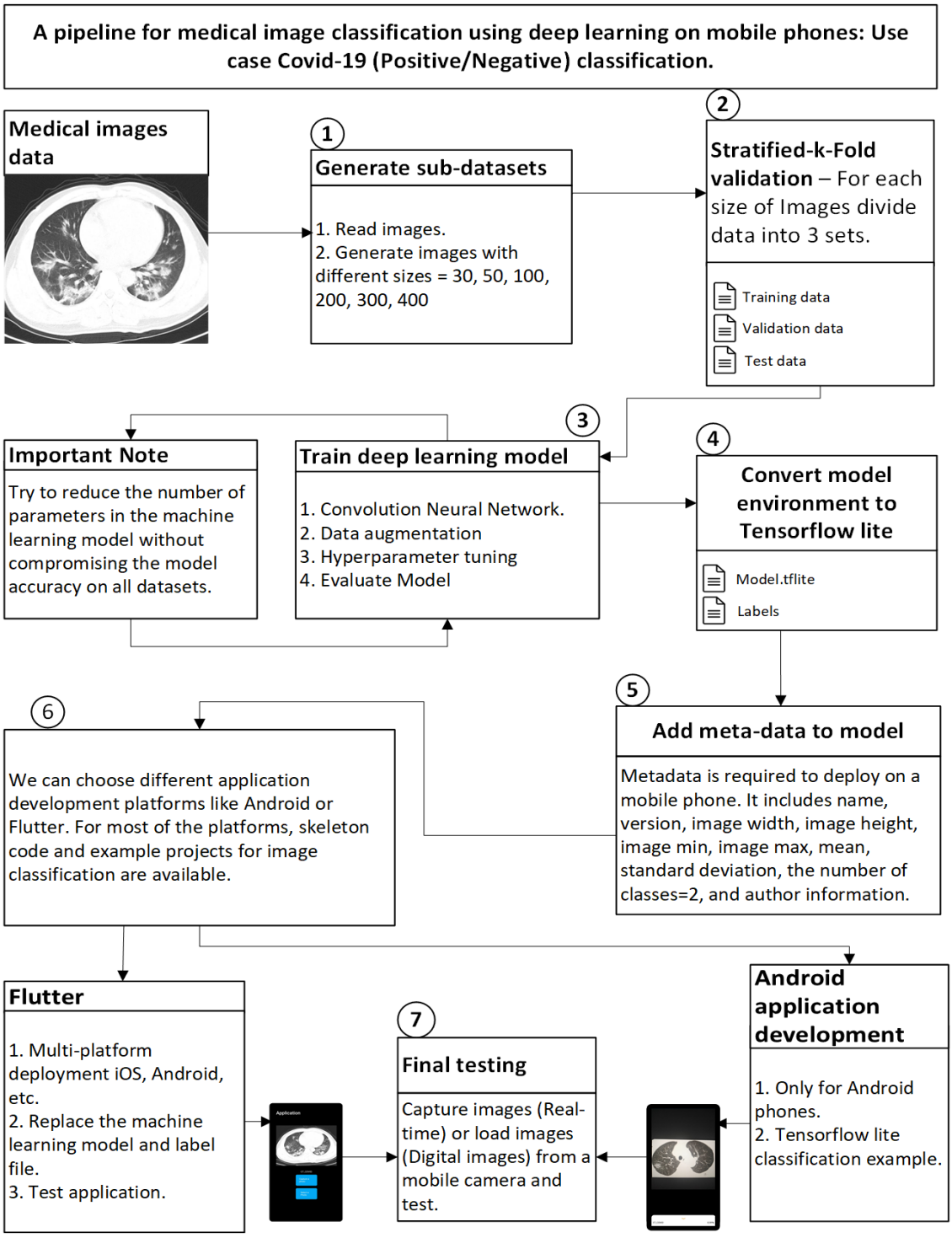


Fig. 1: This diagram shows the overall pipeline for implementing deep learning image classification model for mobile phone devices.

may overfit. If there are too few layers, the model may not learn applicable features. Second, reduce the number of parameters (one could imagine a systematic dropout algorithm that trade-off model size and accuracy) to enable execution within a mobile phone's memory [38], while also minimizing any associated performance penalties. This is another step that is typically adjusted "by hand," and as a starting point, we recommend following the steps taken in our use case below

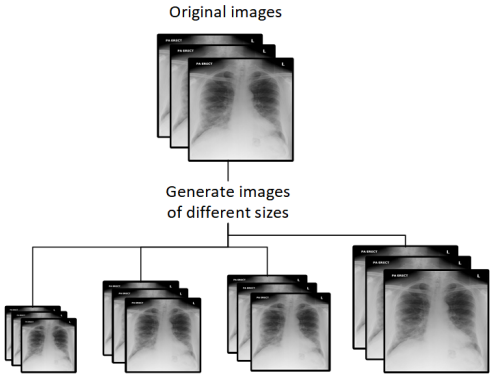


Fig. 2: Choose different images sizes and generate sub-datasets.

in Section 3. Other key considerations before model fitting are included in this subsection and are clearly implemented in the shared code linked below Section 3.

Data augmentation Data augmentation is another key step for maintaining model performance in real-world settings. It artificially stimulates and induces noise and other transformations on the training images when fitting the model [39,40]. In unpredictably noisy applications like ours using potentially multiple mobile phone cameras and hospital settings, such steps are essential.

Parameters for the data augmentation process are controlled via an image train generator and must also be specified. More specifically, when dealing with the medical images captured on mobile phones in real-time, it is essential to consider all options and possibilities with data augmentation, eventually settling on a few different combinations of settings. The key data augmentation parameters in our context are:

- `rescale`: Rescaling (normalizing the pixels values in the image) is the default parameter used in image preprocessing for training the model. Original pictures are in RGB format, with pixel values ranging from 0-255. Such numbers would be too large for the model to cope, resulting in an inflating gradient during the backpropagation phase when training the model. So we multiply the data by `1/255` to change the pixels values between 0-1.
- `rotation range, horizontal flip, and vertical flip`: These parameters randomly rotate and flip training data and should be included because mobile photos can be taken in various rotations. The rotated image is appended with pixels that degrade the classification accuracy in the data augmentation phase. For medical images, the horizontal or vertical flip is fine. However, when the image is rotated, we lose a vast amount of information, depending on the rotation range. One vital point to notice here is that rotated images generated by train generators have appended pixels and can degrade performance. In contrast, images captured by rotating phone cameras do contain all the information.
- `brightness_range`: The brightness range in the train generator increases or decreases the image's color brightness to produce multiple images. When the application is deployed in real-time, there is a possibility that the image's brightness captured from a mobile phone is different from the one on which the model is trained. So, this parameter is compulsory in the data augmentation phase to train the model on images of varying brightness.
- `zoom range, shear range, width shift range, and height shift range`: Zoom range randomly zooms inside pictures, shear range applies shearing transformations to image, width, and height shift change image dimension horizontally and vertically [41]. In a typical image classification task, these parameters can play an essential role in making the model robust. In the case of medical images, it is possible that if these factors are added, the

model's performance will suffer, as evidenced by the findings. In medical images, the difference between the positive and the negative case is subtle. For example, in the dog/cat classification problem, we can distinguish them quickly, but in cases/controls, there is just a white pattern in the image. The transformed image produced using these parameters can convert the positive case into negative and vice versa.

Table 1 contains our recommendation of 4 train generators and their respective parameter settings in Keras. Figure 3 shows the resulting directory structure after following the above steps.

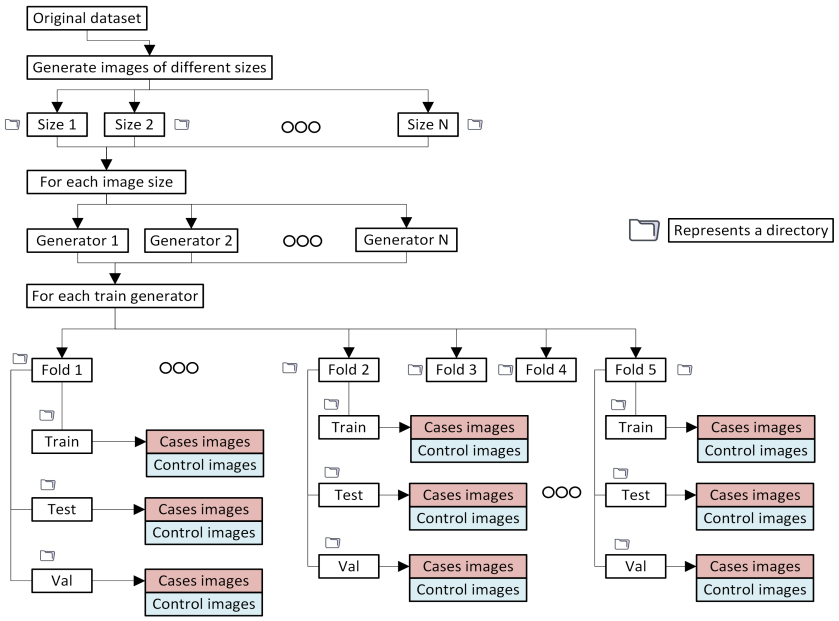


Fig. 3: This diagram shows a directory structure for training the model. Generate images with multiple sizes and make a folder for each size. There can be various train generators for each image size, which do not require a separate folder. For each image size, make one folder for each fold. To use data augmentation or train generator, make three folders (train, test, validation) containing sub-folders representing each category.

Parameter reduction Once one identifies the best model by Stratified-k-fold validation, we must reduce parameters to avoid crashing the mobile application. Hyper-parameter tuning does not, but the number of layers and neurons in each layer affects the model's size. Consider the best model from the previous step has two layers with N filters in the first (convolution layer) and M neurons in the second layer (fully connected layer). Increase filters from 1 to N for the first layer, 1 to M (Neurons) for the second layer, calculate model accuracy and the number of parameters in the model. The number of filters or neurons, the size of the filter, dropout, and strong pooling can be used to reduce the model size.

This step is time-consuming, but once the model with fewer parameters is achieved, it can also be deployed on other devices with low computation power like the Raspberry pi. This model contains a compact form of knowledge learned by the model having high parameters.

Image Generator Parameters	Generator 1	Generator 2	Generator 3	Generator 4
`rescale = 1./255`	✓	✓	✓	✓
`rotation_range = 40`		✓	✓	✓
`brightness_range = [0.2,1.0]`		✓	✓	✓
`horizontal_flip = True`		✓	✓	✓
`vertical_flip = True`		✓	✓	✓
`fill_mode = 'nearest'`		✓	✓	✓
`featurewise_std_normalization = True`			✓	✓
`featurewise_center = True`			✓	✓
`zoom_range = 0.2`				✓
`shear_range = 0.2`				✓
`width_shift_range = 0.2`				✓
`height_shift_range = 0.2`				✓

Table 1: Recommendations for data augmentation settings in Keras image data processing. The first column shows parameter settings, and ✓in subsequent columns denotes inclusion in the 4 recommended image generators.

2.4 Step 4: Convert model environment to TensorFlow Lite

At this step, we have two options: The first is to convert the model to TensorFlow Lite, and the second is to convert the model to TensorFlow Lite and quantize the model (A quantized model executes some or all of the operations on tensors with integers rather than floating-point values) [42].

2.5 Step 5: Specify appropriate metadata

Metadata gives information about the model in addition to its fit weights and architecture. It includes the model name, the input size (image size), and the output size (# of categories), and must be specified. Table 2 gives a starting point for metadata settings. After this step, we have two files: the TensorFlow Lite model and a label text file. One should also verify that the order of categories in the label text file matches the model's prediction order.

name	model's name
version	v1
image width	50
image height	50
image min	0
image max	1
mean	[0]
std	[255]
num_classes	2
author	X

Table 2: Model's metadata parameters and dummy values

2.6 Step 6: Specify appropriate application development platform

Different application development platforms can be used, like a Flutter (`Quantized TensorFlow Lite model`) or Android Studio (`TensorFlow Lite model`). An already developed application template can build deep learning applications in this stage. Explore several options and

decide on the flow of the application. For example, images can be classified in real-time using a camera feed or captured images. This step is related to application development and will not affect the final result.

2.7 Execution and final considerations

At this point, we have a model deployed on potentially multiple mobile phones capable of medical image classification. Select 3 to 5 images from all categories and test the TensorFlow Lite (Quantized/ Unqunatized) model performance on the computer, and that accuracy is the baseline accuracy. For mobile phones, we recommend testing the final model in two ways. The first is to build the application and load those images from the mobile gallery for evaluation (Flutter-based template) using a quantized model. The second option is real-time processing (TensorFlow template) using an unquantized model.

Just like preprocessing is required to train the model on a computer, there is also a preprocessing engine in mobile to preprocess the images, so there can be severe issues when images are tested from the camera feed. The first is the underlying hardware, and the result for the same image using the same model on a computer and mobile phone can yield different results, and there is no solution to mitigate this problem.

The sizes of the images can vary (See figures 4a and 4b), the distance at which the phone should be placed to classify the image can vary, and lastly, the location of images when captured through the phone. If the size of the image varies, then mobile phone distance can be changed such that the frame contains the image. We address each issue separately (See figure 4). Lastly, the background of images can vary (See figures 4c and 4d).

If the model's accuracy when pictures are loaded from the gallery is low, then the model will not perform when pictures are captured and classified in real-time. So, it is recommended to test the model performance on flutter application before shifting it to real-time.

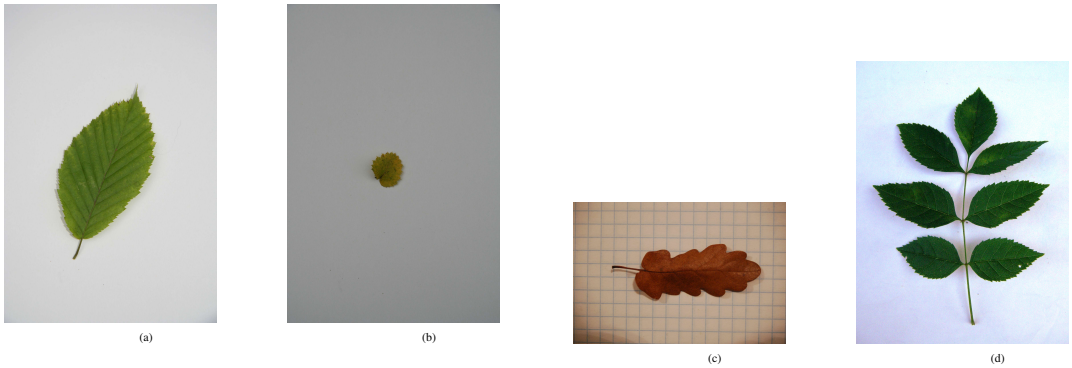


Fig. 4: Panels 4a/4b: Inconsistent images of Hornbeam. Panels 4c/4d: Difference of background of various catagories.

Figure 5 shows how the TensorFlow Lite application template perceives an image.

If classification accuracy is not sufficiently high, return to step 2.3 and repeat the process. To enhance performance, modify the picture size, machine learning model (number of neurons and layers), and hyper-parameters. This final step of testing on additional images is essential for robust performance, and even though the model will run without it, we do not recommend skipping it. Figure 6 shows the possibilities in which the proposed pipeline can be used depending on the type of the dataset.

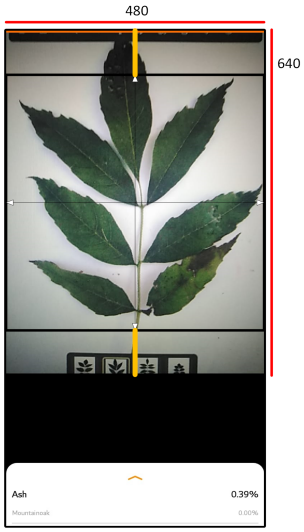


Fig. 5: This diagram shows how real-time picture is perceived by mobile phone. The width of the frame in the TensorFlow Lite template is 480 pixels, and the height is 640 pixels. The captured picture is cropped to 480 by 480, as shown by the black box in the middle. The two yellow lines are equal and show that a particular region is ignored.

3 Use Cases

Medical images have the potential to contribute as a less expensive and more rapid option that can deliver results in minutes instead of days [43]. The added value is not a replacement for the clinical diagnosis of but to rapidly augment physician information with an uncertain and rapidly evolving virus, thereby improving patient care and outcomes.

The system specifications for the following results are: Intel(R) Core(TM) 7-9750H CPU @ 2.60Hz, 16 GB RAM with NVIDIA GeForce RTX 2060 GPU, running Microsoft Windows 10. The development specifications are Cuda compilation tools release 10.0, V10.0.130, Deep Learning framework Keras 2.4.3, Python 3.6.8, and Tensorflow 2.3.1. The mobile phone specifications are a HUAWEI Y7 Prime 2019, Android version 8.1.0, EMUI version 8.2.0, and Model number DUB-LX1.

3.1 Dataset 1: Chest X-ray images

Dataset 1 consisted of images of size (1024,1024,3), which we reduced (Step 1, section 2.1) to $(50, 50, 3)$, $(100, 100, 3)$, $(200, 200, 3)$, and $(300, 300, 3)$. Data splitting for Stratified-5-fold validation (Step 2, section 2.2) resulting in training data (70%), validation data (10%), and test data (20%). The model architecture and training (Step 3, section 2.3) used a CNN with the architecture and hyper-parameters shown in table 9 and 10. This data already contained augmented positive/negative X-ray scan images, so we skipped data augmentation (Section 2.3). Cross-validation, with normal images and augmented images are distributed randomly in the train, validation, and test sets, produced the classification results in Table 3 with image size of 200 yielding the highest accuracy. Parameter reduction (Section 2.3) yielded a model with 4 filters in layer 1 (Convolution layer) and 8 neurons (Fully connected layer) in layer 2, resulting in a reduction in model size from 2,374KB to 24KB. Table 3 shows that the best performance was found with an image size of 200. Taken together with the results from parameter reduction, the model parameters for metadata were finalized and are shown in Table 4.

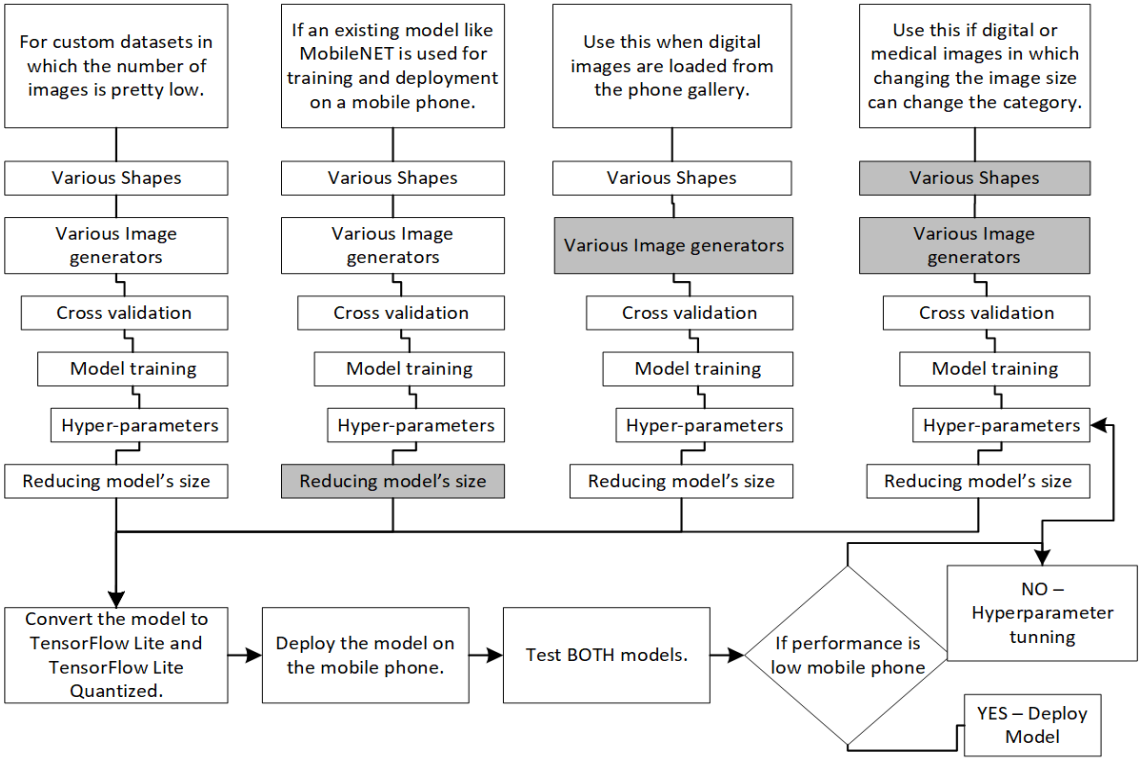


Fig. 6: This figure shows the four possibilities of using the proposed pipeline depending on the type of images. The grayed boxes skip a particular skip step for a particular images type. For custom datasets, all steps are compulsory. If the existing model is used, there is no need for parameter reduction because the model's size is already optimal. When images are loaded from the gallery image generator, digital images will not increase the accuracy. For digital and medical images like CT scans, the image's size is mostly fixed. If the model's performance when tested on a mobile phone is low, tune model architecture and perform hyper-parameter optimization by mutating batch size, number of epochs, and the number of layers in the model.

Dataset 1	
Size	**Accuracy**
50	95.88 (+- 1.80)
100	95.77(+- 1.65)
200	96.54(+- 1.87)
300	59.80(+- 19.41)

Table 3: Dataset 1 [44] cross validation accuracy for different image sizes.

This reduced model was converted to TensorFlow lite (Section, 2.4), and the associated code snippets and scripts can be found in this paper's code repository. These steps involved running the TensorFlow lite converter, generating an appropriate label file with classes, and adding the Table 4 metadata to the TensorFlow Lite model. This resulted in a model with well-defined input size, range of input values, and output (see Table 4). Then, the TensorFlow Lite Android wrapper code generator was used to create platform-specific wrapper code, which efficiently deploys and executes the model code on the mobile phone [45]. Next, we elected to use the Flutter image classification template (Section 2.6), allowing users to capture an image with their camera and pass it from their phone's image gallery application to the model, which then gives the positive or negative prediction.

name	**Dataset 1 Model**
version	**v1**
image width	**200**
image height	**200**
image min	**0**
image max	**1**
mean	**[0]**
std	**[255]**
num_classes	**2**
author	**X**

Table 4: Model's metadata parameters for the best model of dataset 1.

3.2 Dataset 2: Chest CT images

Dataset 2 [46] contained CT-scan images of size $(1024, 1024, 3)$, which we reduced (Step 1, section 2.1) to $(30, 30, 3)$, $(50, 50, 3)$, $(100, 100, 3)$, $(200, 200, 3)$, and $(300, 300, 3)$. The model architecture and training (Step 3, section 2.3) used two CNN models shown in Table 11 (Model 1) and 13 (Model 2) with the data augmentation, and cross-validation to train the model. Table 5 shows the result of classification for dataset 2.

DataSet 2 / Model 1	**Gen1**	**Gen2**	**Gen3**	**Gen4**
Shape=30	63.11(+- 2.98)	59.69(+- 6.21)	61.05(+- 6.48)	59.20(+- 5.11)
Shape=50	63.93(+- 4.33)	57.82(+- 2.84)	59.60(+- 2.80)	57.87(+- 6.51)
Shape=100	**65.82(+- 4.63)**	62.85(+- 4.13)	57.65(+- 2.70)	53.41(+- 6.60)
Shape=200	59.56(+- 3.58)	56.23(+- 7.67)	54.22(+- 5.46)	53.56(+- 3.58)
Shape=300	60.16(+- 9.06)	52.37(+- 6.17)	56.04(+- 6.30)	55.08(+- 2.91)
DataSet 2 / Model 2	**Gen1**	**Gen2**	**Gen3**	**Gen4**
Shape=30	50.95(+- 6.31)	54.89(+- 5.13)	60.68(+- 3.05)	55.77(+- 2.27)
Shape=50	**63.98(+- 1.88)**	60.41(+- 3.96)	58.87(+- 4.76)	56.13(+- 5.77)
Shape=100	59.18(+- 3.80)	55.30(+- 4.00)	56.45(+- 3.53)	55.41(+- 10.60)
Shape=200	62.94(+- 5.49)	53.93(+- 4.03)	56.30(+- 3.41)	50.39(+- 4.00)
Shape=300	60.74(+- 6.91)	50.25(+- 3.97)	51.92(+- 0.81)	53.33(+- 5.81)

Table 5: We used 5-fold validation, 5 different sizes of input images, and 4 generators. If the accuracy is too low, try multiple models.

The best accuracy for dataset 2 was for model 1 with image input size 100, model 1, and train generator 1. Parameter reduction (sub-section, 2.3) reduced the size of the model from 2,374KB to 323KB. Figure 9 (See supplementary material, 6) shows the heatmap of accuracies for each combination of filters and neurons in the first layer and the second layer. Convert model environment to TensorFlow lite version (Step 4, section 2.4).

Specify appropriate metadata (Step 5, section 2.5) adds metadata to the TfLite version of the model. The best accuracy for dataset 2 was for model 1, image size 100, so only change the metadata fields mentioned in Table 6.

name	**Dataset 2 Model**
image width	**100**
image height	**100**

Table 6: Model's metadata parameters for the best model of dataset 2.

After this step, we have a label file and a model with metadata. Specify appropriate application development (Section, 2.6) used flutter image classification template. Examples of the final application at work can be seen in Figures 7a and 7b.

In execution and final considerations (Step 7, section 2.7), we considered the first 5 images from both (CT_COVID and CT_nonCOVID) categories for the final testing on mobile phone, and those images were not part of model training/testing. We tested the model performance on the mobile phone in real-time, and the final accuracy was 0.6, which means the model cannot be deployed, but the performance increased to 0.80 percent when images were loaded from the gallery (Flutter App).

3.3 Dataset 3: Leaves

The dataset [47] consists of five leaves: Ash (24), Beech (28), Hornbeam (30), Mountainoak (20), and Sycamoremaple(20). Dataset is reduced (Step 1, section 2.1) to $(50, 50, 3)$, $(100, 100, 3)$, $(200, 200, 3)$, and $(300, 300, 3)$. The model architecture and training (Step 3, section 2.3) used two CNN models shown in Table 11 (Model 1) and Table 12 (Model 2) with the data augmentation, and cross-validation to train the model. Table 7 shows the result of classification for dataset 3.

DataSet 3	50 - Model 1	100 - Model 2	200 - Model 2	300 - Model 2
Gen1	87.13 (+- 19.63)	86.57 (+- 15.74)	98.04 (+- 2.39)	97.047 (+- 2.41)
Gen2	95.03 (+- 4.87)	91.28 (+- 7.73)	99.04 (+- 1.90)	96.04 (+- 1.97)
Gen3	96.73 (+- 1.64)	94.09 (+- 7.36)	96.04 (+- 3.73)	99.047 (+- 1.90)
Gen4	92.67 (+- 2.94)	91.28 (+- 8.18)	94.14 (+- 5.61)	91.33 (+- 12.92)

Table 7: We used 5-fold validation, 4 different sizes of input images, and 4 generators.

Parameter reduction (sub-section, 2.3) is skipped. Convert model environment to TensorFlow lite version (Step 4, section 2.4) and produce two files: `model.tflite` and `quantizedmodel.tflite`.

Specify appropriate metadata (Step 5, section 2.5) adds metadata to the TfLite version of the model.

After this step, we have a label file and a model with meta data. `model.tflite` is deployed on TensorFlow lite template and `quantizedmodel.tflite` is deployed on Flutter template (Section, 2.6).

In execution and final considerations (Step 7, section 2.7), we considered the first 4 images from all categories for the final testing on mobile phone, and those images were not part of model training/testing. We tested the model performance on the mobile phone when the image size was 50 for generator 3, but the final accuracy was 0.2, which means the model cannot be deployed. At this stage, repeat the process with variation in the model architecture (Step 2, section 2.2) and test the model performance. For shape 224, generator 1, and model 2 (See table 12), the performance increased to 0.75 percent when images were loaded from the gallery (Flutter App). For shape 224, generator 3, and model 2 (See table 12), the performance was 0.75 in real-time (TensorFlow App).

3.4 Dataset 4: Colorectal adenocarcinoma

This is a set of 7180 image patches (9 different categories) from N=50 patients with colorectal adenocarcinoma [48]. The dataset is challenging to train, test, deploy on the phone, and real testing.

Dataset is reduced (Step 1, section 2.1) to $(50, 50, 3)$, $(100, 100, 3)$, $(200, 200, 3)$, and $(300, 300, 3)$. The model architecture and training (Step 3, section 2.3) used one CNN models shown in Table

DataSet 4	50	100	200	300
Gen1	73.19 (+- 6.27)	71.20 (+- 4.11)	18.00 (+- 9.54)	16.80 (+- 8.81)
Gen2	69.19 (+- 10.24)	**75.59 (+- 3.20)**	22.79 (+- 3.91)	22.39 (+- 2.65)
Gen3	68.40 (+- 4.96)	70.00 (+- 4.89)	17.20 (+- 4.11)	20.39 (+- 3.44)
Gen4	70.0 (+- 6.32)	67.20 (+- 7.44)	20.79 (+- 3.24)	22.40 (+- 6.49)

Table 8: Cancer classification result.

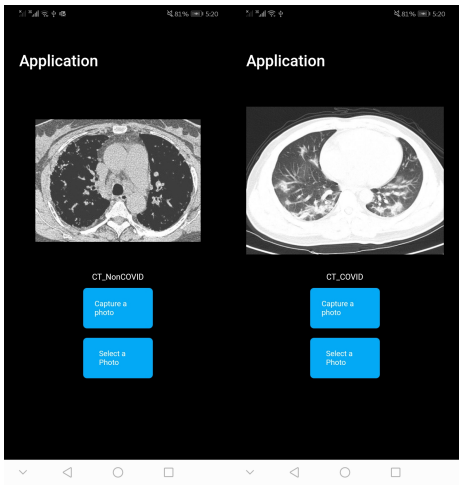
12 (Model 1) with the data augmentation, and cross-validation to train the model. Table 8 shows the result of classification for dataset 4.

Parameter reduction (sub-section, 2.3) is skipped. Convert model environment to TensorFlow lite version (Step 4, section 2.4) and produce two files: `model.tflite` and `quantizedmodel.tflite`.

Specify appropriate metadata (Step 5, section 2.5) adds metadata to the TfLite version of the model.

After this step, we have a label file and a model with meta data. `model.tflite` is deployed on TensorFlow lite template and `quantizedmodel.tflite` is deployed on Flutter template (Section, 2.6).

In execution and final considerations (Step 7, section 2.7), we considered the first 4 images from all categories for the final testing on mobile phone, and those images were not part of model training/testing. We tested the model performance on the mobile phone in real-time, and the final accuracy was 0.2, which means the model cannot be deployed. We increased the image size to 200, and the performance increased to 0.56 percent when images were loaded from the gallery (Flutter App). One point to notice here is the test accuracy was 0.99 on the computer.



(a) (b)

Fig. 7: Screenshot of application 1 7a 7b for covid detection using CT scan, deployed on android phone. Panels 7a/7b: Covid negative/positive Chest CT-scans, respectively.

The following paragraph elaborates the time to execute the pipeline.

For reading images (Step 1, 2.1), write a script, and it may take an average time of 20 - 30 minutes, depending on the way the dataset is stored. Below we calculated the time to train the model for images of various sizes for dataset 2, containing about 744 images. Time to train machine learning model (Step 3, 2.3) was 10, 17, 50, 180, and 600 minutes (total time 14 hours) for images having dimensions 30, 50, 100, 200, and 300. The parameter reduction step (Step 4, 2.4)

is time-consuming, and for one model having two layers, it took about 1 to 2 days. Converting the model to TensorFlow lite and adding metadata (Step 5, 2.5) takes about 5 minutes. Modifying the image classification template (Step 6, 2.6) for a specific dataset will take about 30 minutes. Building and deploying the application will take about 6 hours for one phone. So the total time for running the pipeline for one dataset is about 3 days.

4 Discussion

This section contains the limitations and future directions of the proposed pipeline. There are a few limitations associated with the proposed approach. For example, we considered only android phones (a specific vendor) running a particular version of the Android operating system. There is a high probability that the final classification performance would be identical for phones running the Android operating system due to the same android operating system, but for iPhone or Raspberry Pi, the results may vary. Such applications can also be developed for the iPhone using a different application development framework, one of the future directions for the proposed framework.

5 Conclusion

This study proposed a pipeline for deploying a deep learning model for medical image classification on mobile phones. The scope of the solution is not only limited to covid-19, but we can also use it for breast cancer or any other medical dataset, making the mobile phone a diagnostic tool for medical images classification. Complex models and other high-end application development skills can also lead to image segmentation. It is essential to highlight the usability and application of the proposed pipeline. Imagine traveling in a deep forest, which has insects and plants that can cause rashes. If someone suffers from skin allergies from a plant, they cannot call a doctor or find any medical assistance. Nevertheless, having a mobile phone application can tell which medicine is appropriate for a particular injury. Such applications can empower doctors in clinical settings where they may require knowledge from other sources to diagnose some diseases better. It will also give access to the public to use that application because there are about 4.3 billion people who use mobile phones.

6 Supplementary information

The documentation associated with the manuscript is available at the following link. https://github.com/MuhammadMuneeb007/A-pipeline-for-image-classification-using-deep-learning-on-mobile-phones

The code segments associated with the documentation are available at the following link. https://muhammadmuneeb007.github.io/A-pipeline-for-image-classification-using-deep-learning-on-mobile-phones/Find%20a%20dataset.html#directory-form

The files, associated applications, and directories are available at the following link. https://1drv.ms/u/s!AlFVll05llt7gwheV2i4SN3rba13?e=My0QKq

This section contains the material referenced in the section 3.

Model 1 architecture for dataset 1	
Layers	Parameters
Layer 1 - Con2D	30 Filters * (kernel size = (3,3))
Layer 2 - MaxPool2D	(pool size = (2,2))
Reshape	-
Layer 3 - FullyConnected	(50 Neurons)
Relu	-
Layer 4 - FullyConnected	(2 Neurons)
Softmax	-

Table 9: Model 1 architecture for dataset 1.

Model's Hyper-parameters	
Hyper-parameters	Value
Batch size	10
Epochs	50
Validation size	10%
Optimizer	SGD
Loss	categorical/binary_crossentropy
Metrics	Accuracy

Table 10: Hyper-parameters for all dataset 1 and 2.

Model 1 architecture for dataset 2	
Layers	Parameters
Layer 1 - Con2D	32 Filters * (kernel size = (3,3))
Layer 2 - MaxPool2D	(pool size = (2,2))
Reshape	-
Layer 3 - FullyConnected	(128 Neurons)
Relu	-
Layer 4 - FullyConnected	(2 Neurons)
Softmax	-

Table 11: Model 1 architecture for dataset 2.

Model 2 architecture for dataset 3	
Layers	Parameters
Layer 1 - Con2D	32 Filters * (kernel size = (3,3))
Layer 2 - MaxPool2D	(pool size = (2,2))
Layer 3 - Con2D	32 Filters * (kernel size = (3,3))
Layer 4 - MaxPool2D	(pool size = (2,2))
Layer 5 - Con2D	32 Filters * (kernel size = (3,3))
Layer 6 - MaxPool2D	(pool size = (2,2))
Layer 7 - Con2D	64 Filters * (kernel size = (3,3))
Layer 8 - MaxPool2D	(pool size = (2,2))
Reshape	-
Layer 9 - FullyConnected	(128 Neurons)
Relu	-
Layer 10 - FullyConnected	(50 Neurons)
Relu	-
Layer 11 - FullyConnected	(20 Neurons)
Relu	-
Layer 12 - FullyConnected	(Categories Neurons)
Softmax	-

Table 12: Model 2 architecture for dataset 3 and 4. We increased the number of convolutional layers to extract the information because the model 13 did not work.

Model 2 architecture for dataset 2	
Layers	Parameters
Layer 1 - Con2D	32 Filters * (kernel size = (3,3))
Layer 2 - MaxPool2D	(pool size = (2,2))
Layer 3 - Con2D	64 Filters * (kernel size = (3,3))
Layer 4 - MaxPool2D	(pool size = (2,2))
Reshape	-
Layer 5 - FullyConnected	(256 Neurons)
Relu	-
Layer 6 - FullyConnected	(128 Neurons)
Relu	-
Softmax	-

Table 13: Model 2 architecture for dataset 2.

Acknowledgments

This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. CIRA-2019-050 to SFF.

References

1. S. Mitra and B. U. Shankar, “Medical image analysis for cancer management in natural computing framework,” *Information Sciences*, vol. 306, pp. 111–131, Jun. 2015. [Online]. Available: https://doi.org/10.1016/j.ins.2015.02.015
2. P. D. Velusamy and P. Karandharaj, “Medical image processing schemes for cancer detection: A survey,” in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*. IEEE, Mar. 2014. [Online]. Available: https://doi.org/10.1109/icgccee.2014.6922267

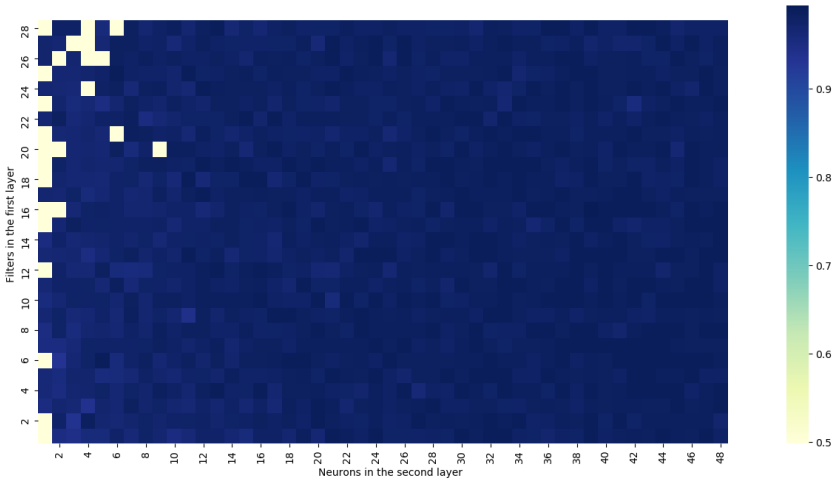


Fig. 8: Heatmap of accuracies. Y-axis and X-axis represent the number of neurons in the first layer and the second layer.

3. S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, "A survey of MRI-based medical image analysis for brain tumor studies," *Physics in Medicine and Biology*, vol. 58, no. 13, pp. R97–R129, Jun. 2013. [Online]. Available: https://doi.org/10.1088/0031-9155/58/13/r97
4. W. Lin, T. Tong, Q. Gao, D. Guo, X. Du, Y. Yang, G. Guo, M. Xiao, M. Du, and X. Q. and, "Convolutional neural networks-based MRI image analysis for the alzheimer's disease prediction from mild cognitive impairment," *Frontiers in Neuroscience*, vol. 12, Nov. 2018. [Online]. Available: https://doi.org/10.3389/fnins.2018.00777
5. J. Islam and Y. Zhang, "Brain MRI analysis for alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks," *Brain Informatics*, vol. 5, no. 2, May 2018. [Online]. Available: https://doi.org/10.1186/s40708-018-0080-3
6. S. Bhattacharya, P. K. R. Maddikunta, Q.-V. Pham, T. R. Gadekallu, S. R. K. S, C. L. Chowdhary, M. Alazab, and M. J. Piran, "Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey," *Sustainable Cities and Society*, vol. 65, p. 102589, Feb. 2021. [Online]. Available: https://doi.org/10.1016/j.scs.2020.102589
7. S. Liang, H. Liu, Y. Gu, X. Guo, H. Li, L. Li, Z. Wu, M. Liu, and L. Tao, "Fast automated detection of COVID-19 from medical images using convolutional neural networks," *Communications Biology*, vol. 4, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1038/s42003-020-01535-7
8. T. A. Soomro, L. Zheng, A. J. Afifi, A. Ali, M. Yin, and J. Gao, "Artificial intelligence (AI) for medical imaging to combat coronavirus disease (COVID-19): a detailed review with direction for future research," *Artificial Intelligence Review*, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s10462-021-09985-z
9. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005
10. Reference, "Welcome to colaboratory - colaboratory," https://colab.research.google.com/notebooks/intro.ipynb, (Accessed on 08/16/2021).
11. ——, "Microsoft azure notebooks," https://notebooks.azure.com/, (Accessed on 08/16/2021).
12. T. V. Dittimi and C. Y. Suen, "Mobile phone based ensemble classification of deep learned feature for medical image analysis," *IETE Technical Review*, vol. 37, no. 2, pp. 157–168, Feb. 2019. [Online]. Available: https://doi.org/10.1080/02564602.2019.1576550
13. B. Hunt, A. J. Ruiz, and B. W. Pogue, "Smartphone-based imaging systems for medical applications: a critical review," *Journal of Biomedical Optics*, vol. 26, no. 04, Apr. 2021. [Online]. Available: https://doi.org/10.1117/1.jbo.26.4.040902
14. N.-M. Cheung, V. Pomponiu, D. Toan, and H. Nejati, "Mobile image analysis for medical applications," *SPIE Newsroom*, Jul. 2015. [Online]. Available: https://doi.org/10.1117/2.1201506.005997
15. A. Karargyris, O. Karargyris, and A. Pantelopoulos, "DERMA/care: An advanced image-processing mobile application for monitoring skin cancer," in *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. IEEE, Nov. 2012. [Online]. Available: https://doi.org/10.1109/ictai.2012.180

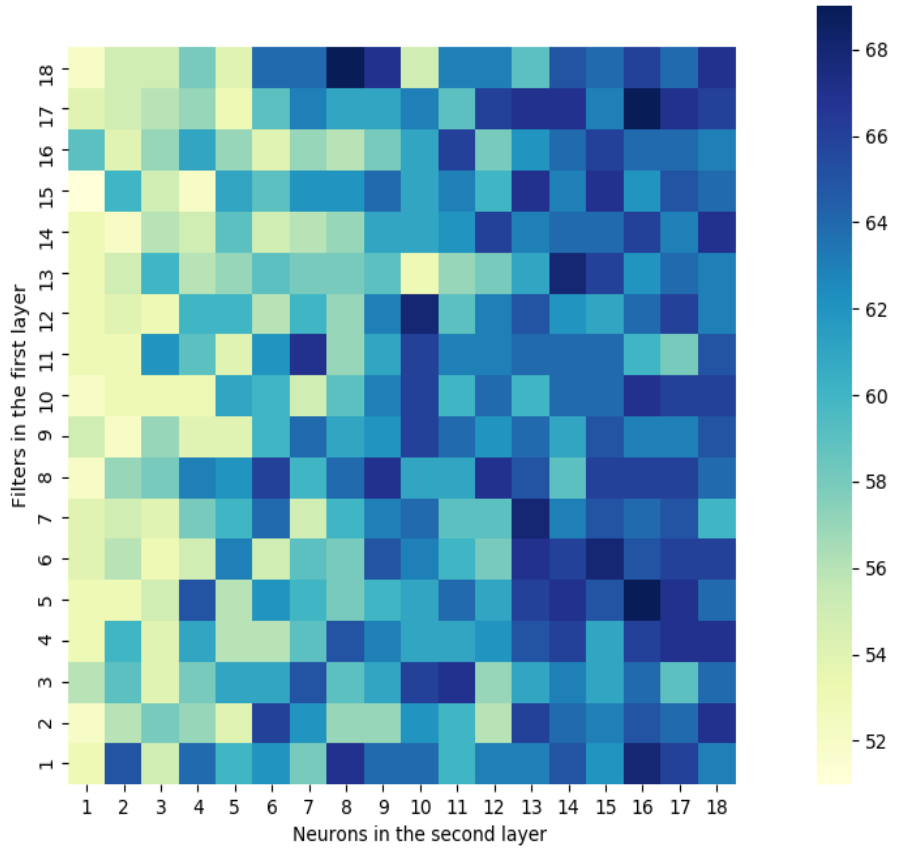


Fig. 9: Heatmap of accuracies. Y-axis and X-axis represent the number of neurons in the first layer and the second layer.

16. S. Pang, S. Wang, A. Rodríguez-Patón, P. Li, and X. Wang, "An artificial intelligent diagnostic system on mobile android terminals for cholelithiasis by lightweight convolutional neural network," *PLOS ONE*, vol. 14, no. 9, p. e0221720, Sep. 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0221720
17. M. Curiel and L. Flórez-Valencia, "Challenges in processing medical images in mobile devices," in *Trends and Advancements of Image Processing and Its Applications*. Springer International Publishing, Nov. 2021, pp. 31–51. [Online]. Available: https://doi.org/10.1007/978-3-030-75945-2_2
18. A. Johny and K. N. Madhusoodanan, "Edge computing using embedded webserver with mobile device for diagnosis and prediction of metastasis in histopathological images," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s44196-021-00040-x
19. C. Morikawa, M. Kobayashi, M. Satoh, Y. Kuroda, T. Inomata, H. Matsuo, T. Miura, and M. Hilaga, "Image and video processing on mobile devices: a survey," *The Visual Computer*, vol. 37, no. 12, pp. 2931–2949, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s00371-021-02200-8
20. R. Rajendran and J. Rajendiran, "Image analysis using smartphones for medical applications: A survey," pp. 275–290, Jul. 2019. [Online]. Available: https://doi.org/10.1002/9781119439004.ch12
21. J. K. Carroll, A. Moorhead, R. Bond, W. G. LeBlanc, R. J. Petrella, and K. Fiscella, "Who uses mobile phone health apps and does use matter? a secondary data analytics approach," *Journal of Medical Internet Research*, vol. 19, no. 4, p. e125, Apr. 2017. [Online]. Available: https://doi.org/10.2196/jmir.5604
22. Z. F. khan and S. R. Alotaibi, "Applications of artificial intelligence and big data analytics in m-health: A healthcare system perspective," *Journal of Healthcare Engineering*, vol. 2020, pp. 1–15, Sep. 2020. [Online]. Available: https://doi.org/10.1155/2020/8894694

23. M. Straczkiewicz, P. James, and J.-P. Onnela, "A systematic review of smartphone-based human activity recognition methods for health research," *npj Digital Medicine*, vol. 4, no. 1, Oct. 2021. [Online]. Available: https://doi.org/10.1038/s41746-021-00514-4
24. K. Nwe, M. E. Larsen, N. Nelissen, and D. C.-W. Wong, "Medical mobile app classification using the national institute for health and care excellence evidence standards framework for digital health technologies: Interrater reliability study," *Journal of Medical Internet Research*, vol. 22, no. 6, p. e17457, Jun. 2020. [Online]. Available: https://doi.org/10.2196/17457
25. Q. Bai, Q. Dan, Z. Mu, and M. Yang, "A systematic review of emoji: Current research and future perspectives," *Frontiers in Psychology*, vol. 10, Oct. 2019. [Online]. Available: https://doi.org/10.3389/fpsyg.2019.02221
26. R. C. Jisha, J. M. Amrita, A. R. Vijay, and G. S. Indhu, "Mobile app recommendation system using machine learning classification," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 940–943.
27. K. Zhu, Z. Liu, L. Zhang, and X. Gu, "A mobile application recommendation framework by exploiting personal preference with constraints," *Mobile Information Systems*, vol. 2017, pp. 1–9, 2017. [Online]. Available: https://doi.org/10.1155/2017/4542326
28. K. Choi, K.-A. Toh, and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern Recognition*, vol. 44, no. 2, pp. 386–400, Feb. 2011. [Online]. Available: https://doi.org/10.1016/j.patcog.2010.08.009
29. B. Ríos-Sánchez, D. C. da Silva, N. Martín-Yuste, and C. Sánchez-Ávila, "Deep learning for face recognition on mobile devices," *IET Biometrics*, vol. 9, no. 3, pp. 109–117, Feb. 2020. [Online]. Available: https://doi.org/10.1049/iet-bmt.2019.0093
30. D. M. Hedderich and S. B. Eickhoff, "Machine learning for psychiatry: getting doctors at the black box?" *Molecular Psychiatry*, vol. 26, no. 1, pp. 23–25, Nov. 2020. [Online]. Available: https://doi.org/10.1038/s41380-020-00931-z
31. Reference, "aylward.org - open-access medical image repositories," https://www.aylward.org/notes/open-access-medical-image-repositories, (Accessed on 08/18/2021).
32. ——, "Smir - sicas medical image repository," https://www.smir.ch/, (Accessed on 08/18/2021).
33. S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, "A survey of image classification methods and techniques," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. IEEE, Jul. 2014. [Online]. Available: https://doi.org/10.1109/iccicct.2014.6993023
34. B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete bayesian network analysis?" *Computational Statistics*, vol. 36, no. 3, pp. 2009–2031, Jun. 2020. [Online]. Available: https://doi.org/10.1007/s00180-020-00999-9
35. Y. Aslam and S. N, "A review of deep learning approaches for image analysis," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 709–714.
36. S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, Dec. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0276-2
37. F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.
38. X. Xia, E. Shihab, Y. Kamei, D. Lo, and X. Wang, "Predicting crashing releases of mobile applications," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, Sep. 2016. [Online]. Available: https://doi.org/10.1145/2961111.2962606
39. C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, Jul. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0
40. L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *ArXiv*, vol. abs/1712.04621, 2017.
41. Reference, "Building powerful image classification models using very little data," https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html, (Accessed on 08/22/2021).
42. Y. Deng, "Deep learning on mobile devices: a review," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, S. S. Agaian, S. P. DelMarco, and V. K. Asari, Eds. SPIE, May 2019. [Online]. Available: https://doi.org/10.1117/12.2518469
43. Reference, "Plain radiograph/x-ray - insideradiology," https://www.insideradiology.com.au/plain-radiograph-x-ray/#:~:text=It%20usually%20takes%20less%20than,about%2C%20and%20your%20general%20health., (Accessed on 06/02/2021).
44. A. M. Alqudah, "Augmented covid-19 x-ray images dataset," 2020. [Online]. Available: https://data.mendeley.com/datasets/2fxz4px6d8/3
45. Reference, "Adding metadata to tensorflow lite models," https://www.tensorflow.org/lite/convert/metadata, (Accessed on 09/06/2021).
46. J. Zhao, Y. Zhang, X. He, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.
47. Reference, "Leaves," https://ftp.cngb.org/pub/gigadb/pub/10.5524/100001_101000/100251/Leaf_outlines.zip, (Accessed on 03/12/2022).

48. ——, “100,000 histological images of human colorectal cancer and healthy tissue — zenodo,” https://zenodo.org/record/1214456#.YhxvROhBzIU, (Accessed on 03/09/2022).

Authors

Muhammad Muneeb obtained his M.Sc. in Computer Science from the Khalifa University, Abu Dhabi, UAE. I am currently working as a research associate in the same institute under the supervision of Dr. Samuel. I like to work on inter-discipline problems. Have interests in algorithms, automation, genetics, medical image analysis, and optimization.

Samuel F. Feng obtained his PhD in Applied and Computational Mathematics from Princeton University (2012), his MA from Princeton University (2009), and his BA from Rice University (2007). From 2012 to 2014 he was a Ruth L. Kirschstein NRSA Postdoctoral Fellow at the Princeton Neuroscience Institute, working on stochastic models of decision making in psychology and neuroscience. In 2014 he joined the Department of Mathematics at Khalifa University, Abu Dhabi, UAE, where is currently an assistant professor.

Andreas Henschel earned his M.Sc. and Ph.D. in Computer Science from the Technical University of Dresden (Germany), in 2002 and 2008, respectively. He joined Masdar Institute as a Postdoctoral researcher in 2009. In 2011, he became Assistant Professor at Masdar Institute, UAE. He spent one year at the Massachusetts Institute of Technology MIT), USA as a Visiting Scholar.

PUZZLE SOLVING WITHOUT SEARCH OR HUMAN KNOWLEDGE: AN UNNATURAL LANGUAGE APPROACH

David Noever[1] and Ryerson Burdick[2]

[1]PeopleTec, Inc., Huntsville, AL, USA
[2]University of Maryland, College Park, MD, USA

ABSTRACT

The application of Generative Pre-trained Transformer (GPT-2) to learn text-archived game notation provides a model environment for exploring sparse reward gameplay. The transformer architecture proves amenable to training on solved text archives describing mazes, Rubik's Cube, and Sudoku solvers. The method benefits from fine-tuning the transformer architecture to visualize plausible strategies derived outside any guidance from human heuristics or domain expertise. The large search space ($>10^{19}$) for the games provides a puzzle environment in which the solution has few intermediate rewards and a final move that solves the challenge.

KEYWORDS

Natural Language Processing (NLP), Transformers, Game Play, Deep Learning.

1. INTRODUCTION

The transformer architecture provides a scalable mechanism for natural language generation (NLG) to encode long-range dependencies needed to output plausible text narratives. Transformers [1] have rapidly advanced to rival or overtake other deep learning architectures such as convolutional neural networks (CNN). Initially developed to handle long-term language dependencies, this approach over-weights important relations via the "attention" method rather than attempting to localize dependencies (CNN) or grow dense networks for all weights. While the resulting sparse network extends available long-term connections needed to relate distant parts-of-speech or sentence context, the net effect has grown to massive models now in the trillions of connection weights [2]. This approach has since found application in other fields unrelated to the original language modeling, such as non-local effects needed for visual context problems. Among the early successes, the Generative Pretrained Transformer (GPT-2) from Open AI [3] remains one of the most robust architectures for fine-tuning applications. In these cases, the original training set gets specialized to diverse domains outside of its initial text data [4]. As a result, previous work has applied GPT-2 to play chess [5], Go [6], and other complex strategy games without knowing the explicit rules but instead learning the text patterns necessary to transfer learning from archival play. Since no move constraints get introduced to the transformer (e.g. legal vs. illegal moves), the trained model results in gameplay without human knowledge [7]. Because of its origins in natural language modeling, GPT-2 serves as a viable mimic of human narratives (sometimes called a "stochastic parrot"), particularly for the specialized use case called here as "unnatural language" generation. Figure 1 highlights some example applications of learning text archives for puzzles including Rubik's cube, Sudoku, and maze solvers.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 21-31, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120902

1.1. Puzzles and Games

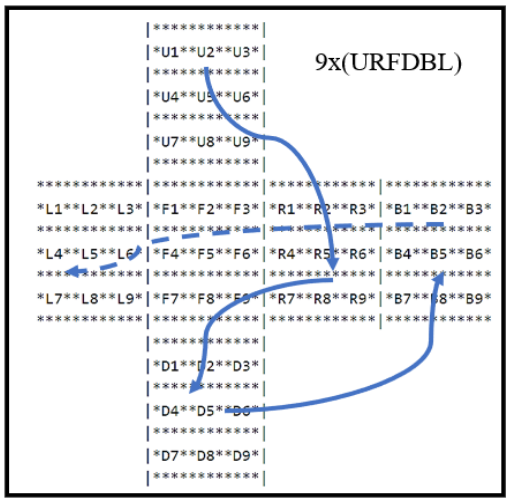


Figure 1. Rubik's Cube String Notation and
Syntax for Position and Colors

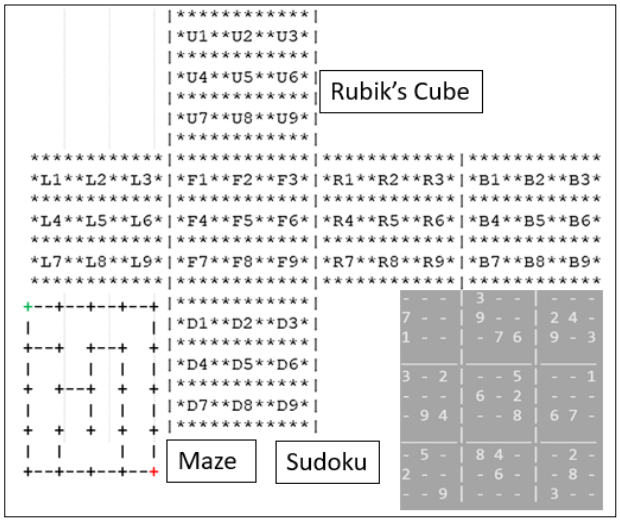


Figure 2. Example sparse reward puzzles in text notation

The application of AI and machine learning approaches to gameplay offers a rich history ranging from Deep Blue in chess (1997) to AlphaZero [7]. One appealing aspect follows from the obvious scoring metrics associated with scoring humans vs. machines. In economics and game theory, a key distinction among the types of games amenable to AI implicitly favors perfect information games, such as chess, checkers, Go, etc. The board state is known equally to the human and machine players and gameplay progresses sequentially. The sequential play alternates its moves in a way different from simultaneous plays like Rock, Paper, Scissors, which are also perfect information but not alternating moves. Recent advances in Monte Carlo tree search [7] have conquered human experts even in imperfect information games like poker, in which players can bluff while concealing their true game state until forced to reveal winners and losers in the final move of turning over cards or folding their hands. A third game category has recently attracted AI attention and might be informally classed as open-ended worlds like the video play in DOTA and StarCraft 2. Playing these games effectively as a tree search problem requires

enormous computing resources and must handle the wide universe of available strategies (“where almost anything goes”). The present research examines a fourth possible category well known to the reinforcement learning community as games or puzzles that offer sparse rewards. These problems are generally characterized by large state spaces and a relatively small number of states which have an associated reward signal. Infrequent rewards often make gradient-based search and other methods that depend upon a smooth reward signal impractical.

1.2. Sparse Rewards

One notable example of a sparse rewards task is the Rubik's cube. The Rubik's cube is a puzzle with 6 rotating faces, each composed of 9 smaller squares ("cubies") which take one of 6 colors. The objective is to rotate the faces until each face contains 9 squares of the same color. The Rubik's cube is an extreme example of a sparse rewards task [8-9] it has a large state space consisting of approximately 4.9×10^{19} possible configurations, and only the goal state has an associated reward signal. This causes a sudden stepwise gain in rewards when making the final solving move.

A less extreme example of a sparse rewards task is the numerical puzzle game, Sudoku. The objective of Sudoku is to fill in missing cells of a 9×9 grid with the numbers 1-9, subject to the conditions that no number may appear twice in the same row, column, or 3×3 block. Because of these conditions, Sudoku is also known as a constraint satisfaction game. Like the Rubik's Cube, Sudoku has an enormously large state space, as there are approximately 6.671×10^{21} valid Sudoku grids alone [10], and a reward signal is only achieved during the final step of the solving process.

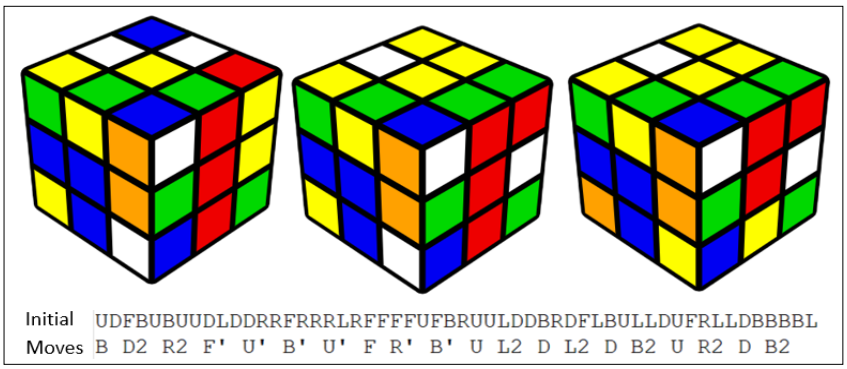


Figure 3. Solution Cube Notation for Visualizing Moves

It is worth noting that traditional Monte Carlo tree search techniques have exhaustive computing needs compared to GPT-2. For example, AlphaGo uses 1920 CPUs and 280GPUs (or $3000 in electricity costs) for each game [11]. The research explores solving these sparse reward games without reinforcement learning or Monte Carlo tree search. Instead, we apply the long-range rewards (weights) found in current language transformers based on their attention strategies applied to text generators. The best-known examples of games with text generators largely focus on fine-tuning the GPT-2. Previous work has applied GPT-2 to perfect information games (e.g. chess, Go). For Sudoku and Rubik’s Cube, deterministic (search) algorithms deliver sufficient quantities of good training data such that traditional deep learning techniques can solve the games using computer visions approaches and convolutional neural networks [12-13]. We propose to solve the games using text-based (ASCII) archives and fine-tune the transformer architecture to visualize another strategic solution to the sparse rewards challenges.

```
<|startoftext|>[WP]
004300209005009001070060043006002087190007400050
008300060000010500350869004291030O[RESPONSE]
864371259325849761971265843436192587198657432 25
748391668973412571352869454291637 8<|endoftext|>
```

Figure 4. Example Sudoku Starting and Final States

2. METHODS

This research compares solving three classes of games using language modeling: Rubik's Cube, Sudoku, and mazes. For each game or puzzle, language representations are generated from archives of available gameplay and fine-tuning large pattern recognition models. While the models were originally trained for language generation tasks, they can be fine-tuned to generate plausible game moves. One common element of the approach stems from the game moves in a string (ASCII text) format. Another notable feature is their visualization, so the language model can be viewed as another game player and not an abstract symbol generator alone. In other words, one can assess the model through a score and rate the strategies it employs.

2.1. Rubik's Cube Representation

For a Rubik's task, we generated a dataset consisting of 5,000 pairs of initial cube configurations and corresponding solutions. To generate the initial configurations, a scrambling formula was created by randomly generating a sequence of moves to perturb the cube from the completed state. These scrambling formulas were anywhere between 1 and 5 moves in length, and an equal number of samples were generated for each possible scramble formula length. Once an initial configuration was determined, the cube state was represented by an encoding string following text formats[14]. As illustrated in Figures 2-3, this encoding uses the cube string positions for an unfolded cube with ordered positions (9 digits) for the following faces: Up (U), Right (R), Front (F), Down (D), Back (B) and Left (L). The string order proves important [15] since a fully solved cube would have 9x(URFDBL) for the completed color faces. The position U1 can be any of the 6 standard colors (red, yellow, orange, blue, white, green). A starting state like "RBL…" means the right color (say, green) is in fixed position U1, the back color (say, red) is in position U2, etc. Finally, once all scrambling formulas were converted to encoding strings, duplicate cube states were removed from the dataset and the remaining samples were split into a training set containing 2404 samples and a test set containing 601 samples.

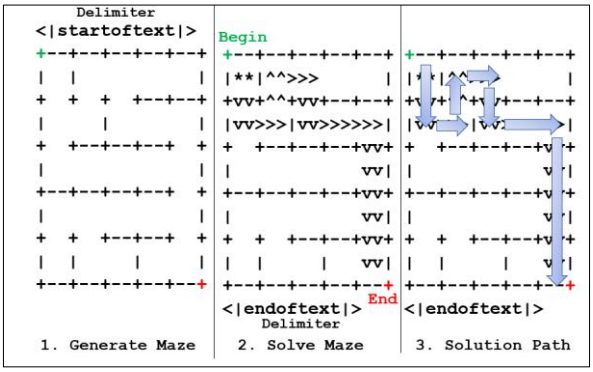


Figure 5. Maze generator and transformer solutions

After the initial Rubik's cube configurations and corresponding encodings were generated, a solution was determined using the Kociemba algorithm [15]. The Rubik's solution syntax introduces each move as space-separated letters with punctuation and numbering conventions describing the turn. A single letter alone means to turn that face in the URFDBL dictionary of choices clockwise by 90 degrees (quarter turns). A letter with an apostrophe means the opposite counterclockwise turn by 90 degrees. If the letter has a number 2, the face gets a half-turn (180 degrees). An example initial state and solution of single moves is shown in Figure 3. We visualize each step of the cube solution using the Visual Cube application [16] and validate solutions using the PyCuber python library [17].

2.2. Sudoku Representations

For Sudoku, we collected one million solved games [18], which consists of a similar split view of the initial and final state. To divide the start and finished puzzle, we insert a word prompt [WP] to demark the first digit of the 81 in the 9x9 puzzle (Figure 4). A zero value represents a blank or open slot. The second demarcation [RESPONSE] serves as a delimiter for the puzzle solution. The visualization of a solved puzzle was customized in a console application that pushes each new digit onto the string for replacing the next available open gap (zero). The puzzle's starting and ending delimiters (<|…|>) allow the generated text of a proposed solution to be parsed and truncated to simplify interpretation.

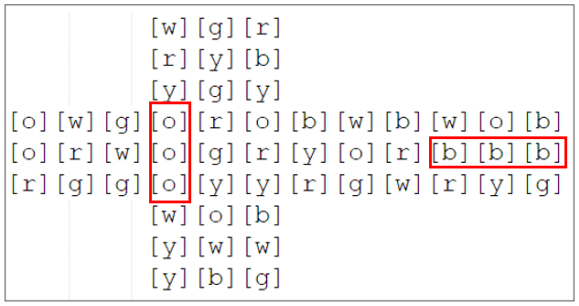


Figure 6. Rubik's Cube Transformer Solving for Single Rows

2.3. Maze Representations

For solving mazes, we generated 10,000 random mazes and embedded their ASCII text solutions between the start and stop delimiters. To generate mazes of 4x4 and 5x5 [19], we use (+) and (-) signs to outline the text grid boundaries, the use (|) pipe symbology to define walls. As shown in Figure 5, we encode both the unsolved and solved mazes in a single training text example for each maze. The training solutions follow the search methods outlined as breadth or depth-first techniques [20]. Each example maze begins with the upper left corner as the starting position (**); the direction of maze navigation follows a text arrow notation (^^=up; >>=right; vv=down; <<=left). As with the other cases, the training set represents a series of maze pairings (unsolved and solved) with one maze in a single row submitted to the transformer.

3. RESULTS

For each puzzle, this work found a visual representation of the language model at play. Where possible, the gameplay is shown as animated versions with sequences of moves.

3.1. Cube Solver

On the Rubik's Cube data, the transformer was unable to solve the complete puzzle more than one in seven attempts. Out of the 601 generated responses for the test examples, 11 were invalid (~1.8%), 576 were incorrect (~95.8%), and only 14 were correct (~2.3%). The small proportion of invalid generated responses indicates that despite being trained initially on natural language, the transformer has adapted well to the "unnatural" language of Rubik's cube formulae; even when it was unable to solve the cube, the overwhelming majority of the time the transformer produced an output which corresponds to a valid Rubik's formula. Figure 6 shows the solution for single rows as an incomplete solution but progressively improved cube state.

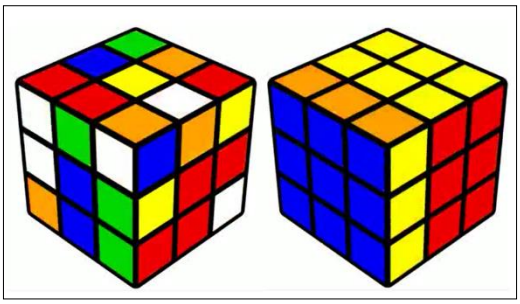


Figure 7. Transformer (left) vs. Kociemba (right) algorithm

Given the short fine-tuning period (~2000 epochs) and the small number of training examples (~2400), it is significant that the transformer was able to solve the Rubik's puzzle at all. Interestingly, though the majority (9/14) of correct generated responses were only 1-3 moves in length, the remaining correct responses were long: one response was 52 moves long, three were 53 moves long, and the longest was 61 moves. Given the small sample size, it is difficult to generalize about the transformer's performance. Regardless, the existence of these solutions suggests the transformer may have learned certain solving patterns present in the Kociemba algorithm.

A video comparing Rubik's Cube solutions is found online [21]. Figure 7 compares the Kociemba algorithm (right) to the transformer solution (left) at the same time step. The algorithm solution shows a quarter turn before reaching the end with all six aligned colored faces after 71 steps. The transformer generates 64 steps before reaching the token limit (1024) for generated text outputs as an inherent GPT-2 limit. To illustrate the sparse rewards, neither the algorithmic nor transformer solution capitalizes on a partial reward, such as solving one color for a face or multiple faces in an intermediate step. The transformer did, however, occasionally solve for single rows and columns in instances where it was unable to solve the puzzle before reaching the token limit. An example of the Rubik's Cube transformer solving for rows and columns is shown in Figure 6.

3.2. Sudoku Solver

Figure 8 shows the GPT-2 gameplay for Sudoku from a randomly selected initial state to a partial (but flawed) final solution. The orange diamonds show the repeated digits as errors in completing the square with unique numbers both in the interior square and the overall rows and columns. A validation algorithm that checks for repetitions (1-9) in every row, column, and sub-square could potentially serve as an overlay on generated text games, much in the same way that Chess game generators playing against humans filter out invalid moves. Because GPT-2 models include the

training text formatting in their transformer architecture, the Sudoku training set may benefit from the native grid or matrix rather than string input which masks the sub-grid orientation. The resulting transformer would generate complete puzzle grids rather than require additional visualizations as shown in Figure 8 for a console (command-line) player.

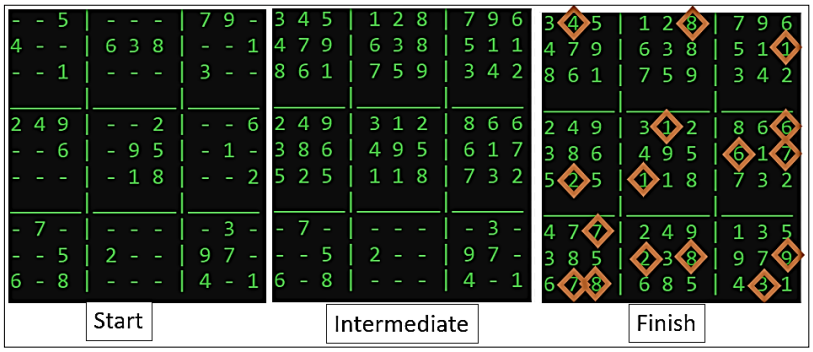


Figure 8. Sudoku Solution Stages using GPT-2

3.3. Maze Solver

Figures 5 and 9 show transformer solutions to the 5x5 (Fig. 5) and 4x4 (Fig. 9) maze sizes. Unlike the Sudoku case, the maze training set preserves formatting for its basic maze grid without removing all end-of-line breaks as a single string. In this way, the maze resembles a narrative paragraph versus the Sudoku sentence format. The trained transformer outputs both a viable unsolved maze and its proposed solution as a pair bracketed by starting and ending delimiters. Since all outputs are generated unconditionally and without a prompt for a starting maze, the output appears as both a scenario generator (viable unsolved maze) and a solution generator (moves to complete the puzzle). Given the token limit of 1024 for generated text, the proposed maze sizes stop at 6x6 grids if the formatting is 4 spaces per grid as shown in Figure 9 and if the unconditional output includes both the starting maze and its paired solution. If a prompt or conditional model is run, the maze sizes naturally extend but the combinatorial moves limit the solution's viability.

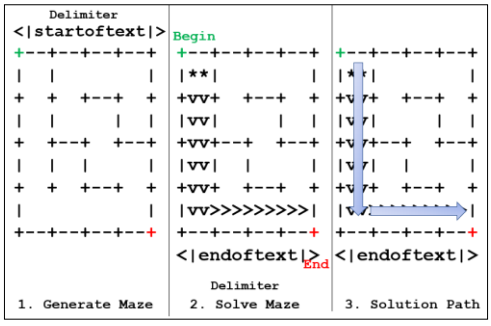


Figure 9. Transformer solution to text mazes in 4x4 size

4. DISCUSSION

Many other games with sparse reward signals have received attention from the reinforcement learning community, including Sokoban, Montezuma's Revenge, and Mountain Car [13,22]. Unlike these games, both Rubik's Cube and Sudoku are well-suited to the application of text generators because they conveniently allow for the examination of sparse rewards problems from

within the confines of games with sequential play and discrete representations of state. Additionally, for both games, deterministic (search) algorithms can provide sufficient quantities of training data such that traditional deep learning techniques, e.g. CNNs, can solve them. Compared to denser reward games, the maze, Rubik's, and Sudoku puzzles require considerable exploration across a flat fitness or optimization landscape. In the case where a solution might take more computing resources to iterate exploratory steps, the attention mechanism behind GPT-2 offers a method to attack the contextual problem of knowing where the numbers or colored faces might relate to each other in the constrained volume of the cube or number squares. Figure 10 illustrates the Sudoku weights for layer 9 as an example of long-term attention and context between a starting number and its long-range dependencies. However, the transformer's ability to solve beyond the 1024 token limit of generated solutions limits the exploration to easier game starting points only. No transformer output for either game achieved a finished state from an arbitrarily random ("hard scrambled") state in the allotted number of steps. Instead, the transformer trained on nearly completed states (e.g. perturbed from a finished state) showed promise in accomplishing its goal to solve the puzzles. Just as with the chess and Go Transformers, the goal of generating plausible gameplay shows possible application but succeeds with supervision and filtering of illegal actions. The secondary goal of demonstrating rule-acquisition (plausible moves) suggests that explicit human knowledge of strategies or heuristics may not be needed specifically for opening or closing moves when the completion times fall within the attention limit of the transformer's context.

Well-known techniques in reinforcement learning emphasize turning a sparse reward game into a denser environment. These approaches feature human domain expertise to craft heuristics, such that the exploration space shrinks or partial rewards provide a stepping stone to reach the solution. A simple example would be solving a maze problem by recursive backtracking or applying the right-hand rule [23]. In the case of Rubik's Cube solvers, many intermediate steps might qualify as partial rewards, such as the layered method, cross, or daisy creations [24]. As a bookkeeping strategy, human Sudoku solvers favor keeping track of which numbers are still possible for each square, thus iteratively narrowing the search space. The hard-coding of such heuristics however ranges outside the scope of the transformer architecture and its powerful capabilities to take raw text games as its only input without domain knowledge when fine-tuned to a new text source and format.

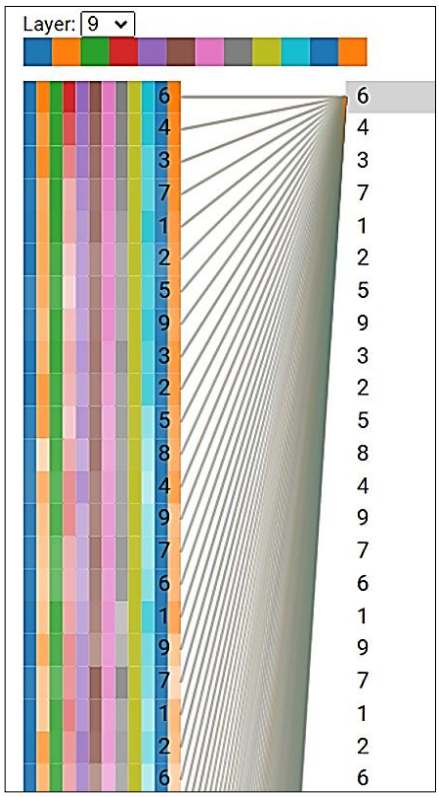


Figure 10. Layer visualization of long-range dependence for a single Sudoku game

One intriguing outcome of exploring transformers with sparse rewards is to suggest new approaches. The attention mechanism itself builds in overweighted connection strengths across longer-range contexts, a critical feature for language models. Ironically, one can posit that attention weights create a sparse reward landscape appropriate for generating interesting narrative text since a frequency-based word approach emphasizes common but less telling words (such as stop words “a”, “the”, etc.). In this way, attention-based models effectively balance the training dataset based on token interest and context rather than frequency. For games, the reinforcement learning community similarly maps flat gradient landscapes to maximize the ratio of rewarding exploitation steps compared to fruitless exploration ones. A simple strategy in sparse rewards substitutes “curiosity-driven” exploration, such that incremental rewards appear when going to points previously not visited. In Sudoku, one can imagine a similar exclusion priority or constraint geared towards not aimlessly substituting [1-9] digits when a row, column, or sub-square already has it. This approach prioritizes a restricted action. In the linguistic origins of GPT-2, the same reward or weight structure might favor novel word choices to avoid repetitive phrases.

The capability of transformers and other text generation methods to play games extends far beyond mazes, Rubik's Cube, and Sudoku. Previous research has highlighted their potential to generate plausible moves for other games which have historically served as benchmarks for game-playing algorithms, notably Chess [5] and Go [6]. Other board games and puzzles offer additional angles from which to examine environments with sparse reward signals (Figure 11). Hex, a board game that has previously drawn attention from the AI community, is one such game. Like Rubik's and Sudoku, it is a perfect information game where the only obvious reward signal is triggered after the final, game-winning move. Unlike Rubik's and Sudoku, Hex is a competitive, 2-player game. It is also amenable to Smart Game Format (SGF), a common

standardized notation for the textual representation of game states. Other candidate games and puzzles include TwixT, which is similar to Hex in both game layout and objective, and Tantrix, which offers sparse rewards in a competitive setting with more than 2 players.

5. CONCLUSIONS

Without encoding puzzle heuristics, the application of GPT-2 can generate viable moves in three sparse reward games: mazes, Rubik's Cube, and Sudoku. These examples offer a novel text-based method to learn plausible moves without human instruction, heuristics, or explicit domain-specific rulesets. These puzzles provide appealing visualization environments to track algorithmic progress incrementally and score winning strategies, identify novel solutions, and augment the traditional black-box understanding inherent in large-scale transformers. Just as attention-based methods provide long-range context, future efforts for improving transformers in gameplay should emphasize larger token limits (>2048 in GPT-3) or condensed game notations for archives.

ACKNOWLEDGEMENTS

The authors would like to thank the PeopleTec Technical Fellows program and the Internship Program for encouragement and project assistance.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[2] Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509.
[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9. https://github.com/openai/gpt-2
[4] Woolf, Max, (2020), GPT-2-Simple, a Python Package, https://github.com/minimaxir/gpt-2-simple
[5] Noever, D., Ciolino, M., & Kalin, J. (2020). The Chess Transformer: Mastering Play using Generative Language Models. *arXiv preprint arXiv:2008.04057*.
[6] Ciolino, M., Noever, D. & Kalin, J. (2020). The Go Transformer: Natural Language Modeling for Game Play. arXiv preprint arXiv:2007.03500.
[7] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354-359.
[8] Demaine, E. D., Eisenstat, S., & Rudoy, M. (2017). Solving the Rubik's Cube Optimally is NP-complete. *arXiv preprint arXiv:1706.06708*.
[9] Darbandi, A., & Mirroshandel, S. A. (2020). A Novel Rubik's Cube Problem Solver by Combining Group Theory and Genetic Algorithm. *SN Computer Science*, *1*(1), 1-16.
[10] Felgenhauer, B., & Jarvis, F. (2006). Mathematics of sudoku I. *Mathematical Spectrum*, *39*(1), 15-22.
[11] Rajput, V. (2021) Deep Learning model compression, Medium, https://medium.com/codex/reducing-deep-learning-size-16bed87cccffRider
[12] Gaddam, D. K. R., Ansari, M. D., & Vuppala, S. (2021). On Sudoku Problem Using Deep Learning and Image Processing Technique. In *ICCCE 2020* (pp. 1405-1417). Springer, Singapore.
[13] McAleer, S., Agostinelli, F., Shmakov, A., & Baldi, P. (2018). Solving the Rubik's cube without human knowledge. *arXiv preprint arXiv:1805.07470*.
[14] Liu, L., Liu, X., Gao, J., Chen, W., & Han, J. (2020). Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*.
[15] Kociemba, H. (2019) Cube Explorer, http://kociemba.org/download.htm
[16] Rider, C. (2017), Visual Cube, http://cube.rider.biz/visualcube.php
[17] Liaw, W., (2021) PyCuber: Rubik's Cube package in Python, https://github.com/adrianliaw/PyCuber
[18] Park, K., (2016), "1 million Sudoku games", Kaggle.com, https://www.kaggle.com/bryanpark/sudoku

[19] Rosettacode.org, “Maze solving” task, Accessed (2021), see https://rosettacode.org/wiki/Maze_solving
[20] Sinck, A. (2016) ASCII Art Maze Solver, https://github.com/asinck/Ascii-Art-Maze-Solver
[21] Noever, D. (2021) Cube Animation Solutions, https://deeperbrain.com/demo/rubix_transformer.mp4
[22] Moore, A. W. (1990). Efficient memory-based learning for robot control.
[23] Roberts, E. Recursive Backtracking, Stanford Computer Science, CS 106B, https://cs.stanford.edu/people/eroberts/courses/cs106b/handouts/16-RecursiveBacktracking.pdf
[24] Youcandothecube.com (accessed 2021), Rubiks Cube Solution, https://www.youcandothecube.com/videos/rubiks-cube-video-solution

AUTHORS

David Noever has 27 years of research experience in machine learning and data mining. He received his Ph.D. from Oxford University, as a Rhodes Scholar, in theoretical physics and B.Sc. from Princeton University. While at NASA, he was named 1998 Discover Magazine's "Inventor of the Year," for the novel development of computational biology software and internet search robots, culminating in co-founding the startup company cited by Nature Biotechnology as first in its technology class. He has authored more than 100 peer-reviewed scientific research articles and book chapters. He also received the Silver Medal of the Royal Society, London, and is a former Chevron Scholar, San Francisco.

Ryerson Burdick is a researcher in the Gemstone Honours Program, University of Maryland, College Park USA. In 2022 he will receive his bachelor’s degree in computer science with a minor in neuroscience. His research focuses on the intersection of human and artificial intelligence, including natural language processing, ethical AI, computer vision, and structured output learning. His work contributes to the growing field of data-driven psychiatric diagnosis and ultimately work towards reducing misdiagnosis and bringing about targeted treatment.

SUB-IMAGE HISTOGRAM EQUALIZATION USING COOT OPTIMIZATION ALGORITHM FOR SEGMENTATION AND PARAMETER SELECTION

Emre Can Kuran[1], Umut Kuran[2] and Mehmet Bilal Er[2]

[1]Department of Software Engineering,
Bandırma Onyedi Eylül University, Balıkesir, Turkey
[2]Department of Computer Engineering, Harran University, Şanlıurfa, Turkey

ABSTRACT

Contrast enhancement is very important in terms of assessing images in an objective way. Contrast enhancement is also significant for various algorithms including supervised and unsupervised algorithms for accurate classification of samples. Some contrast enhancement algorithms solve this problem by addressing the low contrast issue. Mean and variance based sub-image histogram equalization (MVSIHE) algorithm is one of these contrast enhancements methods proposed in the literature. It has different parameters which need to be tuned in order to achieve optimum results. With this motivation, in this study, we employed one of the most recent optimization algorithms, namely, coot optimization algorithm (COA) for selecting appropriate parameters for the MVSIHE algorithm. Blind/referenceless image spatial quality evaluator (BRISQUE) and natural image quality evaluator (NIQE) metrics are used for evaluating fitness of the coot swarm population. The results show that the proposed method can be used in the field of biomedical image processing.

KEYWORDS

Contrast Enhancement, Coot Optimization Algorithm, Knee X-Ray Images, Biomedical Image Processing.

1. INTRODUCTION

There are factors that affect the quality of the image such as contrast, noise and illumination. Contrast is the difference between darkest and brightest parts belong to an image. Hence, higher contrast makes the image regions more separable and it is important to enhance the distorted contrast [1]. Contrast enhancement is significant in the field of computer vision and used for several applications such as retinal image enhancement [2], underwater image enhancement [3] and chest x-ray enhancement [4]. Researchers proposed many techniques for enhancing the contrast of images. Histogram equalization (HE) is one of the most simplest methods proposed [5]. It simply increases the dynamic range of an image by redistributing pixel intensities. For this purpose, HE makes use of probability density functions (PDF) and cumulative distribution functions (CDF). It first computes the image histogram. After calculating the values for PDF and CDF functions according to the histogram of the image, it applies transformation on the output image using the appropriate transformation function. Different HE methods based on the classical HE is proposed in order to overcome limitations of the HE. Brightness preserving bi-histogram

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 33-46, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120903

equalization (BBHE) separates image histogram into two parts according to the mean of the histogram and equalizes these parts independently [6]. Equal area dualistic sub-image histogram equalization (DSIHE) does not make use of mean but median using the same methods as BBHE [7]. Recursive sub-image histogram equalization (RSIHE) also uses median of the histogram, but it continues to separate image histogram until achieving a certain recursive level [8]. Exposure based sub-image histogram equalization (ESIHE) considers bright and dark regions separately for HE [9]. Mean and variance based sub-image histogram equalization (MVSIHE) [10] divides image histogram into four regions according to mean and variance difference, and employs a delta parameter to fuse input image and output image.

Algorithms like HE, BBHE, DSIHE, RSIHE and ESIHE suffers from different artifacts that occur in the output image but MVSIHE is one of the most outstanding methods among other HE techniques according to [11], [12]. Although it has a relatively good performance, its performance mainly depends on the delta parameter, which controls the rate of the image fusing [10]. Since contrast enhancement algorithms are widely employed in the field of medical imaging and biomedical image processing [13]–[15], we tried to provide optimal image quality for the knee x-ray images in this study. We have selected delta parameter and segmentation thresholds of MVSIHE algorithm using coot optimization algorithm (COA) [16]. Metaheuristic algorithms try to minimize/maximize value of the determined fitness function with respect to the problem type. For the fitness function, we have employed Blind/referenceless image spatial quality evaluator (BRISQUE) [17] and Natural Image Quality Evaluator (NIQE) [18] which are used for measuring image quality. These metrics are robust and insensitive to changes, since they are trained on wide variety of images. Besides that they are capable of assessing different kinds of distortions in the image. COA is also another novel optimization algorithm with a good performance as it is claimed in the original study. As it is pointed out in [11], MVSIHE preserves main brightness of the resultant image and also does not cause artifacts, however, we can't ensure that we found the optimum solution. Hence, we focused on improving its performance via parameter selection using COA. Rest of this paper is organized as follows. In Section 2, the used materials and methods are explained. In Section 3, experimental results are given and the proposed method is discussed with its advantages and disadvantages. In Section 3, a conclusion is made.

2. MATERIALS & METHODS

2.1. Coot Optimization Algorithm

The COA is a novel optimization algorithm proposed in [16], which is inspired from behavior of the coot birds. COA tries to simulate collective behaviors of the coots. The coots are directed by a few coots on the water surface. They have four distinct behaviors from observations: random movement, chain movement, position adjusting with respect to the group leaders and leading the group towards optimal area. We need a mathematical model to implement these behaviors.

First of all, a random population of coots is generated at the beginning. Assume that we have a multi-dimensional problem need to be solved for D dimensions, a population of N coots can be generated using Equation 1.

$$\text{PosCoot}(i)=\text{random}(1, D)\times(UB-LB)+LB, \quad i=1, 2, \dots, N \tag{1}$$

In Equation 1, the position of the coots in multi-dimensional space is generated randomly, with respect to the upper bounds UB and lower bounds LB that determined for each dimension.

Hence, the coots are prevented to overflow or underflow these limits. This initial random population is also evaluated according to a selected fitness function given in Equation 2.

$$F(i)=Fitness(PosCoot(i)),\quad i=1, 2, \dots, N \tag{2}$$

In order to model random movement of coots, first, a random position is produced according to the Equation 3. As the second step, the new position of the coot is computed according to the Equation 4.

$$R=random(1, D)\times(UB-LB)+LB \tag{3}$$

$$PosCoot(i)=PosCoot(i)+A\times RN2\times(R-PosCoot(i)) \tag{4}$$

In Equation 4, RN2 is a random number in the range of [0, 1]. A and B are determined according to the Equation 5:

$$A=1-\left(T(i)\times\frac{1}{IterMax}\right),\ B=2-\left(T(i)\times\frac{1}{IterMax}\right)\quad i=1, 2, \dots, IterMax \tag{5}$$

In Equation 5, T(i) is the current iteration, IterMax is the maximum number of iterations. In order to move a coot towards another coot to implement chain movement, average position of the two coots is employed as given in Equation 6.

$$PosCoot(i)=0.5\times(PosCoot(i-1)+PosCoot(i)) \tag{6}$$

Coots also select a leader coot and follow them using Equation 7:

$$L_{ind}=1+(i\,MOD\ N_L) \tag{7}$$

In Equation 7, L_{ind} is the index of the leader and N_L is the number of leaders that determined as a parameter. A probability p is also defined. Finally, the rules given in Equation 8 is employed for determining leader positions.

$$LeaderPos(i)=\begin{cases}B\times R3\times\cos(2R\pi)\times(gBest-LeaderPos(i))+gBest & R4<P\\ B\times R3\times\cos(2R\pi)\times(gBest-LeaderPos(i))+gBest & R4\geq P\end{cases} \tag{8}$$

In Equation 8, R3 and R4 are random numbers in the range of [0, 1], gBest is the current global best, π is 3.14. Pseudocode of the COA is given in Figure 1.

```
Initialize the first population of coots randomly by Equation 1
Initialize the termination criteria, probability p, number of leaders and
number of coots
Ncoot=Number of coots-Number of leaders
Random selection of leaders from the coots
Calculate the fitness of coots and leaders
Find the best coot or leader as the global optimum while the end criterion is
not satisfied
  Calculate A, B parameters by Equation 5
  If rand< P
    R, R1, and R3 are random vectors along the dimensions of the problem
  Else
    R, R1, and R3 are random number
  End
  For i=1 to the number of the coots
    Calculate the parameter of K by Equation 7
    If rand>0.5
      Update the position of the coot by Equation 8
    Else
      If rand<0.5 i~=1
        Update the position of the coot by Eq 6
      Else
        Update the position of the coot by Eq 4
      End
    End
    Calculate the fitness of coot
    If the fitness of coot < the fitness of leader(k)
    Temp=leader(k); leader(k)=coot; coot=Temp;
    end
  End
  For number of Leaders
      Update the position of the leader using the rules given in Equation 8
    If the fitness of leader < gBest
      Temp= gBest; gBest =leader; leader=Temp; (update global optimum)
    end
  End
  Iter=iter+1;
end
Postprocess results
```

Figure 1. Pseudocode of the COA.

2.2. Mean and Variance based Sub-Image Histogram Equalization Algorithm

MVSIHE algorithm can be divided into 5 stages in the following order: histogram segmentation, histogram bin modification, histogram equalization, normalization and image fusing.

2.2.1. Histogram Segmentation

Firstly, input image histogram is divided into two sub-histograms using a threshold k. The probability density function (PDF) of these two parts are computed. Then, for the first separation level k, two variables namely ω_0 and ω_1 can be given as in Equation 9.

$$\omega_0=\sum_{i=0}^{k} \mathrm{PDF}(i), \quad \omega_1=\sum_{i=k+1}^{I_{max}} \mathrm{PDF}(i) \tag{9}$$

In Equation 8, i is the processed intensity level and I_{max} is the maximum intensity level that is

possible (256 for 8-bit). Mean of each part μ_0 and μ_1 can be given as in Equation 10.

$$\mu_0=\sum_{i=0}^{k} PDF(i), \quad \mu_1=\sum_{i=0}^{I_{max}} PDF(i) \tag{10}$$

In Equation 9, I_{max} is the maximum intensity level and i is the intensity level. Whole image mean can be given as in Equation 11 and variance of the two parts can be defined by Equation 12, respectively.

$$\mu_1=\mu_0\omega_0+\mu_1\omega_1 \tag{11}$$

$$\sigma^2(k)=\omega_0(\mu_0-\mu)^2+\omega_1(\mu_1-\mu)^2 \tag{12}$$

The MVSIHE algorithm finds the maximum value of the variance σ^2. After finding optimum threshold k_{opt} (or k_{H2}), same procedure from Equation 9 to Equation 12 is repeated for two distinct histograms. Threshold of the lower sub-histogram, namely, k_{H1} and threshold of the upper sub-histogram, namely, k_{H3}, are employed to determine other separation points. Thus, the segmented histogram is given in Equation 13 with its four sub-histograms.

$$H[I_{lowb},I_{upb}]=\bigcup_{i=1}^{4} sub^{i,4}[I_{lowb},I_{upb}] \tag{13}$$

In Equation 13, I_{lowb} is the lower bound intensity level, I_{upb} is the upper bound intensity level for the four sub-histograms.

2.2.2. Histogram Bin Modification

First of all, PDF for the sub-histogram i can be expressed as in Equation 14.

$$PDF_{sub^{i,4}}=\frac{sub^{i,4}}{n_{i,4}} \tag{14}$$

In Equation 14, $n_{i,4}$ is the number of pixels in the sub-histogram i. MVSIHE applies a histogram bin modification to overcome the domination of high frequency intensity levels and to balance high frequency and low frequency intensity levels [16]. Histogram bin modification is given by Equation 15.

$$MODIFIED\_PDF_{sub^{i,4}}=\left(\frac{e^{PDF_{sub^{i,4}}}-e^{-PDF_{sub^{i,4}}}}{e^{PDF_{sub^{i,4}}}+e^{-PDF_{sub^{i,4}}}}\right) \tag{15}$$

In Equation 15, e is the exponential function. Cumulative distribution function (CDF) of each sub-histogram is then calculated using Equation 16.

$$CDF_{sub^{i,4}}(x)=\sum_{j=I^{i,4}_{lowb}}^{I^{i,4}_{upb}} MODIFIED\_PDF(j), \qquad \text{for } x=I^{i,4}_{lowb}, \ldots, I^{i,4}_{upb} \tag{16}$$

2.2.3. Histogram Equalization

HE is applied to each sub-histogram separately instead of global HE. A transformation function which considers the upper and lower boundaries of the sub-histogram is used in this case. Hence, a scaled HE is made and each of the intensity levels is equalized in its own range. Equation 17 is employed for this purpose.

$$f_{sub^{i,4}}(x)= I^{i,4}{}_{lowb}+\left(I^{i,4}{}_{upb}- I^{i,4}{}_{lowb}\right)\times CDF_{sub^{i,4}}(x), \qquad \text{for } x=I^{i,4}{}_{lowb}, \dots , I^{i,4}{}_{upb} \tag{17}$$

After each sub-histogram is equalized, they are merged to generate the final image.

2.2.4. Normalization

Since the transformation is applied distinctly for each sub-histogram, brightness saturation and artifacts might occur when the distinct histograms are merged, due to non-uniform illumination. Therefore, a normalization is applied on the image according to the Equation 18:

$$T(X)=\frac{X-X_{min}}{X_{max}-X_{min}}\times\left(X_{upb}-X_{lowb}\right)+X_{lowb} \tag{18}$$

In Equation 18, X is the input image which is actually the output of Equation 17, X_{min} is the minimum intensity level in X, X_{max} is the maximum intensity level in X, X_{upb} is the upper bound (which is 255 for 256 levels) and X_{lowb} is the lower bound (0).

2.2.5. Image Fusing

The original input image and output of Equation 18, are fused in this stage with the aim of preserving more information in the final image. The δ parameter is selected for determining the fusing rate, which is in the range of [0, 1], it determines the domination of the input image to the resultant image and vice versa. The fusing is done according to Equation 19.

$$O=\delta\times I_N+(1-\delta)\times I \tag{19}$$

In Equation 19, O is the final image, I_N is the normalized image obtained using Equation 18, and I is the input image, respectively.

2.3. Proposed Method

The proposed method consists of three stages as follows: Determination of COA parameters and defining fitness function, selection of MVSIHE parameters via COA and enhancing the image using the selected parameters. The optimum thresholds for segmentation (k_{H1}, k_{H2}, k_{H3}) and delta (δ) parameter are selected by COA which employs the fitness function given in Equation 26. The illustration of the proposed method is given in Figure 2.

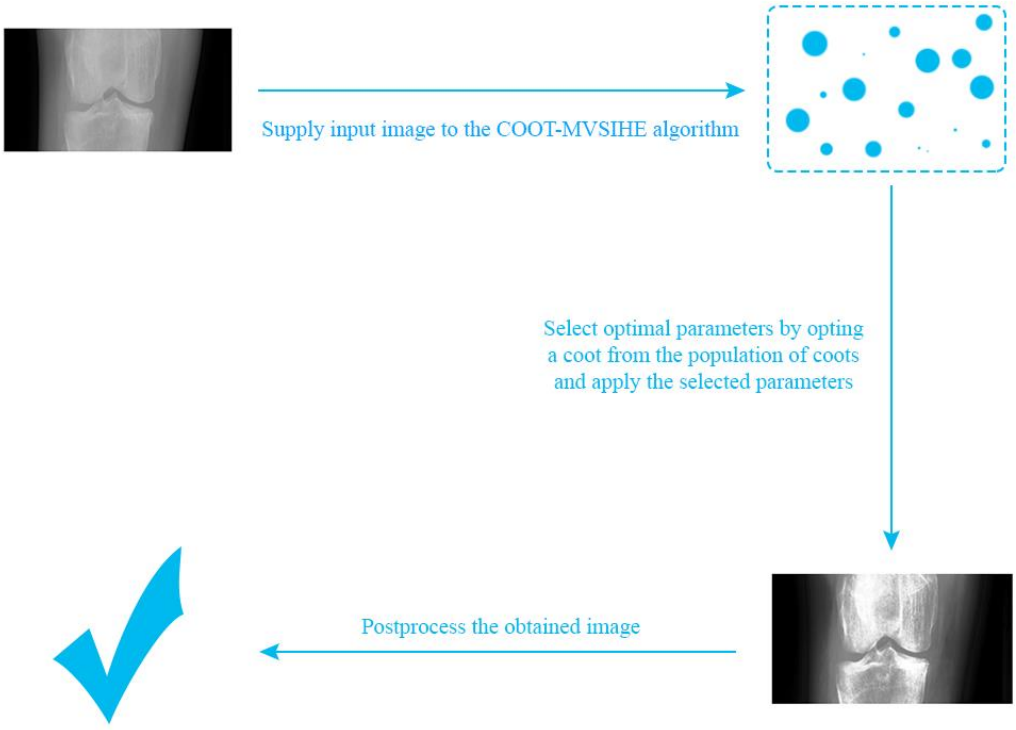


Figure 2. Illustration of the proposed method.

2.3.1. Determination of Coot Optimization Algorithm Parameters

Like in other metaheuristic optimization algorithms, parameters and fitness function of COA need to be determined. Parameters can be selected experimentally. In this study, we selected parameters to be reasonable as possible for an image enhancement task. We opted maximum number of iterations and population size as 10 to be relatively small as compared to some other optimization tasks in the literature since image enhancement is a long process that depends on the image size and bit depth. Upper bounds and lower bounds are specified in order to prevent COA to exceed limits. Number of dimensions is selected according to the problem type. Since we select delta parameter, and three histogram thresholds for MVSIHE, it is defined as four. The selected parameters are given in Table 1 for further information.

Table 1. Parameters of COA.

Parameter (s)	**Value (s)**
Upper bounds	[0, 0, 51, 151]
Lower bounds	[1, 50, 150, 255]
Number of dimensions	4
Maximum number of iterations	10
Population size	10
Number of leaders	[Population Size * 0.1]
Probability	0.5

2.3.2. Fitness Function

The aim of fitness function is to converge to the optimum solution. Each member of the population is evaluated using the determined fitness function with respect to the problem type. Because it is tried to enhance images using COA in this study, fitness function should be opted such that minimum distortion occurs in the resultant images. With this motivation, we selected BRISQUE and NIQE, which are developed to measure quality without needing any reference for various kind of distorted images.

BRISQUE [17] comprises three stages: extracting natural scene statistics (NSS), computing feature vectors and training support vector machines (SVM) [19] for predicting image quality scores. It is known that distribution of the distorted normalized images are different from the distribution of more natural normalized images. Distribution of the relatively less distorted images usually follow a bell curve therefore deviation from this curve can be perceived as a sign of distortion. Mean Subtracted Contrast Normalization (MSCN) is employed in BRISQUE in order to normalize an image. First, to calculate MSCN coefficients, the image is transformed to a luminance matrix as given in Equation 20:

$$\hat{I}(i,j)=\frac{I(i,j)-\mu(i,j)}{\sigma(i,j)+C}, \quad i=1, 2, \ldots, M, \; j=1, 2, \ldots, N \tag{20}$$

In Equation 20, I is the image, M and N are height and width of the image, i and j are spatial coordinates in x-axis and y axis, respectively. C is a small constant that is added for making sure the denominator is not equal to zero. The μ is the local mean field whereas σ is the local variance field. Assume that GB is the Gaussian blur window (GBW) and I is the image, we can compute μ by using Equation 21 and σ by employing Equation 22.

$$\mu=GB*I \tag{21}$$

$$\sigma=\sqrt{GB*(I-\mu)^2} \tag{22}$$

Since distortion also depends on the relationship of the pixels, pair-wise product of MSCN image with a shifted MSCN image is calculated. These pair-wise product images are horizontal, vertical, left-diagonal and right-diagonal. In the second step, the 5 images obtained in the first step (MSCN and pair-wise product images) are employed for feature extraction. MSCN image is fitted to a generalized Gaussian distribution (GGD) whereas pair-wise product images are fitted to an asymmetric generalized Gaussian distribution (AGGD). GGD has two parameters (shape and variance) and AGGD has four parameters (shape, mean, left variance and right variance). Hence, 2 features are obtained from MSCN image and 16 features (4 x 4) are obtained from pair-wise product images. The image is downsized by two (half of its original size), the same feature extraction technique is repeated and thus 36 features are gathered. In the last step of BRISQUE, feature vectors as inputs and their quality scores as outputs are fed to SVM and a model is trained. This model is used for predicting image quality score afterwards.

NIQE [18] consists of five phases: extracting NSS, patch selection, patch characterization, fitting patches to the multivariate Gaussian model (MGM) and applying NIQE index. For the first phase, NSS extraction, the process is similar to the NSS extraction in BRISQUE except that NIQE is only trained on the natural images but not distorted images. Hence, NIQE does not depend on any particular distortion type. In the second phase, the image is divided into P × P patches and patches exceeding a threshold T (which is defined as 0.75 in the original study) are selected according to the average local deviation field δ_L given in Equation 23.

$$\delta_L(t)=\sum\sum_{(i,j)\in patch}\sigma(i,j),\quad t=1, 2, \ldots, N_P \tag{23}$$

In Equation 23, i and j are the spatial coordinates belong to patch, t is the patch index, N_P is the number of patches, and σ(i, j) is the related variance value. In the third phase, similar to the BRISQUE, GGD and AGGD are employed for fitting and with a downsizing, a total number of 36 features are obtained. In the fourth phase, the NSS features obtained in the previous phases are fitted with an MGM model. MGM density can be given as in Equation 24.

$$f(x_1, \ldots, x_k)=\frac{1}{(2\pi)^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-v)^T\Sigma^{-1}(x-v)\right) \tag{24}$$

In Equation 24, $(x_1, \ldots, x_k)$ are the features, Σ is the covariance matrix and v is the mean. In the last phase, the quality of the distorted images can be computed using Equation 25.

$$D(v_1,v_2,\Sigma_1,\Sigma_2)=\sqrt{\left((v_1-v_2)^T\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{-1}(v_1-v_2)\right)} \tag{25}$$

In Equation 25, v_1 and v_2 are the mean vectors whereas Σ_1 and Σ_2 are covariance matrices of natural MVG model and distorted image, respectively.

We have used BRISQUE and NIQE together by multiplying their outputs. The results of this multiplication is opted as the fitness function for a minimization problem. Because as the BRISQUE and NIQE scores increase, the quality of the evaluated images are decreased. We have used these metrics together to take advantage of the powerful sides of them. The preferred fitness function is defined in Equation 26.

$$F(I)=BRISQUE(I)\times NIQE(I) \tag{26}$$

In Equation 26, I is the input image given to the fitness function F.

2.4. Dataset and Preprocessing

Digital knee x-ray images [20] dataset is employed in the study for enhancing knee x-ray images. The images are obtained using PROTEC PRS 500E x-ray machine. This datasets includes images with labelled severity levels (using Kellgren and Lawrence grades) with the help of two distinct medical experts. In this study, we used the images contained in subfolder "MedicalExpert-I" for evaluating our method, since the images are same and only labels differ, the other subfolder is ignored. The images are first converted to gray before evaluating the proposed method. Further details about the dataset can be found in Table 2.

Table 2. Details of the dataset.

Expert ID	Number of images for severity level					Total number of images	Bit depth	Types of Images
	Normal	Doubtful	Mild	Moderate	Severe			
I	514	477	232	221	206	1650	8-bit	PNG
II	503	488	232	221	206	1650	8-bit	PNG

2.5. Evaluation Metrics

2.5.1. Absolute Mean Brightness Error

Absolute mean brightness error (AMBE) calculates the mean brightness error, hence, a lower AMBE indicates a better brightness preservation. AMBE is defined as in Equation 27.

$$AMBE=|M_I-M_O| \tag{27}$$

In Equation 27, M_I and M_O are the mean values of the input image and the output image respectively. AMBE is defined in the range of [0,255].

2.5.2. Peak Signal-to-Noise Ratio

Peak signal-to-noise ratio (PSNR) is employed for measuring the error between input and output image after an operation that might present distortion. Hence, a higher value of PSNR indicates better image quality and less distortion. In order to compute PSNR, first of all, mean squared error (MSE) is computed using Equation 28. Then, PSNR is calculated using Equation 29.

$$MSE=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j)-O(i,j))^2 \tag{28}$$

$$PSNR=10\log_{10}\frac{(2^n-1)^2}{\sqrt{MSE}} \tag{29}$$

In Equation 28, I and O are input image and output image, respectively, and i and j are the spatial coordinates of x-axis and y-axis, respectively. In Equation 29, n is determined according to image type, that is, for 256 gray levels, it is equal to 8 since 2^8 is equal to 256. Higher PSNR value indicates better quality, which is infinite when MSE is equal to 0.

2.5.3. Structured Similarity Index

Structured similarity index (SSI) is a measures the similarity index by considering the input and output image. Higher SSI value indicates more similarity. SSI value is in the range of [-1, 1]. SSI could be calculated by employing Equation 30.

$$SSI=\frac{(2M_IM_O+C_1)(2\sigma_{IO}+C_2)}{(M_I^2+M_O^2+C_1)(\sigma_I^2+\sigma_O^2+C_2)} \tag{30}$$

In Equation 30, M_I, M_O, σ_I, σ_O, σ_{IO}, σ_{IO}, C_1 and C_2 are mean value of the input image, mean value of the output image, standard deviation of the input image, standard deviation of the output image, first constant and second constant, respectively. The constants are usually selected as small values such that they are close to zero which ensures that numerator and denominator to be greater than zero.

3. RESULTS & DISCUSSION

In this section, visual results, performance evaluation results, stability results and convergence curves are provided. Visual results are important since performance metrics are not enough to compare image enhancement algorithms. Performance evaluation results are also given for an objective assessment. Stability results are provided because the results (outputs) of the

metaheuristic algorithms may change slightly at each run and convergence curves are given to indicate performance of the COA algorithm on the dataset.

3.1. Performance Evaluation Results

The performance evaluation results are given in Table 3 in terms of AMBE, PSNR and SSI. First best results are in bold whereas second best results are underlined. The evaluation results show that our method is capable of enhancing different classes of images in the dataset. Proposed method outperforms most of the other methods compared in this study and exhibits competitive performance as compared to some of them. Although, visual results in Section 3.2 indicate that our method is better at preserving details in the image.

Table 3. Average performance evaluation results for AMBE, PSNR and SSI on 1650 knee x-ray images.

Method	AMBE	PSNR	SSI
HE	22.5258	16.6345	0.6898
BBHE	21.4886	17.5533	0.7266
DSIHE	18.4800	17.3079	0.7021
ESIHE	9.8640	21.1487	0.8525
MVSIHE(0.6)	**5.3629**	<u>24.6394</u>	**0.9185**
COA-MVSIHE	<u>7.9919</u>	**26.2748**	<u>0.9139</u>

3.2. Visual Results

Visual results for the images belong to distinct classes (severity levels) are given in Figure 3 for compared methods. The proposed method provides a balanced enhancement and prevents most of the detail loss as it can be observed from the figure. For example, the enhanced output of the sample output image that belong to the mild class provides more details and higher contrast as compared to the outputs of other methods. Other outputs are also balanced and not too sharp or washed-out (saturated).

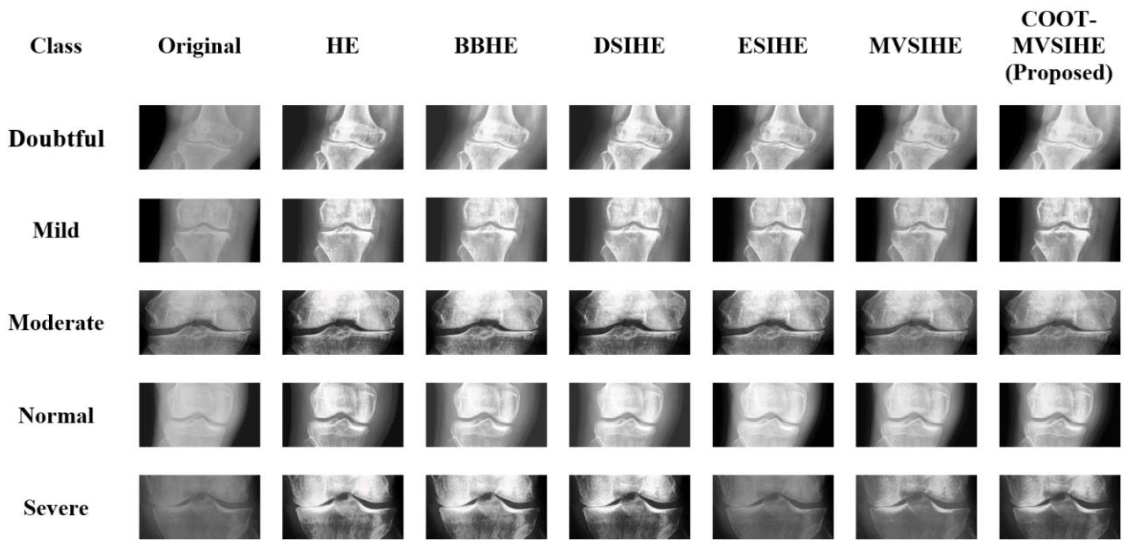
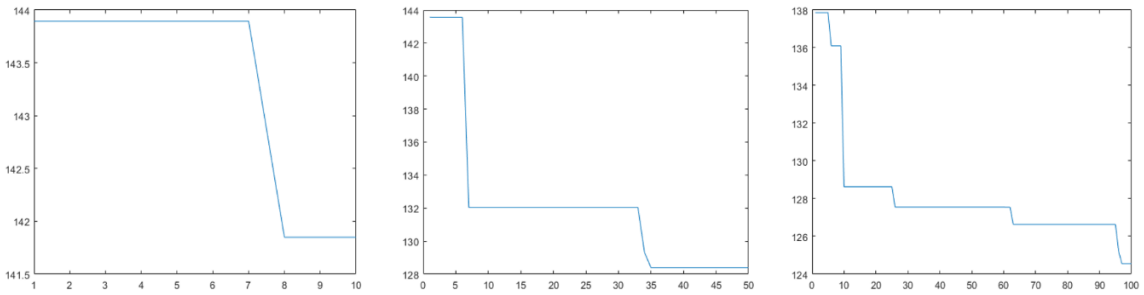


Figure 3. Visual results for images belong to different severity levels (classes) and algorithms.

3.3. Convergence Curves and Stability Results

Convergence curves of the COA algorithm employed for the image enhancement task are given in Figure 4. As it can be seen from the convergence curves that as the number of iterations increases, the COA converges to the minimum (optimum value) better. We can conclude that increasing the maximum number of iterations might increase the chance of finding global optimum.



(a) (b) (c)

Figure 4. Convergence curves for different number of iterations with the same population size of 10: (a) 10 iterations. (b) 50 iterations. (c) 100 iterations.

Since COA is a metaheuristic optimization algorithm, different results might be obtained at different runs. In this study, the stability of the proposed method is tested using 10 sample images by performing 10 runs. The parameters given in Table 1 is used for the test. The stability results are given in Table 4. We used variance, standard deviation and range (difference between maximum and minimum value) for such aim. The standard deviation and variance values is lower than one and close to the zero for AMBE and PSNR. SSI varies relatively higher as compared to the other metrics. Since these metrics show variation of the data, these given results should be considered for the use of proposed method in further studies.

Table 4. Stability results of the proposed method for 10 samples and 10 runs.

Measurement	AMBE	PSNR	SSI
Variance	0,793405	0,000125	2,836652
Standard Deviation	0,845023	0,010618	1,597807
Range	2,56	0,0357	4,8175

3.4. Discussion

The proposed method is adaptive, that is, it can find the optimal parameters for the enhancement of wide variety of images. It does not need any pretrained structure. Its parameters could be changed according to device type for gaining more performance (such as decreasing number of iterations etc.). However, its stability should be considered when a more stable enhancement technique is significant. Also, changing parameters may increase or decrease the success of the enhancement task with respect to the data type.

4. CONCLUSIONS

In this study, a method combining COA with MVSIHE is proposed for enhancement of the knee x-ray images. The experimental results show that our method outperforms most of the other cutting-edge methods or at least shows a competitive performance. Visual results also indicate that our method provides a balanced enhancement for most of the images in the used dataset. Hence, the proposed method can support various supervised models by performing preprocessing task, which might be employed to increase the classification success in further studies.

REFERENCES

[1] A. Mahmood, S. A. Khan, S. Hussain, and E. M. Almaghayreh, "An Adaptive Image Contrast Enhancement Technique for Low-Contrast Images," *IEEE Access*, vol. 7, pp. 161584–161593, 2019, doi: 10.1109/ACCESS.2019.2951468.

[2] K. Aurangzeb, S. Aslam, M. Alhussein, R. A. Naqvi, M. Arsalan, and S. I. Haider, "Contrast Enhancement of Fundus Images by Employing Modified PSO for Improving the Performance of Deep Learning Models," *IEEE Access*, vol. 9, pp. 47930–47945, 2021, doi: 10.1109/ACCESS.2021.3068477.

[3] D. Dev K. and S. Natrajan, "Underwater Image Enhancement for Improving the Visual Quality by CLAHE Technique," *Int. J. Sci. Res. Eng. Technol.*, vol. 4, no. 4, pp. 352–356, 2015.

[4] Z. Liang, J. X. Huang, J. Li, and S. Chan, "Enhancing Automated COVID-19 Chest X-ray Diagnosis by Image-to-Image GAN Translation," 2020, doi: 10.1109/BIBM49941.2020.9313466.

[5] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital Image Processing, Third Edition," *J. Biomed. Opt.*, vol. 14, no. 2, p. 029901, 2009, doi: 10.1117/1.3115362.

[6] Y. T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Trans. Consum. Electron.*, vol. 43, no. 1, pp. 1–8, 1997, doi: 10.1109/30.580378.

[7] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 68–75, 1999, doi: 10.1109/30.754419.

[8] K. S. Sim, C. P. Tso, and Y. Y. Tan, "Recursive sub-image histogram equalization applied to gray scale images," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1209–1221, 2007, doi: 10.1016/j.patrec.2007.02.003.

[9] K. Singh and R. Kapoor, "Image enhancement using Exposure based Sub Image Histogram Equalization," *Pattern Recognit. Lett.*, vol. 36, no. 1, pp. 10–14, 2014, doi: 10.1016/j.patrec.2013.08.024.

[10] L. Zhuang and Y. Guan, "Image Enhancement via Subimage Histogram Equalization Based on Mean and Variance," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–12, 2017, doi: 10.1155/2017/6029892.

[11] S. H. Majeed and N. A. M. Isa, "Adaptive Entropy Index Histogram Equalization for Poor Contrast Images," *IEEE Access*, vol. 9, pp. 6402–6437, 2021, doi: 10.1109/ACCESS.2020.3048148.

[12] S. H. Majeed and N. A. M. Isa, "Iterated Adaptive Entropy-Clip Limit Histogram Equalization for Poor Contrast Images," *IEEE Access*, vol. 8, pp. 144218–144245, 2020, doi: 10.1109/ACCESS.2020.3014453.

[13] S. Bhattacharya *et al.*, "Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey," *Sustain. Cities Soc.*, vol. 65, p. 102589, Feb. 2021, doi: 10.1016/j.scs.2020.102589.

[14] J. Qiu, H. Harold Li, T. Zhang, F. Ma, and D. Yang, "Automatic x-ray image contrast enhancement based on parameter auto-optimization," *J. Appl. Clin. Med. Phys.*, vol. 18, no. 6, pp. 218–223, 2017, doi: 10.1002/acm2.12172.

[15] S. Mohan and T. R. Mahesh, "Particle Swarm Optimization based Contrast Limited enhancement for mammogram images," in *7th International Conference on Intelligent Systems and Control, ISCO 2013*, 2013, pp. 384–388, doi: 10.1109/ISCO.2013.6481185.

[16] I. Naruei and F. Keynia, "A new optimization method based on COOT bird natural life model," *Expert Syst. Appl.*, vol. 183, p. 115352, Nov. 2021, doi: 10.1016/j.eswa.2021.115352.

[17] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012, doi: 10.1109/TIP.2012.2214050.

[18] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, 2013, doi: 10.1109/LSP.2012.2227726.
[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," 1992, doi: 10.1145/130385.130401.
[20] P. S. Gornale and P. Patravali, "Digital Knee X-ray Images," *Mendeley Data*, 2020. https://data.mendeley.com/datasets/t9ndx37v5h/1 (accessed Jan. 12, 2022).

AUTHORS

Emre Can Kuran was born in İzmir, Turkey, in 1997. He received the B.S. degree in computer engineering from Harran University, Turkey, in 2020, and currently is a M.Sc. student in computer engineering at Harran University, Turkey. He is currently working as a research assistant in Software Engineering, Bandırma Onyedi Eylül University, Turkey. His research interests include artificial intelligence, image processing, computer vision, data science, machine learning and software development.

Umut Kuran was born in Şanlıurfa, Turkey, in 1982. He received the B.S. degree in mathematics from Harran University, Turkey, in 2005, the M.Sc. degree in mathematics from Celal Bayar University, Turkey, in 2007, and in computer science from Rutgers, The State University of New Jersey, USA, in 2013. He is currently a Ph.D. student in mathematics, at Harran University. He is currently working as a full time lecturer in Computer Engineering, Harran University, Turkey. His research interests include artificial intelligence, image processing, neural networks, fuzzy logic, numerical computation and cryptology.

Mehmet Bilal Er was born in Şanlıurfa, Turkey, in 1988. He received the B.S. degree in computer engineering from Eastern Mediterranean University, Cyprus, in 2010, the M.Sc. degree in computer engineering from Cankaya University, Turkey, in 2013, and the Ph.D. degree in computer engineering from Maltepe University, Turkey, in 2019. He is currently an Assistant Professor with the School of Computer Engineering, Harran University, Turkey. His research interests include sound processing, pattern recognition, machine learning and deep learning.

INDIVIDUALIZED EMOTION RECOGNITION THROUGH DUAL- REPRESENTATIONS AND GROUP-ESTABLISHED GROUND TRUTH

Valentina Zhang

Phillips Exeter Academy, Exeter NH, USA

ABSTRACT

While facial expression is a complex and individualized behavior, all facial emotion recognition (FER) systems known to us rely on a single facial representation and are trained on universal data. We conjecture that: (i) different facial representations can provide complementing views of emotions; (ii) when employed collectively in a discussion group setting, they enable accurate FER which is highly desirable in autism care and applications sensitive to errors. In this paper, we first study FER using pixel-based DL vs semantics-based DL in the context of deepfake videos. The study confirms our conjectures. Armed with the findings, we have constructed an adaptive FER system learning from both types of models for dyadic or small interacting groups and further leveraging the synthesized group emotions as the ground truth for individualized FER training. Using a collection of group conversation videos, we demonstrate that FER accuracy and personalization can benefit from such an approach.

KEYWORDS

Emotion recognition, facial representations, adaptive algorithm, training data ground truth .

1. INTRODUCTION

In medical practice, emotion recognition is crucial to accurate clinical decision-making [1]. However, there are several obstacles. Patients' emotional behaviors could oftentimes be affected by an underlying condition such as neurodivergence. On the other hand, physicians could be biased by their own emotions and limited by their cognitive ability.

Deep learning (DL) offers a unique value in this context as a DL-based system does not have an emotional bias and could detect patterns and changes too subtle for human cognition in real-time. Yet, there are many challenges for building an accurate DL-based facial emotion recognition (FER) system. Leveraging a small conversing group setting in autism care, our research investigates the potentials of composing DL-based FER systems with dual-representations and automatically deriving the ground truth for quality training data.

This paper hypothesizes that for comparable accuracy rates, pixel-based DL and semantics-based DL sometimes deliver complementing predictions in FER. In the context of small conversing groups, the two types of DL models could be orchestrated to deliver more accurate group and individual emotion recognition. Our adaptive group emotion recognition system includes three components: individual emotion recognition, adaptive group emotion synthesis, and group vs individual emotion modeling. We aim to understand the relationship between pixel-based DL and semantics-based DL and how they could work together to potentially outperform humans in FER.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 47-57, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120904

The main contributions of this paper are (i) a comparatively study of DL models trained with different facial representations; (ii) an adaptive approach toward accurate individual and group FER leveraging discussion group context; and (iii) a proposal to use group emotion as ground truth labels for FER personalization.

The remainder of the paper is organized as follows: In section 2, a few closely related works are presented. In section 3, we outline the two neural networks we used for this work and how their different performances inspired us. Section 4 describes our system architecture, face detection mechanism and working model for adaptive group emotion recognition in detail. Recognizing the established group emotion as a robust ground truth, section 5 outlines how it could be leveraged to improve emotion recognition for individuals and automatic training data labelling. The section then analyzes the test results of our experiment. Section 6 discusses the threats to the validity of our research. Section 7 gives the conclusion and future work of the paper.

2. RELATED WORKS

In the area of individual FER, this work benefitted from studying the pioneering work such as [3][4][14] and holistically understanding their principles and limitations. In the area of group FER, this work has been informed by a diverse set of existing research settings ranging from a four-person UNO game [10] to public crowds [12]. In the area of comparative research in different facial representations and different DL architectures, this work drew its inspirations from [13]. Last but not least, this work relies on [15][16] for the complete and up-to-date survey of all published research works in FER and group FER. With deep appreciation, in this section, we review these representative papers that are most related to and influenced our work.

Alex Krizhevsky et al [14] provides a first detailed description and analysis of architecture and design variables of DL-based image classification. Ian Goodfellow et al [2] has an early yet insightful discussion on the challenges of facial emotion recognition and outlines some important considerations in designing a successful solution. Octavio Arriaga et al [4] explains one of the first and very successful CNN-based individual FER systems. The system does not rely on facial landmarks. Tatsuya Hayamizu et al. [10] is one of the earliest works in group emotion recognition. It also studies a 4-person group, but it relies on classic statistics-based AI techniques instead of DL. Liwei Wang et al. [13], presents a first systematic approach quantitatively characterizing what representations do deep neural networks, and how similar are the representations learned by two networks with identical architecture but trained from different initializations.

3. ANALYSIS OF TWO TYPES OF FER MODELS

When training a DL model for image analysis tasks, there are two general approaches. One is to use the full images as training input data, which is called in this paper as pixel-based DL or image-based training. The other approach is semantics-based, which extracts semantics from the images and uses extracted features such as facial landmarks as the training data. We choose to comparatively study these two different approaches as the human emotion recognition system operates similarly.

In the human brain, a section called the fusiform face area looks at the whole face holistically, which is similar to how a model trained on full images would function. On the other hand, a part of the human brain called the occipital face area recognizes the eyes, nose, and mouth as individual pieces, which would be very similar to a model trained on facial landmarks. The landmark-based training is generally considered more effective because it helps the neural

networks to focus on the essence of the problem, which is the outline of a face. In facial emotion recognition where the outlines to emotion mapping may not be well defined [2], we assume that image-based training may outperform landmark-based training in some cases. This section explains our experiment for validating our assumption.

Our first model is a standard fully-convolutional neural network composed of 60 convolutional and separable convolutional layers, ReLUs, Batch Normalization, Dropout, Flatten, and Global Average Pooling layers. It is trained with the ADAM optimizer, and achieved a validation accuracy of 60% on the FER-2013[3] dataset.[4]

Our second model is trained on facial landmarks which are extracted by solving the shape prediction problem. In this approach, each face consists of a few shapes which outline the face, eyes, mouth, and nose. Through extracting these shapes, this model will ignore the other features and details of the face, which may have introduced noise into the data of pixel-based training. We use a standard 68-point facial landmark system. There are 6 landmarks for each eye, 9 for the nose, 20 for the lips, and the remaining 27 outline the face.

Landmark-based training uses information about these 68 landmarks on the face to correlate the shape of these landmarks with a certain emotion. Four pieces of information from each landmark is extracted: the x and y coordinates, its distance from the mean of all the points, and its vector angle. Collectively, this representation of the 68 landmarks gives a comprehensive summary of facial features. Our second model is a simple neural network consisting of two hidden layers of 128 neurons using the Rectified Linear Unit activation function, and Adamax optimizer. This produces a validation accuracy of 58%.

For our experiment, we analyzed a well-known deepfake video of Robert Downey Jr. created from a speech by Elon Musk using DeepFaceLab 2.0 Quick96 at 1 million iterations. We apply our two trained models to every frame of the video and produce a probability value for each of our seven core emotions. Subsequently, we correlated the results from both models and constructed correlation heatmaps as shown in Fig. 1.

The correlation analysis reveals some interesting results. On the two heatmaps, the correlation between the original and deepfake videos' corresponding emotions is shown as a diagonal orange line of square cells. Our data shows that the landmark-based training model detected a much higher correlation between the original and the deepfake, even in the "Disgust" and "Surprise" components, which are very minimal in the videos.

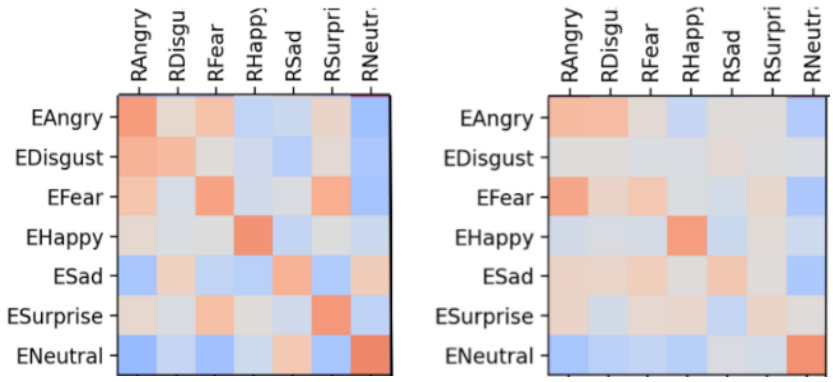


Fig. 1. Emotion Correlation by Image-Trained (left) vs. Landmark-Trained Model (right)

We believe that this is due to the fact that subtle facial expressions are difficult to accurately capture through facial landmarks. On the other hand, the image-based training model detected details that were lost during the creation of the deepfake. To confirm this belief, we employed three human evaluators of the video. Given the manually labeled video from each of the three evaluators, we computed the two-judges agreement to obtain the true labels (e.g., a label was marked as a true positive if at least two of the three evaluators classified it as such). As our assumption predicted, the emotion ground truth on the two faces, as shown in Fig. 2, do not match well. Generally, Elon Musk appeared to have many more positive emotions than the deepfake did.

Our manual analysis also confirmed some subtle or mixed emotions lost in the deepfake video.

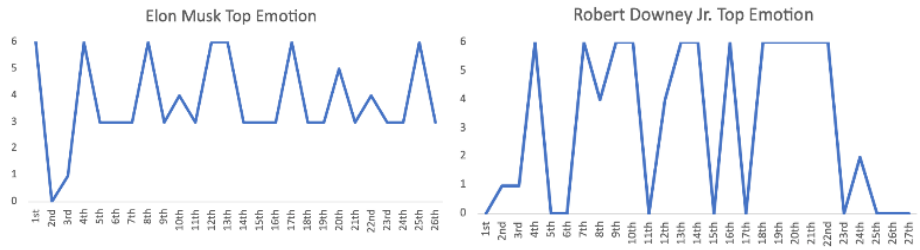


Fig. 2. Ground Truth for Deepfake (left) vs. Original (right)

To analyze our assumption that our two models have complementing strengths in detecting facial emotions, we constructed the complementarity metrics for the recognized emotions as shown in TABLE I.

Tabel I. Dl Models Complementary Metrics

Deepfake	Landmark Correct	Landmark Incorrect
Image Correct	12	20
Image Incorrect	16	4

Exclusive	%
Landmark	42%
Image	33%

Original	Landmark Correct	Landmark Incorrect
Image Correct	8	10
Image Incorrect	6	28

Exclusive	%
Landmark	42%
Image	25%

With the two tables on the left, we report the intersection of sets of true positive emotion detected by two models. For example, 23% of the true positives in the original video and 15% of the true positives in the deepfake are detected by both the image-trained model and landmark-trained model. The relatively small overlap (no more than 51%) suggests that these representations complement each other.

The two tables on the right show the difference in the sets of true positives detected by the two representations. For example, 42% of the true positives detected by the landmark-trained model were not correctly identified by the image-trained model for both the original and the deepfake.
In summary, while the image-trained model picks up more textual details and recognizes more subtle emotions, the landmark-trained model detects less noise and has better accuracy in detecting well-articulated emotions. Thus, there is a potential to create a more effective emotion recognition system by combining the two models.

4. ADAPTIVE FER LEVERAGING GROUP EMOTION CHANGE CADENCE

With its wider adoption, DL-based FER today is used in technologies interacting with humans where accurate detection of individual and group emotion becomes desirable or even necessary [5]. Our motivating interest in better autism care is one example. Group emotion is a complex function of group members' emotions, group context, and environmental context. While the context information is relatively easy to acquire, people reflect their emotions facially in different ways and varying degrees of intensity. It is a challenge to design a system that accounts for individual behaviors.

Taking on the challenge, we construct a system combining the strength of both models from the previous section. Since group emotion is defined by the cadence of individual emotion changes in an engaging environment, our basic idea is to use the cadence to find an optimal trade-off between the two models.

We chose small (4 people) conversing groups as the experiment context of our system for four reasons: First, compared to static images, videos are more redundant for robust emotion recognition. Second, emotional changes within a conversation group tend to be synchronous which simplifies our system design. Third, small groups are easier for manual emotion evaluation and annotation. Lastly, we found a good amount of 2x2 grid view conversation videos for analysis.

The architecture of our system is shown in Fig. 3. There are three important components: noise reduction in emotional change recognition, group emotion synthesis and change alignment, and adaptive weights of two models' outputs.

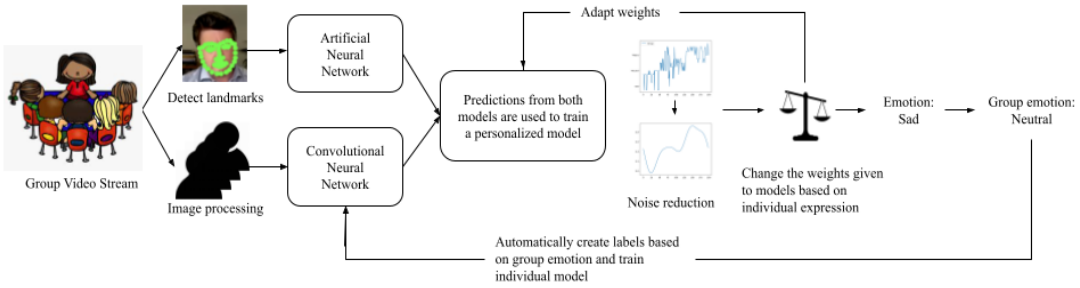


Fig. 3. Architecture of the Adaptive FER System

4.1. Noise Reduction in Emotional Change Recognition

One important limitation observed of the DL-based FER models is that they are sensitive to noisy inputs. For example, image quality jitters from frame to frame in the video stream could cause brief false emotion readings. To combat this, we computed the weighted arithmetic mean of detected emotions for every frame in a second. This averaging allows us to ensure that correlation between changes in emotion are in fact caused by participant's emotions, and reduces the impacts from various input noises.

4.2. Group Emotion Synthesis and Change Alignment

A group's emotion should reflect all its members' emotions. It is more diverse than individual emotions and could consist of more than one dominating emotion reflecting a polarizing group sentiment. To account for that, for any given point in time, our system sums up the probabilities of each emotion type from all group members and uses Euclidean distance to check for deviation. This is adequate for the small group size we analyze. For larger groups, a clustering algorithm such as K-means could be used to separate polarized sections.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

Fig. 4. Euclidean Distance

In detecting emotion changes, we separate the group-wise events where most members change emotions from the individual events where only one member changes. This allows us to treat group-level noises differently from individual-level noises. For group-level noise, we use DFT to convert the time series into frequency-domain and the noise reduction is achieved by eliminating the low-energy frequencies.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k n / N} \qquad k = 0, \ldots, N-1,$$

Fig. 5. Discrete Fourier Transform

For individual-level noises, we want to handle them more carefully because we do not want to miss any real emotion readings obscured by personalized behaviors. We adopted a collection of heuristics such as checking for recurring patterns before dismissing it as a noise.

4.3. Adaptive Weights of Two Models' Output

By our observations, the most common inconsistency in recognized emotions is the intensity in which people express their emotions. One's faint twitch of the cheek may convey the same amount of happiness as another's wide grin. As we demonstrated in earlier discussion, the image-based model is better equipped for recognizing these subtleties which the landmark abstraction could overlook. Thus, if not enough changes in emotions were detected in a certain group member, we increase the weight of the image-trained model to account for more subtle changes. On the other hand, if the emotion readings are noisier than the group average, the group member's facial expressions may be exaggerated, or the image quality may be low. In this case, we amplify the landmark-trained model to focus on the key emotions.

In Fig. 6, we describe our adaptive algorithm. For example, as a baseline, both models are given the same weight. Every minute, we track the amounts of changes in emotion during that minute. Out of the four participants, the ones that were detected to have above-average amounts of changes in emotion were given an extra weight to the landmark-based model and the same weight change is reduced from the image-based model. The opposite was done to the participants who were detected to have below-average amount of changes in the same time frame. Overtime, the weight for each participant settles into equilibrium as the group emotion dynamics converges. To facilitate the convergence, the weight adjustment value is a function of emotion change amount distribution among the group members. The higher the deviation, the higher the adjustment value.

```
Initialize weights for image-based model to 0.5
Initialize weights for landmark based model to 0.5
for every 10 seconds
        for every participant in meeting
                if the amount of emotion changes of member >
                  group number of emotion changes
                        Increase weight for landmark based model
                        Decrease weight for image-based model
                else
                        Decrease weight for landmark based model
                        Increase weight for image-based model
Emotion of each participant = image model prediction * image model weight +
                        landmark model prediction * landmark model weight
Group emotion = average of individual emotions
```

Fig. 6. Pseudo-code for the Adaptive Algorithm

To illustrate the results of our system, the graph to the left in Fig. 7 shows that the adaptive algorithm starts taking corrective action after 20 seconds and has eliminated three transient noises the non-adaptive algorithm classified as "Sad". The graph to the right in Fig. 7 shows that the adaptive algorithm corrected the false "fear" while bringing to surface a subtle angry emotion that would have been missed otherwise.

An issue our adaptive system does not handle well is when an individual's facial structure or neurodivergence makes the person show unintended emotions. In some recordings we reviewed as a part of this work, there were participants consistently misclassified as having a sad or angry emotion component. A more advanced online learning algorithm could aim to detect such persistent patterns in people's emotions and systematically remove the misleading structural component. Alternatively, considering our discussion group context, one could automatically generate personalized DL training data using the group emotion as the ground truth. The next section explores the latter as a solution.

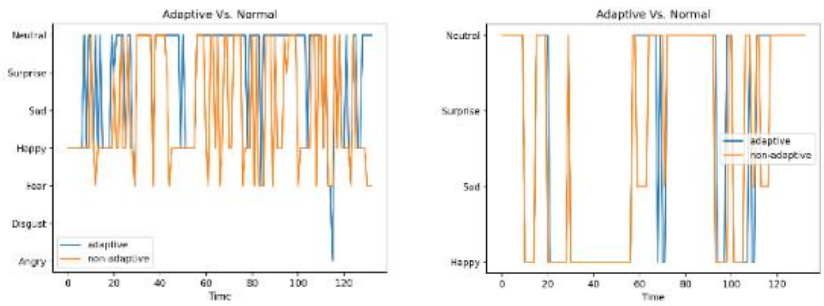


Fig. 7. Emotions Recognized by Adaptive vs Non-adaptive Algorithm

5. GROUP EMOTION AS THE GROUND TRUTH

Group emotion recognition has applications in social psychology [6], shot selection [7], image retrieval [8], surveillance [9], event detection [10], and event summarization [7]. We propose that group emotion be also used for personalizing individual emotion recognition.

For our group emotion recognition purpose, we classify discussion groups into four types: (i) unengaged groups, (ii) engaged groups, (iii) synchronous groups, and (iv) homogenous groups.

An engaged group can establish more information about the emotion correlation among group members than an unengaged group. A synchronous group is a strongly engaged group whose members share the same emotion change cadence, but the specific emotions at a given time may not be the same. At the highest level of engagement, in a homogenous group, group members tend to share the same emotions at any given time.

In the previous section, we discussed how group emotion cadence assists our adaptive algorithm to recognize individual emotion more accurately. It assumes that emotions of the members of the group are all affected by group-wise events, which would indicate that it is a synchronous group. This assumption required us to identify and eliminate the unengaged members from the group emotion calculation.

If we further restrict our application context to a homogenous group, such as in classrooms, or movie theatres, or concert halls, we hypothesize that the group emotion could be treated as the ground truth and used to label new training data. For example, when a group member's images are labelled with this ground truth, DL model could be trained against the member's personal facial expression patterns. In this section, we discuss our experiment designed to validate this idea.

Our experiment consists of the following four steps:

Step 1. We identified group discussion videos of knowledge-sharing nature where group members' emotions are highly synchronized without any divergence of opinions.
Step 2. Using the first half of the video, we compute the group emotion Gx using 3 of the 4 people in the group and label all image frames of the fourth person using Gx as the ground truth.
Step 3. Train our emotion recognition model with the labelled images obtained in Step 2.
Step 4. Using the second half of the video, we apply the newly trained model to the fourth person and check its performance against the models described earlier in this paper.

In principle, Step 3 could be achieved through active learning techniques so that the training could happen online in real-time. However, for this work, we have not tried that because our main goal is to show the validity of using group emotion as the ground truth and the viability of such an approach.

For the validation in Step 4, while a quantitative and general analysis is very difficult, we consider as a qualitative indicator whether automatic labeling increases the correlation between the individual and group emotion. Under our assumption that our group discussion videos do not produce diverging emotions and the group members are evenly engaged throughout the videos, more correlation between the individual and the group would indicate that automatic labeling improves the accuracy of the model. Our correlation metrics are calculated using the Pearson algorithm. For the videos we experimented with, we observed an anecdotal correlation improvement of 15% - 20%. While this is encouraging, we believe more data is needed for a thorough quantitative analysis.

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

Fig. 8. Pearson Correlation Coefficient

As a specific example, Fig. 9 compares two time series obtained in our experiment. The blue line is generated by the adaptive FER system described in the previous section. The orange line is

generated by the model trained with the automatically labeled data. The main difference between the two is that the blue line was trained with universal data (FER-2013) and the orange line was trained with personal data. Rather than relying on a universal facial emotion model, the orange line is able to identify the inherent "Sad" component in this particular group member's facial expression. Additionally, the false readings of "Fear" and "Angry" due to unrelated facial changes were also detected and compensated.

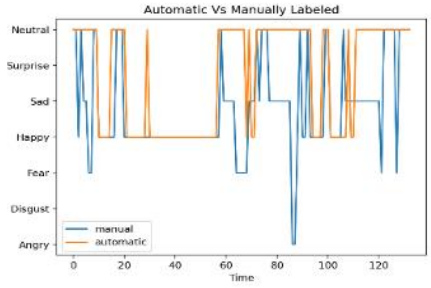


Fig. 9. Model Trained with Auto Labels vs Manual Labels

One important prerequisite for this approach is that the group videos used to collect training data need to exercise the full range of emotion states. This allows the personalized DL model to be trained with labels of all possible emotion values.

Labeling training data is an expensive, error-prone, yet critical step in developing DL-based systems. Our experiment shows an example of how this step could be automated to some extent through the knowledge of group context. Another interesting observation made in our experiment is that data labelled this way captures individual facial emotion patterns which could lead to more accurate and personalized emotion detection. We believe this finding could be applied to other group behavior analysis contexts where DL could play a bigger role in both synthesizing group-level behavior properties and creating individualized DL models.

6. THREATS TO VALIDITY

Construct validity. The main threat is related to how we assess the complementarity of the facial representations: image vs landmark. We support this claim by performing two different analyses: (i) complementarity metrics; and (ii) correlation test.

Internal validity. This is related to possible subjectiveness when evaluating the group emotion in the video fragments used. To mitigate such a threat, we employed three evaluators who independently checked the emotion. Then, we computed two-judges agreement on the evaluated videos. We also qualitatively discuss false positives and borderline cases.

External validity. The results obtained in our study used a small number of selected videos that may not generalize to other small conversing group contexts. To mitigate this threat, we applied our approach to a collection of group discussion videos of different subject areas, such as art, technology, politics, entertainment and sports. Another threat in this category is related to the fact that we apply our approach on pre-recorded videos only. While we do not yet have data to show the effectiveness of our approach in a live group meeting context, the focus of this paper is to show a general technique rather than to build a tool.

7. CONCLUSIONS

In this paper, we show that accurate emotion recognition can be informed by different facial representations. We evaluated the performance of two dominant facial representations and showed their complementary values. Our adaptive group emotion recognition system is flexible and could be reused for different group sizes and contexts. This avoids retraining which eliminates a large time sink native to some DL approaches, and broadens the applicability of our approach. Moreover, as an on-going effort, the adaptive algorithm used by our system is being replaced with an adaptive machine learning (ML) model. Further analysis is being done to assess the relative effectiveness of this ML model against our hand-crafted adaptive algorithm. We hypothesize that the two also present complementary values to some degree, and their complementarity metrics should be studied.

Our approach also highlights the values of accurate group emotion analysis. We showed that by establishing the recognized group emotions as the ground truth, individual emotion patterns such as resulting from neurodivergence could be better analyzed and modeled through automatic training data labeling. This finding speaks to the general possibility of automating the creation of certain training data in various group meeting contexts.

ACKNOWLEDGEMENTS

The author wants to express her heartfelt appreciation to Dr. Neha Keshav and Dr. Ned Sahin at Brain Power LLC for their invaluable guidance throughout this research. The work also received generous support from MIT BeaverWorks Summer Institute. Thank you sincerely.

REFERENCES

[1] Nancy R. Angoff, Making a Place for Emotions in Medicine, 2 Yale Journal of Health Policy L. & Ethics (2002).
[2] Tuan Le Mau, etc. Professional actors demonstrate variability, not stereotypical expressions, when portraying emotional states in photographs, Nature Communications 19 August 2021.
[3] Ian J. Goodfellow, et al. “Challenges in Representation Learning: A report on three machine learning contests.” http://deeplearning.net/icml2013-workshop-competition, July 2013.
[4] Octavio Arriaga et al., Real-time Convolutional Neural Networks for Emotion and Gender Classification, 2016.
[5] J. Bullington. Affective computing and emotion recognition systems: the future of biometric surveillance. Proceedings of the 2nd annual conference on Information security curriculum development. ACM, 95–99. 2015.
[6] P.M. Niedenthal and M. Brauer. 2012. Social functionality of human emotion. Annual review of psychology 63 (2012), 259–285.
[7] A. Dhall et al., 2015. Automatic group happiness intensity analysis. IEEE Transactions on Affective Computing 6, 1 (2015), 13–26.
[8] A. Dhall, A. Asthana, and R. Goecke. 2010. Facial expression based automatic album creation. International Conference on Neural Information Processing. Springer, 485–492.
[9] T. Vandal, D. McDuff, and R. El Kaliouby. 2015. Event detection: Ultra large-scale clustering of facial expressions. In IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Vol. 1. IEEE, 1–8.
[10] Tatsuya Hayamizu, Group Emotion Estimation using Bayesian Network based on Facial Expression and Prosodic Information, 2012 IEEE International Conference on Control System, Computing and Engineering.
[11] James Bergstra and David D. Cox. Hyperparameter optimization and boosting for classifying facial expressions: How good can a “null” model be? Workshop on Challenges in Representation Learning, ICML, 2013.

[12] V. Franzoni, G. Biondi, and A. Milani, "Crowd emotional sounds: spectrogram-based analysis using convolutional neural network." in SAT@ SMC, 2019, pp. 32–36.
[13] Liwei Wang, et al.. Towards Understanding Learning Representations: To What Extent Do Different Neural Networks Learn the Same Representation. 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
[14] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097–1105, 2012.
[15] Emmeke A. Veltmeijer et al., Automatic emotion recognition for groups: a review. DOI 10.1109/TAFFC.2021.3065726, IEEE Transactions on Affective Computing.
[16] Wafa Mellouk et al., Facial emotion recognition using deep learning: review and insights. The 2nd International Workshop on the Future of Internet of Everything (FIoE) August 9-12, 2020, Leuven, Belgium.

AUTHOR

Valentina Zhang is a student at Phillips Exeter Academy, an intern at Brain Power LLC developing AI-based technologies for autism care, and an assistant at MIT EEEi workshops building a conductive learning environment for children with neurodivergence. Besides her current responsibilities, Valentina is an alumna of MIT Beaver Works Summer Institute specializing on machine learning in medicine and a speaker at Global AI Student Conference.

A DOMAIN ONTOLOGY FOR MODELING THE BOOK OF PURIFICATION IN ISLAM

Hessa Abdulrahman Alawwad

College of Computer Sience and Information, Imam Mohammad Ibn Saud Islamic University, (IMSIU), Riyadh, Saudi Arabia

ABSTRACT

This paper aims to fill the gap in major Islamic topics by developing an ontology for the Book of Purification in Islam. Many trusted books start with the Book of Purification as it is the key to prayer (Second Pillar after Shahadah, the profession of faith) and required in Islamic duties like performing Umrah and Hajj.

The strategy for developing the ontology included six steps: (1) domain identification, (2) knowledge acquisition, (3) conceptualization, (4) classification, (5) integration and implementation, and (6) ontology generation. Examples of the built tables and classifications are included in this paper.

Focus in this paper is given to the design and analysis phases where the technical implementing of the proposed ontology is not within this paper's objectives. Though, we presented an initial implementation to illustrate the steps of our strategy.

We make sure that this ontology or knowledge representation on the Book of Purification in Islam satisfy reusability, where the main attributes, concepts, and their relationships are defined and encoded. This formal encoding will be available for sharing and reusing.

KEYWORDS

Domain Ontology, Book of Purification, knowledge representation, OWL, Protégé.

1. INTRODUCTION

Sharing and exchanging knowledge are two of the basic purposes of communication. People request and give information in daily communication. Comprehending the topics they share and their relationships with other people happens smoothly and naturally with a person's knowledge-building process. Teaching is one of the best-known examples of knowledge exchange.

With the increased use of the internet, people have carried their communicative natures to the digital community, where question-answering systems are one method of revealing their need for information-sharing, and where people, or an intelligent agent that acts on their behalf, share and request knowledge from one or many sources.

As information is distributed by many sources, it is hard for machines to understand it and, therefore, to give the right answer to a specific question. To make the process of understanding this data or information possible, we need to model domain knowledge that comprises the main concept described with attributes and relationships with other concepts.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 59-67, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120905

The Semantic Web was originated by the creator of the World Wide Web, Tim Berners-Lee, who described it as the shift of the web from its conventional, hyperlinked documents, to a huge semantic information storage system where machines can understand and retrieve knowledge from it. Machines should conduct automated reasoning using inference rules on the structured data in order to exploit the Semantic Web. In order to achieve that, knowledge representation must be modeled [1].

Modeling concepts along with their relationships in a way that makes it easy for machines to comprehend them introduces the importance of ontology. Ontology is a philosophical term that has been discussed for a long time, and it refers to the subject of existence, defining a being in a manner that answers the questions: How does this being mean to exist? How does the existence of one being differ from that of another? In computer science, ontology is a structured set of concepts that makes sense for information [2].

Ontology in the context of knowledge sharing, as Tom Gruber has described, is a specification of conceptualization [3]. Representing the information in an ontology makes it both understandable for machines and humans and available for share and reuse.

Ontologies provide a framework for extracting conclusions from the structured information [4]. They mainly reduce the conceptual confusion among those who share the information. They can be applied to many areas, for example knowledge-sharing and reuse, artificial intelligence (AI), software design, and the Semantic Web.

Work on expert systems in the 1980s led AI developers to model knowledge in a standardized semantic manner that machines would be able to understand and comprehend. They also wanted this knowledge to be shared and reused. As ontologies offer this paradigm of knowledge representation and sharing, it was an emerging solution to such an environment.

Ontologies are defined using semantic markup languages like the Resource Description Framework (RDF) and the Web Ontology Language (OWL) and are structured as taxonomies. RDF models the information objects as HTTP Uniform Resource Identifiers (URIs), comprise the information found on the web, and start with “http:.” The RDF models the information or resources conceptually and describes them as a statement of (subject, predicate, object) form.

The Resource Description Framework Schema (RDF/S) adds, to some extent, a semantic to the RDF description, so it is an enhanced approach of describing resources based on the RDF definition. RDFS describes the resources along with the relationships among them by defining the domain as classes, subclasses, and properties. The OWL offers much more richness in the semantic description of the relationships between concepts like stating the disjointedness between two classes.

The level of expressiveness of the semantic can be enhanced by the use of the OWL, which is built on top of RDF and RDFS, to model complex relationships. Querying the data stored in the RDF representation is possible using an RDF query language called SPARQL. This could be used to test the built ontology against the domain for which it was developed.

Some authors in [5] identified three major scenarios for the use of ontology: first, supporting the communication process between people, where an unambiguous yet informal ontology would be sufficient; second, achieving interoperability in communication between computer systems by translating between the different modeling techniques, paradigms, languages, and software tools where the ontology is used as an interchange format; third, improving both the process and

quality of engineering software systems by meeting the reusability, reliability, and maintainability.

2. BACKGROUND

Arabic and/or Islamic ontologies have been active fields of academic research. The WordNet Project [6] introduced an ontology for mapping between different languages. In Quranic ontology, Quran is the religious book of Islam and a revelation from Allah. Knowledge representation defines key concepts in Quran, along with the relationships between these concepts. A few ontologies have been developed [7] [8].

A few ontologies also have been developed for Hadith, the second source of Islamic legislation, which is a record of the words, actions, and silent approval of the Prophet Muhammad PBUH. TibbOnto [9] presents the Prophet's medicine in a semantic ontological representation based on an authentic Tibb Al-Nabawi Hadith. As an example of Islamic ontology, some authors in [10] developed an ontology of the business model, diverse roles, distinct concepts, and representation rules of the Islamic banking industry.

Another group of researchers of Hadith science defined a large Arabic Islamic ontology [11] They followed different steps including defining concepts, relationships, functions, axioms, and instances. They defined five criteria to shed light on the importance of the ontology of clarity, consistency, extensibility, minimal encoding deformation, and minimal ontological commitment. Their ontology was created with Portege's editor and the OWL and included 183 concepts and 145 relations.

With Islam-related research, researchers need to carefully choose their main source or reference for ontology development, whether the Quran, Hadith, Qiyas, or consensus. Commentary books (Kutub Al Shuruh) are also reliable sources of Islamic legislation. The problem with Islamic ontology is the lack of coverage of many Islamic topics. One study on Ketab Al-Salaat [12] proposed an ontology for Salaat (the Second Pillar of Islam) on the Protégé tool based on a test-driven ontology-development methodology (TODE). They have developed a prototype application on the JENA semantic web toolkit for the utilization of the proposed ontology. Their ontology (Figure 1) consisted of 113 concepts and 85 properties.

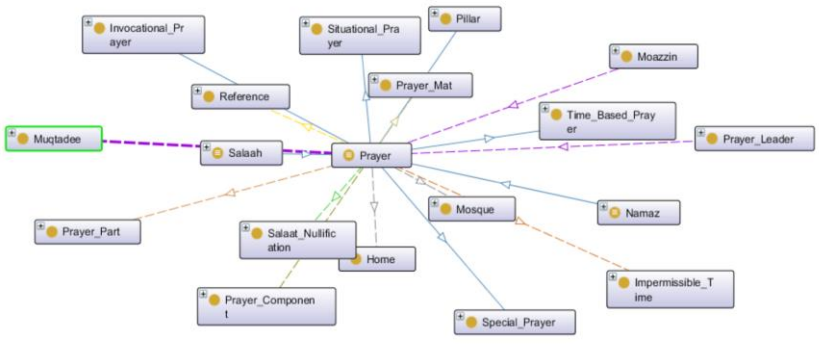


Figure 1. The proposed ontology of Salaat along with properties and relations [12].

Authors in [13] also proposed an ontology for Salaat, their constructing process encompassing three stages: first, determining the domain and scope of the ontology, which are Islamic knowledge and Salaat, or prayer, respectively; second, reusing the existing ontologies where they

used Quran indexes; and third, defining the classes, the class hierarchy, and the properties of classes. Finally, they evaluated their method. Their ontology (Figure 2) covered 48 concepts and 51 properties.

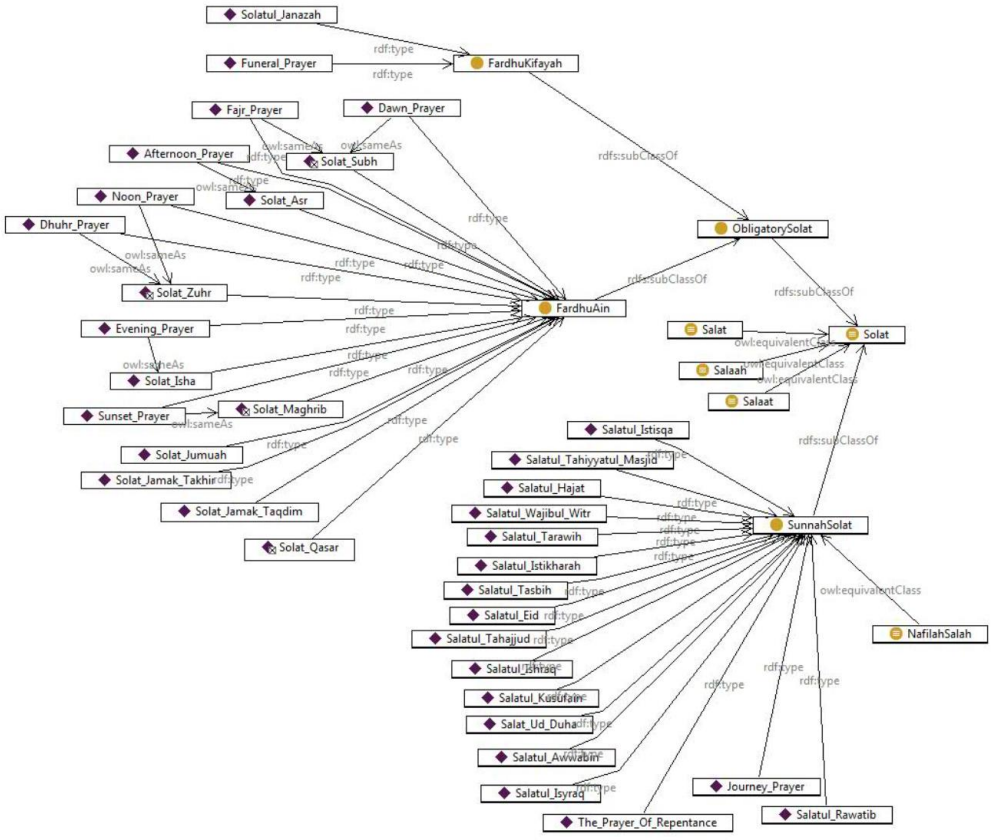


Figure 2. The proposed ontology of Solat along with properties and relations [13].

Authors in [14] defined an ontology for Hadith understanding with a specification word, “Samiaa,” which indicates the hearing. The domain of knowledge was Sahih Al-bukhari, which is recognized as the most reliable book of Hadith. They performed five stages of Hadith understanding. In the first phase, they extracted Sahih Al-bukhari from Hadith software. In the second phase, they identified the word Samiaa as the root word and then identified the noun form of Samia and its morphologies. Then, they extracted all Hadith that contained the word Samiaa and analyzed them based on commentary books. Their overall Hadith ontology is shown in Figure 3.

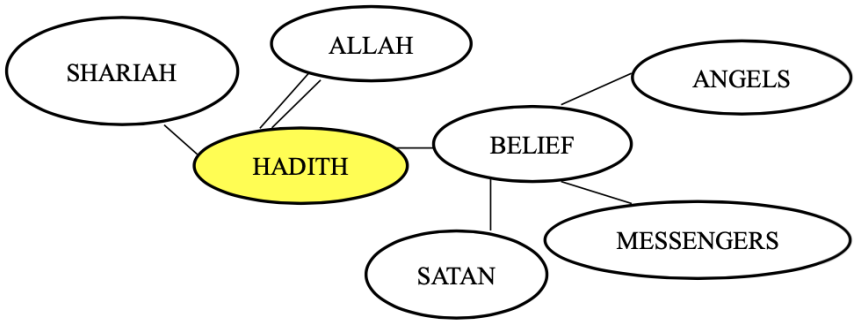


Figure 3. Ontology of Hadith [14].

Other authors [15] proposed an Arabic prophetic ontology about prophets and messengers in Islam based on the Holy Quran, AlHadith, and commentary books. They followed five steps for their ontology building: the first was specification, which means understanding the concepts and their relationship in the Quran, Hadith, and commentary books by meeting experts and reusing related ontologies. Then, they represented the conceptualization in an intermediate representation. Then, they started to build the ontology by defining the class hierarchies and properties of the classes. The resulting ontology contained 151 classes and 44 properties.

3. PROBLEM DEFINITION

This paper aims to fill the gap in major Islamic topics by developing an ontology for the Book of Purification in Islam. Many trusted books start with the Book of Purification, as it is the key to prayer (the Second Pillar after Shahadah, the profession of faith) and is required in performing Umrah and Hajj, for example.

4. ARCHITECTURE DESCRIPTION

This paper's focus is to ensure that this ontology or knowledge representation on the Book of Purification in Islam has satisfying reusability, where the main attributes, concepts, and their relationships are defined and encoded. This formal encoding will be available for sharing and reuse.

This paper also focuses on meeting the interoperability, reliability, and maintainability attributes. The defined ontology will ease the process of communication between the software systems (interoperability). This formal encoding will also help in the automation of consistency checking, which will make the system more reliable (reliability). The use of the defined ontology in a software system can render maintenance easier in many ways. Building systems by defining an explicit ontology improves documentation of the software, which, in turn, reduces maintenance costs [16].

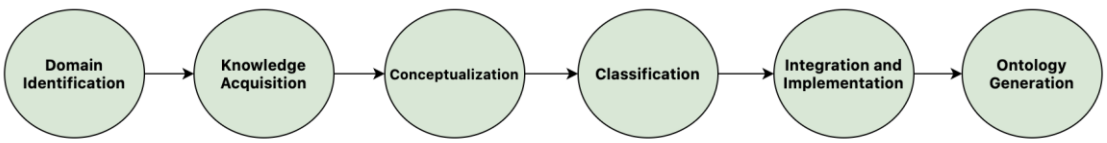


Figure 4. Design and development of the Book of Purification ontology.

Figure 4 presents our methodology for the development of the Book of Purification ontology.

4.1. Domain Identification

This first step is concerned with identifying the domain of this ontology and specifying the reference for constructing the ontology. It encompasses the process of defining the domain, scope, and purpose of the ontology and reviewing any existing domain ontologies.

The ontology model for the Book of Purification in Islam provides a semantically structured model that is understandable by both humans and machines for the concepts of purification and facilitates knowledge-sharing and retrieving in the domain of purification.

As this paper aims to develop an ontology for the Book of Purification, the reference that will be used to identify the ontology is the book Alruwd Almurbea. This is one of the Shuruh books written by Imam Mansoor Alhanbali. Other references, like Sahih al-bukhari, Sunan Ibn Majah, and Sunan Abi Dawud, can be included in future work.

4.2. Knowledge Acquisition

This step is concerned with acquiring the knowledge needed to build the ontology from the specified reference. This step includes meeting experts and revising knowledge.

4.3. Conceptualization

Table 1. Sample of the concept table for the Book of Purification ontology.

Concept	Description
Islam	This is the main class that encompasses all the rest of classes.
Pillars	Five basic acts in Islam, considered mandatory by believers, and are the foundation of Muslim life. They are summarized in the famous hadith of Gabriel.
Worship	In Islam, worship refers to ritualistic devotion as well as actions done in accordance to Islamic law which is ordained by and pleasing to Allah (God). Worship is included in the Five Pillars of Islam, primarily that of salat, which is the practice of ritual prayer five times daily.
Salah	The second of the five pillars in the Islamic faith as daily obligatory standardized prayers. It is a physical, mental, and spiritual act of worship that is observed five times every day at prescribed times.
Purification	Cleanliness and being free from dirt. While in the context of Islamic Jurisprudence (Shari'ah), It means the removal of impurities and dirt.
Inner Purification	It is the purification of the heart from polytheism, sins etc.
Physical Purification	It is the purification of the body from dirt and impurities.

In this step, we specify conceptualization by defining tables. These tables will be used to directly build the ontology in section 4.5: (1) The concept table (table 1 presents a sample) contains the domain concepts along with the hierarchy of these concepts. Each concept will be defined in a <concept, description> glossary. (2) The properties table (table 2 presents a sample) is used for

identifying the internal structure between concepts by specifying the domain and range for every property along with a description for it. (3) A relation table defines constraints, types, and cardinality. (4) Create the individuals of the concepts and their defined relationship.

Table 2. Sample of the properties table for the Book of Purification ontology.

Object properties	Domain (Concept)	Range (Concept)	Description
achieved_by	Wudu	Purification	Purification from these is achieved by performing al-wudu
achieved_by	Ghusl	Purification	Purification from these is achieved by performing Ghusl
purify_through	Body	Ghusl	It is possible to purify oneself through wudu or ghusl.

4.4. Classification

Figure 5. Part of the defined classes along with subclasses.

Classification is the main step of building the ontology, it consists of four steps. (1) Identifying the class hierarchy: the class is the general specification that can be used to instantiate individuals. Figure 5 presents part of the ontology of the Book of Purification and the classes along with subclasses. (2) Defining the properties of classes. Properties represent the relationships in the ontology; this is the second step after defining the classes and subclasses. Property could be either data property or object property. Object property has the type owl:ObjectProperty and links individuals to individuals, whereas data property has the type owl:DatatypeProperty and links individuals to data values. Figure 6 presents part of the defined

object and data properties. (3) Creating the individuals of our ontology. Individuals are the instances of our classes. Figure 7 represents individuals for the class Natural_Discharges under the Minor Hadath class. (4) Creating the axioms of our ontology. In order to describe the nature of the relationship between the classes, attribute and instance axioms are used. Axioms are defined for the classes to describe how a certain class relates to another. It could be either a subclass (where a class is a subclass of another class), an equivalent class, or a disjointed class.

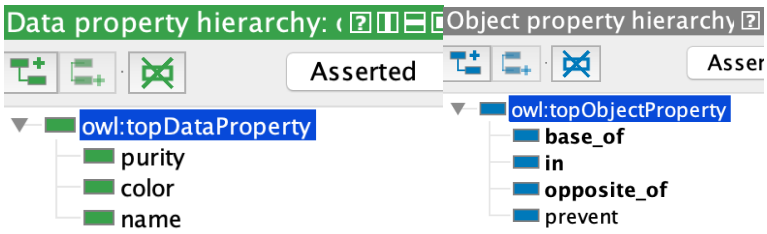


Figure 6. Part of the defined object and data properties.

The class hierarchy in Figure 7 represents some of these axioms, where Minor and Major Hadath are subclasses of the class Hadath.

Axioms for attributes represent how an attribute relates to another, and whether the relationship is transitive, inverse, or equivalent. Axioms for individuals state whether two individuals are the same or different.

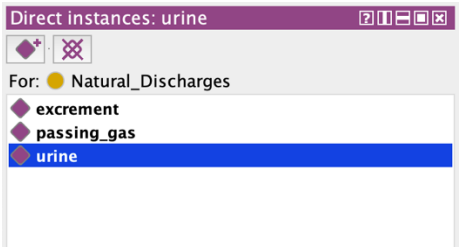


Figure 7. Individuals of the proposed ontology for the Book of Purification.

4.5. Integration, Implementation and Ontology Generation

The last two steps include implementing the knowledge defined in the previous steps and possibly re-using an existing ontology like Salaat to expand the applicability of our ontology.

The implementation process will be achieved using the standard ontology language OWL and the ontology editor Protégé, which includes reasoning tools to check the consistency of the built ontology.

Since the technical implementation of the proposed ontology is not within our objectives in this paper, focus is given to the design, analysis, and initial work of the last two steps. The intensive work on Steps 5 and 6 will be left for future work.

5. CONCLUSIONS

In this paper, we have discussed the gap that needs to be filled with encoding the major Islamic topics in the formal encoding that is available for share and reuse by means of ontology.

We make sure that the proposed ontology or knowledge representation on the Book of Purification in Islam is defining and encoding the main attributes, concepts, and their relationships.
We have presented in detail our strategy in building the ontology, which encompasses six steps, including domain identification, knowledge acquisition, conceptualization, classification, integration and implementation, and ontology generation.

As designing and analyzing the ontology was the focus of this paper, we started with an initial implementation to represent the steps in our strategy in this paper.

Other systems such as Q&A systems can reuse our ontology. It can be tested and evaluated by user-driven and application-driven approaches.

REFERENCES

[1] G. Kuck, "Tim Berners-Lee's Semantic Web," SA Journal of Information Management, vol. 6, no. 1, 2004.
[2] A. Abdelkader, "Creation of Arabic Ontology for Hadith Science," International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, p. 3269–3276, 2019.
[3] A. L. Knowledge Systems, "What is an Ontology?," [Online]. Available: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html. [Accessed 2020].
[4] F. Echarte, J.J. Astrain, A. Córdoba and J. Villadangos, "Ontology of folksonomy: a new modeling method," in Semantic Authoring, Annotation and Knowledge Markup, 2008.
[5] Jasper and M. Uschold, "A framework for understanding and classifying ontology applications," in he IJCAI99 Workshop on Ontologies and Problem-Solving Methods (KRR5), 1999.
[6] S. E. W. B. P. V. D. Farwell, A. Pease and C. Fellbaum, "Arabic WordNet and the Challenges of Arabic," in Arabic NLP/MT Conference, London, U.K, 2006.
[7] "Al-Bayan: An Arabic Question Answering System for the Holy Quran," [Online]. Available: https://www.aclweb.org/anthology/W14-3607.pdf. [Accessed 2020].
[8] "Arabic Quranic Search Tool Based on Ontology," [Online]. Available: http://eprints.whiterose.ac.uk/101257/1/alqahtani16nldb.pdf. [Accessed 2020].
[9] A. Al-Rumkhani, M. Al-Razgan and A. Al-Faris, "TibbOnto: Knowledge Representation of Prophet Medicine (Tibb Al-Nabawi)," Procedia Computer Science, vol. 82, p. 138–142, 2016.
[10] "Characteristics and Development Criteria for Islamic Banking Ontology," [Online].
[11] A. Abdelkader, "Creation of Arabic Ontology for Hadith Science," Advanced Trends in Computer Science and Engineering, vol. 8, p. 3269–3276, 2019.
[12] N. Islam and K. Laeeq, "Salaat Ontology: A Domain Ontology for Modeling Information Related to Prayers in Islam," Indian Journal of Science and Technology, vol. 12, pp. 1-12, 2019.
[13] S. Saad, N. Salim, H. Zainal and Z. Muda, "A process for building domain ontology: An experience in developing Solat ontology," in International Conference on Electrical Engineering and Informatics, Bandung, 2011.
[14] J. Junaidi, I. H. Jamal, N. M. Ghazali, H. Ahmad and R. A. Salam, "Expanding Hadith Understanding Using Ontology," Advanced Science Letters, vol. 23, no. 5, p. 4611–4614, 2017.
[15] H. A. Al-Sanasleh and B. H. Hammo, "Building Domain Ontology: Experiences in Developing the Prophetic Ontology Form Quran and Hadith," in nternational Conference on New Trends in Computing Sciences (ICTCS), Amman, 2017.
[16] M. Uschold and R. A. Jasper, " A Framework for Understanding and Classifying Ontology Applications," in the IJCAI99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, 1999.

AN IMPROVED NLP FOR SYNTACTIC AND SEMANTIC MATCHING USING BIDIRECTIONAL LSTM AND ATTENTION MECHANISM

Fadya Abbas

Department of Computer Engineering, Nahrain University, Iraq

ABSTRACT

Dealing with extensive amounts of textual data requires an efficient deep learning model to be adapted. However, the following reasons; the highly ambiguous and complex nature of many prosodic phrasing also enough dataset suitable for system training is always limited, cause big challenges for training the NLP models. This proposed conceptual framework aims to provide an understanding and familiarity with the elements of modern deep learning networks for NLP use. In this design, the encoder uses Bidirectional Long Short-Term Memory deep network layers, to encode the test sequences into more context-sensitive representations. Moreover, the attention mechanism is mainly used to generate a context vector that is determined from distinct alignment scores at different word positions, hence, it can focus more on a small words' subset. Hence, the attention mechanism improved the model data efficiency, and the model performance is validated using an example of data sets that show promise for a real-life application.

KEYWORDS

NLP, Deep Learning, LSTM, Attention Mechanism, Data Efficiency.

1. INTRODUCTION

Syntactic and semantic analysis is becoming more crucial with time as it can provide valuable information needed in many wide-world applications. Several Natural Language Processing (NLP) based syntactic and semantic analysis methods that trained using the following deep learning models: RNN, GRU and LSTM models [1]. Bidirectional LSTM (BiLSTM) deep neural network layers have been used more in NLP algorithms, particularly in the Encoder stage. Basically, the BiLSTM layers are implemented in a Bidirectional manner, to form two learning models in different directions [2]. The main advantage of the BiLSTM layer is to encode the text sequences into more context-sensitive representations. Recently, attention mechanism designs led to the development of modern NLP architectures. [3] This research conducted three syntactic experiments. Also, their data is associated with different percentages of noise. These experiments use the following three models; the conventional LSTM, bi-directional LSTM, and bi-directional LSTM with Conditional Random Field (CRF) which has given the best performance among the other two. [4] This research also presents an architecture for argument classification that uses Bidirectional LSTM and CRF decoding for finding optimal sequences. This method is based on the combination of syntactic features and external word representations from FastText, where the FastText is used to obtain better results for words that do not exist in the dictionary.

The Attention mechanism essentially generates a context vector that is computed from various alignment scores at various word positions, so it can focus on a small words' subset. It weighs all

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 69-75, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120906

inputs individually that are fed to the decoder to create the target sequence. This leads to a better contextual understanding resulting in a maximum prediction score in target sequence generation. Also, attention mechanism is used widely in the language translation fields [5].

However, these algorithms still need to be implemented in the syntactic and semantic analysis in terms of data efficiency perspective. Hence, the proposed NLP algorithm uses the attention layer to extract only the valuable data and remove the others; therefore, it can improve the training limitation problem and leverage the model into more data-efficient.

Figure. 1 demonstrates the overall system architecture, particularly, it shows that the input clauses are encoded into sequences of distributed vectors in the Encoder stage. The learnable parameters; E, W, M, and W' are the adjustable weights between the model hidden layers. The final layer of the Encoder sends the output Y_{t+1} information to the Decoder after an extensive processing of data dependencies in the attention layer (with the output h_{t+1}). Finally, the Decoder can analyse the entire contextual understanding of all the previous words via probability distribution of the Softmax function. Combining each hidden layer's output, the weight matrices, and bias (b) can mathematically be expressed as follows:

$$\text{Encoder: } Y_{t+1} = Softmax(E_{t+1}\,.\,x_{t+1} + W_t\,.\,Y_t + b_{t+1}) \quad (1)$$

$$\text{Decoder: } O_{t+1} = Softmax(M_{t+1}\,.\,h_{t+1} + W'_{t-1}\,.\,O_t + b_{t+1}) \quad (2)$$

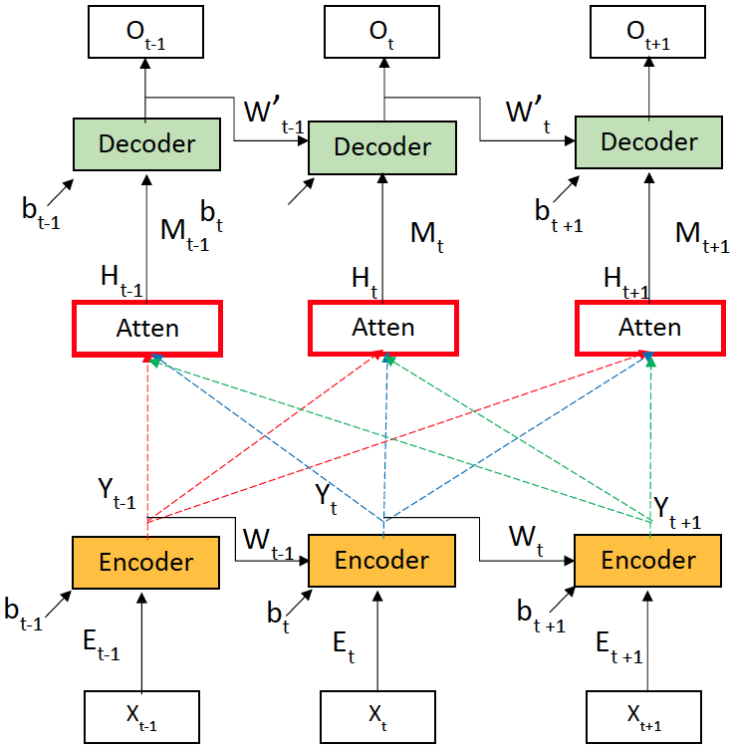


Figure 1. Overall NLP architecture.

Where the Softmax is the activation function at the neural network layer output. With attention assistance, the Decoder can determine the extent of the match between the t-th input and the corresponding t-th output information. More details on the main parts and the pipeline data processing of the above diagram are clarified below.

2. ARCHITECTURE DETAILS

The following Architecture-based approaches are organized into four main levels using text sequences' analysis. Each level constructs a model and shows the algorithm that uses syntactic elements of a text and how to initiate the relevant learning weights to eventually ends with a prediction category for the final text sequence patterns.

2.1. Word Embeddings and Tokenization

The text is processed in this model and represented as a sequence of embedding tokens to express the semantic features of the sequences; given the text X = (x_1; x_2; : : : ; x_n) in which consisting of n clauses, and with each clause, xt =(w_1;w_2; : : : ;w_m) containing m words. First, we obtain a hidden representation and the tokenization process of the sequences using the pre-trained model called Bidirectional Encoder Representations from Transformers (BERT); T_t = BERT(x_t). In this way, the input clauses are encoded into a sequence of the distributed vectors T = [T_1; T_2; : : : ; T_n]. In BERT, four types of embedding: token embedding, segment embedding, position embedding, and topic embedding. The token embedding performs the embedding vector of each word, segment embedding is utilized to distinguish sentences, position embedding learns embedding at each word position in order to represent the sequences' order information, finally, the topic feature embedding is used to capture the underlying topic information [2, 6]. With these mentioned combined features, it can generate different embeddings for polysemous words and can model words and context by characters which leads to better contextual management, especially with the words that have multiple grammatical structures.

2.2. The Encoder - BiLSTM Layers Implementation

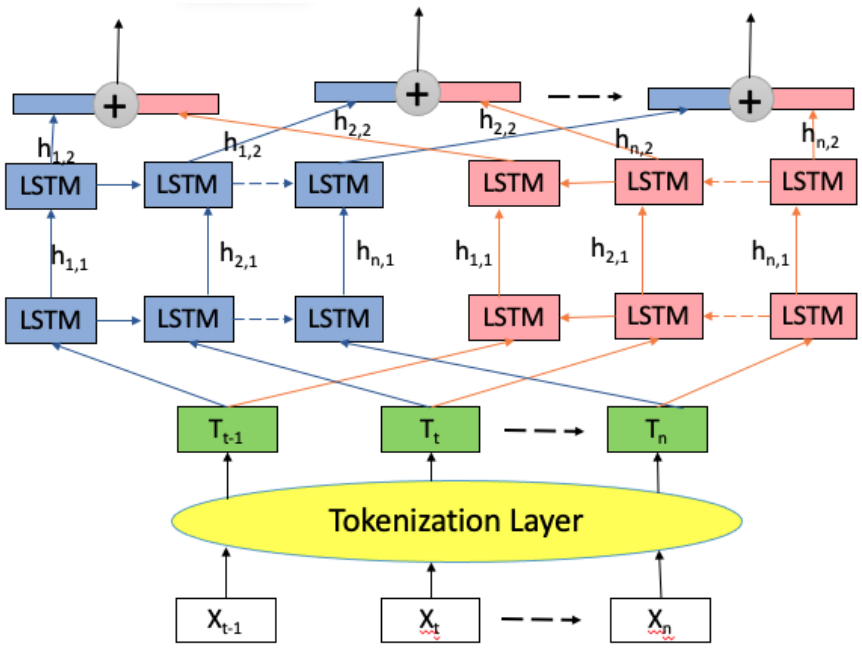


Figure 2. Encoder architecture - using BiLSTM.

Recently, the LSTM architecture has already offered great prediction results in many engineering applications. In this work, BiLSTM is used in the Encoder stage to enable additional training by crossing over the input data sequences twice. This can be happened by including two LSTM

layers, one takes the forward direction sequences, and the other takes the backwards direction sequences, which offers better predictions. In this way, The approach can capture contextualized word representations in a clause. So, BiLSTM model leverages the Encoder task in extraction of the contextual word representations. The goal of BiLSTM is to model (1) complex semantical and syntactical characteristics of words (2) lexical ambiguity or polysemy, words with similar pronunciations could have different meanings at different locations or contexts [7].

At time step t-1 the forward language model can predict the next token T_t for the given previous tokens of the input sequence using the joint probability distribution as in (3). For the backward direction it is following the same manner except it is in a reversed order as shown in (4) [8, 9]:

$$p(T_1, T_2, .., T_n) = \prod_{i=1}^{N} p(T_i \mid T_1, T_2, .., T_{i-1}) \text{Forward direction} (3)$$

$$p(T_1, T, .., T_n) = \prod_{i=1}^{N} p(T_i \mid T_{i+1}, T_{i+2}, .., T) \text{Backward direction} (4)$$

Hence, the two LSTMs can process sentence inputs twice in reversed directions to encode the input sequence T into more context-sensitive representations. For each token representation $T_i$the bi-directional vector representations can be computed by their two hidden layers$\vec{h}_{ij}$and $\overleftarrow{h}_{ij}$, in which their weights can be updatedusing μ of each LSTM model as

$$\overrightarrow{h_{ij}} = \overrightarrow{LSTM}\left(T_i, \overrightarrow{h_{i-1,j}};\ \mu_{LSTM}\right) \text{Forward direction} (5)$$

$$\overleftarrow{h_{ij}} = \overleftarrow{LSTM}\ (T_i, \overleftarrow{h_{i+1,j}};\ \mu_{LSTM}) \text{Backward direction} (6)$$

Where $\vec{h}_{i-1,j}$and $\overleftarrow{h}_{i+1,j}$ are the forward and backward LSTM hidden layers respectively. By concatenating these two directional outputs the final representation to be sent to the attention layer is as follows:

$$Y_{t+1} = \left[\overrightarrow{h_{ij}} + \overleftarrow{h_{ij}}\right] (7)$$

2.3. The Attention Layer

The main objective of the attention layer is to detect long-range dependency between the word pairs in a sentence using the attention weights. Therefore, the model can capture pair-wise relations of the input tokens in a sequence. So, It sets large weights on the important tokens to the word. In the other words, the attention layer enhance the contextualization by generating attention vectors, that can determine the relevance of the t-th word in a sequence that concerning other words. To obtain the attention output, Feed Forward Neural (FFN) Network receives the attention vectors and outputs the attention's result to the Decoder's model. In fact, the Encoder output and Decoder input embeddings are both fed to the attention to performs attention between the them. This obtains the relevance of the input's tokens concerning its targets tokens as the Decoder determines the actual vector representation between the mapping target and the source [3].

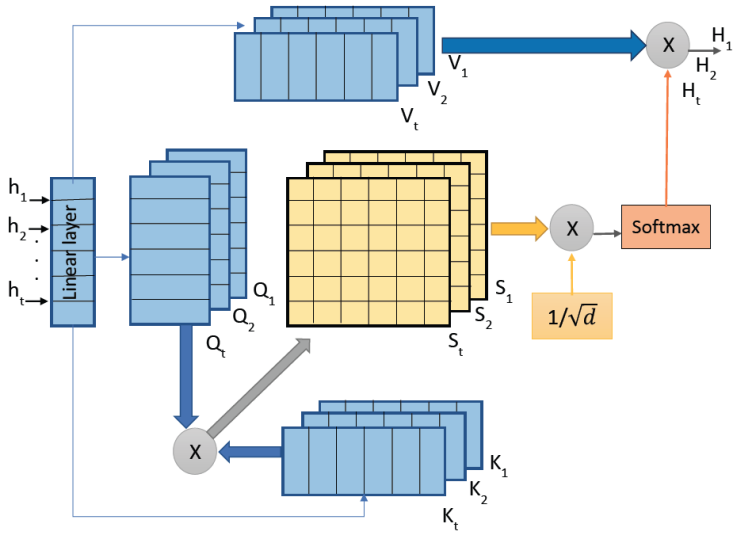


Figure 3. The attention mechanism.

It is worth noting that the attention layer is formed by multihead attentions, which contains of t attention heads, each learns different representation in a sequence. The main task of multihead attention is to map the hidden state sequence received from the BiLSTM into t different query, key and value matrices via a linear projection. Therefore, for the t-th head, the queries, keys and values can be represented by Q, K and V matrices respectively. Finally, the context vectors for each attention head (H_t) can be calculated using the following formulation [10]:

$$H_t = Attention(\, Q\, W_t^Q, K\, W_t^K, V\, W_t^V) \qquad (8)$$

$$H_t = Softmax\left(\frac{(Q\, W_t^Q)\,(K\, W_t^K)}{\sqrt{r}}\right) V.W_t^V \quad (9)$$

Where the $W^Q{}_t$, $W^K{}_t$, and $W^V{}_t$ are the weights of Q, K, and V matrices respectively. The softmax function is used to find the probability of the matrices correlation, where $\sqrt{r}$ is just to scale the final attention score.

Multihead attention enhances the model's capability to attend sequence parts from a different perspective by implementing attention parallelly multiple times. This performs a self-attention vector for each token by weigh the word with higher than others by the high resultant dot product.

This attention mechanism will be implemented in parallel for multiple times is computed which result in multiple attention vectors for each token to determine the attention vector. Finally, the resulting heads or attention outputs are concatenated and then transformed to the next LSTM layer (in the Decoder).

2.4. The Decoder

After capturing the information using the previous multihead attention layer, the sequence representations are sent to the next stage which is the Decoder. In fact, multihead attention can help the Decoder in better learning the soft alignment between the summary and the source document. The Decoders predicts the final summary representation also learns the final objective which is maximizing the likelihood of the conditional probability. Therefore, as shown in the figure. 1 the Decoder is also using LSTM Layer in order to predict the final summary

representation. This layer is implemented to model the output distribution over the class tags then generates an output sequence of predicted tags (S_t) [7, 8]:

$$S_{t+1} = LSTM(H_t, S_t;\ \theta_{LSTM}) (10)$$

Subsequently, the output of Decoder's LSTM is sent to the Softmax layer over tag vocabularies. The Softmax layer determines the normalized probability distribution over thw labels of the possible phrase for each word:

$$p^{pred}(O_t = l_t | S_{1:N} = \frac{\exp(W_t S_t + b_t)}{\sum_{t=1}^{N} W_t S_t + b_t}) \quad (11)$$

Where p^{pred} represents the possibility that the t-th token's label is in the label set, O_t represents the output of the t-th token. W_t and b_t are referred to the weight vectors and bias vectors respectively. l_t is the ground truth of the clause S_t for the tag prediction, and N denotes to the total number of tags.

This model is trained to maximize the log-likelihood of a tag prediction as an objective function in which the following loss function can be determined:

$$Loss_t = \sum_{k=1}^{K} p_k^{real} * \log(p_k^{pred}) (12)$$

$Loss_t$ denotes to the loss of the t-th token. K is the length of label set. p_k^{real} denotes the real possibility that the t-th token's label in the label set is close or equal to 1 if the label is the real label, otherwise is 0.

3. INITIAL RESULTS

The proposed algorithm applied to the training data set represented by four chapters of the Moby Dick book. The test procedure passes 25 words as test data to the NLP algorithm and the algorithm predicts the 26th word. The vocabularies of all words that occurred less than five times in the training data set have been discarded.

Table 1. Accuracy of syntactic and semantic matching.

NLP Algorithm	Syntactic [%]	Semantic [%]	Total accuracy [%]
Without Attention Mechanism	59	54	57
With Attention Mechanism	61	58	60

Table. 1. shows the precision syntactic and semantic matching. It has experimented on ten sentences frequency. As it can be seen from the table, the attention mechanism can improve the syntactic and semantic word relationships accuracy prediction.

4. CONCLUSIONS

The main key contribution of this work is to obtain a syntactic and semantic prediction. The proposed NLP architecture is represented by combining two powerful deep learning models; BiLSTM and attention mechanism, showing how to train distributed representations of words and phrases to make precise analogical reasoning possible. It is successfully trained the models on

several terms and phrases. With more improvements to the current model that the author planned to achieve in future work, the technique can be used to train the large-of-words models and for the syntactic and semantic prediction fields. Thanks to the attention mechanism for its computationally efficient architecture to deal with critical training data set.

REFERENCES

[1] Q. A. Shreda and A. A. Hanani, "Identifying Non-functional Requirements from Unconstrained Documents using Natural Language Processing and Machine Learning Approaches," in IEEE Access, doi: 10.1109/ACCESS.2021.3052921.

[2] Z. Ke, J. Sheng, Z. Li, W. Silamu and Q. Guo, "Knowledge-Guided Sentiment Analysis Via Learning From Natural Language Explanations," in IEEE Access, vol. 9, pp. 3570-3578, 2021, doi: 10.1109/ACCESS.2020.3048088.

[3] R. Liu, B. Sisman, F. Bao, J. Yang, G. Gao and H. Li, "Exploiting Morphological and Phonological Features to Improve Prosodic Phrasing for Mongolian Speech Synthesis," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 274-285, 2021, doi: 10.1109/TASLP.2020.3040523.

[4] A. Jettakul, C. Thamjarat, K. Liaowongphuthorn, C. Udomcharoenchaikit, P. Vateekul and P. Boonkwan, "A Comparative Study on Various Deep Learning Techniques for Thai NLP Lexical and Syntactic Tasks on Noisy Data," 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2018, pp. 1-6, doi: 10.1109/JCSSE.2018.8457368.

[5] D. Vasić and M. K. Vasić, "Syntax-aware Neural Semantic Role Labeling for Morphologically Rich Languages," 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2020, pp. 1-6, doi: 10.23919/SoftCOM50211.2020.9238179..

[6] Z. Kong, C. Yue, Y. Shi, J. Yu, C. Xie and L. Xie, "Entity Extraction of Electrical Equipment Malfunction Text by a Hybrid Natural Language Processing Algorithm," in IEEE Access, vol. 9, pp. 40216-40226, 2021, doi: 10.1109/ACCESS.2021.3063354.

[7] C. Fan, C. Yuan, L. Gui, Y. Zhang and R. Xu, "Multi-Task Sequence Tagging for Emotion-Cause Pair Extraction Via Tag Distribution Refinement," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 2339-2350, 2021, doi: 10.1109/TASLP.2021.308983.

[8] Y. Hu, X. Qiao, L. Xing and C. Peng, "Diversified Semantic Attention Model for Fine-Grained Entity Typing," in IEEE Access, vol. 9, pp. 2251-2265, 2021, doi: 10.1109/ACCESS.2020.3046787.

[9] A. A. Syed, F. L. Gaol and T. Matsuo, "A Survey of the State-of-the-Art Models in Neural Abstractive Text Summarization," in IEEE Access, vol. 9, pp. 13248-13265, 2021, doi: 10.1109/ACCESS.2021.3052783.

[10] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian and N. Al-Nabhan, "T-BERTSum: Topic-Aware Text Summarization Based on BERT," in IEEE Transactions on Computational Social Systems, doi: 10.1109/TCSS.2021.3088506.

AUTHOR

Fadya Abbas

Received the BSc. Degree in computer engineering from Nahrain University, Baghdad, Iraq, in 2011. Her research interest in Deep Learning includes Natural Language Processing and Computer Vision and their applicability in different real-life applications. She is currently based in Edmonton, AB, Canada, applying to many fruitful Machine Learning courses – Deep Learning and computer programming.

COMPARING METHODS FOR EXTRACTIVE SUMMARISATION OF CALL CENTRE DIALOGUE

Alexandra N. Uma and Dmitry Sityaev

Connex One, 27 Quay Street, Manchester, M3 3GY, United Kingdom

ABSTRACT

This paper provides results of evaluating some text summarisation techniques for the purpose of producing call summaries for contact centre solutions. We specifically focus on extractive summarisation methods, as they do not require any labelled data and are fairly quick and easy to implement for production use. We experimentally compare several such methods by using them to produce summaries of calls, and evaluating these summaries objectively (using ROUGE-L) and subjectively (by aggregating the judgements of several annotators). We found that TopicSum and Lead-N outperform the other summarisation methods, whilst BERTSum received comparatively lower scores in both subjective and objective evaluations. The results demonstrate that even such simple heuristics-based methods like Lead-N can produce meaningful and useful summaries of call centre dialogues.

KEYWORDS

Information Retrieval, Text Summarisation, Extractive Summarisation, Call Centre Dialogues.

1. INTRODUCTION

In the last decade, the rate of adoption of AI technology in the contact centre space has been on the increase. The popularity of speech analytics products is not surprising. The technology allows call centre managers to quickly assess the performance of their call centre agents. It also allows analysts to extract business insights from conversations. Businesses can also use information extracted by speech analytics to improve future customer journeys and experiences.

Whilst modern Automatic Speech Recognition (ASR) solutions reach a very high level of accuracy for call transcription, producing accurate and succinct call summaries is still a very challenging task, and the area of conversation summarisation remains an active research field. Call summaries are usually short abstracts summarising the interaction between a customer and an agent. It is not uncommon to capture such important information as a customer's reason for a call, their concerns, agent's handling of the call and final call resolution. Once call conversation transcripts have been obtained, call summarisation can be viewed as a text summarisation challenge.

Various approaches have been proposed and applied towards text summarisation. However, they do fall broadly into two categories: extractive summarisation and abstractive summarisation. Extractive summarisation is based on selecting the most important sentences from the text and presenting them in the summary verbatim (i.e. word for word). Abstractive summarisation is based on creating new paraphrased sentences that summarise the text.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 77-87, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120907

In this paper, we focus on extractive methods for call summarisation. There are several advantages associated with extractive summarisation methods. Firstly, most extractive summarisation methods do not require labelled data (labelling is a very time consuming exercise). Secondly, most extractive summarisation methods do not require any training (with some exceptions). Thirdly, algorithms and models used in extractive summarisation do not normally present challenges for production deployment.

Whilst the task of call summarisation can be addressed through exploring techniques commonly used in text summarisation, it is important to bear in mind that call transcripts exhibit certain characteristics. In particular, call centre conversations normally involve two or more people (e.g. an agent and a customer). Call centre transcripts arising as a result of the application of the ASR technology normally lack punctuation, and punctuation often needs to be restored in order for the summary to be made more comprehensible (Figure 1 illustrates a typical call centre dialogue summarisation pipeline). Additionally, conversations are characterised by such phenomena as hesitations, speech restarts, ill-formed sentences, etc. These aspects need to be additionally addressed when producing call conversation summaries.

In this paper, we evaluate several techniques that can be used to produce call summaries based on call transcripts. We present results from objective and subjective evaluations carried out on our data and point out benefits and limitations of various approaches. The paper is organised as follows. Section 2 briefly surveys previous work on extractive summarisation. Section 3 provides an explanation of how different methods work and the evaluation criteria and tools used. Section 4 provides experimental set up and results for the objective evaluation, whilst Section 5 provides experimental set up and results for the subjective evaluation. Section 6 offers a discussion of the results, stating the limitation of the methods tested. Finally, Section 7 provides a summary of the paper outlining the conclusions, limitations as well as directions for future work.

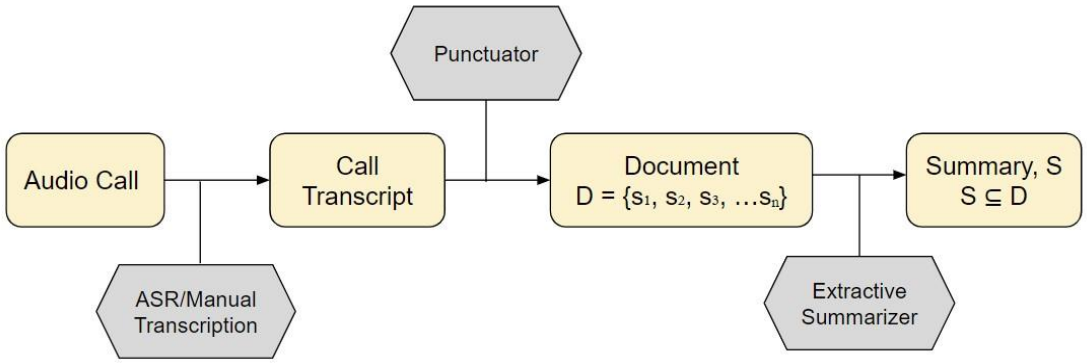


Figure 1. A Typical Call Centre Dialogue Summarization Pipeline

2. RELATED WORK

As mentioned above, research on text summarisation is broadly divided into two approaches: extractive summarisation and abstractive summarisation. Below we survey some of the previous work carried out on extractive summarisation.

[1] was one of the first attempts to introduce extractive summarisation for texts. The method is based on extracting important sentences using such features as word frequency and phrase frequency. Common words of high frequencies are ignored in this approach.

[2] and [3] explored the application of HMMs (Hidden Markov Models) for the task of extractive summarisation. The method is based on the model computing the likelihood that a sentence should be included in the summary or not. Only the sentences with maximum posterior probability are selected.

[4] and [5] approached the task of text summarisation using topic models. Latent Semantic Analysis (LSI) technique is used to identify semantically important sentences. Thus, sentences selected for the summary are characterised by minimum redundancy.

[6] attempted to apply fuzzy logic towards the task of text summarisation. 8 most important features are selected and calculated for each sentence first. Fuzzification and inference rules are then applied, with the defuzzification step producing a sentence score. A set of sentences with highest scores is then extracted for the summary.

Neural networks have also been explored for producing summaries. [7] have successfully used RNNs (Recurrent Neural Networks) to carry out extractive summarisation of documents.

More recently, transformer-based models have been considered and applied to text summarisation. [8] used BERT (Bidirectional Encoder Representations from Transformers) model to summarise lecture notes. [9] describes further improvements to the BERT model by way of creating a fine-tuned summariser.

[10] specifically address the task of summarisation of call transcripts for call centres. They propose a novel method which combines call channel separation, topic modelling, sentence selection and punctuation restoration.

A good summary of text summarisation techniques can be found in [11], [12], [13], [14], [15].

3. CONCEPTS AND TERMINOLOGY

3.1. Lead-N

Lead-N is an extension of the well-used summarisation baseline Lead-3. Lead-N first filters away sentences that contain fewer than 7 non-stop words, then selects the first N number of sentences for the summary. For our purposes, as per user specification, we set the number of sentences for each model at 7. Henceforth, this model would be referred to as Lead-7.

3.2. Text Rank

Text rank is a graph-based ranking algorithm proposed by [16] . Text data is split into sentences, and a similarity matrix is created. The similarity matrix is then converted into a graph where sentences are nodes and similarity scores are edges. Top ranked sentences then selected for extractive summary.

3.3. KLSum

The KLSum (Kullback-Leibler Sum) algorithm iteratively adds sentences to the summary by selecting at any time step, a sentence that minimises the KL divergence between the candidate summary and the unigram distribution of the original document. An extensive analysis of this model was carried out by [17].

3.4. BERTSum

BERTSum is a transformer-based extractive summarisation model proposed by [8]. BERTSum first filters away "too long" or "too short" sentences from the document, then encodes the remaining sentences using the BERT large model, representing each sentence as the averaged embeddings of its tokens. The model then clusters these embedded sentences into N clusters, where N is the number of sentences required for the summary. A summary of the call document is created by taking the centroid sentence for each cluster.

3.5. TFIDFSum

TFIDFSum scores each sentence as the sum of the tf-idf scores of the tokens in that sentence. The sentences with the highest scores are then selected for the summary. This method is often used as a baseline summarisation method.

3.6. TopicSum

TopicSum, like TFIDFSum, scores each sentence by scoring its tokens. However, rather than using tf-idf scores for tokens, a topic model is applied to the document, and the score of each sentence is the sum of the scores assigned to its tokens by all the topics. We found the Latent Dirichlet Allocation (LDA) topic model [18] with the number of topics set to 15, to perform best for our purposes.

3.7. RBMSum

The RBMSum approach proposed by [19] is based on extracting features from each sentence (sentence position, sentence length, etc.). These features are then enhanced using an RBM (Restricted Boltzmann Machine), and sentence scoring is then created using the sum of the enhanced features.

3.8. ROGUE-L

ROGUE (Recall-Oriented Understudy for Gisting Evaluation) is a metric often used in objective evaluation of text summaries [20]. The ROUGE-L version of this metric is a function of the longest common subsequence of between the produced summary and the reference summary (also often referred to as the gold summary). Summaries that share a longer sequence with gold summaries tend to have a higher ROUGE-L score. This metric is widely preferred for extractive summarisation evaluation.

3.9. MOS

MOS (Mean Opinion Score) is a popular measure for representing the quality of a system or a stimulus. It is a rating which is obtained by averaging scores from a subjective evaluation test.

3.10. Label Studio

Label Studio is an open source data labelling tool for labelling video, audio, image, time series, as well as text data[21]. We used this tool to collect annotator judgement in Experiment 2.

4. EXPERIMENT 1

In this section, we describe the objective evaluation of the methods discussed in Section 3 using the ROUGE-L metric (see section 3.8). Section 4.1 outlines the details of the experiment such as the dataset used and process of creating reference summaries. Section 4.2 provides the results of the experiments, and we briefly discuss them with the reference to the evaluation metric. A detailed discussion of the results is provided in Section 7.

4.1. Method

In this subsection, we discuss the dataset used for this experiment and the procedure for evaluating the models with relation to gold summaries.

4.1.1. Data

For this evaluation experiment, we selected 15 calls with an average duration of 15 minutes at random from the *mobile phone*s domain. For each of these calls we (a) create call documents (b) produce a gold summary from the call document. These steps are detailed below:

A. Creating Call Documents

1. Manual Call Annotation. We listened to the calls and manually annotated them using lower case letters only and no punctuation. This is the format typically used by ASR to output call transcriptions.
2. Punctuation Restoration. The next step was to restore punctuation to the call transcripts. We did this using a python package from Hugging Face [22]. This BERT-based punctuator comprises the BERT-base encoder with an additional linear layer. This layer takes the encoded input (the text stream) and for each token predicts whether or not it is followed by a punctuation mark. With punctuation in place, sentence segmentation was carried out using the Spacy Sentencizer [23].

B. Producing Gold Summaries of Calls

Reference summaries for each single document version of the call were produced by manually selecting the 7-10 most relevant sentences for each call. We aimed at including the following information in the summary: (1) the reason for the call, (2) pertinent information unique to the call, and (3) the call resolution. We consider these reference summaries as the ‘gold standard’ for our purposes.

4.1.2. Procedure

The evaluation procedure was as follows:

1. We produced summaries for the 15 documents using the models described in Sections 3.1 - 3.7.
2. We computed the ROUGE-L scores between these summaries and the manually generated gold standard summaries. The results are presented below in Section 4.2.

4.2. Results

Table 1 contains the results of evaluating all the models discussed in Sections 3.1 - 3.7 on their ability to produce summaries that closely match the gold summaries using ROUGE-L. As can be seen from the table, although TFIDFSum has a higher recall, Lead-7 outperforms the other models in precision and F1 score.

Table 1. ROUGE-L evaluation of models relative to gold summaries

Model Name	Precision	Recall	F1
Lead-7	**0.532**	0.405	**0.449**
TextRank	0.499	0.414	0.441
TFIDFSum	0.460	**0.428**	0.429
TopicSum	0.459	0.423	0.427
BERTSum	0.510	0.340	0.397
KLSum	0.521	0.329	0.386
RBMSum	0.465	0.280	0.340

5. EXPERIMENT 2

In this experiment, we evaluated the effectiveness of the summarisation methods using subjective judgements of human annotators. The MOS was used to aggregate these judgements.

5.1. Method

This subsection contains a discussion of the data, data preparation and procedure for this subjective evaluation experiment. In this subsection, we discuss the dataset used for this experiment and the procedure for the subjective evaluation of the models.

5.1.1. Data

For this evaluation experiment, we selected 8 calls from 5 domains – *mobile phones*, *life insurance*, *debt collection*, *home improvements*, and *solar panel funding*. The average duration of these calls is 11 minutes. The data was processed in the same manner as the data in Experiment 1 (see 4.1.1) and the summaries produced were used for the experiment outlined in 5.1.2.

As a preliminary step, we reduced the number of models from 7 models to 4. The 4 models were selected by examining the summaries produced by the models during Experiment 1. This was done by subjectively ranking the models in order of how well the summaries produced met the gold standard – in other words we asked and answered the question, 'how well did they meet the criteria by which we created gold summaries' (see Section 3.1.1). Based on this, we chose Lead-7, BERTSum, TopicSum and RBMSum. Although TextRank performed competitively with respect to ROUGE-L scoring in Experiment 1, the summaries produced by TextRank for the *life insurance* domain were poor (see Section 6.1). For this reason, it was decided not to include TextRank in subjective evaluation.

We produced summaries of the calls outlined above using the four methods selected.

5.1.2. Procedure

Label Studio was used to collect subjective judgements from the annotators about the summaries produced by various models. Our goal was to conduct a 2 hour experiment to ascertain which model's summaries the participants preferred. The participants were drawn from the data science and transcription teams of the company – these participants work on various aspects of call centre data intelligence.

The procedure of this experiment was as follows:

1. We carried out an initial pilot experiment to ensure that the task setup was intuitive. This pilot experiment had 1 participant. This participant was asked to listen to 8 calls (one at a time) and rank each of the four summaries of any given call on a scale of 1-10.
2. We carried out the main experiment with 6 participants. The participants were asked to repeat the steps from the pilot experiment. The names of the models were removed to facilitate an unbiased annotation. It is also important to note that the participants did not have access to the text transcript of the call, or the gold summaries. They had to make their judgement solely on the basis of how well they thought the written summary captured the audio recording of the call.
3. The judgments of the participants were aggregated using the Mean Opinion Score (MOS). The results of this are shown in Table 2 in the section that follows.

5.2. Results

Table 2 contains the aggregated results of the subjective judgements of human annotators. The Table places the model name side by side with the mean opinion score for each model (see Section 3.9). As can be seen from the table, TopicSum emerges as the preferred summarisation model, with the highest MOS score of 5.96.

Table 2. Aggregated results of the subjective judgements of human annotators.

Model Name	Mean Opinion Score
TopicSum	**5.96**
Lead-7	5.14
RBMSum	4.20
BERTSum	3.66

6. DISCUSSION

In this section, we discuss the results of the experiments in two parts. Firstly we discuss the suitability of the 7 models evaluated in Section 5. For each method, we analyse the results of Section 6, stating the benefits and limitations of the model. Secondly, we briefly compare subjective and objective evaluation of summaries.

6.1. Comparing extractive summarisation methods of call centre dialogue summarisation

Table 1 shows that Lead-7 received the highest F1 score compared to other methods. This is likely because by favouring sentences that occur at the beginning of the call, Lead-7 usually

captures the reason for the call, which is an important part of gold summaries. For subjective evaluation, Lead-7 was shown to be the second most preferred method. Taken together, these results show that Lead-7 can be a very competitive baseline for call centre dialogue summarisation. On the negative side, a post-hoc analysis revealed that Lead-7 received lower subjective ratings scores when the sentences in the summary were too long resulting in a "wordy summary". Lead-7 was also revealed to frequently miss the call resolution, a side-effect of selecting sentences from the beginning of the call document.

From Table 1, we see that Lead-7 outperformed TextRank by a small margin when evaluated using ROUGE-L. In fact, a side by side examination of summaries produced by the two models showed that Lead-7 and TextRank produced summaries with the highest degree of token overlap. One possible reason for the competitive ROUGE-L scores for TextRank is that by preferring sentences with the highest similarity to other sentences, TextRank selects sentences with the most diverse words, hence capturing an extensive vocabulary, and Rouge-L being an n-gram based scorer rewards this behaviour. An examination of the summaries produced by TextRank showed them to be highly coherent but with the disadvantage of lacking in topic diversity. Furthermore, for longer calls where several topics are discussed, TextRank often settles on one aspect of the call, ignoring other usually important aspects. For these reasons, TextRank was not included in Experiment 2.

As already mentioned, the heuristic model TFIDFSum works by ranking sentences according to the portion of important tokens they contain: where token importance is computed as token term-frequency inverse document frequency. The competitive ROUGE-L score obtained for this method suggests that by maximising the sum of this token tfidf, TFIDFSum encourages the selection of longer sentences, especially ones that contain weighty words. Also, because high tfidf words are frequently occurring, they are likely to be found across the entire document. Thus, the technique of choosing high-weighted sentences also encourages high coverage summaries. TopicSum, designed to improve on the word importance scoring of TFIDFSum also achieves similar ROUGE-L scores and produces very similar summaries. However, TopicSum produced summaries that contained more call-specific details and for this reason, it was selected for subjective evaluation over TFIDFsum.

Table 1 shows KLSum to have a poor ROUGE-L score – it ranks in the bottom two models according to the metric. As noted in Section 3.2, KLSum aims at producing a summary that best matches the entire document (in our case, the whole call). It does so by selecting a candidate summary with the least divergence from the unigram distribution of the document. This approach, while encouraging diversity, has no notion of topic or word importance and leads to a poor recall of important but succinct topical sentences. Analysis of KLSum summaries revealed they were lacking in coherence, often drifting from topic to topic. This led us to the exclusion of KLSum from Experiment 2.

BERTSum also received a low ROUGE-L score compared to other methods. BERTSum creates N clusters for each document, and selects N sentences closest to each centroid as the document summary. Thus, similarly to KLSum, BERTSum also encourages diversity, however a notion of topicality is enforced by clustering in this case. An examination of BERTSum summaries showed them to also be coherent but lacking in crucial details pertinent to the calls. This lack of detail, possibly a side-effect of the initial filtering process (see Section 3.4) was noted by the annotators in Experiment 2 and is the likely reason for the comparatively low ROUGE-L score for the model.

The last model discussed in the section is RBMSum, which has the worst ROUGE-L score. An examination of the summaries produced by RBMSum shows that RBMSum, like BERTSum,

selects shorter sentences, hence producing relatively shorter summaries. That notwithstanding, RBMSum summaries neatly capture the highlights of the calls.

6.2. Comparing objective and subjective evaluation

In comparing the objective and subjective evaluation results, we observe two key differences:

1. Although RBMSum received the lowest ROUGE-L score in the objective evaluation, in Experiment 2, the annotators rated RBMSum higher than BERTSum.
2. TopicSum was preferred over Lead-7 by human annotators, even though it scored lower than Lead-7 according to the ROUGE-L metric.

This seems to indicate, in line with findings of the perspectivist view of NLP [24] [25], that the gold standard might be just one opinion of what constitutes a good summary. This suggests that while metrics like ROUGE-L are useful for comparing models, the choice for best summary might be user/purpose dependent.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we experimentally compared the use of several extractive summarisation models in building a call centre dialogue summarisation pipeline without depending on gold summaries for model training. To summarise the audio recordings, we convert each call to a document of sentences and apply an extractive summarisation model to extract the most important sentences from this document. We observed that models that take word importance into consideration produce summaries which are most similar to the gold summaries. We also observed that even simple baselines like Lead-7 can produce good summaries of calls. We evaluated the quality of the summaries by aggregating subjective judgements of human annotators. Comparing objective and subjective evaluation of the summaries suggest that both are needed to ascertain the suitability of summarisation models.

We limited the scope of our work in this paper to extractive summarisation techniques as these techniques are often unsupervised and as such neither need labelled training nor present serious challenges for production deployment. Our experiments are also limited in the number of calls annotated, and the expertise of the annotators who, while being well-versed in call centre data transcription and information extraction are not call centre agents or managers. Further research would involve (1) evaluating our results on a larger set of annotated calls of varying lengths with annotation guidelines produced by call centre managers, and (2) extending the scope of our work to include supervised extractive summarization techniques which we will train on dialogue datasets like TweetSum[26] and evaluate on our call centre calls.

ACKNOWLEDGEMENTS

The authors would like to thank everyone at the Connex One data science team, and the transcription team lead for their participation in this research.

REFERENCES

[1] Luhn, Hans Peter (1958) “The automatic creation of literature abstracts”, IBM Journal of Research and Development, Vol. 2, No. 2, pp 159-165.
[2] Mittendord, E. & Schauble, P. (1994) “Document and passage retrieval based on hidden markov models”, Proceedings of the 17th ACM-SIGIR Conference, pp 318-327.

[3] Conroy, John & O'Leary, Dianne (2001) "Text summarization via hidden markov models", Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp 406-407.
[4] Gong, Yihong & Liu, Xin (2001) "Generic text summarization using relevance measure and latent semantic analysis", Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp 19-25.
[5] D. Wang, S. Zhu, T. Li & Y. Gong (2009) "Multidocument summarization using sentence-based topic models", Proceedings of the ACL-IJCNLP Conference, pp 297-300.
[6] L. Suanmali, N. Salim, M. S. Binwahlan (2009) "Fuzzy logic based method for improved text summarization", Computing Research Repository, arXiv: 0906.4690v1.
[7] R Nallapati, F. Zhai & B. Zhou (2017) "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents", Proceedings of the 31st AAAI Conference on Artificial Intelligence, pp 3075-3081.
[8] Miller, Derek (2019) "Leveraging BERT for text summarization on lectures", Computing Research Repository, arXiv: 1906.04165v1.
[9] Liu, Yang (2019) "Fine-tuning BERT for extractive summarization", Computing Research Repository, arXiv: 1903.10318v2.
[10] Biswas, Pratik & Iakubovich, Aleksandr (2020) "Extractive summarization of call transcripts", Computing Research Repository, arXiv: 2103.10599v2.
[11] Jones, Karen Sparck (2007) "Automatic Summarizing: The state of the art", Information Processing and Management, Vol. 43, No. 6, pp 1449-1481.
[12] Jezek, Karel & Steinberger, Josef (2008) Automatic Text Summarization, FIIT STU Bratislava, Ustav Informatiky a softveroveho inzinierstva.
[13] Gupta, Vishal & Lehal, Gupreet Singh (2010) "A survey of text summarization extractive techniques", Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, pp 258-268.
[14] Gaikwad, Deepali & Mahender, C. Namrala (2016) "A review paper on text summarization", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 3, pp 154-160.
[15] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. Trippe, J. Gutierrez & K. Kochut (2017) "Text summarization techniques: A brief survey", Computing Research Repository, arXiv: 1707.02268v3.
[16] Mihalcea, R., & Tarau, P. (2004) "TextRank: Bringing Order into Text". Empirical Methods in Natural Language Processing (EMNLP)
[17] Aria Haghighi & Lucy Vanderwende (2009) "Exploring content models for multi-document summarization". In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09), pp 362–370.
[18] Blei M. David, Andrew Y. Ng, and Michael I. Jordan (2003) "Latent dirichlet allocation". Journal of Machine Learning Research. Vol 3, pp 993–1022.
[19] Sukriti Verma., & Vagisha Nidhi (2018). Extractive Summarization using Deep Learning. Res. Comput. Sci., Vol 147, pp 107-117.
[20] Lin, Chin-Yew (2004), "ROUGE: A Package for Automatic Evaluation of Summaries.". Association for Computational Linguistics. pp 74-81
[21] Label Studio, https://labelstud.io/
[22] Hugging Face Punctuator: https://huggingface.co/felflare/bert-restore-punctuation
[23] Spacy Sentenciser, https://spacy.io/api/sentencizer
[24] Basile, Valerio, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio and Alexandra Uma (2021) "We Need to Consider Disagreement in Evaluation." Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future, pp 15–21
[25] Uma, Alexandra Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank and Massimo Poesio (2021), "Learning from Disagreement: A Survey." Journal of Artificial Intelligence Research (JAIR), Vol 72, pp 1385–1470
[26] Feigenblat, Guy, R. Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki and Ranit Aharonov. "TWEETSUMM - A Dialog Summarization Dataset for Customer Service." Findings of the Association for Computational Linguistics: EMNLP 2021, pp 245-260

AUTHORS

Alexandra Uma Alexandra received a PhD in Computer Science from Queen Mary University of London in 2021. She is currently working as a data scientist at Connex One

Dmitry Sityaev He received MPhil in General Linguistics from the University of Oxford in 1999 and MSc in Data Science from the University of London in 2019. He is currently working as Director of AI at Connex One.

LEARNING TO PRONOUNCE AS MEASURING CROSS-LINGUAL JOINT ORTHOGRAPHY-PHONOLOGY COMPLEXITY

Domenic Rosati

scite.ai Brooklyn, New York, USA

ABSTRACT

Machine learning models allow us to compare languages by showing how hard a task in each language might be to learn and perform well on. Following this line of investigation, we explore what makes a language "hard to pronounce" by modelling the task of grapheme-to-phoneme (g2p) transliteration. By training a character-level transformer model on this task across 22 languages and measuring the model's proficiency against its grapheme and phoneme inventories, we show that certain characteristics emerge that separate easier and harder languages with respect to learning to pronounce. Namely the complexity of a language's pronunciation from its orthography is due to the expressive or simplicity of its grapheme-to-phoneme mapping. Further discussion illustrates how future studies should consider relative data sparsity per language to design fairer cross-lingual comparison tasks.

KEYWORDS

Phonology, Orthography, Linguistic Complexity, Grapheme-to-Phoneme, Transliteration.

1. INTRODUCTION

The aim of this work is an initial computational exploration of what makes a language "hard to pronounce" as suggested in the discussion for the SIGMORPHON 2020 shared task of grapheme-to-phoneme (g2p) transliteration [1]. Specifically, we will look at learning to transliterate several languages into the international phonetic alphabet (IPA), a form of g2p transliteration, as an indicator of language complexity. In order to do this, we train a character-level transformer model on g2p transliteration for 22 languages and measure the ratio of grapheme-to-phoneme used in languages against the character-level accuracies per word that is achieved by training the models under medium-resource assumptions. In the analysis below, we show that this method can empirically demonstrate intuitive notions of cross-lingual complexity such as phonetic orthographies being easier to learn than non-phonetic ones. We also show that data sparsity is a key issue in making cross-lingual comparisons that hasn't been well addressed in previous literature. Specifically, we will argue that it is unfair to compare languages with different inventories using the same training and testing set sizes.

Transliteration is defined here as the process of finding a mapping from one source language orthography to a target orthography. Sometimes the target transliteration alphabet is a simplification or variation of the source language orthography such as using a simplified alphabet, or an orthography designed to elucidate pronunciation as can be found in simplified English pronunciation schemes. Other times transliteration is used to map language artifacts native to one language to a transliteration that elucidates its pronunciation in another alphabet for

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 89-98, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120908

readers of the target language. For example, in romanization a transliteration of a name natively belonging to one language is written in a “roman” or Latin-based script such as English in order to be readable by a reader of a Latin script or is written under constraints such as only having access to a Latin script-based keyboard. In this work we specifically focus on the g2p transliteration task that preserves a source language's native orthography and outputs letters drawn from IPA. It is our belief that setting up the task in this way ensures that we are learning informative end-to-end pronunciation transliterations that are useful for comparing languages. Previous g2p task set ups have allowed human or automated romanization of source orthographies (see descriptions of this in [1]) which we believe is detrimental to measuring language complexity. This is particularly important in orthographies such as used by Mandarin where romanization such as in Pinyin is giving the language a “phonetic” head start. Since we don’t allow English to be reduced to a simpler phonetic variety, why should we allow other languages to be?

2. PREVIOUS WORK

For natural language processing, transliteration models provide many unique and desirable qualities for inspecting language learning. First g2p language models have had a particular historical importance due to their role in speech to text and automatic speech recognition systems. Second, phonology and orthography are well studied areas. Theoretical models about how sequences of characters should be composed orthographically and how pronunciation occurs are readily available for researchers to guide their modelling of the task of transliteration including how feature representations, loss, and evaluation functions should be designed in order to preserve features such as phonetic similarity [2]. Third, unlike the high dimensionality of word and audio feature spaces, transliteration models have relatively low complexity in their feature spaces as they are bounded by comparatively low alphabet sizes. Because of this transliteration models may provide interesting opportunities for inspecting tasks that traditionally would have required more costly audio inputs or outputs. Additionally, the transliteration process is itself interesting as humans commonly need to engage in understanding and explaining the pronunciation of written texts and the process of learning to transliterate requires some competency with the source and target orthographies that may elucidate the dynamics of the language that is being modelled. Finally, unlike other language modelling tasks, transliteration models often focus on achieving good results in low resource scenarios where 1k (low resource) or 10k (medium resource) training samples are available. Insights for modelling with little data then may be drawn from work in transliteration.

In our work, one aspect that we’d like to study is what transliteration models can tell us about language complexity. Language complexity is a well-studied field that asks whether some languages or dimensions of language such as its syntax or phonology can be more complex or harder to learn than others [3]. Typically, this is based on classical linguistic analysis of inventories of the number of features and rules available within a phonology, orthography, or syntax system. Recent studies, notably [4] and [5], have proposed that cross-lingual complexity can be framed as a learning problem: How hard is it to learn a model of reading, writing, or pronouncing? Based on the accuracy of a particular modelling applied across languages we can potentially draw empirical conclusions of how complex one language might be with respect to another. In this respect, [4] is the closest analysis to this work that tries to understand how hard one language might be to transliterate (grapheme-to-phoneme) or decode (phoneme-to-grapheme). In our work, we look particularly at g2p transliteration to draw conclusions about joint orthography and phonology complexity.

3. METHODS

For our g2p transliteration task, we chose to use the ipa-dict dataset of which consists of monolingual wordlists with IPA transliteration in 28 languages [6]. The alternative dataset under consideration was the Wiktionary project which has IPA phonetic transliterations available and is much more comprehensive in terms of the number of transliteration records and languages available. A version of this has been curated in the Wikipron project that has been used in various SIGMORPHON workshops. However, for the purposes of an initial investigation the smaller ipa-dict dataset was chosen as we found it exemplary of many languages in an easy to access way.

Given the ipa-dict data, we pre-processed the dataset by removing punctuation, splitting tokens by individual characters, and separated each individual dataset by language. Removing punctuation and splitting tokens by individual characters is problematic from a phonetic perspective since punctuation can change pronunciation and individual phonemes can be composed of multiple characters for instance in the case of gnocchi, an Italian pasta, whose phonetic transliteration /ˈɲɔk.ki/ transforms the “gn” onto the voiced palatal nasal /ɲ/ and ch into the voiceless velar plosive /k/. Similarly, the comma in English can transform how adjacent words are blended during speech. In the case of tokenizing IPA transcriptions, we split units that have a single phonetic meaning such as the case of diacritics and other components that modify the preceding letter. However, we don’t consider these major obstacles for two reasons. In the tokenization case, we are looking to model transliteration from a sequence-to-sequence perspective where multiple characters can be decoded as single phonemes or single characters can be decoded as multiple phonemes depending on the context of proceeding characters and language under transliteration. For punctuation, since we are focusing here on word-to-word transliterations we don’t consider punctuation a major feature in this task.

Our aim to model transliteration is to look at how the process of transliteration can be learned across languages. Therefore, we split the dataset into the 22 languages which have over 10k transliteration records. We select a random sample of 10k transliteration records for each language. By focusing on a 10k set, we are emulating the medium resource sample size given in previous SIGMORPHON conferences [1]. We do this so that each transliteration model has the same number of samples to train from. In order to train the model that we describe below, we split the dataset into a training set of 8k samples, an evaluation test set of 1k samples, and a final test set of 1k samples.

The architecture we chose to model transliteration was the popular transformer architecture [7]. We chose the transformer architecture because it provides a proven approach to model sequence transduction where positional context is considered across the sequence using attention and self-attention that has been proven out in both character level [8] and transliteration contexts [9] achieving state-of-the-art g2p performance. The transliteration process can be understood as sequence transduction where a source language orthography for a given word is translated into a target language orthography. With attention, we theorize that we will be able to adequately translate a phonetic letter given both previous and preceding characters in the input sequence. Additionally, with self-attention, we theorize that the decoder can learn additional phonetic transliteration procedures that ensure that certain phonetic sequences are more likely than others. For instance, learning to attend to certain sequence dependencies like vowel harmony. Finally, with multi-head attention and multiple layers we theorize that multiple levels of phonetic and orthographic features can be captured in understanding the input sequence and producing the output sequence.

Table 1 describes the parameters used to train a transformer model using the architecture presented in [7]. The learning rate was set to a uniform rate of 0.005 using an Adam optimizer

and dropout rate of 0.1 was used. The implementation of Vaswani [7] was made in PyTorch and the code is available on request. One thing to note is that the transformers used are relatively shallow and trained for a relatively low duration of 20 epochs. The choice of a shallow model was due to the relatively limited amount of training data and speed at which the model was able to converge on a relatively stable loss for each language. Larger models were underfitting and slow to converge. However, in the discussion below we theorize that this could potentially be resolved with self-supervision pretraining or a larger dataset such as Wiktionary. A large batch of 512 was chosen due to [8] which suggests that for character-level transformers batch size is critical especially in the case of low resource settings where transliteration tasks often take place.

Table 1. Parameters used to train each transliteration model.

Dropout Rate	Learning Rate	Batch Size	Embedding dimension	Layers	Feed Forward Size	Attention Size	Number of Attention Heads
0.1	0.005	512	32	2	32	32	2

4. RESULTS

We trained 22 transliteration models in order to compare the ability to learn transliteration for different languages. Once the models were trained, they were evaluated on their respective test sets using a simple character-level accuracy metric as seen in [10]. The accuracy metric for each predicted sequence is the mean of per-token comparison of the predicted IPA transliteration and the ground truth transliteration. Simple character level transliteration was chosen over the more common word and phoneme error rate (See [9]) for demonstrative purposes in this paper rather than to prove performance against some baseline. In future studies, word and phoneme error rate could be more illustrative of performance and provide standard comparative metrics. In Table 2, we present these accuracies across each language we trained a transliteration model for. Notably, Esperanto, Malay, Swahili, and Vietnamese are the most accurately transliterated while Cantonese, Japanese, and Mandarin (hans indicates simplified and hant indicates traditional) are the least accurately transliterated. In order to explore these results further we had the following questions on what might contribute to the differences of accuracies.

Following [4], we theorized that the range of letters used by the source language, it's graphemes, and the range of IPA letters used, its phonemes, are potentially illustrative of the complexity involved in the language. We also show the ratio of grapheme to phonemes used including a convenience column that shows the distance from a 1:1 ratio. These numbers are presented below in Table 2 with German and French (Quebec) appearing to use the most IPA sounds and Malay and Esperanto using the least. The smallest source alphabets were Swahili, English (both UK and US), and Malay and the largest were Mandarin. As for the ratio of sounds used to source alphabet length, German, English (UK), and Swahili had the highest ratio of sounds used per letter with Esperanto, French (France), Spanish (Both Mexico and Spain) having the closest to a 1:1 ratio and Mandarin and Cantonese having the lowest ratio. Figure 1 explores these results graphically presenting the ratio of IPA letters used to source language letters used plotted against the accuracy of each model. One clear limitation of this approach is that a simple ratio of grapheme to phoneme does not robustly indicate how close a language is to representing phonemes in a 1:1 way. An example of this is Arabic, which appears in relatively close alignment but in this dataset is not using any graphemes for vowels as is common in standard written Arabic.

Table 2. Transliteration model for languages, their vocabulary length, and ratio of number of IPA tokens used to source vocabulary tokens used.

Language	Accuracy	IPA Vocab Length	Source Vocab Length	Ratio of IPA to Source Length	Distance from 1:1 ratio
Mandarin (hant)	**71.98%**	41	**23283**	0.002	**1.00**
Mandarin (hans)	73.88%	41	20505	0.002	1.00
Cantonese	79.95%	33	14672	0.002	1.00
Japanese	73.66%	32	5510	0.006	0.99
Vietnamese (Southern)	95.53%	43	90	0.478	0.52
Vietnamese (Northern)	95.65%	43	90	0.478	0.52
Vietnamese (Central)	96.32%	45	90	0.500	0.50
Odia	95.00%	38	63	0.603	0.40
Arabic	87.10%	32	38	0.842	0.16
Esperanto	97.08%	27	29	0.931	0.07
French (France)	92.12%	43	46	0.935	0.07
Spanish (Mexico)	95.68%	32	33	0.970	0.03
Spanish (Spain)	94.85%	33	33	1.000	0.00
Malay	96.87%	30	27	1.111	0.11
Finnish	91.81%	38	34	1.118	0.12
French (Quebec)	90.80%	54	47	1.149	0.15
Norwegian	84.74%	48	34	1.412	0.41
English (US)	80.30%	37	26	1.423	0.42
Swedish	85.42%	48	33	1.455	0.45
Swahili	96.63%	40	24	1.667	0.67
English (UK)	83.92%	47	26	1.808	0.81
German	85.55%	**84**	32	**2.625**	**1.63**

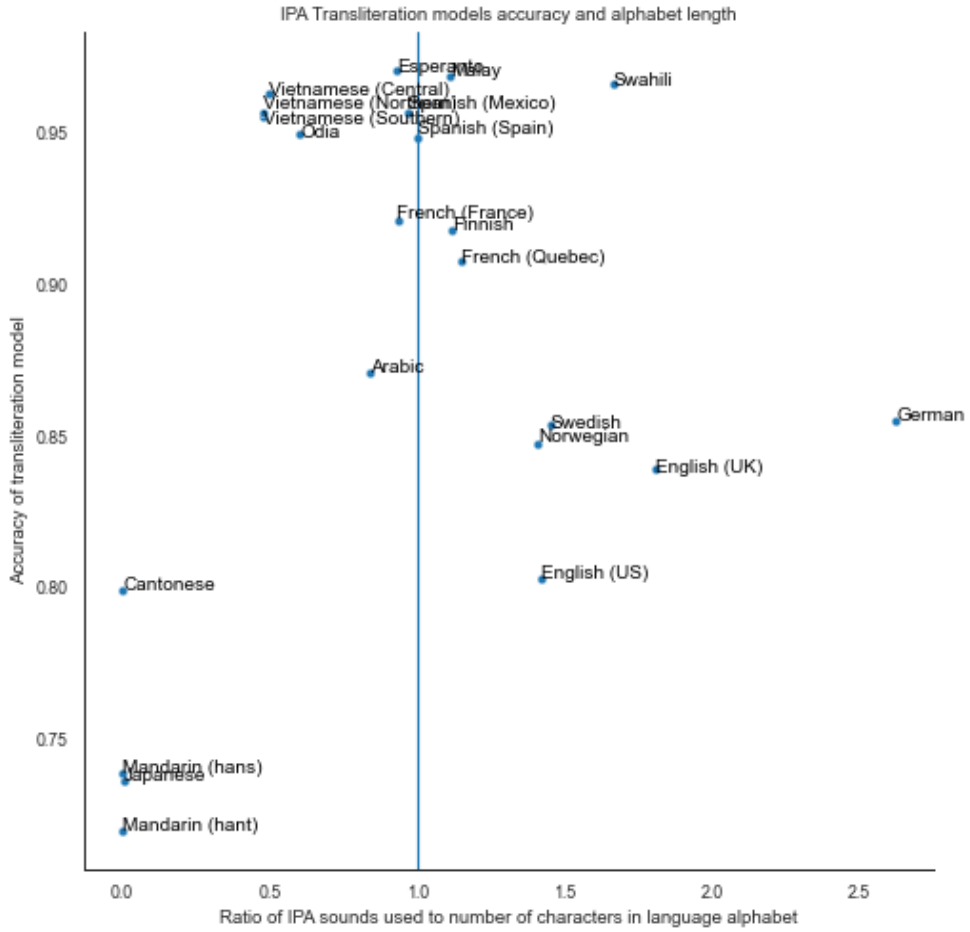


Figure 1. IPA Transliteration model accuracy and the ratio of IPA length to source language alphabet. The blue line indicates a 1:1 ratio for grapheme-to-phoneme.

5. DISCUSSION

When looking at learning to transliterate we are interested in two core competencies. The first is learning to read, deciphering a source alphabet and making some transformation of it, and the second is learning to pronounce, understanding how a source alphabet maps to phonemes. In the above experiments, we are asking how learning to read and pronounce under the g2p task differ across languages. Why does Mandarin appear the hardest to learn and Esperanto, a constructed language, the easiest? The above results provide an empirical justification for two intuitive hypotheses presented below: logographic writing systems are harder to learn to transliterate than phonetic writing systems, second that languages with high redundancy for phonemes or close to a 1:1 ratio of letter to phoneme are the easiest to learn.

First, we should say that the ratio of grapheme-to-phoneme is a very simplistic measure of language complexity, as is demonstrated in the case of Arabic noted above. However, we feel justified in using it at the very least to illustrate a few things about language complexity because

it allows us to look at the size of an input vocabulary (the graphemes) and output vocabulary (the phonemes) as the model being trained would. That is without explicit knowledge about how to evaluate the complexity of a language. We will see this below when we discuss the notion of complexity as it relates to source vocabulary size and data sparsity. Secondly, the below discussions are based on. Another related limitation that should be presented before discussing results is that Latin-script based orthographies are overrepresented in our comparison. Table 3 shows that we include 4 logographic orthographies, 1 abugida, 1 abjad, and 16 Latin-based orthographies. In future works, we should try to study more languages that do not use Latin-based orthographies.

Given the limitations above, the major discrepancies in accuracy results are the differences between languages with logographic writing and languages without. This is an intuitive result and follows naturally from the fact that the number of values the input alphabet can take is orders of magnitudes larger than non-logographic orthographies. If each unique character is understood as a class under a multi-class classification scheme, then we can easily see that some languages are much sparser than others. Summarizing results from the SIGMORPHON 2020 workshop [1] discusses the lack of performance in Korean. The authors note that some orthographies, notably Korean in their dataset, are much sparser than others. Meaning the mapping between the number of training samples and source graphemes is very low or sparse compared to other languages. In order to illustrate the dynamics of this in our dataset we have tabulated the sparsity of each language we are using in Table 3 which shows the number of samples per unique character per language. In this case Mandarin only has 0.34 examples per letter and Japanese only has 1.45 samples per class. The next lowest number of samples per letter in a source language alphabet is Vietnamese with 88 samples per class. We know that the accuracy of a classifier will naturally depend on the number of training samples per class producing underfitting in a situation without enough data. However, we can't simply say that large source vocabularies require more training data to learn transliteration as Vietnamese is one of the easiest languages to learn, achieving an accuracy of 96.32% on Vietnamese (Central) despite only having 88 samples per letter. (Interestingly, Vietnamese was also the lowest baseline word error rate in the SIGMORPHON 2020 task [1]). Again, English (US) has the lowest accuracy score of 80.30% among non-logographic orthography despite having 307 samples per letter. Based on this we can observe that logographic orthographies are harder to learn to transliterate than phonetic ones. We should be wary to suggest that there is a linear relationship between number of letters in a source vocabulary and its ease of learning, at least in the case where non-logographic orthographies are involved.

Table 3. Samples per unique character in alphabet for each language given a training set of 8000 tokens.

Language	Type of Orthography	Accuracy	Number of Unique Characters	Number of Samples per Unique Character
Mandarin (hant)	Logographic	71.98%	23283	0.34
Mandarin (hans)	Logographic	73.88%	20505	0.39
Cantonese	Logographic	79.95%	14672	0.54
Japanese	Logographic	73.66%	5510	1.45
Vietnamese (Southern)	Latin based	95.53%	90	88.88

Vietnamese (Northern)	Latin based	95.65%	90	88.88
Vietnamese (Central)	Latin based	96.32%	90	88.88
Odia	Abugida	95.00%	63	126.98
French (Quebec)	Latin based	90.80%	47	170.21
French (France)	Latin based	92.12%	46	173.91
Arabic	Abjad	87.10%	38	210.52
Finnish	Latin based	91.81%	34	235.29
Norwegian	Latin based	84.74%	34	235.29
Spanish (Mexico)	Latin based	95.68%	33	242.42
Spanish (Spain)	Latin based	94.85%	33	242.42
Swedish	Latin based	85.42%	33	242.42
German	Latin based	85.55%	32	250
Esperanto	Latin based	97.08%	29	275.86
Malay	Latin based	96.87%	27	296.29
English (US)	Latin based	80.30%	26	307.69
English (UK)	Latin based	83.92%	26	307.69
Swahili	Latin based	96.63%	24	333.33

In order to explain why certain languages are easier to learn than others we hypothesize that first and intuitively the phonetic alphabets are easiest to learn. Among those languages with close to a 1:1 mapping between a grapheme and a single phoneme or with many redundancies in terms of multiple graphemes mapping to a single phoneme, or which are highly expressive with respect to how source letters can be transliterated are the easiest to learn. In Figure 1, we see those languages with close to a 1:1 ratio of grapheme to phoneme appear to do the best. Languages which are more expressive in terms of a phonetic writing system, meaning they have multiple and even redundant graphemes to phonemes, also appear to do well. Languages that appear to do poorly like English and German have a low ratio of letters in the source language to phonemes they express. This means that the languages have letters that can map to multiple different phonemes introducing which would have the effect of introducing ambiguity in the g2p task.

Based on the findings above, we can propose an initial hypothesis that there appears to be a pattern where languages with the closest grapheme-to-phoneme ratio or language with the most expressive phonetic orthographies tend to be the easiest to learn. Related to this, we should note that under the transliteration task cross-lingual comparison of languages appears to be constrained by data sparsity. The ease of learning a language from a computational point of view is tied to the number of samples we can observe for each transliteration pair. Languages with large source orthographies, especially logographic ones, are dramatically more sparse than other

ones. With this in mind, we should question the fairness of comparing languages under g2p transliteration tasks where data sparsity is dramatically different. Perhaps this can explain some of the results on why logographic languages appear to be harder to learn. We speculate that if each language was given a training set proportional to their source vocabulary, then we could have a much more robust way to compare the complexity of languages under the machine learning scheme. We urge future authors who are using joint orthography and phonology datasets such as those studying g2p to consider the sparsity of data when comparing languages. Additionally, future work should try to design tasks where the relative data sparsity is fairer.

6. CONCLUSION

In this work we have suggested that exploring the process of transliteration can yield interesting insights into the process of language learning especially for machine models. In order to further pursue this work further a few interesting notes should be made. First, as seen with logographic languages and languages with high phoneme-source letter ambiguity the number of training samples must be higher for effective learning therefore in future work more comprehensive datasets should be used such as Wiktionary and comparison based not on absolute number of samples but on number of samples per class should be made. Additionally, in order to prove out the theory that the simplest (closest to a 1:1 mapping between phoneme and grapheme) or most expressive phonetic orthographies are the easiest to pronounce we should sample many different orthographies and devise a metric for complexity that isn't simply a ratio of grapheme over IPA letter.Due to the simplicity of the ratio of absolute number in grapheme and phoneme inventories, future work should develop measures based on theorical results from linguistics on how each phoneme maps to each grapheme. Finally, once a fairer complexity measure and task is developed for cross-lingual comparison, we should look at performing an ablation study that illustrates how complexity of learning to pronounce a language change under training and modeling strategies that are known to improve the ability to transliterate such as encoding phonetic similarity in features, self-supervision, data augmentation, and joint or cross language modeling.

REFERENCES

[1] K. Gorman, L. F. E. Ashby, A. Goyzueta, A. McCarthy, S. Wu, and D. You, "The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Online, Jul. 2020, pp. 40–50. doi: 10.18653/v1/2020.sigmorphon-1.2.

[2] R. Y.-H. Lo and G. Nicolai, "Linguistic Knowledge in Multilingual Grapheme-to-Phoneme Conversion," in *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Online, Aug. 2021, pp. 131–140. doi: 10.18653/v1/2021.sigmorphon-1.15.

[3] S. Moran and D. Blasi, "Cross-linguistic comparison of complexity measures in phonological systems," in *Measuring Grammatical Complexity*, Oxford: Oxford University Press, 2014. doi: 10.1093/acprof:oso/9780199685301.003.0011.

[4] X. Marjou, "OTEANN: Estimating the Transparency of Orthographies with an Artificial Neural Network," in *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, Online, Jun. 2021, pp. 1–9. doi: 10.18653/v1/2021.sigtyp-1.1.

[5] T. Pimentel, J. Valvoda, R. Hall Maudslay, R. Zmigrod, A. Williams, and R. Cotterell, "Information-Theoretic Probing for Linguistic Structure," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Jul. 2020, pp. 4609–4622. doi: 10.18653/v1/2020.acl-main.420.

[6] *ipa-dict - Monolingual wordlists with pronunciation information in IPA*. open-dict-data, 2022. Accessed: Jan. 21, 2022. [Online]. Available: https://github.com/open-dict-data/ipa-dict

[7] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Jan. 21, 2022. [Online]. Available: https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.htm

[8] S. Wu, R. Cotterell, and M. Hulden, "Applying the Transformer to Character-level Transduction," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, Apr. 2021, pp. 1901–1907. doi: 10.18653/v1/2021.eacl-main.163.

[9] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Transformer based Grapheme-to-Phoneme Conversion," *Interspeech 2019*, pp. 2095–2099, Sep. 2019, doi: 10.21437/Interspeech.2019-1954.

[10] M. Hammond, "Data augmentation for low-resource grapheme-to-phoneme mapping," in Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Online, Aug. 2021, pp. 126–130. doi: 10.18653/v1/2021.sigmorphon-1.14.

AUTHORS

Domenic Rosati, MIS (Dalhousie 2017) is a research scientist at scite.ai. His interests include investigating foundational aspects of natural language understanding and reasoning.

USE OF MACHINE LEARNING FOR ACTIVE PUBLIC DEBT COLLECTION WITH RECOMMENDATION FOR THE METHOD OF COLLECTION VIA PROTEST

Álvaro Farias Pinheiro[1], Denis Silva da Silveira[2]
and Fernando Buarque de Lima Neto[1]

[1]Polytechnic School, University of Pernambuco, Recife, Brazil
[2]Department of Administrative Sciences,
Federal University of Pernambuco, Recife, Brazil

ABSTRACT

This work consists of applying supervised Machine Learning techniques to identify which types of active debts are appropriate for the collection method called protest, one of the means of collection used by the Attorney General of the State of Pernambuco. For research, the following techniques were applied, Neural Network (NN), Logistic Regression (LR), and Support Vector Machine (SVM). The NN model obtained more satisfactory results among the other classification techniques, achieving better values in the following metrics: Accuracy (AC), F-Measure (F1), Precision (PR), and Recall (RC) with indexes above 97% in the evaluation with these metrics. The results showed that the construction of an Artificial Intelligence/Machine Learning model to choose which debts can succeed in the collection process via protest could bring benefits to the government of Pernambuco increasing its efficiency and effectiveness.

KEYWORDS

Data Mining, Artificial Intelligence, Machine Learning & Public Debt Collection.

1. INTRODUCTION

According to Witten, it is estimated that data from organizations double every 20 months, and this large amount of data is increasingly difficult to use for decision making [1]. According to Wirtz, this problem is even more pronounced in the public sector [2]. In this context, the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques has been shown to be increasingly appropriate to solve this problem [1]. And, as pointed out by Gousios, there is more and more data to be used in software engineering associated with data science techniques [3] to find descriptive and predictive information, in several areas, from credit analysis [4] to the billing process [5].

According to Hunt, the debt collection process is increasingly using Artificial Intelligence for better results. And the application of these techniques in the public sector that deal with debt collection becomes even more relevant for two reasons: (1) the need to differentiate debtors from tax evaders; and (2) the legal obligation to collect, regardless of the amount. The first is related to the need for greater assertiveness in the process, and the second is related to efficiency. Both are expected contributions using AI [6].

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 99-108, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120909

Arising from the above argument, the following challenges are commonly encountered in government agencies to adequately carry out public debt collections: (1) dealing with the large volume of data [1], and (2) using reliable and agile mechanisms to carry out debt [6].

This article is organized into six sections. This section presented the objective for the development of this work. The second section dealt with motivation. The third section described the theoretical foundation. The fourth section presented the research method used. The fifth section presented the results. And the sixth section presented the conclusions, with applicability for future work. Finalizing the document with the references.

2. MOTIVATION

The Attorney General of the State of Pernambuco (AGS/PE) has been using a tool for Business Intelligence (BI), called Qlik Sense since 2019, and with it, descriptive analyses were performed to obtain knowledge of the data, which are available to the organization stored in the Oracle database of this institution, and data inserted and maintained by transactional application called the Justice Automation System (JAS) implemented since 2006.

With these data, added to several others stored in the SQL Server database of this body, from applications integrated in the platform called Portal-AGS/PE, it was possible to better understand the evolution of active debt over the years, and in this understanding, it was observed the need to better understand how data behaves.

AGS/PE being the state government agency, responsible for collecting active debts, with the knowledge obtained with the BI tool, observed the annual increase in the amount of debt and the number of debtors, which motivated the use of Business Analytics (BA) to identify the causes and consequences and in the search for its solutions, with a focus on improving the collection processes.

With the purpose of assisting in this process, in this article, we use AI and ML techniques to identify which debts are most appropriate for the protest collection modality. Thus, the AGS/PE Active Debt Center will have a less costly tool for public coffers than electronic court. This was possible by training smart techniques based on data from the Active Debt Registration (ADR).

3. THEORETICAL BASIS

This section provides an overview of the techniques used to perform Supervised Machine Learning. In addition, it also presents the metrics used for the analysis of classification techniques.

3.1. Supervised Learning Techniques

Neural Network (NN) allows you to perform complex computations through a training function on a dataset. Thus, NNs can be seen as approximations of nonlinear functions, e.g., classification or regression. There are several parameterization and models, usually have an input layer and one, output, and one or more intermediate layers [7].

Logistic Regression (LR) allows estimating the probability associated with the occurrence of a given event in front of a set of explanatory variables, being a statistical technique that aims to model, from a set of observations, the logistic relationship between responses and a series of numerical or categorical explanatory variables [8].

Support Vector Machine (SVM) allows you to generate a representation of examples as points in space, mapped so that the examples in each category are divided clearly and accurately. Thus, new input cases are then mapped appropriately as belonging to one of the categories of the output space. Therefore, what an SVM does is find a separation line, a hyperplane, between data from multiple classes. That is, the hyperplane seeks to maximize the distance between the closest points relating to each of the existing classes [9].

3.2. Classification Analysis Metrics

Accuracy (AC) is the ratio between true positives (TP) and true negatives (TN) for the sum of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [10].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision (PR) is the ratio between true positives (TP) for the sum of the number of true positives (TP) and false positives (FP) [10].

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall (RC) is the ratio between true positives (TP) for the sum of the number of true positives (TP) and false negatives (FN) [10].

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

F-Measure (F1) is twice the ratio between multiplying precision by the recall to precision and recall [10].

$$F1 = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Receiver Operating Characteristic (ROC) is a metric for comparing the performance of the models, represented by the area under the ROC curve. The ROC curve is plotted as a diagram of true positive values (TP) as a function of the false positive ratio (FP). The closer the value is to 1, the better the classifier performance [10].

Finishing, True Positives (TP) are tests that are passing because the application is behaving as expected. True Negatives (TN) are tests failing due to real failures. False Positives (FP) are tests that pass, but they shouldn't pass. And, False Negatives (FN) are tests failing, however, due to inconsistency in the test itself [10].

4. METHOD

It is through the scientific method that the researcher's way of proceeding to the conclusion of research to achieve a goal is defined [11]. Thus, the goal of this section is to describe in the broadest sense, the application of the above techniques based on the data obtained belonging to AGS/PE and referring to the registration of active debts.

The first step of the method was the identification of the problem. Thus, at this stage, we sought to identify problems in the tax foreclosure that prevented the financial recovery of the state.

Therefore, a classification of the debts entered was carried out, to be able to learn, based on the occurrences, which debts were appropriate to the protest collection modality. Being more specific, several pieces of training of techniques already named was carried out to identify in the Active Debt Registration (ADR) which debts are most appropriate.

The idea here was: the characteristics that pointed to a greater assertiveness in the automated sending of lots of ADR to protest, would be decreased in the number of ADR returned because they did not fit the rules of this modality.

The second step was data mining because to apply Artificial Intelligence / Machine Learning it is necessary to first perform data mining. The Cross-Industry Standard Process for Data Mining (CRISP/DM) technique was used, performing the activities, as described by Chapman, which are: (1) understanding of the business; (2) understanding of the data; (3) data preparation; (4) data modeling; (5) evaluation of the data; and (6) deployment. Repeating the process as many times as necessary, until the mined data are satisfactory [12].

The goal of this step was to determine which data set is the most appropriate to solve the problem, and how this data should be standardized, and balanced, to avoid bias in the learning process. And for this problem, which aims to predict which active debt (ADR) should be protested electronically or not, the following data were selected:

- CAD: number of the certificate of the active debt.
- COMPANYNAME: name (company name).
- OCCURRENCE, type of occurrence of the debt, being canceled, returned, paid, protested, withdrawn, and held.
- SITUATION: debt situation, being it: awaiting regularization, with active installment, settled, exhausted installment, under-defense and under-defense, and guaranteed judicial.
- PROCESS: the type of the process: active, canceled, and liquidated.
- BALANCE: the amount of the outstanding balance.
- NOTARYPUBLICOFFICE: the identification of the protest office.
- PROTOCOL: the protest protocol number.
- DTOCCURRENCE: the date of occurrence.
- DTSUBSCRIPTION: the date of registration of the active debt.
- IRREGULARITY: the description of the irregularity, in most occurrences, were: invalid transferor, incompatible ZIP Code, insufficient address, uninformed, other jurisdiction, and insufficient time.
- PROVIDENCE: the type of providence, which can be a cancellation with charge, cancellation without charge, and not informed.

After data selection, the next phase was data elaboration, and the activities of cleaning, creation, integration, and formatting of the data were carried out, to identify which are the independent variables, and the ones identified were: OCCURRING, SITUATION, PROCESS, BALANCE, NOTARYPUBLICOFFICE, DTOCCURRENCE, DTSUBSCRIPTION, and IRREGULARITY.

With the purpose of supporting the definition of the new variables created, with the goal of being used in the calculation of dependent variables. They are, OCCURRENCEDAY and SUBSCRIPTIONDAY, were created.

In this process, we chose to remove the PROTOCOL variable because it was not adding any value to the problem. And, the CAD variable because it is a unique identifier that does not have duplicity and does not contribute to the process of extracting characteristics, was anonymized, and left as a goal only for the identification.

This is to keep the debtor's identification in absolute secrecy, due to the confidential nature of the taxpayer's data and the legal requirements of the General Data Protection Act (GDPA). Therefore, as the data from this sample are real, we chose not to show the debtor, and the COMPANYNAME variable was also extracted, thus ensuring its anonymity.

The variables SITUATION, IRREGULARITY, and PROVIDENCE were divided into several binary variables with the application of normalization. Finally, the variable OCCURS was marked as the target variable with the values representing success or failure in the protest process.

After this elaboration, all independent variables were normalized to balance the values that were at different or very heterogeneous scales. For this process, the method of obtaining the Y of X was used, based on the rule Y = (X-MIN)/(MAX-MIN), to be used as data entry in the tested models, avoiding bias.

In the next step, the selection process was applied, using the vertical selection technique, selecting only the records whose PROCESS criterion was equal to 'active', ignoring the 'canceled' and 'settled' because there is no point in calculating the dependent variable of processes that are not in progress. Following the same logic, the tuples that had the variables with null values were also extracted, resulting in 6966 tuples available for learning.

With the cleaning and transformation of the data performed in all variables, to better adapt the classification techniques for supervised training, the Google Colaboratory tool [14] was used for the test stage of the models, aiming to verify which of the chosen techniques is the most appropriate.

The tests of the models were performed with varying hyperparameters, to identify which configuration was the most applicable to the data treated. Emphasizing that the classification techniques applied to the data were NN, LR, and SVM, and all training was performed using the Orange Canvas tool [13].

During the training of the selected models, the evaluation stage was performed. In this step, it was possible to observe how the results obtained with the techniques met the specified problem. In this study, the evaluation was performed through techniques that seek to influence better decision-making, through training for classification through metrics, such as Accuracy, Precision, Recall, F-measure, and area under the Receiver Characteristic Operating (ROC) curve. The results obtained in this evaluation phase are presented in the results section.

Finally, the last phase performed was the implementation, in this phase all the learning obtained, through data mining, training, testing, and validations with the choice of the best model based on the best values in the analyzed metrics were saved, generating a pickle file to be able to use it when necessary, allowing the serialization of objects to be used in python applications.

5. RESULTS

In this section, we present the results of the application of the techniques with the evaluation by the metrics Accuracy (AC), F1, Precision (PR), and Recall (RC). Thus, after the application of the techniques, it was observed that the Neural Network that obtained the highest RATE of AC was 98%, with F1 with 97%, PR with 97%, and RC with 98%.

Table 1 shows the settings of the hyperparameters that were applied to the neural network until they reach the best value. The column 'L' represents the number of layers used in the experiment,

the column 'N' the number of neurons per layer, the column ‘F’ is the function, the column 'S' the method applied, and the column 'R' the learning rate used that was 0.0001 for all, with the best results in bold.

Table 1. Values used in NN hyperparameters and the results evaluated by AC, F1, PR and RC metrics

L	N	F	S	R	AC	F1	PR	RC
1	32	ReLu	Gradient	0.0001	0.978	0.971	0.971	0.978
1	64	ReLu	Gradient	0.0001	0.978	0.971	0.970	0.978
1	128	ReLu	BFGS	0.0001	0.978	0.970	0.969	0.978
1	256	ReLu	Adam	0.0001	0.978	0.970	0.969	0.978
1	512	ReLu	Adam	0.0001	0.978	0.976	0.974	0.978
2	64	Adam	Gradient	0.0001	0.977	0.968	0.967	0.977
2	64	ReLu	Gradient	0.0001	0.979	0.972	0.972	0.979
2	64	Hyperbolic	Gradient	0.0001	0.978	0.971	0.971	0.978
2	64	ReLu	BFGS	0.0001	0.979	0.977	0.975	0.979
3	64	ReLu	Gradient	0.0001	0.978	0.971	0.970	0.978
3	128	ReLu	Gradient	0.0001	0.978	0.971	0.971	0.978
3	128	ReLu	Gradient	0.0001	0.979	0.972	0.972	0.978
4	256	ReLu	Gradient	0.0001	0.979	0.972	0.972	0.978
4	**256**	**ReLu**	**Adam**	**0.0001**	**0.980**	**0.978**	**0.977**	**0.980**

Table 2 shows the regularization hyperparameters represented by column 'R' and Strength by column 'S' used in the LR model with the application of the metrics Accuracy (AC), F1, Precision (PR), and Recall (RC), with the best results in bold.

Table 2. Values used in LR hyperparameters and the results evaluated by AC, F1, PR and RC metrics

R	S	AC	F1	PR	RC
L1	80	0.978	0.977	0.971	0.975
L1	70	0.978	0.977	0.971	0.975
L1	60	0.978	0.977	0.971	0.975
L1	50	0.978	0.977	0.971	0.975
L1	40	0.978	0.977	0.975	0.975
L2	50	0.977	0.967	0.965	0.975
L2	40	0.979	0.977	0.975	0.975
L2	30	0.978	0.977	0.975	0.975
L2	20	0.979	0.977	0.975	0.975
L2	10	0.978	0.978	0.976	0.978
L1	30	0.978	0.978	0.976	0.978
L1	20	0.979	0.978	0.976	0.978
L1	10	0.979	0.978	0.976	0.978
L1	**1**	**0.980**	**0.978**	**0.977**	**0.980**

Table 3 shows the Cost hyperparameters represented by column 'C' and Regression by column 'R' used in the SVM model with the application of Accuracy (AC), F1, Precision (PR), and Recall (RC) metrics, with the best bold results.

Table 3. Values used in VMS hyperparameters, and the results evaluated by AC, F1, PR and RC metrics

C	R	AC	F1	PR	RC
0.10	0.40	0.974	0.970	0.970	0.977
0.10	0.30	0.974	0.970	0.970	0.977
0.10	0.20	0.974	0.970	0.970	0.977
0.10	0.10	0.974	0.970	0.970	0.977

0.50	0.50	0.975	0.970	0.970	0.977
0.50	0.40	0.975	0.960	0.960	0.977
0.50	0.30	0.975	0.970	0.970	0.977
0.50	0.20	0.975	0.970	0.970	0.977
0.50	0.10	0.975	0.970	0.970	0.977
1.00	0.50	0.975	0.971	0.971	0.978
1.00	0.40	0.976	0.971	0.971	0.978
1.00	0.30	0.976	0.971	0.971	0.978
1.00	0.20	0.976	0.971	0.971	0.978
1.00	**0.10**	**0.978**	**0.971**	**0.971**	**0.978**

Table 4 presents the results of the metrics applied to compare performance between the 3 classification techniques used: the neural network, logistic regression, and the support vector machine. With the configuration of 4 layers, with 256 neurons in each layer, using the ReLu activation function with the Adam technique and learning rate of 0.0001, the neural network, proved to be the best model among the 3 tested, as can be observed in the following table, with the best results in bold.

Table 4. Comparison of the results obtained with the experiments between the techniques of NN, LR and SVM

Technique	**AC**	**F1**	**PR**	**RC**
LR	0.978	0.971	0.970	0.978
SVM	0.978	0.970	0.971	0.978
NN	**0.980**	**0.978**	**0.977**	**0.980**

The data set used in this experiment had 6966 records, with the variable OCCURRING being the target, with 5 categorical variables, 7 numerical, and 3 textual variables. Being used for this experiment 6 variables for extraction of characteristics, the independent variables: SITUATION, BALANCE, EVENTS, SUBSCRIPTIONDAY, IRREGULARITY, and PROVIDENCE.

Targeting the VARIABLE OCCURS and the target variables, only for identification, CAD_ANONYMOUS, and NOTARYPUBLICOFFICE. For the set, only the paid occurrence was discarded because there were not enough instances to balance the training, since the data set included only the data from the years 2018, 2019, and 2020. Being used in training and evaluating a subset corresponding to 80% of the data, resulting in 5571 records, with the remaining 1392 for validation.

To perform the tests, the Cross-Validation technique was used with the number of folds equal to 3 in a stratified way. Thus, observing the ROC curve of these 3 techniques compared, it is possible to notice that the neural network model had the most representative area, as can be seen in Figures 1 and 2. In the validation it was also possible to observe that the neural networks model was able to predict with a certain rate of success, the data that have not yet been presented to the model, obtaining 98% accuracy.

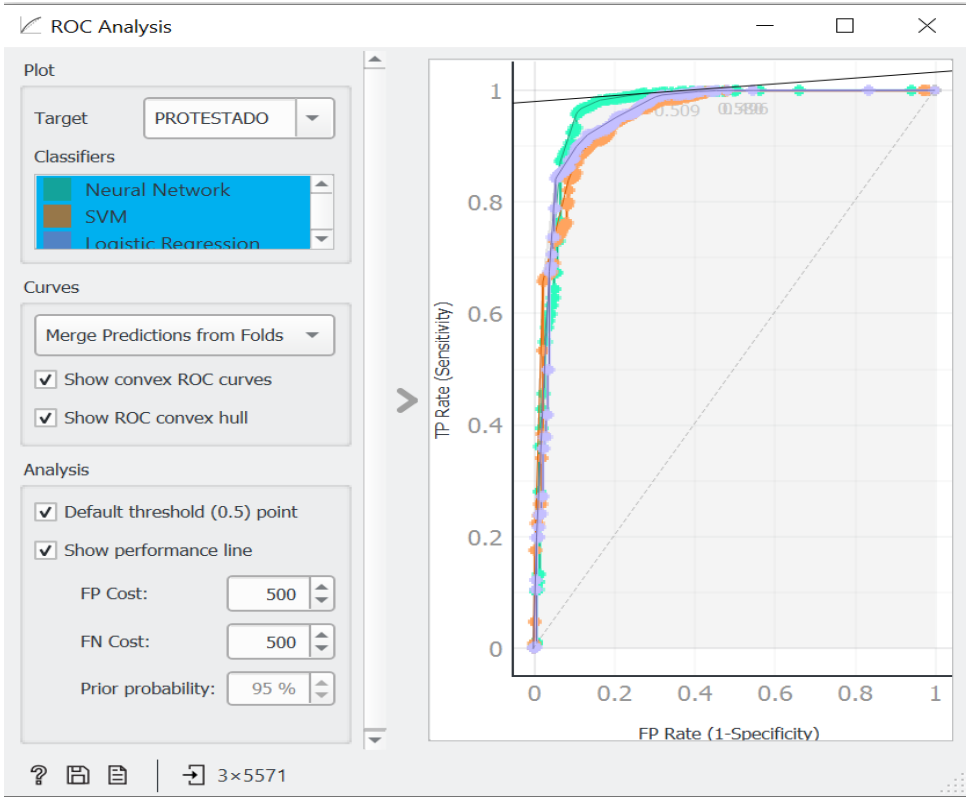


Figure 1. ROC curve resulting from the evaluation performed between NN, LR and SVM

	Neural Network	OCORRENCIA	TABELIONATO	CDA_ANONIMIZADO
313	0.00 : 0.00 : 0.95 : 0.05 : 0.00 → PROTESTADO	PROTESTADO	RECIFE	724
76	0.00 : 0.00 : 1.00 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	GLORIADEGOITA	723
729	0.00 : 0.00 : 0.88 : 0.12 : 0.00 → PROTESTADO	PROTESTADO	RECIFE	718
530	0.00 : 0.00 : 1.00 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	RECIFE	717
666	0.00 : 0.01 : 0.99 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	JABOATÃODOSGUARARPES	714
1185	0.00 : 0.00 : 1.00 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	GOIANA	710
981	0.00 : 0.00 : 1.00 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	JABOATÃODOSGUARARAPES	700
1157	0.00 : 0.00 : 1.00 : 0.00 : 0.00 → PROTESTADO	PROTESTADO	ESCADA	699
706	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	MORENO	6957
992	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	SÃOLOURENÇODAMATA	6953
1100	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	MORENO	6951
999	0.19 : 0.00 : 0.81 : 0.00 : 0.00 → PROTESTADO	DEVOLVIDO	SIRINHAEM	6944
973	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	OLINDA	6941
1023	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	CARUARU	6940
119	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	OLINDA	6934
168	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	OLINDA	6925
250	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	NAZAREDAMATA	6911
283	0.00 : 1.00 : 0.00 : 0.00 : 0.00 → DEVOLVIDO	DEVOLVIDO	PEDRA	6906

Model	CA	F1	Precision	Recall
Neural Network	0.984	0.983	0.983	0.984

Figure 2. Result of the application of the NN model with the AC, F1, PR and RC metrics

6. CONCLUSION

This article sought to analyze the best Model of AI/ML that can be used to recommend CADs that should be protested or directed to another type of collection, optimizing the debt collection process, due to the gain of assertiveness, using a database of debts from the state of Pernambuco. With the results, it was possible to verify that the classification methods used achieved good accuracy results, being above 97%. Thus, it is possible to infer that the proposed model can be considered dependable since all search metrics achieved a satisfactory result.

The method used to select characteristics allowed better attributes to be used in the learning process and this, consequently, can produce more assertive results. Thus, the AGS may reduce or even eliminate the number of CADs that are returned by the notary offices, allowing them, which do not suit this form of collection, can be directed in advance to the most appropriate method of collection to their characteristics.

6.1. Future Works

As future work, training will be carried out with the completeness of the data, which are since 2006, the year of implementation of the justice system, since for this experiment was used a sample corresponding to the years 2018, 2019, and 2020.

With the use of the complete database, it will be possible to apply the model to the CADs that are indicated as not suitable for protest can be directed to other debt collection modalities. Being possible the from the return of failure, to indicate what would be the other modality of judicial collection, considering the rule for sending to electronic filing, which can be represented by logic: $Y = ((X1 \cap (X3 > 4000)) \cup (X2 \cap (X3 > 2000))$, where (Y) represents the range of life, (X1) represents ICMS, (X2) Other taxes, (X3) outstanding balance, allowing for electronic judgement only if the outstanding balance exceeds $4,000.00 in the case of TAX ON MOVEMENT or $2,000.00 in the case of other taxes, being able to send to BANKNOTE PROTEST SERVICE otherwise.

ACKNOWLEDGEMENTS

This work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Financing Code 001, and with the collaboration of the Attorney General of the State of Pernambuco, in particular the Coordination of Active Debt of the Attorney of Finance.

REFERENCES

[1] Witten, I. H., Frank, E., Hall, M. A., & Pal C. J. (2011) Mining: Practical tools and techniques of registering. Morgan kaufmann.
[2] Wirtz, B. W., & Müller, W.M. (2019) An integrated artificial intelligence structure for public management. Public Management Review.
[3] Gousios, G., & Spinellis, D. (2017) Github Mining Software Engineering Data at the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). IEEE.
[4] Koutanaei, F. N., Sajedi, H., & Khanbabaei, M. (2015) A hybrid data mining model of resource selection algorithms and set learning classifiers for credit score, Journal of Retailing and Consumer Services.
[5] van de Geer, R., Wang, Q., & Bhulai, S. (2018) Data-driven consumer debt collection via machine learning and approximate dynamic programming.
[6] Hunt, R. M. (2007) Collecting consumer debt in America.

[7] Rumelhart, D. E. & Geoffrey E. H., & Ronald J. W. (1986) Learning representations by back-propagating errors.
[8] Tolles, J., & William, J. (2016) Logistic Regression Relating Patient Characteristics to Outcomes. JAMA.
[9] Cortes, C., & Vapnik, V. N. Support-vector networks. Machine Learning. 20 (3): 273–297. CiteSeerX, 1995.
[10] Powers, D. M. (2011) Evaluation: from precision, recall and f-measure to roc, informed, striking.
[11] Richardson, R. J. (2017) Social research - Methods and Techniques. 4. Ed. São Paulo: Atlas.
[12] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000) Crisp-dm 1.0 step by step data mining guide.
[13] Demšar, J., Jet, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., & Štajdohar, M. (2013) The Journal of Machine Learning Research.
[14] Schumann, M. A (2015) Book about Colab: and related activities. New York, N.Y.: Printed Matter.

AUTHOR

Álvaro Farias Pinheiro is an Analyst in Information and Communication Technology Management at the State Agency for Information and Communication Technology of the State, Coordinator of Systems, Digital Automation and Innovation of the Attorney General of the State of Pernambuco, and Institutional Relations Director of the Pernambuco State Civil Servants Association. He is a member of the National Artificial Intelligence Technical Group. He is currently a Ph.D. Student in Computer Engineering at the Polytechnic School of Engineering of Pernambuco, University of Pernambuco. He holds an MBA in Artificial Intelligence applied to Marketing at Unyleya Faculty. He holds a Master's in Software Engineering from the Recife Center for Advanced Studies and Systems. He holds a Specialization in Software Engineering Development Methodologies from the Brazilian Union of Technology. He holds a Bachelor's degree in Information Systems with Emphasis in Software Engineering from Recife Integrated Faculty.

Denis Silva da Silveira is an associate professor in the Management Department at Federal University of Pernambuco. He received his Ph.D. degree in Production Engineering (2009) at Federal University of Rio de Janeiro, with postdoctoral research at New University of Lisbon (2016). He has experience in Information Systems and Business Process Management. His research interests include Business Processes Management, Information Systems Architecture, Conceptual Modelling, Semantic Models and User Models.

Fernando Buarque de Lima Neto is Ph.D. in Artificial Intelligence from the University of London (2002), with a degree from the Imperial College London - DIC, in Artificial Neural Networks (2002), a master's degree in Computer Science from the Federal University of Pernambuco (1998) and graduation in Computer Science from the Catholic University of Pernambuco (1990). He is an Associate Professor at UPE at the Polytechnic School of Pernambuco, and a permanent member of the Doctoral Program in Computer Engineering. The lines of research are (1) Artificial/Computational Intelligence, (2) Modeling/Simulation of Real Complex Problems, and (3) Decision Systems explainable via Computational Semiotics.

APPROACHES IN FAKE NEWS DETECTION : AN EVALUATION OF NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING TECHNIQUES ON THE REDDIT SOCIAL NETWORK

Moosa Shariff, Brian Thoms, Jason T. Isaacs, Vida Vakilian

Department of Computer Science, California State University, Channel Islands

ABSTRACT

Classifier algorithms are a subfield of data mining and play an integral role in finding patterns and relationships within large datasets. In recent years, fake news detection has become a popular area of data mining for several important reasons, including its negative impact on decision-making and its virality within social networks. In the past, traditional fake news detection has relied primarily on information context, while modern approaches rely on auxiliary information to classify content. Modelling with machine learning and natural language processing can aid in distinguishing between fake and real news. In this research, we mine data from Reddit, the popular online discussion forum and social news aggregator, and measure machine learning classifiers in order to evaluate each algorithm's accuracy in detecting fake news using only a minimal subset of data.

KEYWORDS

Machine Learning, Natural Language Processing, Reddit Social Network

1. INTRODUCTION

Fake news has been considered a significant threat to democracy, journalism, and freedom of expression [1]. In [2], researchers detail the potential for fake news to reduce trust in governments and impact global politics, most notably during the "Brexit" referendum and the 2016 U.S. presidential election. Fake news has also sowed doubt and added additional hurdles when it comes to managing personal health and well-being, as recently experienced from disinformation campaigns surrounding the COVID-19 pandemic and statistics on vaccinations [3]. Fake news can lead to real-world consequences and pose significant challenges to information systems where the goal is delivering relevant, timely and accurate information.

The general population continues to spend more time online each day consuming information, with some estimates that people in the U.S. spend an average of close to 2.5 hours a day on social media [4]. More so, the Internet has emerged as a primary source for entertainment and information that is rapidly replacing traditional media outlets. Consequently, broadcasting features of the Internet, primarily through social media technologies, allow any user to post original content or share content and claim it as 'news'. In many cases, these opinion pieces can often be incomplete information or more deceitful in nature, and in other cases, they can spawn from artificial means such as computer bots. Exacerbating challenges to minimizing fake news is

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 109-124, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120910

the sheer volume and velocity of this information and how quickly it can disseminate within and across social networks. Consequently, as reported in [5], the more exposure a user has to information that is inaccurate or false, the greater the likelihood that they perceived that information as accurate. Additionally, individualistic methods for evaluating fake news range drastically, as reported in [6], and become even more difficult to combat when a person has a personal interest in the story, as reported in [5]. For these reasons, and more, computing solutions are required to minimize exposure to fake news early on and provide users with quick tools for evaluation.

In this research, we focus on Reddit, which has one of the highest percentages of users who receive news, according to Pew Research Center [7], and investigate machine learning models for predicting veracity in Reddit posts using only a minimal subset of data

2. BACKGROUND AND RELATED WORK

2.1. Social Media

According to a report by Elisa Shearer and Jeffrey Gottfried in, "News Use Across Social Media Platforms 2016", an estimated 62% of Americans get news on social media, with around 50% of this population having viewed this news on social media [8]. While interest in news and current events is generally encouraging, access to accurate and reliable information is critical online, where misinformation can be difficult to determine and quick to spread.

Reddit is a web-based platform with features for social news aggregation, content rating, and discussion forums. According to Statista, Reddit has 430 million monthly active users which is slightly higher than the 330 million users of Twitter and it has a higher engagement rate [9]. It had over 199 million posts and 1.7 billion comments in 2019 and is the 5th most visited site in the U.S. More specifically, as reported in [8], 70% of users on Reddit get news from Reddit subreddits (see Figure 1), which is the highest followed by Facebook and Twitter. In other words, individuals flock to Reddit for news, more so than they would to Facebook or Twitter.

Table 1. News by Platform

Platform	Users Receiving News
Reddit	70%
Facebook	66%
Twitter	59%
Tumblr	31%
Instagram	23%
YouTube	21%
LinkedIn	19%
Snapchat	17%
Vine	14%

2.2. Fact-Checking Challenges

Fact-checking is a common technique performed by journalists and involves the verification of claims and sources related to information. With access to a greater number of datapoints than ever before, due in large part to social networking technology and ubiquitous computing, manual fact-

checking is laborious and time-consuming with numerous challenges. Detailed in [10], challenges involve (i) retrieval of all potentially relevant documents, (ii) verification of source reliability, (iii) prediction of source bias and (iv) determination of a document's veracity.

Numerous projects exist today with the goal of impeding the spread of false information online including popular websites. Within the U.S. alone, popular systems such as FactChecker.org, PolitiFact, Snopes and RealClearPolitics. These systems build atop ongoing research in the field, such as work in [11], which uses natural language processing (NLP) to quickly extract and order sentences in ways to aid in the classification of factual claims, and [12] which uses probabilistic classifiers that can both validate credibility but also aid in identifying what aspects of a document a user should focus on. Much other research exists in this emerging area of information systems, but these two studies highlight two critical components of fake news detection, including the challenges involved in data preparation and subsequent steps in algorithm construction, data modelling and testing.

2.3. Natural Language Programming (NLP) Solutions

On Reddit alone, hundreds of thousands of posts are created each day with many posts receiving thousands of views per hour. According to subredditstats.com [13], the top 10 subreddits, i.e. topics, have over 40,000 posts per day and over 360,000 comments each day. Table 2 highlights some of the more popular subreddits and the exposure these topics can generate. Consequently, any viable solution to monitoring veracity in this space would require a system capable of detecting fake news in real-time.

Table 2. Top Subreddit by Post / Comments

Subreddit	**Subscribers**	**Comments**	**Votes**
Politics	7.6m	5.2m	93m
WallStreetBets	10.5m	4.4m	54m
Teenagers	2.5m	880k	52m
NoStupidQuestions	2.3m	720k	9.2m
m=millions, k=thousands			

In this research, we analyze the language used within the titles of Reddit posts. Titles are particularly interesting as they afford Redditors a quick glimpse into a Reddit post and are aimed at attracting viewers with minimal data. Additionally, Reddit post titles are limited to 300 characters. More so, titles typically use language strategically designed to evade detection. Despite this, language leakage occurs, which is hard to monitor. This leakage includes frequencies and patterns of pronoun, conjunction, and negative emotion word usage [14]. The goal in the linguistic approach is to look for such instances of leakage or, so-called "predictive deception cues" found in the content of a message [15]. This can be achieved by creating a machine learning model using NLP algorithms. In this research we use titles to ascertain Ngrams.

N-Grams are sets of keywords that are strung together in groups of *n* words, where *n* is a positive nonzero integer. N-Grams are either continuous sets of characters or words. The most basic version of an N-Gram is the unigram, which is an n-gram of size 1. The next two n-grams are the bigram and the trigram. Frürnkanz [16] noted that word sequences of only about 2 to 3 words were easiest to apply without causing performance stress compared to conducting n-gram analyses on larger word sets. N-Grams can be created from characters, words or even binary text.

In this research, we use a combination of unigrams, bigrams and trigrams as input to our classifiers.

3. RESEARCH METHODOLOGY

3.1. Cross Industry Standard Process for Data Mining (CRISP-DM)

This research adheres to the Cross-Industry Standard Process for Data Mining (CRISP-DM), which is a framework for data mining [17]. Research in fact-checking systems adheres to CRISP-DM in the following steps:

1. Understand the problem domain,
2. Understanding the underlying data,
3. Preprocess and preparing this data,
4. Model the data,
5. Evaluate each model, and
6. Deploy the system.

Understanding the problem domain was covered in 2.1 and 2.2. In this section and subsequent sections, we discuss how we prepare the data, model the data and evaluate each model.

3.2. Data Collection and Pre-processing

Data was collected from two subreddits, theonion and nottheonion. Theonion is a subreddit of satirical Reddit posts and contains links to articles that have fake news. The nottheonion subreddit contains real news. Data was collected from Reddit using Reddit's Python Reddit API Wrapper (PRAW) and Pushshift.io, an API that provides enhanced functionality and search capabilities over PRAW. Initial data collection is agnostic since early in the data mining process we are not concerned with the context of the data, however, limitations in using PRAW are numerous including limiting the result-set from an API call to 1000, preventing the collection of results between specified dates and limiting API calls to only 1 per second. For this reason, we use Pushshift which provides better search functionality and doesn't have any API call limits. Using Pushshift and PRAW 24,001 initial documents were retrieved from nottheonion subreddit and 16,931 initial documents were retrieved from theonion subreddit on January 7, 2020. For all posts the following data was collected:

- Post Title,
- Post Domain from which the article was obtained,
- Number of Comments Per Post,
- Post Timestamp,
- Post Author,
- Post Score, which is calculated using the number of upvotes or downvotes a post received,
- Author's Karma using reddit.info(), which is calculated using a user's contribution to the Reddit community.

After the data collection phases, the pre-processing phase involved cleaning this data for further analysis. Using the Python programming language, more specifically, the numpy and pandas libraries, duplicate entries and stopwords are removed from the dataset. Stopwords were generated using the Python scikit-learn library and a custom algorithm for determining unigrams and bi-grams from the dataset was used. Lemmatization was used instead of stemming to reduce

words to their root form and to help normalize the dataset. Figures 1 and Figure 2 illustrate the Top 5 bigrams for the onion and not the onion subreddits.

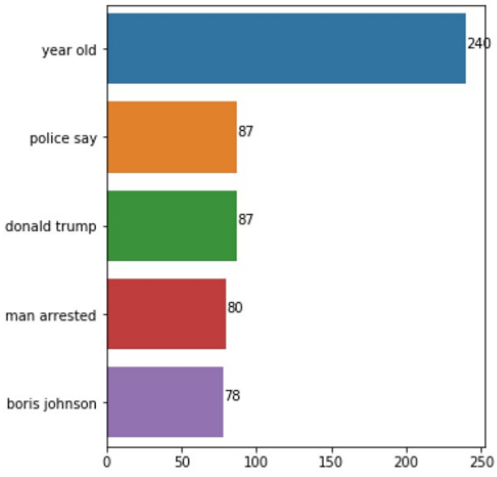


Figure 1. Top bigrams theonion

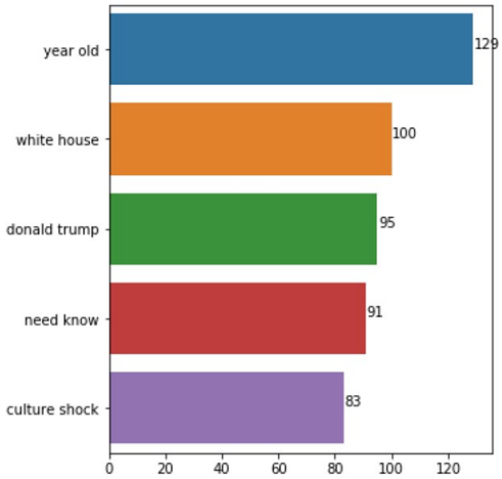


Figure 2. Top bigrams nottheonion

After the data cleansing process, our dataset consisted of 11,201 posts from nottheonion subreddit and 10,605 posts from theonion subreddit. We deemed this to be a good subset since data from each subreddit contained close to the same number of posts. An example title from theonion might look like this: *Neighborhood Rallies To Designate Pothole As Historic Landmark*. An example title from nottheonion might look like this: *Researchers perform magic tricks for birds, who are not amused*.

3.3. Feature Extraction

Feature selection is a critical step in machine learning and allows us to choose those features that contribute the most to the desired output. In other words, which features best afford to classify a post as fake or not. Feature selection not only helps reduce overfitting and improve accuracy but also decreases the training time as there would be lesser data to train. During this stage, we select which parts of the data we want to use from the data we collected. This is done using feature selection algorithms that identify the weight of a feature (i.e. attribute) from our data to affect the classification of our news. Feature selection algorithms reduce the dimensionality of the dataset and select only a minimal subset of features for input into our classifier algorithms. Data is encoded in a numerical format using a Count Vectorizer and TFIDF Vectorizer. A baseline accuracy is calculated prior to parameter hypertuning. KBest and Recursive Feature Elimination from the Python's sklearn feature selection package identify those features which contribute most to our output.

The KBest algorithm provides a score of each feature, where higher scores represent a higher contribution of that feature to the output. Recursive Feature Elimination gives us a ranking of the features by importance and it recursively discards the least important features. Recursive Feature Ranking was also calculated with similar output, which shows lower dependencies on a Reddit post's title and domain name. KBest and Recursive Ranking of feature selection scores can be found in Table 3.

Table 3. Feature Rank

Reddit Feature	KBest Score	Recursive Rank
Author Karma	9458741.764	6
Reddit Score	1874318.802	4
Number of Comments	250962.133	1
Post Title	134261.909	5
Post Author	91729.5	2
Domain	15174.17	3

Table 3 identifies that accuracy scores are significantly determined from a post's karma score and author, followed by the number of comments on the post and then the title of a post. While this feature ranking gives indications to accuracy, they rely on social data affixed to a post after it is already published. Therefore, in this research, we rely primarily on post title and domain for training and testing purposes. As for the authors and their scores, we disregard them because both can be easily spoofed. As detailed in [18], it is estimated that between 9% and 15% of active Twitter accounts were in fact bots and 60 million accounts on Facebook were bots. For Reddit, while numbers are not official, the estimate is around 10% of posts being made by bots.

Finally, features were converted from categorical form to binary form prior to training and testing. The dataset is subsequently split into training and testing datasets (e.g. 80% of the data for training and 20% for testing). After testing, a confusion matrix will be constructed for our best performing model. A confusion matrix helps to describe the performance of a classifier on a set of test data for which the true values are known.

4. CLASSIFIER MODELS AND RESULTS

Classification is the analog of regression when the variable being predicted is discrete, rather than continuous [19]. Classification algorithms are used to predict labels or classes of input data and map data to categorical values. More specifically, a classifier is a function that takes a range of known data as input (i.e. independent variables or predictors) and attempts to group that data into preset categories or classes, which can then be used to classify future unseen data. For each of our models, we focus only on optimized hyperparameters, or those determined through our feature extraction (Section 3.3).

For each model, we calculate three scores. The training score is how well our model performs against the training dataset. The end goal of training is to form a generalized model and prevent overfitting, which occurs when a model fits so well to the data with lots of variance. Validation scores provide insight into average performance over multiple testing iterations. In this research, we focus on n-fold cross-validation, which repeatedly trains on 80% to 90% of our dataset. We set the value of n to 5, based on research in [20], which saw no significant change in the output when increasing the number of cross-validations and a lower number of cross-validations reduces the execution time.

Test Scores are generated once a model has been optimized against relevant hyperparameters suitable for the algorithm. Test scores aim to test unseen data against the validated model and represent how the model would perform in a real-world scenario. Higher scores during testing

indicate a more generalized model. Based on the results during the validation and testing phases, we choose the best performing model in which to refine our final model.

4.1. Model 1: Baseline

Before implementing our classifier models, we use Logistic Regression to obtain a measure for baseline accuracy, our Model 1. Generally, researchers create a baseline model to validate against other models. In statistics, Logistic Regression can be used to model the probability of a certain outcome and is among the top 5 most widely used baseline models [21], largely chosen for its simplicity. For our baseline model, we use Logistic Regression and the title feature from our dataset. The hyperparameters used to formulate the baseline parameters were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 1
- Cost: 1

Scikit-learn's Count Vectorizer can be used to convert a collection of text documents to a vector of term/token counts. For our models, including our baseline model, we use Count Vectorizer to convert Reddit data to a vector of term/token counts. We set the Cost parameter to 1. The cost function for Logistic Regression quantifies the error between predicted values and expected values, which can make the model more complex on one hand but help reduce overfitting on the other hand.

The end accuracy for our baseline model after training, testing and validation was 84.57% and serves as the model to improve upon going forwards. It should be noted that this model showed overfitting with training scores at 99%.

4.2. Model 2 Count Vectorizer and Logistic Regression (1)

Model 2, similar to our baseline model, integrates Count Vectorizer with Logistic Regression but includes more features to train and test (e.g. Title and Domain). Subsequently, two distinct pipelines for our data are generated using a Count Vectorizer on both the Title and the Domain name, to form a single pipeline, which serves as input to our Logistic Regression classifier. The primary parameters and hyperparameters for Model 2 were as follows:

- Features: Title and Domain
- N-gram Range: 1 to 3
- Stopwords: 1
- Cost: 1

Table 4 highlights the results from training and testing and resulted in validation scores over 98%. The high accuracy can be attributed to the fact that domain sources were relatively homogenous, with the majority coming from theonion or a few other sources, all future models omitted domain as a parameter and focus only on the title.

Table 4. Model 2 Output

Measure	Result
Validation Score (%)	98.36
Training Score (%)	99.57
Testing Score (%)	98.27

Elaborating on Model 2, it was decided that we eliminate Domain as a feature for subsequent models. Further analysis of the dataset identified the source for fake news coming largely from two different domains, theonion and clickhole.com. A breakdown of domains and their post breakdown can be found in Table 5 and Table 6.

Table 5. Domains By Reference (Fake)

Domain	Post Count (%)
Theonion.com	7388 (44%)
Clickhole.com	4535 (27%)
Local.theonion.com	1201 (7%)
Politics.theonion.com	1054 (6%)
Youtube.com	457 (3%)
Entertainment.theonion.com	410 (2%)
Sports.theonion.com	382 (2%)
Youtu.be	175 (1%)
Lifestyle.clickhole.com	161 (1%)
i.redd.it	142 (1%)

Table 6. Domains By Reference (Real)

Domain	Post Count (%)
Theguardian.com	727 (3%)
Cnn.com	689 (3%)
Foxnews.com	582 (2%)
Google.com	565 (2%)
Bbc.com	561 (2%)
Independent.co.uk	474 (2%)
Nbcnews.com	471 (2%)
Nypost.com	439 (2%)
Newsweek.com	434 (2%)
Bbc.co.uk	381 (1%)

4.3. Model 3 Count Vectorizer and Logistic Regression (2)

For Model 3, we eliminate the Domain feature from our model and rerun our testing using Count Vectorizer and Logistic Regression. This creates a single pipeline for data input. Results are detailed in Table 4 and show validation scores around 84.94%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 1
- LRC: 1

Table 7. Model 3 Output

Measure	Result
Validation Score (%)	84.94
Training Score (%)	99.84
Testing Score (%)	84.62

4.4. Model 4 TF-IDF Vectorizer and Logistic Regression

Model 4 integrates TF-IDF with Logistic Regression. Term frequency/inverse document frequency (TF-IDF) is one of the most commonly used term weighting schemes in today's information retrieval systems [22]. This is different from Count Vectorizer as it takes into account the occurrence of the word not just in a single document but in the entire set of documents. Common words like 'a', 'the', etc. that appear frequently across all documents, have reduced weights and more weightage is given to words with lower frequency counts. To convert a collection of raw documents to a matrix of TF-IDF features we find the product of term frequency and inverse document frequency.

Results for Model 4 are detailed in Table 8 and show validation scores of 84.04%. The primary parameters and hyperparameters used in this model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Cost: 1

Table 8. Model 4 Output

Measure	Result
Validation Score (%)	84.04
Training Score (%)	91.84
Testing Score (%)	84.00

4.5. Model 5 Count Vectorizer with Support Vector Machine (SVM)

For Model 5, we implement a Support Vector Machine (SVM). SVM is a supervised machine learning algorithm that is capable of performing linear and nonlinear classification. SVM is an optimal classifier in the sense that, given training data, it learns a classification hyperplane in the feature space which has the maximal distance (or margin) to all the training examples, with the exception of a small number of outlier examples [23]. Linear kernels such as SVM are preferred for text classification due to a number of reasons including that most text classification problems are linearly separable [24] and text tends to possess many features, which is good for a linear kernel where non-linear mapping does not improve the performance [25]. Results for Model 5 are detailed in Table 9 and show validation scores at 84.85%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Cost: 0.1

Table 9. Model 5 Output

Measure	Result
Validation Score (%)	84.85
Training Score (%)	99.88
Testing Score (%)	85.50

To elaborate more on Model 5, SVM costs are calculated differently from Logistic Regression, thus the difference in these hyperparameter values. However, Cost is used in a similar fashion to Logistic Regression and attempts to guide how much we want to avoid misclassifying the data. Higher cost values help avoid misclassifying data but increase execution time.

4.6. Model 6 Count Vectorizer with Random Forest Classifier

Random forest classifier is a supervised machine learning algorithm based on ensemble learning. In contrast to ordinary learning approaches which try to construct one learner from training data, ensemble methods try to construct a set of learners and combine them [26]. According to [27] there have been significant improvements in classification accuracy by growing an ensemble of trees and letting them vote for the most popular class. Although it generally gets a better accuracy, this is not true for all predictions and it is slow to generate predictions when multiple decision trees. Results for Model 6 are detailed in Table 10 and show validation scores at 81.27%. It should be noted that Model 6 was the lowest-performing model. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Max Depth: None
- Min Samples Leaf: 1
- Min Samples Split: 5
- Estimators: 200

Table 10. Model 6 Output

Measure	Result
Validation Score (%)	81.27
Training Score (%)	99.92
Testing Score (%)	81.85

4.7. Model 7 TF-IDF Vectorizer with Multinomial Naïve Bayes

Model 7 implements a Naïve Bayes Classifier. Naïve Bayes Classifiers are based on applying Bayes Theorem with the “naïve” assumption of conditional independence between features. Recent work in supervised learning has shown that a surprisingly simple Bayesian classifier with strong assumptions of independence among features, called Naïve Bayes, is competitive with state-of-the-art classifiers such as C4.5 [28]. Naïve Bayes classifiers use Bayes Theorem, which calculates the probability of an event based on the prior knowledge of conditions that might be related to the event. While Naïve bayes classifiers are simple, they can provide are fast and accurate.

The Multinomial Naïve Bayes classifier is a specialized version of Naïve Bayes that is suitable for classification with discrete features (e.g., word counts for text classification). Multinomial Naïve Bayes estimates the conditional probability of a particular term given a class as the relative frequency of the term t in all documents belonging to the class C [29]. It assumes that every word is independent of the other. Instead of calculating the probability of sentences, we now calculate the probability of every single word. These probabilities are multiplied and the highest probability gives us the class it belongs to. Results for Model 6 are detailed in Table 11 and show validation scores at 83.94%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Alpha: [0.1, 0.3, 0.6, 1]

Table 11. Model 7 Output

Measure	Result
Validation Score (%)	83.94
Training Score (%)	90.53
Testing Score (%)	83.51

4.8. Model 8 Count Vectorizer with Multinomial Naïve Bayes

In Model 8, we implement features of Model 3 and Model 7 using Multinomial Naïve Bayes Classifier with Count Vectorizer. Results for Model 8 are detailed in Table 12 and show validation scores at 85.17%. It should be noted that Model 8 was the highest performing model. The primary parameters and hyperparameters defined for our baseline model are as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Alpha: [0.1, 0.3, 0.6, 1]

Table 12. Model 8 Output

Measure	Result
Validation Score (%)	85.17
Training Score (%)	99.53
Testing Score (%)	85.50

Since this model generated the highest performance, we select this model as ‘Best’ and conduct a confusion matrix to further determine the model’s accuracy. Illustrated in Figure 3, the confusion matrix labeled 2301 true negatives, 350 false positives, 440 false negatives and 2361 true positives. 790 predictions were misclassified. Using these values, we calculate accuracy, precision and recall for this model. This results in an accuracy of 85.51%, precision of 87.09%, recall of 84.29% and F1-score of 85.67%. The accuracy for this model is better than our baseline accuracy.

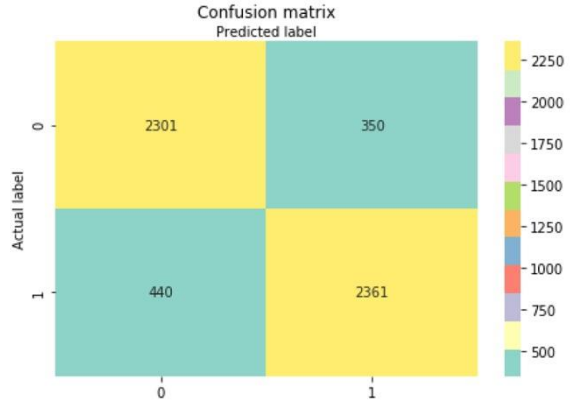


Figure 3. Model 8 Confusion Matrix

5. DISCUSSION

In this section, we reflect on the implications of our results and identify a few key findings surrounding natural language approaches in fake news detection.

5.1. Feature Removal - Domain

Performing KBest and Recursive Feature Ranking allowed us to produce a subset of features to serve as input for our model. It is important to note that most features were omitted since much of the data comprised in those features, such as comments and postscores are values that are generated after a post is submitted. Instead, we rely solely on values that are generated at the time of the post. Furthermore, results from Model 2 indicated a high degree of fitting using Domain plus Title. However, further analysis identified Domain to be a poor feature since it was relatively homogenous and varied little across theonion subreddit. This is most likely due to the fact that most of our data belongs to two to three domains, most of the domains referenced in theonion are from theonion.com (7388), clickhole.com (4535) and the rest of them are referenced a very few number of times. This allowed us to eliminate Domain as a parameter and focus on a complete minimal subset of our data and only use Title. Consequently, computing Logistic Regression using title and Count Vectorizer resulted in an accuracy of 84.57%.

5.2. NLP Considerations

Our research supports the use of n-grams in fact-checking systems, which has shown particular success in previous studies, such as [30], which used trigrams and gradient boosting to achieve 95% accuracy on an open-source Kaggle dataset. Using n-grams helped the classifiers capture complex expressions which in turn helped increase their accuracy. None of the N-grams and stop words from the NLTK package were selected. Additionally, we discovered that models not relying on lemmatization achieved slightly higher performance early on. Therefore, it was decided to proceed using non-lemmatized words.

5.3. Over Fitting

Overfitting takes place when a model achieves high testing scores with low validation scores. Models tend to overfit when a dataset is filled with noise and/or inaccurate data. One solution to avoid overfitting is to use linear algorithms on linear data or setting hyperparameters such as maximum depth if we are using decision trees.

Disregarding our baseline model, overfitting tended to take place in Model 3, Model 5, Model 6 and Model 7. Overfitting is more common in linear models such as Logistic Regression and Support Vector Machines. It was surprising to find Model 7 overfitting since Random Forest classifiers are generally better at preventing overfitting, but this was not the case. More so, our best-performing algorithm also showed overfitting in its training and results. It would be worthwhile to explore different approaches for Model 4 and Model 8, which were the only two models not to experience overfitting.

5.4. Hyper Parameters

Hyperparameter optimization or tuning is a critical aspect of machine learning. A hyperparameter is a parameter whose value is used to control the learning process. Each model required different hyper-tuning parameters depending on the algorithm and its requirements. For example, the alpha hyperparameter for our Multinomial Naïve Bayes classifiers differed from Model 5 (0.6) and Model 6 (1.0). The alpha parameter, also known as the additive smoothing parameter controls the shape of our model. A lower score makes the model complex, whereas a higher score makes the model simple and biased. Additive smoothing adds to the probability and is used as a fail-safe for unknown words in the vocabulary.

Hyper-tuning variables for each of the classifiers achieved only slight improvements of the test scores (e.g. 0.05% to 0.1%). All models used a cross-validation of 5 which repeatedly trains the model on the dataset 5 times.

5.5. Count Vectorizer vs. TF-IDF

It was interesting to note that Count Vectorizer performed slightly better than TF-IDF. This is typical since TF-IDF generally performs better on larger datasets. TF-IDF reflects the relative importance of a word for statistical analysis and is generally better than Count Vectorizers because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words. Unfortunately, TF-IDF was only performed with Logistic Regression and multinomial naïve Bayes classifier. In future research, it also makes sense to extend the implementation to support vector and random forest classifiers. The implementation of TF-IDF across other machine learning approaches may result in better results.

6. LIMITATIONS AND FUTURE RESEARCH

The authors acknowledge that a number of limitations in this research exist. First, data collected was from a single day in January of 2020. Future research should look to expand the data collection phase. Next, improvement to each of the models could be made by using bootstrap aggregation, also called bagging and/or boosting which has shown to reduce bias and improve the stability and accuracy in statistical classification and help prevent overfitting. Also, models applying TF-IDF algorithms appeared to resist overfitting, so it would be interesting to apply TF-IDF with Support Vector Machine and Random Forest. Finally, each model could be expanded to collect text from within an article. This might better train each model. In this research, we focus solely on Title since it is can play a critical factor in determining whether or not a user chooses to view a post.

7. CONCLUSION AND FUTURE WORK

In this research, we implement and evaluate multiple classifier models, which can be used to aid in fake news detection, all of which performed well. The best model was built using Count Vectorizer and Multinomial Naïve Bayes and was able to get an accuracy score of 85.51%. While accuracy levels are low, it is close to the accuracies obtained by [31] and [32] on social media platforms. Interestingly in this research, we were able to get an accuracy above the baseline accuracy by the implementation of different machine learning classifiers and through hyper-tuning. Additionally, unlike the other statistical models on fake news, which use a variety of metadata from social media platforms, our findings rely only on Reddit post titles. This work demonstrates that titles can play a significant role in classifying information as real or fake and marks a good starting point for detecting fake news.

REFERENCES

[1] Frearson, J. (2018). "The rise of fake news is a threat to our democracy - and our message," *Business Reporter*, Jan. 2018. Retrieved online via: https://www:businessreporter:co:uk/2018/01/13/rise-of-fake-news-is-a-threat-to-our-democracy.

[2] Zhou, X. and Zafarani, R. (2018). "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys* (CSUR), 2018.

[3] van der Linden, S. Roozenbeek, J. and Compton, J. (2020). "Inoculating Against Fake News About COVID-19," *Frontiers in Psychology*, v11), 2020.

[4] Tankovska, H. (2021). "Daily time spent on social networking by internet users worldwide from 2012 to 2020," Statista. Originally published on Feb 8, 2021. Retrieved online on https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/.

[5] Balmas, M. (2014). "When Fake News Becomes Real: Combined Exposure to Multiple News Sources and Political Attitudes of Inefficacy, Alienation, and Cynicism," *Communication Research*, v41(3), pp. 430–454.

[6] Zafarani, R., Zhou, X., Shu, K. and Liu, H. (2019). "Fake News Research: Theories, Detection Strategies, and Open Problems," In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (KDD '19), Association for Computing Machinery, New York, NY, USA, pp. 3207–3208.

[7] Shearer, E. and Mitchell A. (2021). "News Use Across Social Media Platforms in 2020," Pew Research Center, Published January 12, 2021.

[8] Shearer E. and Gottfried, J. (2016). "News use across social media platforms. News use across social media platforms," *Pew Research Center*, May 2016. Retrieved from https://www:journalism:org/2016/05/26/news-use-across-social-media-platforms-2016/.

[9] Tankovska, H. (2021). "Global social networks ranked by number of users 2021," Statista. Originally published on Feb 9, 2021. Retrieved online on https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[10] Nadeem, M., Fang, W. Xu, B. Mohtarami, M. and Glass, J (2019). "Fakta: An automatic end-toend fact checking system," In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[11] Hassan, N., Arslan, F., Li, C., and Tremayne, M. (2017). "Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster," In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '17). Association for Computing Machinery, New York, NY, USA, pp. 1803–1812.

[12] Nguyen, TT, Weidlich, M., Yin, H., Zheng, B., Nguyen, QVH, and Stantic, B. (2019). "User guidance for efficient fact checking," In *Proceedings of the VLDB Endowment*. v12(8). April 2019, pp. 850–863.

[13] Reddit.com, "Subreddit Status," Retrieved online from https://subredditstats.com/ on June 9, 2021.

[14] Feng, VW and Hirst, G. (2013). Detecting deceptive opinions with profile compatibility. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 338346, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[15] Conroy, NK, Rubin, VL, and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news," In *Proceedings of the Association for Information Science and Technology*, 52(1), January 2015.

[16] Frürnkanz, J. (1998). "A Study Using N-Gram Features for Text Categorization," *Austrian Research Institute for Artificial Intelligence*, pp. 98-30, 1998.

[17] Shearer C. (2000). "The CRISP-DM model: the new blueprint for data mining," *Journal of Data Warehousing*, v5, pp. 13-22.

[18] Lazer, DMJ, Baum, MA, Benkler, Y., Berinsky. AJ, Greenhill, KM, Menczer, F, Metzger, MJ,
Nyhan, B, Pennycook, G, Rothschild, D., Schudson, M., Sloman, SA, Sunstein, CR, Thorson, EA, Watts, DJ, Zittrain, JL (2018). "The Science of Fake News," *Science* , March 09, 2018, v359(6380), pp. 1094-1096.

[19] Pereira, F, Mitchell, T., and Botvinick, M. (2009). Machine learning classiers and fMRI: A tutorial overview. NeuroImage, 45(1): S199{S209, March 2009.

[20] Forman, G. and Cohen, I. (2004). "Learning from little: Comparison of classifiers given little training," In *Lecture Notes in Computer Science*, pp. 161-172. Springer Berlin Heidelberg,.

[21] Lin, W., Hu, Y. and Tsai, C. (2012). "Machine learning in financial crisis prediction: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v42(4), pp. 421-436.

[22] Aizawa, A. (2003). "An information-theoretic perspective of tf-idf measures," *Information Processing and Management*, v39(1), pp. 45-65, January 2003.

[23] Li, Y., Bontcheva, K. and Cunnigham, H. (2009). "Adapting SVM for data sparseness and imbalance: a case study in information extraction," *Natural Language Engineering*, v15(2), pp. 241271, April 2009.

[24] Joachims, T. (1998). "Text categorization with support vector machines: Learning with many relevant features," In *Machine Learning*: ECML-98, pp. 137-142, Berlin, Heidelberg, 1998.

[25] Hsu, CW, Chang, CC and Lin, CJ. (2008) "A practical guide to support vector classification," *Technical Report Department of Computer Science and Information Engineering*, National Taiwan University.

[26] Zhou, ZH. (2012). "Ensemble Methods: Foundations and Algorithms," *Chapman and Hall/CRC*, 1st Edition.

[27] Breiman, L. (2001). "Random forests," *Machine Learning*, v45(1), pp. 5-32.

[28] Friedman, N., Geiger, D. and Goldszmidt, M. (1997). "Bayesian network classifiers," *Machine Learning*, v29(2/3), pp. 131-163.

[29] Kamel S. (2019). "Arabic Language Processing: From Theory to Practice," *Springer International Publishing*.

[30] Wynne, HE and Wint, ZZ. (2019). "Content Based Fake News Detection Using N-Gram Models," In *Proceedings of the 21st International Conference on Information Integration and Webbased Applications & Services* (iiWAS2019). Association for Computing Machinery, New York, NY, USA, pp. 669–673.

[31] Ajao, O, Bhowmik, D. and Zargari, S. (2018). "Fake news identification on twitter with hybrid cnn and rnn models," In *Proceedings of the 9th International Conference on Social Media and Society*, SMSociety '18. Association for Computing Machinery, pp. 226-230, New York, NY, USA.

[32] Ruchansky, N., Seo, S. and Liu, Y. (2017). "Csi: A hybrid deep model for fake news detection," In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, Association for Computing Machinery, pp. 797-806, New York, NY, USA, 2017.

AUTHORS

Moosa Ahmed Shariff is a recent graduate of the Master of Computer Science program at California State University Channel Islands. His research interests explore the implementation of machine learning algorithms on large data sets using python and visualization software. After graduating CSU Channel Islands, his career goals are to acquire a software engineering position with an emphasis on data science.

Dr. Brian Thoms is Associate Professor of Computer Science at California State University, Channel Islands where he teaches courses on Human Computer Interaction and Database systems. Dr. Thoms received his PhD in Information Systems and Technology from Claremont Graduate University. His research interests explore data mining in the domains related to social and healthcare systems. He is also co-founder and CIO for Health e-Services, a health storage and analytics company.

Dr. Jason T. Isaacs is Associate Professor of Computer Science at California State University, Channel Islands where he teaches courses on embedded systems and software engineering. Dr. Isaacs received his Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara. His research interests include multiagent control systems, UAV path planning, localization and mapping, and sensor networks.

Dr. Vida Vakilian is an Assistant Professor of Computer Science at California State University, Channel Islands. From 2015 to 2019, she was an Assistant Professor at California State University, Bakersfield. Dr. Vakilian received her Ph.D. degree in Electrical Engineering from University of Montreal, Canada in 2014. Her research interests include wireless communication, signal processing and communication-aware robotics. Dr. Vakilian is the recipient of the Enhancing Access to the Radio Spectrum (EARS) Award from the NSF.

AN MULTI-DIMENSIONAL VIDEO REVERSE SEARCH ENGINE USING COMPUTER VISION AND MACHINE LEARNING

Qiantai Chen[1] and Yu Sun[2]

[1]Department of Computer Science
University of California - Irvine, CA 92697
[2]Computer Science Department
California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Online media has become a mainstream of current society. With the rapid development of video data, how to acquire desired information from certain provided media is an urgent problem nowadays. The focus of this paper is to analyse a sufficient algorithm to address the issue of dynamic complex movie classification. This paper briefly demonstrates three major methods to acquire data and information from movies, including image classification, object detection, and audio classification. Its purpose is to allow the computer to analyse the content inside of each movie and understand video content. Movie classification has high research and application value. By implementing described methods, finding the most efficient methods to classify movies is the purpose of this paper. It is foreseeable that certain methods may have advantages over others when the clips are more special than others in some way, such as the audio has several significant peaks and the video has more content than others. This research aims to find a middle ground between accuracy and efficiency to optimize the outcome.

KEYWORDS

Convolutional Neural Network, Image Classification, Object Detection, Audio Classification, Movie Classifier.

1. INTRODUCTION

The topic of video classification has achieved outstanding results in recent years, and diverse ways of approach such as implementing complex algorithms and architecture improved the overall accuracy in a certain genre of video [11][12]. By making the computational device understand and recognize the video content is a major challenge for nowadays technology applications. As the increasing amount of contribution has been done in this area, it is more often to recognize the hidden value for the video classification in both commercial and research areas. The usage of identifying the video content can be employed in numerous applications including autonomous driving, rocket science, internet streaming, home assistant, and analyzing sports video.

One of the areas that desperately needs the video classification is the online streaming application such as YouTube [13] and Netflix [14]. There are approximately more than five hundred of videos uploaded to YouTube every minute. The methods of automatically diagnosing and analyzing the language information, understanding the content, captioning the video with

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 125-136, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120911

categories and description, and coming up with relatively high accuracy compared with humans is a classic concern. On the other hand, image classification not only contributes to the entertainment application but also makes autonomous driving become a reality. There are several existing vehicle models capable of self-driving have been announced around the world for a while now, and some of the brands are really achieving some significant results in this area by utilizing the video classification real time with live feedback.

The traditional approach of video classification is that of creating a decision tree [1][2][3]; many researchers and research papers indicate that there is still a great amount of potential improvement leftover for decision tree's accuracy. Some methods include expanding the size of the dataset as well as combining several layers of discrete spectral information. However, with the rapid development of deep learning style machine learning [15], especially by utilizing the approaches including convolutional neural network, long short-term memory, and gated recurrent unit, modern artificial intelligence outpaces the traditional approaches in both commercial and academic use. These approaches successfully demonstrate significantly greater efficiency and more wide-spread application than decision trees. Some of the research publications on the topic offer even more specific approaches that utilize these modern deep learning techniques. It is common to acknowledge that object grounding is relatively mature technology and also has greater potential when used for both commercial and government areas. There are multiple ways to play around with object grounding [4] and the way is by using the different settings within the models including Zero-Shot Grounding-Net [5] and Video Object Grounding-Net [6]. On the other hand, in one of the research findings, which shows there is a significant margin when they are trying to identify the tennis videos with the sound [7]. In between classifying the video, looking for a potential existing paddling sounds that only could exist through playing tennis could quickly help the researcher to locate the desired frame. Another significant way of identifying the video content is by utilizing the Video Temporal Analysis, there are a lot of exciting methods underlying such as temporal activity detection, language-based video search, and action anticipation, but it is still an interesting topic to unraveling the mysteries due to the reasons that great number of videos has such complex temporal structures, great video variation and problem scale.

The methods that have been conducted in this research were those of creating a total amount of three coding programs to achieve a different result, since the problem set requires a variety of algorithms to solve it. An analysis will be conducted after implementing all three programs. The other method is by gathering the existing programs, and creating a complex algorithm that utilizes pre-existing results for further computation. The research goal is to examine the best feature extraction method and optimize both the effectiveness and efficiency of the program. The part of the program where we implemented the audio classification approach is inspired by the research paper written by Xun Gong and Fucheng Wang [7], who indicate there is a significant improvement in their program by identifying the sounds of hitting a tennis ball. The idea of selecting certain peaks within the voice and match noise, as well as the methods in which they did this, enlightened our thoughts of something similarly involving key film noise that appear in certain movie genres, such as explosions in the action genre. By considering this idea, searching for desired sounds gives the program ability to quickly eliminate a great portion of the sample data. There are some good features we take to achieve our result! Our approach utilizes multi-layered programming techniques, a giant database for object detection and recognition, interactable human interface design, and the ability to present an option for users to upload movies that are currently not listed in our database, thus updating the model further.

There are several different measurements implemented to ensure the results are following the prediction that we are expecting. The way to prove our result is by utilizing cross-validation in the design process, so that the program will automatically emphasize a confidence level of the

accuracy that these certain methods could eventually end up with. Meanwhile, the program of image classification uses application of neural networks, which utilizes the validation and testing datasets to determine if the processed results match with our two other additional datasets. To further mitigate the possibility of edge case, the program also has the ability to compare the model accuracy and select the highest accuracy among all of the models. There is also an option by doing it manually is to customize the model using hyper parameter tuning. Once our analysis of the results outlines recommended models, we select one for each area and move on to the next step of tuning the models in the system. From there it is easy to determine what changes are needed for our general heuristic and to get the highest accuracy possible for the combined system. The final stage for our program is by conducting a comprehensive case study on the system, and analyzing the statistical data to determine if it still meets both the standard of effectiveness and efficiency.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, followed by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In this section, we layout the key technical and research challenges to address.

Challenge 1: Maintaining project speed across multiple machine algorithms is inherently difficult. Even just one classification machine learning algorithm could consume a great amount of time and processing resources. With the time and space complexity of just one, trying to manage a reasonable speed in between the three identification methods that this paper is trying to imply is difficult. Given the complex nature of machine learning algorithms, it is important to maintain them across systems and limited computational resources. Therefore, parallel processing is the most efficient choice, especially when facing the design that algorithms should run independently without interaction or exchanging of information. The general solution is that we collect the results three times from each individual algorithm and then use a supplemental program to pick the final prediction with the highest confidence from the results generated. Parallel processing not only allows us to achieve a relatively high accurate result, but also presents the ability of manipulating the algorithm's speed in different circumstances. In terms of accuracy, parallel processing also allows for cross-referencing of the individual models' results as opposed to just picking the highest confidence. This allows for a decently robust solution.

Challenge 2: Selecting what to sample and how much of it to sample including frequency and length can lead to vastly different results. Selecting a database from the original sources could result in an insufficient amount of data or overwhelming unnecessary computation. Small changes to training or testing data can result in widely varying accuracies due to the nature of classification and other model designs. Movies, which are usually hundreds of gigabytes per film, are especially an issue when you take into account the number of movies that would need to be preprocessed for such a project. The concept for optimizing the right amount of data is usually the hardest thing to do given consideration for both selecting data now as well as updating it in the future with the consistent release of films. Videos themselves can also vary considerably in quality even from the same footage of the film, so it can drastically affect predictive performance. Generally, the particular solution is that we process every frame of the movie so that when the algorithms come in, we will always gain the best dataset. In this approach, we sacrifice the

calculating power and receive a relatively complete set of data. By ignoring certain frames of the video, such as the beginning of the movie where it usually just starts with the black images, movies usually take several frames to switch from one scene to another. One way of mitigating the general solution's risk is by measuring the average change of scene and rescale it under different genres or categories of a movie. Meanwhile, to prevent the inaccurate dataset being processed without our knowledge, a validation dataset and training dataset has been set up to make sure the result is exactly what we expected.

Challenge 3: Many machine learning models are better suited for individual tasks. For example, image classification often runs better using Neural networks than standard classification algorithms. This variance forces researchers to study different potential solutions to find the optimal one. There are so many existing models that have been created and published to the world, and only few of them are worthy to investigate deeper into it, as the nature of the concerns in mathematical, statistical, and computational fields are different in each model. How to decide the best model is definitely one of the hardest parts of the whole experimenting process, since we are just unable to try all of the published data models. Our problem requires us to not just go through the process of selecting an individual model, but doing this three times for three entirely different subjects. The general solution is by going through a selection of models or just randomly picking a common machine learning model which is highly recommended at the time. You then run a few tests to generate the result but not enough to require into the greater depth and understanding of each model and its algorithm. This paper is doing research based on other researchers' results, and coming up with the machine learning model that is already working. We are not only examining the validity of the model but also selecting the model that has proven most useful when applied to similar problems.

3. SOLUTION

The specific solution to build the system and address the challenges above will be presented in this section.

3.1. Overview of the Solution

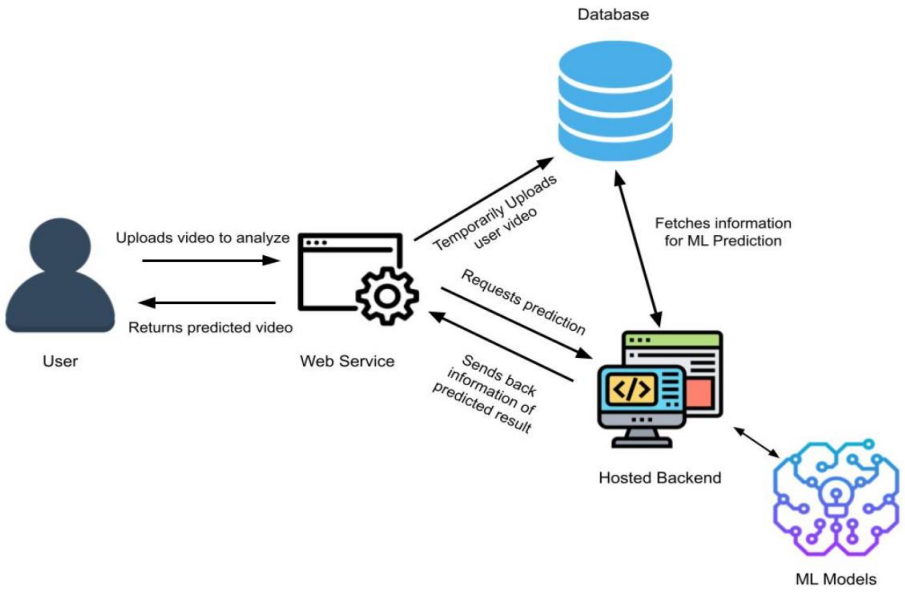


Figure 1. An Outline of the Video Prediction Web Service

The user has the option to upload a video onto a website page, where it connects to both the database and backend system. The interaction between user and system is solidly limited to the web service front end that we implemented only for proper user interaction, although the web service page will eventually return a predicted movie name back to the user. There are several processes going on behind the scenes of our web service page. The web services will transfer the user uploaded video to the database for temporary storage, and access the hosted backend to request a prediction for the temporary video clip. The clip will be accessed later on by the backend and machine learning portions of the application. The step is starting with the database, which passes the movie clip uploaded from the user as an output, and the hosted backend will accept the movie clip as input for further evaluation and process. Once the video has been processed, the machine learning models involved in the program will process the movie clip with each of the different methods and tools to analyze what film it may have originated from. Once the movie clip is processed, the result will transfer back to the backend, which will access the pre-processed original movie dataset stored in the database. After finalizing the result coming out of the system, the hosted backend will essentially present a piece of information regarding the final predicted result from the system back to the web service page, which is the final stage for the whole system, displaying the end result for the user.

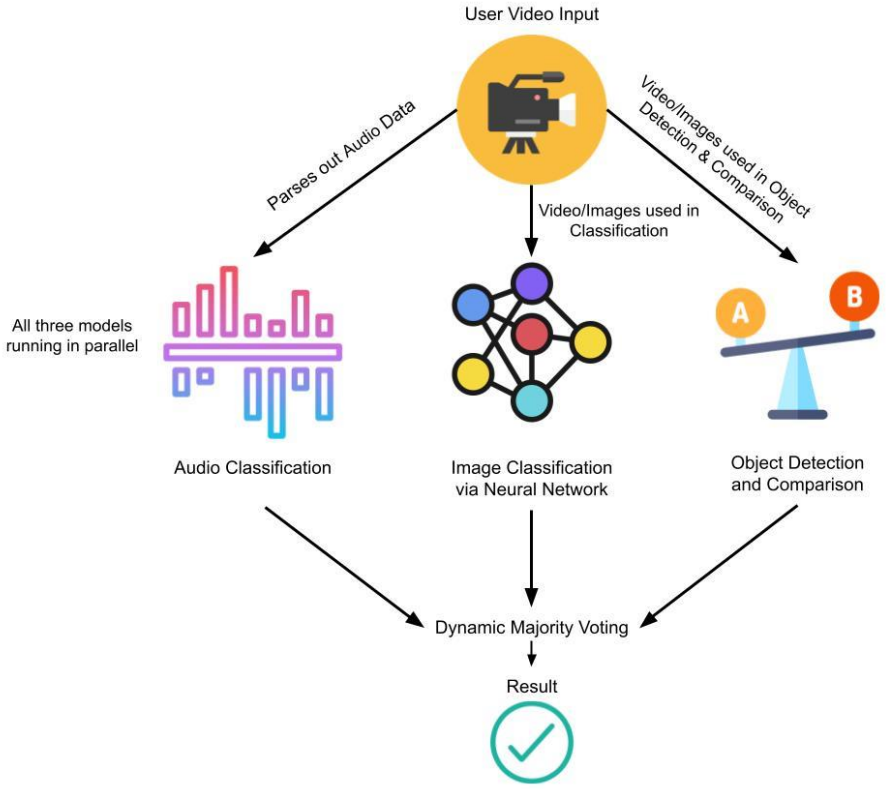


Figure 2. Outline of the custom machine learning process

3.2. Audio Classification

The audio classification is one of the most mature techniques in nowadays, there are many audio streaming companies implement a similar build in functionalities in their application. The hardest part of this research is to determine how we want to extract the audio from either database or user upload. The following code segment emphasis the idea of how we decide to select the audio, and deliver the audio part for further analysis. Conversion from a .mp4 file to .mp3 file is first step

needs to be processed. Every 3 thousand millisecond an audio segment will be selected and converted into a .wav file, which is much easier for computational device to do the machine learning. The algorithm will do the segment-level feature extraction, which gets feature matrix for each dataset. When the feature matrix successfully extracted, the algorithm can perform machine learning including cross validation and convolutional neural network.

```
# Length in Milliseconds, every 3k ms we are going to cur a part of the audio
current_time = 0
interval = 3000
counter = 0

while current_time < len(sound):
    print("Gerring clip at time {}".format(current_time))
    new_clip = sound[current_time:current_time + interval]
    new_clip.export("processedAudio/PacificRim" + str(counter) + ".wav", format="wav")
    current_time += interval
    counter +=1
```

Figure 3. The code excerpt of the custom audio file process

3.3. Image Classification

Since the application is expecting database in great scale, the first thing is to shrink however the database it has to exactly what we require to do the prediction, and avoid unnecessary computer computing recourses to decrease the efficiency of algorithm. The algorithm is already processed the image by rescaling the frame, taking screenshots of data in every three seconds, grey-scaling the image, and categorize the dataset. Meanwhile, the algorithm also initiates a validation dataset, testing dataset, and training dataset. Each dataset creates for purpose of providing dataset for algorithm self-predicting, saving original database for future reference, and getting ready for convolutional neural network machine learning respectively.

```
model = Sequential([
  data_augmentation,
  layers.Rescaling(1./255),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
model.compile(optimizer="adam",
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure 4. Outline of the custom machine learning process

The above code segment is designed to create the model, which consists of three convolution blocks with a max pool layer in each of them. There's a fully connected layer with 128 units on top of it that is activated by a *relu* activation function. In order to address the overfitting issue, the application automatically eliminate duplicate frames.

3.4. Object-based Classification

By utilizing a public object detection architecture named Yolov5 and modify behavior of the code, we are able to detect object position with each of the frame and stored the information in a .json file. The coordinate of the boxing boundaries and type will be placed in .json file. As long as we have the analyzation for each frame, we are able to place each .json file into a new .json file which combined all frame analyzation for one movie. At this point, each movie will have an individual .json file, and its ready to do next step.

```
def drawBBox(x1, y1, x2, y2, typeofobject, imageList):
    try:
        for row in range(y1, y2):
            for col in range(x1, x2):
                imageList[row][col] = itemdic[typeofobject]
                print("Found a {} assigning it a value of {}".format(typeofobject, itemdic[typeofobject]))
    except IndexError:
        print("Invalid xy: ", x1, y1, x2, y2)


def generateDataSet(data, outputValue):
    for imagekey in data:
        myList = [[0 for i in range(640)] for j in range(640)]
        for objectkey in data[imagekey]:
            list1 = list(data[imagekey][objectkey].values())
            if list1[4] not in itemdic:
                itemdic[list1[4]] = len(itemdic) + 1
            drawBBox(list1[0], list1[1], list1[2], list1[3], list1[4], myList)
        flattenList = [j for sub in myList for j in sub]
        allimageslist.append(flattenList)
        outputData.append(outputValue)
    print(itemdic)
```

Figure 5. The code excerpt of plotting objects to matrix

As shown in the code segment above, we first generate a 640 x 640 matrix, which purpose is to place the desired bounding box generate by the coordinates in previous .json file. In the for loop, the code will process a Metrix for each frame, and create a unique dictionary for each type of object detected in Yolov5. By using the coordinates, we now have the object shown in the matrix, so that we are allowed to do the prediction based on the different libraries and tools including cross validation and convolutional neural network. In addition, for purpose of eliminating minor differences within certain frames, the code is also design a algorithm, which can automatically shift, flip, and rescale the bounding box, if it gains a high confidence level of matching object.

4. EXPERIMENTS

This paper conducts two sets of experiments for a series of model solutions that we aim to combine into one approach. This includes testing both the overall accuracy of a selection of models as well as the average speed for a selection of models, for each type of solution. The first experiment ties directly to model accuracy as it is of paramount importance.

4.1. Evaluation of the Searching Effectiveness

In the case of audio classification, each experiment tests with different levels of data, one of the experiments has two elements undergoing cross validation with five splits, while the other experiment receives five elements and the same amount of cross validation splits. Both experiments point out that the Support Vector Machine (SVC) model has the highest accuracy with 76% and 53.06% respectively. Meanwhile, the testing dataset is created by using three seconds slices of a single movie.

For the object detection solution, under the framework of cross validation with doubled data received the highest accuracy of 43.04%, which is achieved under SVC model, however the model emphasizes an outstanding result in terms of effectiveness, the timing of SVC model gives a hint that this is still debatable or considered as an efficient choice.

The most outstanding outbreak achieved in this research is that of the highest accuracy in our image classification method. By utilizing neural networks and several libraries in TensorFlow. Our neural network image classification achieved a 90% when tested against its training dataset while achieving a 71% accuracy in the validation dataset testing. As shown below in right-hand side of the figure 6, the correlation between self-prediction regarding the validation and testing slope becomes more and more readable, more importantly, the slope reaches a stable level as it approaches the end. By increasing the epoch will not highly affect the accuracy in this case, and the potential overfitting issues has been addressed by implementing dropout layer to neural network and data augment will essentially get rid of similar screenshots. One the left-hand side the figure 6 emphasizes the slope correlation before eliminating the over fitting issue.

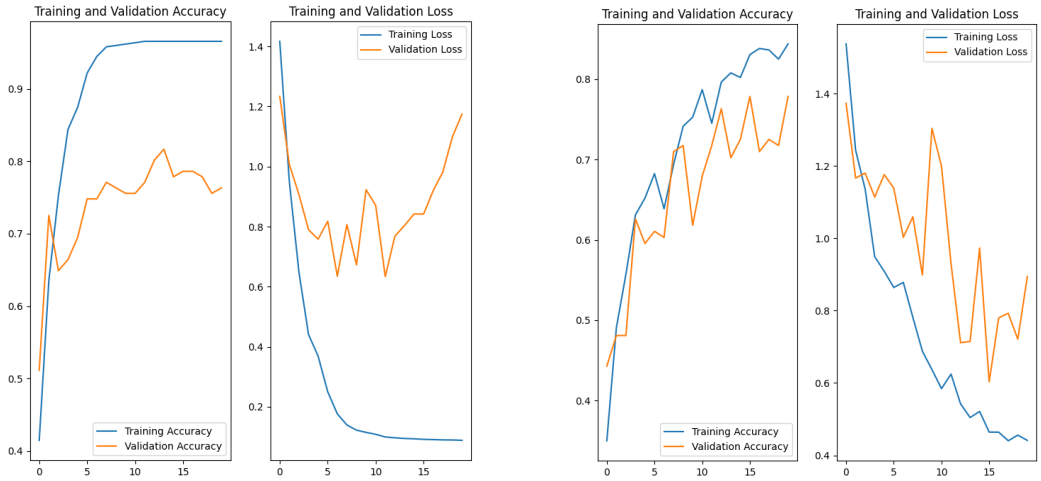


Figure 6. Overfitting model vs. Model with drop out layer

The parallel programming's accuracy is a mean average of all other three approaches, which are approximately at 55.94% accuracy. In fact, the program utilizes a different approach than picking the most votes from three approaches. Since the image classification reaches the highest breakthrough, the program will take object recognition and audio classification into consideration only if in the situation where neural network image classification ends up with a confidence level under 60%.

4.2. The Evaluation of Searching Efficiency

Every program needs to consistently consider the tradeoffs between efficiency and effectiveness. The result for the audio classifier model indicates that although SVC has the most satisfying accuracy, however, in terms of speed, the Gaussian I Bayes (GaussianNB) gives relatively lower accuracy with 47.34% of accuracy and seventy-two times faster than SVC. As shown in the figure 7, the comparison is made under the unit of seconds.

Likewise, the object detection emphasizes a similar outcome to that of audio classification models. The SVC model once again gives the highest accuracy, as described above, and the relatively slowest time when compared to other tested models. The GaussianNB has a lower accuracy of 42.67% but finishes the task ten times faster than SVC.

Image classification under a neural network reaches the highest accuracy across all of our approaches, in terms of efficiency, it is actually considerable given that it takes 0.103 seconds to make the prediction.

The average timing for parallel processing varies on how many steps are involved. If the program's image classification reaches a confidence level higher than 60%, the average for prediction is how long it takes to do the image classification. In case the confidence level drops under 60%, two other approaches will get involved as a backup resource to support the result, and it will basically take approximately a total of 2.25 seconds.

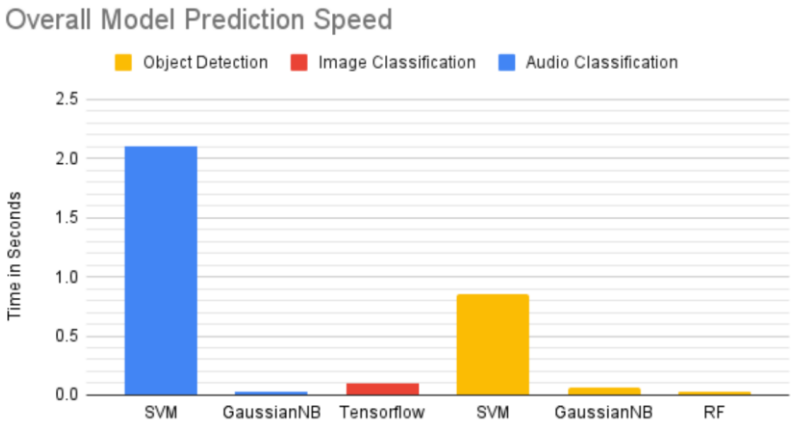


Figure 7. Subsample of machine learning speed tests for each method

4.3. Further Experiment Result

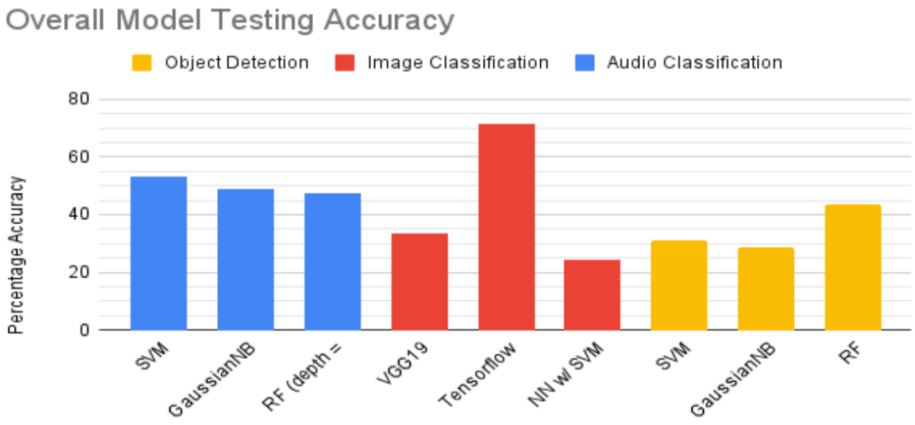


Figure 8. Subsample of machine learning accuracy tests for each method

To select a model, we compare various models and then run multiple tests across them such as cross-validation to pick the most accurate one. We then do hyperparameter tuning to tune it to exactly what we need. For Image Classification we compared SVM, RandomForest, passive aggressive, and convolutional neural networks with results of 65%, 77%, 28%, and 91% respectively.

For object detection, we run the prediction under different numbers of dataset. When cross validation at level 2 with no data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), RandomForest (no max depth) with result 31.4%, 25.26%, 31.86%, 33.33% respectively. When cross validation at level 7 with no data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), RandomForest (no max depth) with result 31.1%, 28.57%, 32.23%, 43.25% respectively. When cross validation at level 7 with data duplication, we compared SVM, GaussianNB, RandomForest (max depth 2), with result 43.04.%, 42.67%, 37.17% respectively.

In case of audio classification, we compared cross validation at level 5 with 5 elements. The following models has been tested, SVM Classifier (MidTerm Features), Random Forest (MidTerm Features, max_depth = 2), Random Forest (MidTerm Features, no max_depth), Passive Aggressive (MidTerm Features), GaussianNB (MidTerm Features) with result 53.06%, 48.97%, 46.53%, 37.96%, 47.34% respectively.

5. RELATED WORK

X. Gong and F. Wang [7] noticed that the result of tennis video detection will be significantly more efficient by applying the audio detection. This audio-based approach assists the detection process with a video including a tennis event. With a shortcoming that the support vector machine supports a maximum of two classification problems at the single time, the author implements a method of combination of support vector machine and decision tree support vector machine to tackle the audio multi classification problem. L. Wang, H. Liu, and F. Sun [8] implemented a soft coding bag of dynamic systems to perform a satisfactory performance in the area of extreme learning machines. The paper emphasizes the system efficiency when it applies to the public database with algorithm and comparison in between the industry common method and this BoS approach. The main difference between this work is that we are using different

dataset. V. Lopez-Vazquez., et al. [9] emphasizes that classic deep learning approaches are able to tackle the complex neural networks in the area of unexplored environments with even low-quality images. Moreover, the author indicates that with a higher rate of enhancement image could potentially assist the work of detection of features and gain better classification accuracy. In addition, deep learning approaches have been used to detect underwater animals in this paper. J. Gao., et al. [10] implemented several approaches including temporal boundary regression, temporal unit regression network, and more to address current problem in long untrimmed videos with generate temporal actional proposals.

6. CONCLUSIONS

Three approaches are capable of delivering a valid result with high confidence analysis for the video clip prediction. By utilizing the parallel programming, the final state of application by combining three approaches will finalize and leverage the result both in terms of effectiveness and efficiency. The accuracy level of image classification emphasizes that in ordinary situations, the prediction is strong enough to provide a meaningful result. The application requires a comprehensive dataset in order to make a prediction at a high level. By asking the user to upload their video clips and movies will enhance the database and the categorized dataset will save the result for future references, which means that while the user uses it, the application is learning and analyzing at the same time. All of the classification approaches present at least 90% accuracy when the database analyzes it in advance, even if there are minor differences when capturing the frame and voice, since the application has been designed to handle it by rescaling, rotating, shifting the frame. Even with the video clip that the database has never seen before, the image classification will brief the prediction and confidence level and the application will decide whether or not to involve the other two approaches. The prediction for the validation dataset is proven to be trusted and accurate. Speaking of image classification, in the future, the application will not limit to the current defined architecture and libraries, and push the accuracy to state-of-art performance by implementing the application under other existing techniques such as PyTorch. Likewise, the performance of object detection and audio classification will also rebuild to advance the result.

In the application, all of the approach is still not applying state-of-art performance models, which means that once it is getting published, this application has already become a past tense. The growth rate of deep learning technologies just requires tons of research readings and learning from the other's work. The limitation of human resources causes the application has not tried all of the existing methods. On the other hand, mobile devices are the most popularly used device currently and sadly we have not decided when to release the mobile application due to the limitation of human resources to maintain the system in the future.

In the future release, several models will be given consideration and implementation for a single approach, as we want to select from new models that are developed and could potentially increase accuracy, but also back up the experiment with stronger evidence. Also, enhancing the user experience on a mobile devices' browser will be strongly considered.

REFERENCES

[1] S. Moral-García, C. J. Mantas, J. G. Castellano, M. D. Benítez, and J. Abellán, "Bagging of credal decision trees for imprecise classification," *Expert Systems with Applications*, vol. 141, p. 112944, 2020.

[2] F. Alam, R. Mehmood, and I. Katib, "Comparison of decision trees and Deep Learning for object classification in autonomous driving," *Smart Infrastructure and Applications*, pp. 135–158, 2019.

[3] M. A. Friedl, C. E. Brodley, and A. H. Strahler, "Maximizing land cover classification accuracies produced by decision trees at Continental to Global Scales," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 969–977, 1999.
[4] X. Yang, X. liu, M. Jian, X. Gao, and M. Wang, "Weakly-supervised video object grounding by exploring spatio-temporal contexts," *Proceedings of t^he 28th ACM International Conference on Multimedia*, 2020.
[5] J. Ye, X. Lin, L. He, D. Li, and Q. Chen, "One-stage visual grounding via semantic-aware feature filter," *Proceedings of t^he 29th ACM International Conference on Multimedia*, 2021.
[6] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Learning visual affordance grounding from demonstration videos," *arXiv.org*, 12-Aug-2021. [Online]. Available: https://arxiv.org/abs/2108.05675. [Accessed: 23-Dec-2021].
[7] X. Gong and F. Wang, "Classification of tennis video types based on machine learning technology," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, 2021.
[8] L. Wang, H. Liu, and F. Sun, "Dynamic texture video classification using Extreme Learning Machine," *Neurocomputing*, vol. 174, pp. 278–285, 2016.
[9] V. Lopez-Vazquez, J. M. Lopez-Guede, S. Marini, E. Fanelli, E. Johnsen, and J. Aguzzi, "Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories," *Sensors*, vol. 20, no. 3, p. 726, 2020.
[10] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.
[11] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).
[12] Brezeale, D., & Cook, D. J. (2008). Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(3), 416-430.
[13] Arthurs, J., Drakopoulou, S., & Gandini, A. (2018). Researching youtube. *Convergence*, *24*(1), 3-15.
[14] McDonald, K., & Smith-Rowsey, D. (Eds.). (2016). The Netflix effect: Technology and entertainment in the 21st century. Bloomsbury Publishing USA.
[15] Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. nature, 521(7553), 436-444.

Automatized bioinformatics data integration in a Hadoop-based data lake

Júlia Colleoni Couto, Olimar Teixeira Borges, and Duncan Dubugras Ruiz

School of Technology, PUCRS University,

Abstract. When we work in a data lake, data integration is not easy, mainly because the data is usually stored in raw format. Manually performing data integration is a time-consuming task that requires the supervision of a specialist, which can make mistakes or not be able to see the optimal point for data integration among two or more datasets. This paper presents a model to perform heterogeneous in-memory data integration in a Hadoop-based data lake based on a top-k set similarity approach. Our main contribution is the process of ingesting, storing, processing, integrating, and visualizing the data integration points. The algorithm for data integration is based on the Overlap coefficient since it presented better results when compared with the set similarity metrics Jaccard, Sørensen-Dice, and the Tversky index. We tested our model applying it on eight bioinformatics-domain datasets. Our model presents better results when compared to an analysis of a specialist, and we expect our model can be reused for other domains of datasets.

Keywords: Data integration, Data lake, Apache Hadoop, Bioinformatics.

1 Introduction

Data integration is a challenging task, even more nowadays where we deal with the V's for big data, such as variety, variability, and volume (Searls [1]; Lin et al.[2]; Alserafi et al. [3]). Regarding the variety of data to be integrated into data lakes, having different types of data can be considered one of the most difficult challenges, even more because most datasets may contain unstructured or semi-structured information (Dabbèchi et al. [4]). According to Hai, Quix, and Zhou [5], it is very onerous to perform interesting integrative queries over distinct types of datasets. Another challenge is the high-dimensionality data that may be stored in the data lake. To compute the similarity for that high-dimensional data is expensive. Checking whether the tables are joinable or not is time-consuming because of the large number of tables that may have in a data lake (Dong et al. [6])

To manually analyze different datasets for data integration, a person must check the attributes and at least a dataset sample. To perform a more elaborated work, the person must look for the data dictionary of each dataset, and sometimes it is not easily available. According to Sawadogo and Darmont [7], it is a problem since it is time-consuming, error-prone, and can lead to data inconsistency. Among the methods for data integration, the logic-based ones that consider the dataframes as sets, such as the based on the overlap of the values, could provide useful solutions (Levy [8]).

This paper presents a model we developed to perform heterogeneous data integration, taking advantage of a data lake we build based on Hadoop. The integration model is based on schema matching techniques, such as row content-based overlapping. To do so, we first define the datasets for the experiments, related to the domain of bioinformatics. Then, we build a data lake to ingest, store, process, and visualize the data. We use Apache Nifi for data ingestion and the HDFS (Hadoop Distributed File System) for data storage. We process the data using Python, and we create visualizations of the data using Neo4J.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 137-153, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120912

Our main contribution is a model that allows to quickly ingest different kinds of textual datasets, transform them into dataframes, and, using an approach based on in-memory set similarity for data integration, we suggest the top-k points of integration for the data. We present experiments with eight bioinformatics datasets, and we compare our approach with manual data integration performed by a domain specialist. Our paper can also be used as a guide to building a data lake from scratch.

In what follows, we investigate automatized data integration. Section 2 presents an explanation about the main topics that are essential for our study and related work. Section 4 describes the methodology we followed, and Section 5 presents the results we achieved. Section 6 discusses our results, challenges, and presents an example with the usefulness of our model. Finally, Section 7 summarizes our conclusions and future work.

2 Background

In this section, we briefly present the basic concepts related to our study. We summarize data integration, data lake, and present the datasets we used in our experiments. We finish by discussing the related work.

2.1 Data integration

Data integration deals with the problem of combining different data sources to provide the user with a unified view (Lenzerini [9]). There are different approaches for data integration, and we base our work on the top-k overlap set similarity problem: For all the attributes in all the dataframes, find the top fits for data integration, according to the intersection among the attributes' distinct values (Zhu et al. [10]). We based our work on an in-memory set similarity approach since the integration is executed using Python notebooks. As similarity metrics, we use the most well-known distance measures for sets similarity according to Ontanón (2020) [11]: Tverski's (Tversky [12]), Sørensen's index (Sørensen [13]), and Jaccard (Jaccard [14]), compared to the Szymkiewicz-Simpson overlap coefficient (Vijaymeena and Kavitha [15]). We develop our data integration based on a data lake.

2.2 Data lake

In a previous work (Couto et al. [16]), we define a *data lake* as a central repository for raw data storage, processing, and analysis, that can be used for unstructured, semi-structured, and structured datasets. A data lake can be composed of different software with its own tasks in an integrated ecosystem. It means we can have different software for data ingestion, storage, processing, presentation, and security, and they have to work together. The most used tool to create a data lake is Apache Hadoop [16]. Forster [17] states that Hadoop is the most used distributed platform for storage, processing, and analysis of big data. Hadoop is an Open-Source Software (OSS) developed in Java and maintained by the Apache Software Foundation [18]. Hadoop is based on the Google MapReduce paradigm and in Google File System, and it is mainly used for distributed processing in computer clusters. We populate our data lake with bioinformatics datasets.

2.3 Bioinformatics datasets

Bioinformatics is the product of the union of computer science and biology (Lesk [19]), where we use software to make inferences about datasets of modern molecular biology, so we can connect the data and extract valuable predictions. There are a lot of bioinformatics

datasets available, having the most variate information, formats, types, and size. Our study selected eight datasets to populate our data lake and work on automatized data integration. Table 1 presents the characteristics of each dataset, ordered by size from the smaller (DRUGBANK) to the larger (IID). Table 1 shows that we selected heterogeneous datasets, having varied sizes (from 1 MB to 1,8GB), from 13k entries to almost 1 million entries, with the number of attributes varying from 6 to 253. The datasets are also presented in different formats, such as TXT, XML, TSV, JSON, and via API.

Table 1. Characteristics of the bioinformatics datasets

Dataset	Size (MB)	Entries	Attributes	Format
DRUGBANK	0,95	13580	9	XML
DRUGBANK PROTEIN	1,40	26965	7	XML
OMIM	1,80	17092	14	TXT
DRUGCENTRAL	2,50	17390	19	TSV
MONDO	4,00	43233	12	JSON
DISGENET	10,30	84037	16	TSV
UNIPROT	30,20	565255	7	API
REACTOME	37,90	826877	6	TXT
IID	1800,00	975877	253	TXT

- OMIM (McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD), 2021 [20]): Online Mendelian Inheritance in Man - human genes and genetic phenotypes. We used the *genemap2* dataset.
- DISGENET (Pinero et al. [21]): Collections of genes and variants associated with human diseases.
- REACTOME (Jassal et al. [22]): We are using the *UniProt2Reactome* dataset. It is composed of reactions, proteins, and pathways.
- MONDO (Mungall et al. [23]): Ontology for disease definitions.
- DRUGBANK (Wishart et al. [24]): Pharmaceutical knowledge base, we split it into two dataframes: DRUGBANK and DRUGBANK_PROTEIN.
- IID (Kotlyar et al. [25]): Integrated Interactions Database - database of detected and predicted protein-protein interactions. We used the human data annotated dataset.
- DRUGCENTRAL (Avram et al. [26]): Online drug compendium - we use the drug-target interaction dataset.
- UNIPROT (Consortium [27]): We are using the *reviewed Swiss-Prot XML* dataset, a non-redundant and manually annotated database containing protein sequences.

2.4 Related work

The work of Cockell et al. [28] describes Ondex, a Data Integration Platform. They represent the data as a graph, where the nodes present the concepts, and the edges present the relations. They developed parsers to import the data to OXL format. Then, they use mappers and transformers to join different types of datasets, remove nodes that are not connected and add information to the network. Finally, they manually traversed using Ondex to search for interesting examples of drug repositioning. They use the following datasets: DRUGBANK, UNIPROT, HPRD, KEGG, PFam, SymAtlas, G-Sesame, OpenBabel, and BLAST. They use the cross-references presented on Uniprot to include accession numbers from other linked datasets (e.g., ENSEMBL, GO, OMIM, PRINTS).

Sellis et al. [29] use web services and ontologies to create Semantic Web services, to integrate three biological databases: EMBL, MEDLINE, and Array Express. They answer the following query: "for a given Nucleotide Number (EMBL database), find all experiments (Array Express database) and all publications (MEDLINE) which have taken place." They use OWL-S - an ontology-based on Web Ontology Languages that describe web services.

In his work, Hendler [30] discusses themes related to data integration, discovery, linked data, and the combination of structured data and unstructured data, and the author presents some theoretical approaches to deal with issues that come from heterogeneous datasets integration. For instance: use of natural language processing, graph databases, alignment using a third dataset.

Petermann et al. [31] developed a model named Business Intelligence with Integrated Instance Graphs, which they use for graph-based data integration and analysis. Their model has three types of graphs (separate graph databases): one for Unified Metadata (UMG, where the nodes are the classes and the edges are the associations), one for Integrated Instance (IIG, where the nodes are data objects and the edges are the relationships) and the last one for Business Transactions (BTG). In their process, they first perform metadata acquisition and integration for the UMG, then instance integration to create the IIG, generation of BTGs, and graph analytics. They use Neo4J.

Bradshaw et al. [32] developed an automatic and semi-automatic semantic data integration approach, based on concept bags, for synonyms and non-synonymous concepts. Concept bags are similar to word bags used in data mining. They compute the similarity between data elements and medical terms. To check the similarity, they use the Jaccard algorithm. They convert text or named entities in concept codes and then compare it using a vector-based analysis method. They use the following datasets: UMLS (315 entries), REDCap (899649 entries), Medical terms (60 entries). They state that their method presents the same or better performance when compared to other approaches.

Zhu et al. [10] develop JOSIE: an algorithm for JOining Search using Intersection Estimation. They use inverted indexes (mapping from words to their location - for quick search in text files). They work with the join table search problem: for a column C in a table, find other tables in the data lake where the intersection between column and C is high. They use two data lakes: Open Data and WebTables. They compare their results with MergeList-D and ProbeSet-D.

Zhang and Ives [33] develop JUNEAU, an approach to support multiple table relatedness measures, such as augmenting training data, finding potential features to extract, clean data, and finding joinable or linkable tables. They use pruning, top-k, and approximation strategies to return the tables that are most related.

When we compare our work with JOSIE [10] and JUNEAU [33], which are the most related, the main difference is related to our algorithm performing the data integration of more than two dataframes and simultaneously outputting it to take advantage of the raw data in the data lake.

3 Problem Statement

As stated by Khalid and Zimányi [34], managing and querying a data lake is a difficult task, mainly because the data is heterogeneous, may have replicas or versions, have a considerable volume, and present quality issues. In this sense, data integration, which represents 80-90% of the challenges for data scientists (Abadi et al. [35]), is a fundamental task to enable querying a data lake. However, integrating heterogeneous data into data lakes is a complex task, mainly due to the variety of data types (Hendler [30], Alrehamy

and Walker [36] that can compose the data lake. If we only talk about textual data, there are countless extensions and possible formatting, such as: .txt, .docx, .csv, .xls, .xml, .json and so on. Furthermore, the analysis for the integration depends on experts, often data scientists, who need to spend time inspecting data profile information, such as the types of each attribute, a sample of that data, or studying the data dictionary - when the dictionary is available. Finally, data integration is essential for extracting a more holistic view and information from the data lake, enabling us to make simple to complex queries and add value to the information.

Therefore, for the problem of automatic data integration in data lakes, the input would a number of heterogeneous datasets and a threshold that limits the integration points of interest. The output would be the points of integration among the datasets, and the evaluation measures would be the ones based on an expert evaluation of the integration points.

Regarding the complexity of the problem, according to Alserafi et al. [3], the equation to calculate the total number of comparisons that needed to be performed to find the columns candidate to data integration is

$$comparisons = \left[d \times \frac{d-1}{2} \right] \times m^2 \tag{1}$$

where d represents the number of datasets, and m represents the average number of attributes for each dataset. Considering our datasets (previously presented in Table 1), we have 344 attributes in total, considering 9 dataframes, then $m = 38$. Thus, we would have to perform about 51984 comparisons among the attributes.

4 Methods

The purpose of this study is to create a model for automating the integration of datasets. To do so, we use similarity measures to check the possibility of data integration in a Hadoop-based data lake. We started by creating the *system architecture* for the data lake, based on Docker containers. Then, we worked on *data management*. Finally, we present the *algorithm* we developed.

4.1 System architecture

Our data lake is supported by an Ubuntu 20 64-bit Linux server, having the following configuration: 16GB RAM DDR3, Processor Intel® Core(R) I7-4790 CPU@3.6GHz x 8, 1TB disk capacity. The data lake is composed of ten Docker containers:

1. Apache Nifi: a framework used for data ingestion.
2. Python - Jupyter Notebook: a programming language and a web application to run Python code, used for data processing.
3. Neo4J: a graph database used to visualize the integration among the dataframes.
4. Hadoop Namenode: the master node in the HDFS architecture
5. Hadoop History Server: keeps the logs of all the jobs that ran on Hadoop.
6. Hadoop Resource Manager: contains the YARN (Yet Another Resource Negotiator), a service that manages the resources and schedules/monitors the jobs.
7. Hadoop Node Manager: launches and manages the containers on a node.
8. Hadoop Datanodes: Three containers (Datanode1, Datanode2, Datanode3). The worker's nodes in the HDFS architecture, where the data is stored.

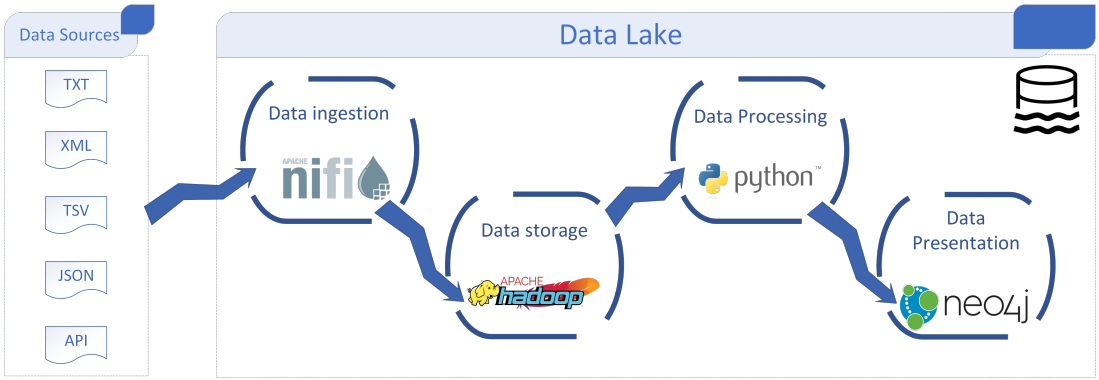


Fig. 1. Composition of the data lake

4.2 Data management

Data management includes data ingestion, storage, processing, and presentation, as illustrated in Figure 1. We ingested data into the data lake by creating processes in Apache Nifi. We create one process for each dataset, where the process searches for the dataset on HTTP (for MONDO, REACTOME, DISGENET, IID, and DRUGCENTRAL), or in a local folder (for OMIM and DRUGBANK, because they are not available directly due to the need of registering and licensing). Then we unzipped some of the datasets, and we renamed them all for standardization. Apache Nifi then sends the datasets to Hadoop, where they are stored in HDFS. For UNIPROT, we use the API directly on Jupyter. Then, we start data processing using the Python - Jupyter Notebook docker. Lastly, we create a graph visualization, based on Neo4J, to present the results.

4.3 Algorithm

We start the experiments by turning the datasets into Python Pandas dataframes. Pandas is a Python library for data analysis and manipulation. The standardization of the datasets as dataframes assure a unified entry for the algorithm, solving issues regarding one dataset being derived under one condition and the others being on other conditions. To create the DRUGBANK dataframe, we based on the solution provided by [37]. We also use other libraries, such as HDFS, that provide a pure HDFS client, bioservices that provide API access to UNIPROT, and the package *py_stringmatching* that implements the similarity metrics.

After creating the dataframes, one of the authors, a specialist in data science, analyzed the datasets to manually map the attributes candidates for points of integration. To do so, the specialist analyzed the names of the columns and a sample of data for each column, using data profiling techniques. The specialist took about four hours to finish this analysis, and we present the manual mapping in Figure 2. Figure 2 presents the manual data integration points, based on a graph visualization, where the nodes or vertices are the names of the dataframes, and the edges are the attributes' names. The orientation of the arrow indicates that, for instance, the attribute 'lbl' from the Mondo dataframe is a point of integration to the dataframe Disgenet, meaning that a percentage of 'lbl' is also present in another attribute of Disgenet. We developed this Figure to be later compared with the results of the algorithm we developed for data integration so that we could compare a user specialist analysis with the algorithm output.

Our algorithm is based on the concept of intersection or overlap between two attributes in sets of data. We first identify the unique values of each attribute for each dataset. Then

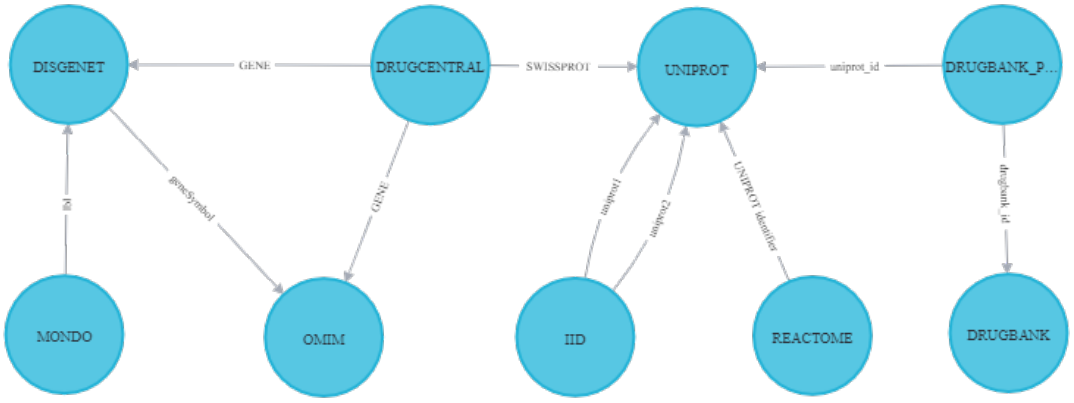


Fig. 2. Manually mapped integration

we compare each dataset attribute with all the other datasets' attributes to check if the unique values of the content of each attribute are contained in any other attributes of all of the other datasets. The attribute with fewer unique values indicates the orientation of the data integration. For instance, let us analyze the following case that includes dataframes (df) and attributes (att):

- df1['att01'] has 10 unique values;
- df2['att06'] has 20 unique values;
- 10 values from df1['att01'] are also present on df2['att06'].

In that case, we can notice that 100% of df1['att01'] are also present in df2['att06'], being that a good point for data integration. The notation would be: df1['att01'] $\longrightarrow$ df2['att06']. Regarding the minimum value for data intersection, we defined a threshold of 0.4 (in a range from 0–1)] to identify good integration points, but it is configurable according to the user's needs. It means that if two columns in a dataframe have 40% or more of data in common, the two columns are candidates for data integration, and the dataframes where the columns come from are integrable.

To define the best threshold for our experiments, we tested different values and compared the results with the specialist's analysis. We started with 0.9, and after each execution, we compared our results with the specialist's manual integration. When we noticed that the selected threshold retrieved at least all of the integration points defined by the specialist, we stopped decreasing the threshold, determining the value of 0.4.

Figure 3 details the activities diagram for the Algotrithm 1 we developed. Figure 3 shows that we can configure restrictions to select the attributes to be analyzed, such as the minimum number of unique values that an attribute must have to enter in the comparisons, and if we want to perform comparisons with attributes that contain only numeric values. Other restrictions include: removing attributes with only nulls or NaN and removing attributes with binary values (0 or 1, T or F). The binary values would not present real candidate points for data integration among the datasets since they mostly represent True or False values. For instance, the IID dataset presents dozens of attributes that are named after diseases, where the value = 0 corresponds to False and values = 1 corresponds to True (e.g.: 'bone disease', 'overnutrition', 'asthma', 'lymphoma').

The algorithm starts by creating dataframes for all the datasets, then it selects the first source dataframe, which will be compared to the other dataframes. Then the attributes of the source dataframe are compared with the attributes of the first candidate dataframe

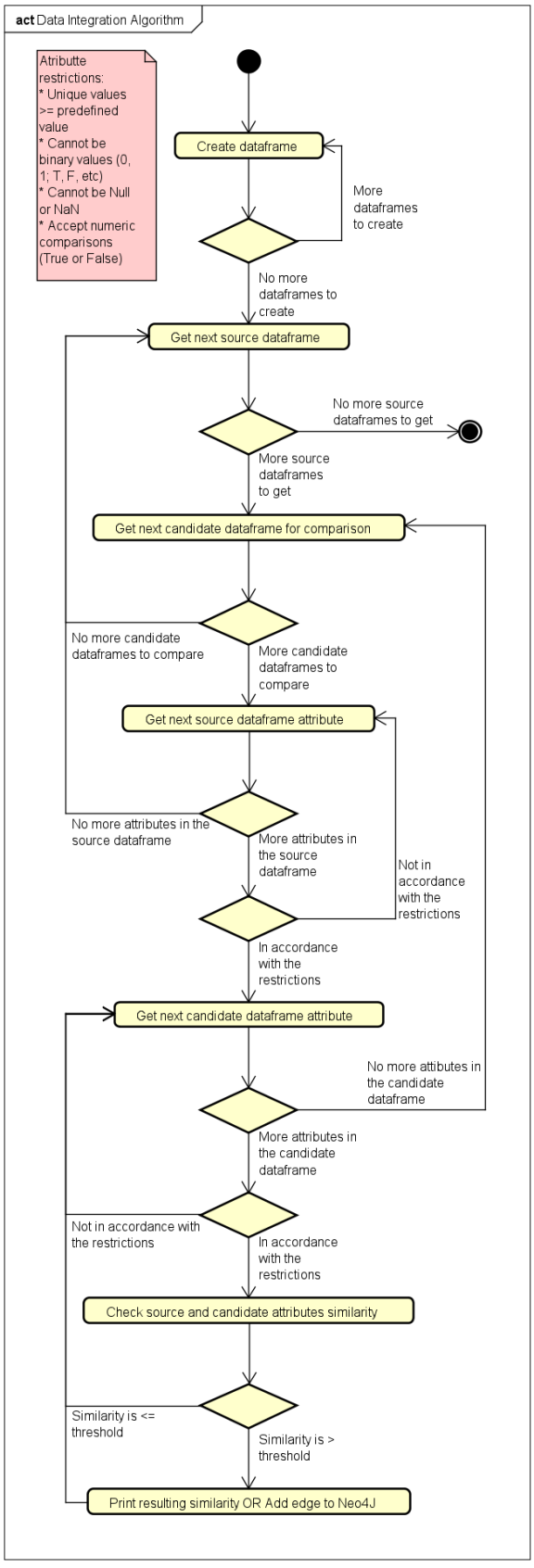


Fig. 3. Algorithm for data integration in UML Activity Notation

Algorithm 1 Pseudo-code for the data integration algorithm

Require: $_1$*datasets*, $_2$`minimum number of unique values`, $_3$`accept numeric comparisons` (*True or False*)
Ensure: `dataframeLeftName + columnName, dataframeRightName + columnName: Overlap:`*value* / `Jaccard:`*value* / `Sørensen-Dice:`*value* / `Tversky:`*value*

```
for dataframes do
    while columns in the dataframe = True do
        if compare numeric values = False then
            if value is numeric = True then
                next column in while
            end if
        end if
        if unique values ≤ predetermined value OR target column has already been compared = True then
            next column in while
        else
            if column values ≠ binary values then
                for dataframes + 1 do    ▷ repeats the same logic as the previous for for the next dataframe
                    ... for
                    if minimum number of unique values between compared columns ≠ 0 then
                        calculate Overlap, Jaccard, Sørensen-Dice, and Tversky
                        if Overlap, Jaccard, Sørensen-Dice, and Tversky > 0.4 then
                            return Output = Ensure
                        end if
                    end if
                end for
            end if
        end if
    end while
end for
```

to check the similarity. It happens until we do not have more source dataframes to be compared to the candidates.

Our algorithm also handles so that there are no redundant comparisons among dataframes and attributes. Firstly, we assure that a dataframe is not compared to itself by identifying its previously defined name in the algorithm. Secondly, when we compared each attribute of the first dataframe, we stored its description in a variable. Before comparing the dataframe's attribute with another, we check that there are no attributes with the same description. Therefore, we exclude the possibility of redundant comparisons between dataframes and attributes.

The algorithm returns a list having the names of the dataframes, attributes, and resulting values for the Szymkiewicz-Simpson overlap coefficient – Equation 2, which is the main result, compared to other similarity metrics (Jaccard – Equation 3, Sørensen-Dice – Equation 4, and Tversky – Equation 5). The resulting values for the similarity metrics range from 0 (attributes are not at all similar) to 1 (attributes contain the same data). Next, we present the equations related to the metrics, where X represents the attribute of the source dataframe and Y represents the attribute of the dataframe candidate to be compared.

The Overlap Equation calculates the size of the intersection divided by the smaller of the size of the two attributes or sets:

$$overlap(X, Y) = \frac{|X \bigcap Y|}{min(|X|, |Y|)} \tag{2}$$

The Jaccard measures the size of the intersection between two sets divided by the size of the union:

$$jaccard(X,Y) = \frac{|X \bigcap Y|}{|X \cup Y|} \tag{3}$$

The Sørensen-Dice similarity score returns twice the intersection divided by the sum of the cardinalities.

$$dice(X,Y) = \frac{2 \times |X \bigcap Y|}{|X| + |Y|} \tag{4}$$

The Tversky index is a generalization of the Sørensen-Dice's and the Tanimoto coefficient (aka Jaccard index) coefficient, but introduces the use of the parameters α and β, where $\alpha = \beta = 1$ produces the Tanimoto coefficient and $\alpha = \beta = 0.5$ produces the Sørensen–Dice coefficient:

$$tversky(X,Y) = \frac{|X \bigcap Y|}{|X \bigcap Y| + \alpha|X - Y| + \beta|Y - X|}; \alpha, \beta >= 0 \tag{5}$$

Our model also presents the option to insert nodes and edges in a Neo4J database to better visualize the relationships among the dataframes.

5 Results

After analyzing the first results presented by the algorithm, we identified that some suggested integration points are numeric values that, in our dataframes, do not represent actual data integration points. For instance:

- UNIPROT['Lenght'] it is the length of the canonical sequence and it varies from 3 to 4 numeric chars;
- OMIM['Entrez_Gene_ID'] the National Center for Biotechnology Information (NCBI) gene ID, values from 1 to 115029024;
- DRUGCENTRAL['STRUCT_ID'] the structure ID, and has values from 1 to 5390;
- DISGENET['YearInitial'] and DISGENET['YearFinal'] are years from 1924 to 2020;
- DISGENET['NofPmids'] the PubMed id, and has values from 1 to 67;
- DISGENET['NofSnps'] the Single nucleotide polymorphisms (SNP) id, has values from 1 to 284.

Because of that, we decided to add a parameter in the algorithm do set if we want to make numeric comparisons. We set the parameter to false since, in our case, it does not represent actual data integration points, but to be able to generalize for different domains and different types of datasets, that kind of comparison must be useful.

Regarding the similarity metrics, as Tversky, Sørensen-Dice, and Jaccard present correlated values for our data (Tverski's index with α and $\beta = 0.5$ was equal Sørensen-Dice and twice the Jaccard coefficient), we show only the Jaccard and the overlap values in Table 2.

After carefully comparing the Jaccard and Overlap results with the manual mapping and reviewing the actual dataframes, we identified that the Overlap provides better insights about the relationships that could be created among the dataframes.

For instance, for the relationship DISGENET["diseaseType"] and MONDO ["lbl"], the Jaccard index is equal to zero, while the Overlap is 0,667. We checked the data, and we really found a point for integration in that case. Another example is DRUGBANK["drugbank_id"] and DRUGBANK_PROTEIN["drugbank_id"], which represent the

Table 2. Final data integration mapping

DF1	Column DF1	DF2	Column DF2	Overlap	Jaccard
UNIPROT	Entry	← REACTOME	UNIPROT identifier	0,409	0,058
UNIPROT	Gene names	← OMIM	Approved_Symbol	0,466	0,015
UNIPROT	Gene names	← DISGENET	geneSymbol	0,483	0,009
UNIPROT	Gene names	← IID	symbol2	0,484	0,017
UNIPROT	Gene names	← DRUGCENTRAL	GENE	0,486	0,002
UNIPROT	Gene names	← IID	symbol1	0,488	0,017
DRUGCENTRAL	ACCESSION	→ DRUGBANK_PROTEIN	uniprot_id	0,488	0,018
UNIPROT	Organism	← DRUGBANK_PROTEIN	organism	0,499	0,206
DRUGCENTRAL	ACCESSION	→ IID	uniprot1	0,509	0,072
DRUGCENTRAL	ACCESSION	→ IID	uniprot2	0,510	0,071
DISGENET	geneSymbol	← DRUGCENTRAL	GENE	0,528	0,110
REACTOME	UNIPROT identifier	← DRUGBANK_PROTEIN	uniprot_id	0,546	0,030
REACTOME	UNIPROT identifier	← IID	uniprot2	0,565	0,107
REACTOME	UNIPROT identifier	← IID	uniprot1	0,567	0,105
DRUGBANK_PROTEIN	uniprot_id	→ IID	uniprot1	0,583	0,147
DRUGBANK_PROTEIN	uniprot_id	→ IID	uniprot2	0,590	0,147
OMIM	Approved_Symbol	← DRUGCENTRAL	GENE	0,609	0,082
DRUGCENTRAL	GENE	→ IID	symbol1	0,615	0,076
DRUGCENTRAL	GENE	→ IID	symbol2	0,617	0,075
REACTOME	UNIPROT identifier	← DRUGCENTRAL	ACCESSION	0,624	0,019
DISGENET	diseaseType	→ MONDO	lbl	0,667	0,000
REACTOME	Species	→ DRUGCENTRAL	ORGANISM	0,750	0,051
UNIPROT	Entry	← DRUGCENTRAL	ACCESSION	0,829	0,004
OMIM	Approved_Symbol	→ IID	symbol1	0,866	0,704
DISGENET	geneSymbol	→ IID	symbol1	0,880	0,464
OMIM	Approved_Symbol	→ IID	symbol2	0,885	0,717
DISGENET	geneSymbol	→ IID	symbol2	0,898	0,469
UNIPROT	Entry	← DRUGBANK_PROTEIN	uniprot_id	0,909	0,008
DISGENET	geneSymbol	→ OMIM	Approved_Symbol	0,915	0,536
UNIPROT	Entry	← IID	uniprot1	0,953	0,030
UNIPROT	Entry	← IID	uniprot2	0,960	0,031
DRUGBANK	drugbank_id	← DRUGBANK_PROTEIN	drugbank_id	1,000	0,579

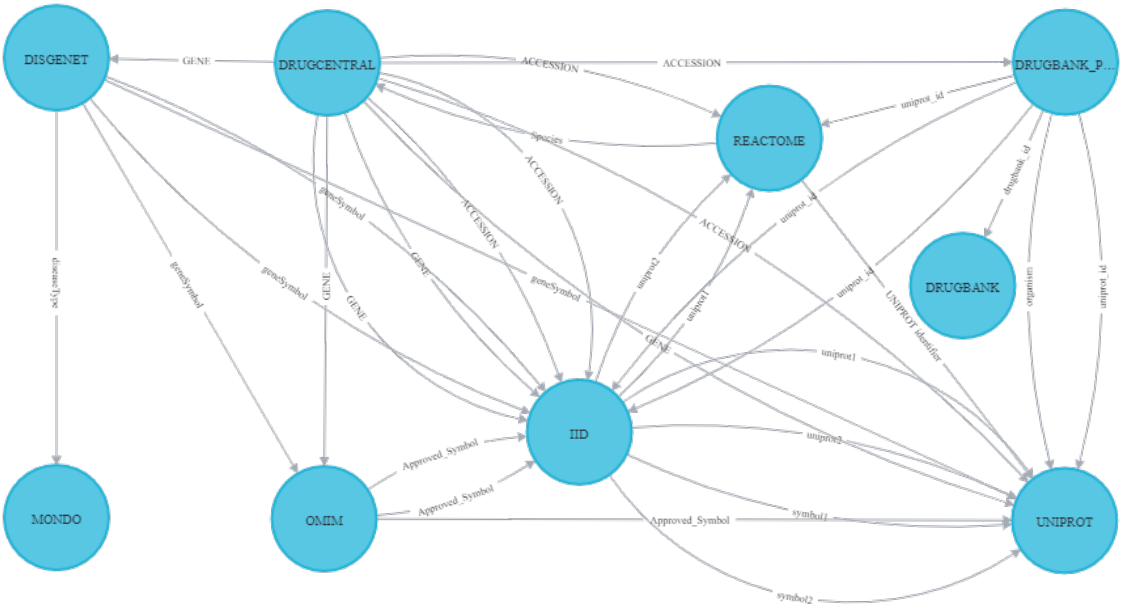


Fig. 4. Final data integration

same data according to the Overlap coefficient and to our manual analysis, and in this case, the Jaccard index is 0,579.

Hence, the Overlap coefficient seems to represent better how similar two attributes are and the level of data integration we could achieve if we integrate two dataframes using the top-ranked pairs of attributes indicated by the algorithm. Thus, we decided that in our case, it is better to use the Overlap Coefficient.

To better visualize the relationships between the dataframes, we create a database on Neo4J, where the nodes are the dataframes, and the edges are the name of the attributes. Figure 4 presents the final data integration resulting from our algorithm for the bioinformatics dataframes. In this graph visualization, similar to the manually mapped data integration visualization, the names of the dataframes are the vertices. The names of the attribute responsible for the integration are presented in the edges that connect the vertices. Figure 4 presents 9 nodes and 32 edges, some with only one edge between them (such as MONDO and DISGENET) and others having a high concentration of edges, such as IID, UNIPROT, and DRUGCENTRAL. The higher concentration of edges pointing to a dataframe means that the dataframe is referenced by a high number of other dataframes, meaning they represent an important point of integration.

When we manually mapped the points for integration, we identified 10 points (see Figure 2), while our model identifies 32 points, presented in Table 2. For instance, we manually mapped an integration between DRUGCENTRAL["SWISSPROT"] and UNIPROT ["ENTRY"], but our model shows that the coefficient for that integration is less than 0,4, while suggesting two better points: DRUGCENTRAL["GENE"] → UNIPROT[" Gene names "], with a overlap of 0,487, and DRUGCENTRAL["ACCESSION"] → UNIPROT["ENTRY"], with an overlap of 0,829.

Additionally, our model discovered 22 more paths of integration that were not manually identified, which we list above:

– From IID and: DISGENET, DRUGCENTRAL, OMIM, REACTOME, and DRUGBANK_PROTEIN

- From UNIPROT and: OMIM and DISGENET
- From REACTOME and: DRUGBANK_PROTEIN and DRUGCENTRAL
- From DRUGBANK_PROTEIN and DRUGCENTRAL

The scalability of the proposed solution takes place in terms of enabling comparisons between all attributes of all datasets. For example, the 19 attributes of DRUGCENTRAL are compared with the 253 attributes of the IID and so on, creating a bigger and bigger search space as we add more datasets for comparison.

Regarding the evaluation, we performed an analysis to answer the following question: *1) What is the average execution-time speedup provided by our model, including the data manipulation and algorithm?* We ran the model 10 times to get the average running time. It takes on average 2 hours and 30 minutes to run in the hardware we described in Section 4.1. Note that we run it in memory, in hardware with a humble configuration.

6 Discussion

We faced some challenges during the development and execution of our model. Initially, we had to elaborate on different ways of treating the datasets, as they had different data types. After this process, the researchers met to define the best way to carry out the comparison process. Effectively, the algorithm creation process started when we defined the four ways to calculate distances (Overlap, Jaccard, Sorensen, and Tversky's). Then, the initial algorithm implemented worked for most columns of the datasets. However, the algorithm generated errors, specifically for columns with information of the "JSON" or "XML" type, being corrected and treated soon afterward. After running the algorithm, we noticed that some of the comparisons generated 100% matches in many cases. Therefore, we verified that there were columns with information of binary values, which meant "False" or "True", but that was not necessarily relevant and similar to each other. We address this issue by removing columns with these data types from our comparison. We also skip Null and empty values, in the comparison steps. Furthermore, when we ran with all the datasets simultaneously, the initial version of the algorithm worked but took longer than we expected. Therefore, we performed refactoring in the algorithm, so we executed in the settings described in Section 4.1, we could obtain better results in a considerably shorter time.

The challenges we faced during algorithm development are all data-related. When we start data analysis with data pre-processing, the data must go through a cleaning phase, which could have ruled out some of the related challenges. However, one of the goals of the algorithm is to receive data from different formats with different types of attributes and be able to perform the necessary initial comparisons. In this way, we allow the algorithm to be executed even by people without specific knowledge in data processing, so they can and still obtain good results for their data integration.

Let us now discuss the utility of our model, by considering the data integration example in the field of bioinformatics. A data scientist has access to a data lake with the same bioinformatics datasets we worked on: OMIM, DISGENET, REACTOME, MONDO, DRUGBANK, IID, DRUGCENTRAL, and UNIPROT. The data scientist received the task to study neglected diseases, such as tuberculosis. To do so, it is necessary to explore the data related to the gene *inhA*, which is related to the organism *Mycobacterium tuberculosis*. Having those two pieces of information, it is easy to find the related data on UNIPROT. Actually, it may be on the top-5 results of a quick search on Google. But how will the person know if and how the data found in UNIPROT can be integrated with the other data sources, so they can find additional information? Well, usually the person

would have to put an effort into understanding the schema of all the datasets, analyze the data dictionary, a sample of data, and so on.

Using our data integration model, we will be able to see that UNIPROT is easily integrated with OMIM, DISGENET, IID, and DRUGCENTRAL by the *gene name*. By integrating with OMIM, we would have more details about genetic phenotypes related to the gene *inhA*; while DISGENET would bring the variants of the genes related to the tuberculosis disease. IID adds information about how a protein related to *inhA* (*Enoyl-[acyl-carrier-protein] reductase [NADH]*) interacts with other proteins. UNIPROT can also be integrated with REACTOME since REACTOME contains a field named *UNIPROT identifier*. Thus, we would have additional information about how the molecules interact in a cell to change the cell or create a certain product; for instance, turn genes on or off.

Additionally, integrating with DRUGCENTRAL would add information about interactions related to the drugs and tuberculosis. The integration with DRUGCENTRAL will allow integration with DRUGBANK, which brings supplementary information about the substance of the drugs and related products. For instance, we will find that *Pretomanid* is a medication for the treatment of tuberculosis. Finally, having the disease type from DISGENET, we could connect with the MONDO ontology, and learn about the different types of the disease, such as endocrine, esophageal, ocular, spinal tuberculosis, and others.

7 Conclusions

In this paper, we presented a model for automatized data integration in a Hadoop data lake, and we present experiments with eight well-known datasets from the bioinformatics domain, having different sizes and formats. We tested the similarity among the dataframes with different similarity measures, and we identified that The Overlap coefficient and Jaccard would be enough for us to validate our proposal.

Because the Overlap coefficient presented better results than the actual data and a specialists analysis, our experiments suggest that the Overlap coefficient is the best option for the in-memory set similarity approach we developed. Based on the Overlap coefficient, we found the top-k overlap set similarity that can help define data integration points for datasets in a data lake. For future work, we plan to implement text similarity strategies to magnify the reach of our results and increase the points for data integration based on semantic and syntactic.

Availability of data and materials

The data that support the findings of this study are available from:

- UNIPROT (UniProtKB - Reviewed (Swiss-Prot)). API available at [38].
- OMIM (genemap2). Available at [20], upon register and request. Release: File generated on 02/07/2020.
- DISGENET: Available at [39]. Release: version 7.0, January 2020.
- DRUGCENTRAL: Available at [40]. Release: 18/09/2020.
- IID: Available at [41]. Release: 2018-11.
- MONDO: Available at [42]. Release: v2021-01-15.
- REACTOME: Available at [43]. Release: Version 75, 07/12/2020.
- DRUGBANK: Available at [44], upon registration. Release: 5.1.8, 2021-01-03.

Restrictions apply to the availability of these data, which were used under license for the current study, and so are not all publicly available. Data are, however, available from the authors upon reasonable request and with permission of the owners, when necessary.

Acknowledgments

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES)* – Finance Code 001. We also thank Professor Anil Wipat and his team at Newcastle University for the support and advice in the early stages of the project and for providing us with the hardware for first creating the data lake.

References

1. D. B. Searls, "Data integration: challenges for drug discovery," *Nature reviews Drug discovery*, vol. 4, no. 1, pp. 45–58, 2005.
2. X. Lin, X. Li, and X. Lin, "A review on applications of computational methods in drug screening and design," *Molecules*, vol. 25, no. 6, p. 17, 2020.
3. A. Alserafi, A. Abelló, O. Romero, and T. Calders, "Towards information profiling: Data lake content metadata management," in *International Conference on Data Mining Workshops*, (Barcelona, ES), pp. 178–185, IEEE, 2016.
4. H. Dabbèchi, N. Z. Haddar, H. Elghazel, and K. Haddar, "Social media data integration: From data lake to nosql data warehouse," in *International Conference on Intelligent Systems Design and Applications*, (Online), pp. 701–710, 2020.
5. R. Hai, C. Quix, and C. Zhou, "Query rewriting for heterogeneous data lakes," in *European Conference on Advances in Databases and Information Systems*, (Budapest, HU), pp. 35–49, Springer, 2018.
6. Y. Dong, K. Takeoka, C. Xiao, and M. Oyamada, "Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach," in *International Conference on Data Engineering*, (Chania, GR), pp. 456–467, IEEE, 2021.
7. P. Sawadogo and J. Darmont, "On data lake architectures and metadata management," *Journal of Intelligent Information Systems*, vol. 56, no. 1, pp. 97–120, 2021.
8. A. Y. Levy, *Logic-Based Techniques in Data Integration*, pp. 575–595. Boston, MA: Springer US, 2000.
9. M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, (New York, US), p. 233–246, Association for Computing Machinery, 2002.
10. E. Zhu, D. Deng, F. Nargesian, and R. J. Miller, "Josie: overlap set similarity search for finding joinable tables in data lakes," in *International Conference on Management of Data*, (Amsterdam, NL), pp. 847–864, ACM, 2019.
11. S. Ontañón, "An overview of distance and similarity functions for structured data," *Artificial Intelligence Review*, vol. 53, no. 7, pp. 5309–5351, 2020.
12. A. Tversky, "Features of similarity.," *Psychological review*, vol. 84, no. 4, p. 327, 1977.
13. T. J. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*. Copenhagen: I kommission hos Ejnar Munksgaard, 1948.
14. P. Jaccard, "Distribution comparée de la flore alpine dans quelques régions des alpes occidentales et orientales," *Bulletin de la Murithienne*, vol. XXXVII, pp. 81–92, 1902.
15. M. Vijaymeena and K. Kavitha, "A survey on similarity measures in text mining," *Machine Learning and Applications: An International Journal*, vol. 3, no. 2, pp. 19–28, 2016.
16. J. Couto, O. T. Borges, D. Ruiz, S. Marczak, and R. Prikladnicki, "A mapping study about data lakes: An improved definition and possible architectures," in *International Conference on Software Engineering and Knowledge Engineering*, (Lisbon, PT), pp. 453–458, KSI Research Inc., 2019.
17. R. R. Forster, *Hive on Spark and MapReduce: A methodology for parameter tuning*. Master thesis, NOVA Information Management School, Lisbon, PT, 2018.
18. Apache Software Foundation, "The Apache Software Foundation." `https://apache.org`, 2021. Accessed 25 Nov 2021.
19. A. Lesk, *Introduction to bioinformatics*. Oxford: Oxford university press, 2019.
20. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD), "Online mendelian inheritance in man, omim®." `https://www.omim.org`, 2021. Accessed: 25 Nov 2021.
21. J. Piñero, J. M. Ramírez-Anguita, J. Saüch-Pitarch, F. Ronzano, E. Centeno, F. Sanz, and L. I. Furlong, "The disgenet knowledge platform for disease genomics: 2019 update," *Nucleic acids research*, vol. 48, pp. D845–D855, 2020.
22. B. Jassal, L. Matthews, G. Viteri, C. Gong, P. Lorente, A. Fabregat, K. Sidiropoulos, J. Cook, M. Gillespie, R. Haw, *et al.*, "The reactome pathway knowledgebase," *Nucleic acids research*, vol. 48, pp. D498–D503, 2020.

23. C. J. Mungall, J. A. McMurry, S. Köhler, J. P. Balhoff, C. Borromeo, M. Brush, S. Carbon, T. Conlin, N. Dunn, M. Engelstad, *et al.*, "The monarch initiative: an integrative data and analytic platform connecting phenotypes to genotypes across species," *Nucleic acids research*, vol. 45, pp. D712–D722, 2017.
24. D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey, "Drugbank: a comprehensive resource for in silico drug discovery and exploration," *Nucleic acids research*, vol. 34, pp. D668–D672, 2006.
25. M. Kotlyar, C. Pastrello, Z. Malik, and I. Jurisica, "Iid 2018 update: context-specific physical protein–protein interactions in human, model organisms and domesticated species," *Nucleic acids research*, vol. 47, pp. D581–D589, 2019.
26. S. Avram, C. G. Bologa, J. Holmes, G. Bocci, T. B. Wilson, D.-T. Nguyen, R. Curpan, L. Halip, A. Bora, J. J. Yang, *et al.*, "Drugcentral 2021 supports drug discovery and repositioning," *Nucleic Acids Research*, vol. 49, pp. D1160–D1169, 2021.
27. U. Consortium, "Uniprot: a worldwide hub of protein knowledge," *Nucleic acids research*, vol. 47, pp. D506–D515, 2019.
28. S. J. Cockell, J. Weile, P. Lord, C. Wipat, D. Andriychenko, M. Pocock, D. Wilkinson, M. Young, and A. Wipat, "An integrated dataset for in silico drug discovery," *Journal of integrative bioinformatics*, vol. 7, pp. 15–27, 2010.
29. T. Sellis, D. Skoutas, and K. Staikos, "Database interoperability through web services and ontologies," in *International Conference on BioInformatics and BioEngineering*, (Athens, GR), pp. 1–5, IEEE, 2008.
30. J. Hendler, "Data integration for heterogenous datasets," *Big data*, vol. 2, pp. 205–215, 2014.
31. A. Petermann, M. Junghanns, R. Müller, and E. Rahm, "Graph-based data integration and business intelligence with biiig," *Proceedings of the VLDB Endowment*, vol. 7, pp. 1577–1580, 2014.
32. R. L. Bradshaw, R. Gouripeddi, and J. C. Facelli, "Concept bag: A new method for computing concept similarity in biomedical data," in *International Work-Conference on Bioinformatics and Biomedical Engineering*, (Granada, ES), pp. 15–23, Springer, 2019.
33. Y. Zhang and Z. G. Ives, "Finding related tables in data lakes for interactive data science," in *International Conference on Management of Data*, (Portland, US), pp. 1951–1966, ACM, 2020.
34. H. Khalid and E. Zimányi, "Using rule and goal based agents to create metadata profiles," *Communications in Computer and Information Science*, vol. 1064, pp. 365–377, Sep, 2019.
35. D. Abadi, A. Ailamaki, D. Andersen, P. Bailis, M. Balazinska, P. Bernstein, ..., and D. Suciu, "The seattle report on database research," *SIGMOD Record*, vol. 48, p. 44–53, Dec, 2019.
36. H. Alrehamy and C. Walker, "SemLinker: automating big data integration for casual users," *Journal of Big Data*, vol. 5, pp. 1–14, Mar, 2018.
37. D. S. Himmelstein, "User-friendly extensions of the DrugBank database v1.0." `https://doi.org/10.5281/zenodo.45579`, Feb. 2016. Accessed 25 Nov 2021.
38. UniProt Consortium, "UniProt KB Reviewed (Swiss-Prot) dataset." `https://www.uniprot.org`, 2021. Accessed: 25 Nov 2021.
39. Integrative Biomedical Informatics Group, "DisGeNET curated gene-disease associations dataset." `https://www.disgenet.org/static/disgenet_ap1/files/downloads/curated_gene_disease_associations.tsv.gz`, 2021. Accessed: 25 Nov 2021.
40. S. Avram, C. G. Bologa, J. Holmes, G. Bocci, T. B. Wilson, D.-T. Nguyen, R. Curpan, L. Halip, A. Bora, J. J. Yang, *et al.*, "DrugCentral dataset." `https://drugcentral.org/download`, 2021. Accessed: 25 Nov 2021.
41. M. Kotlyar, C. Pastrello, Z. Malik, and I. Jurisica, "IID dataset." `http://iid.ophid.utoronto.ca/static/download/human_annotated_PPIs.txt.gz`, 2018. Accessed: 25 Nov 2021.
42. OBO Foundry, "Mondo dataset - json edition." `http://purl.obolibrary.org/obo/mondo/mondo-with-equivalents.json`, 2021. Accessed: 25 Nov 2021.
43. Reactome, "Reactome UniProt to pathways dataset." `https://reactome.org/download/current/UniProt2Reactome_All_Levels.txt`, 2021. Accessed: 25 Nov 2021.
44. OMx Personal Health Analytics, Inc., "DrugBank dataset." `https://go.drugbank.com/releases/latest`, 2021. Accessed 25 Nov 2021.

Authors

J Couto holds a degree in Information Systems (2012), a Master's in Computer Science (2018), and an MBA in Project Management (2016). She worked as a Project Manager, in distributed software projects in the health sector. Currently pursuing a Ph.D. in Computer Science at PUCRS, focusing on automating the integration of big data based on data profiling.

O. T. Borges is a Ph.D. student at PUCRS. He holds a degree in Information Systems (2015) and a Master's in Computer Science (2018). He is a member of the MuNDDoS Research Group (Distributed Software Development Research Group). His current research focus is supporting software development using Artificial Intelligence and Machine Learning techniques to Software Startups.

D. D. Ruiz holds a BS in Electrical Engineering from UFRGS (1983), a master's degree (1987), and a Ph.D. (1995) in Computer Science from the same university (UFRGS) and post-doctorate in Computer Science at Georgia Institute of Technology (2002). He is is a professor in the School of Technology at PUCRS, Brazil, where he leads the GPIN research group, acting primarily in the core of Machine Intelligence and Robotics. He has been working mainly in the areas of business process automation, non-conventional databases, bioinformatics, and database knowledge discovery (KDD).

Data Visualization of Graph-Based Threat Detection System

Ilnaz Nikseresht, Issa Traore, and Amirali Baniasadi

Department of Electrical and Computer Engineering
University of Victoria, Victoria, Canada

Abstract. The Activity and Event Network Model (AEN) is a new security knowledge graph that leverages large dynamic uncertain graph theory to capture and analyze stealthy and long-term attack patterns. Because the graph is expected to become extremely large over time, it can be very challenging for security analysts to navigate it and identify meaningful information. We present different visualization layers deployed to improve the graph model's presentation. The main goal is to build an enhanced visualization system that can more simply and effectively overlay different visualization layers, namely edge/node type, node property, node age, node's probability of being compromised, and the threat horizon layer. Therefore, with the help of the developed layers, the network security analysts can identify suspicious network security events and activities as soon as possible.

Keywords: data visualization, security, intrusion detection system, intrusion prevention system.

1 Introduction

In the network security domain, analysts deal daily with network packets and security events and alerts generated through a various data sources such as firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), system logs, etc. With an exponential increase in the amount and diversity of security data, there is a growing demand for visualization tools to help with analyzing the data and gaining insight into it. Visualization tools help avoid having to spend excessive hours on raw data analysis and allow the security analysts to distinguish unusual patterns and trends in even the most intricate data sources. This will let analysts identify the existing and novel network attacks in the minimum required period. Knowing that the visualization requires data from single or multiple sources, either one or many of the below data sources can be input into the system. Network traces, security events (IDS, IPS, Firewalls, AV), network events (switch, router, or server data), or application logs are some instances of the data sources [1]. It is needless to say that, with an increase in the incorporation of data sources to be fed to the visualization system and their features, the network security analyst's ability to gain insight into irregular traffic trends and events also improves.
The remaining sections in this report are structured as follows. Section 2 gives

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 155-168, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120913

an overview of existing security visualization models and gives a summary of key features of the AEN system architecture. Section 3 outlines the proposed model to address previously stated limitations. Section 4 depicts the experimental evaluation of the proposed scheme. Section 5 makes concluding remarks.

2 Related Work

Shiravi et al. [1] reviewed network security visualization practices by classifying them into five classes, host/server monitoring, internal/external monitoring, port activity, attack patterns, and routing behavior. The first visualization class in this study is host/server monitoring which mainly exhibits the nodes in the network as hosts or servers and tries to identify possible correlation among them with the aim of detecting the malicious nodes. One of the earlier works that was done on small-sized networks in this class was authored by Erbacher et al. [2],[3] where they placed the monitored server in the middle of the visualization layer, and the rest of the hosts are around five concentric circles where each node's ring defines a distinct IP address from that of its monitored server. In [4], Takada and Koike developed a model called Tudumi. Tudumi is a 3D visualization model to observe and examine the user performance on a server, achieved by the utilization of multilayered concentric disks in a small-sized network.To highlight various access techniques such as file transfer or terminal service, this model uses different-sized dashed lines. A model developed by Lakkaraju et al, called NVisionIP [5]-[6], focuses on large networks presented in a grid of 256 by 256, where each of the cells displays the hosts' associations. The horizontal axis of the network represents the network subnets, and the vertical axis of the network represents the subnet hosts. There is a magnifier that enables the analyst to access the nodes of interest in the network. To detect the hidden malware in the network, Fink et al. [7] developed a model which illustrates the correlations between network traffic and host processes. VISUAL [8] is a security visualization system developed to permit the analyst to observe the interaction patterns between an internal network regarding external sources. The internal network is a grid, where each cell represents one of the internal hosts. On the other hand, external sources are depicted out of the internal grid, with the square dimension indicating the activity level. VizFlowConnect [9] employs parallel axes to display the current relations among network nodes in the network, which is achieved by designing three parallel axes. While most left is responsible for nodes originating network transfer to the internal network, the center is responsible for presenting the internal hosts, and the right one is responsible for depicting the destination nodes of internal traffic. The port activity visualization class has been designed to highlight the viruses, trojans, worms, and zero-day exploits activities. The developers of this visualization class believe that with the scaling techniques implemented due to the amount of traffic and extended range of possible port numbers and IP addresses, the malicious actors can be located. Abdullah et al. [10] designed a port-based

sketch of activities in the network based on the given services. It is discussed that by assigning popular ports to principal services, the chances of exposing them to attacks rise.Therefore, they are grouped in bins of 100's, whereas the registered ports are grouped in bins of 1000's, and the rest of the private/dynamic ports are assigned in a single bin. This system provides the ability for the user to view more precise aspects of the irregular activities in the network by depicting the data over time. Potential Doom, a system proposed by Lau [11], involves the rotating cube, which attempts to visualize the coarse trends in large-scale networks, namely port and IP data in a 3D cube, with each axis of three-dimensional representation as a component of a TCP connection. The X-axis represents the destination IP addresses, Y-axis represents the port numbers, and Z-axis represents the source IP addresses. The system is suitable for single attacks and can only be employed for the discovery of port scans. In [12], McPherson et al. present ProtVis which uses a 256x256 grid of various colors to reflect the network activities on grid cells. The port's position on the grid is decided by separating the port number into an (X, Y) position, where X highlights the port number's high byte and Y highlights the port number's low byte. Through time, each point's changes are depicted using a distinct color, where black indicates no change, blue indicates small change, red indicates larger change, and white indicates the most variations. In addition, the system provides a magnifier to support more specific data regarding specific ports. One of the challenges the system faces regarding identifying malicious nodes is when irregular activity is discovered between ports with high and legitimate activities. The attack patterns visualization class targets both detection and presentation of attacks in different steps. This type of visualization is of prime importance, because various attacks behave differently and many of them are implemented in multi-steps, including reconnaissance, scanning, gaining access, maintaining access, clearing tracks, and installing back doors for future access. Because the high number of alerts generated daily by the intrusion detection systems is at times overwhelming for the security analysts, they constitute one of the well-known sources of data for this class. Girardin [13] proposed an unsupervised machine learning system developed with detecting the network's irregular and intrusive actions in mind. The system depicts network state and deviations from natural behavior by using front ground and background colors, size, and relevant positioning on a map. In this way, similar events are grouped, and the map is also arranged in the same way. Nyarko et al. [14] developed NIVA, a newtork IDS visual analyzer. It utilizes data from multiple intrusion detectors and uses links and colors to manifest attacks. The system includes a GUI window and a 3D rendering window which consists of glyphs connected by links. The position of each glyph depends on its IP address, meaning that closer glyphs favor closer IP addresses. Colitti et al. developed BGPLay [15], a system which permits ISPs to observe the reachability of a specific prefix from the viewpoint of a given edge router while including animation to specify routing

adjustments. Wong et al. [16] developed the TAMP system to visualize the BGP irregularities by employing statistical techniques to perform BGP data aggregation. They proposed animation techniques to depict the routing behavior variations over time to assist the network administrators in identifying any abnormal patterns. The principal objective of the AEN system is to assist network analysts in detecting any security irregular patterns or malicious nodes. The system functions by utilizing the raw data from various network nodes and external repositories to discover the current node-to-node relationships and extract unusual ones. Multiple data sources such as network packets, system logs, and intrusion detection alerts are processed to generate the graph.

3 On the AEN Graph Model

3.1 Overview of the AEN Graph Model

The AEN Graph is a new security knowledge graph model developed at the Information Security and Object Technology (ISOT) Lab to capture and analyze stealthy and long term attack trends and patterns. To capture the inherent uncertainty and dynamic nature of network environments and attack occurrences, AEN consists of a dynamic uncertain directed multigraph model. The graph engine is developed using emerging graph database technologies because one of the requirements of the model is the need to load the entire graph in memory for malicious patterns detection and analysis. The graph model is updated continuously and maintained over time, with most of the historical information being preserved by necessity and leveraged in long-term attack detection. As a result, over time the graph is expected to become extremely large in size. The main concern in the graph construction is the method employed to mine different data features responsible for nodes identification, their relationships, and attributes. The mining method can be implemented by 1) direct connection to the data source or 2) from the available data gathered from various tools and services, or 3) by data mining into aggregates, or 4) from the previously known attacks by security experts, or 5) from log analysis performed on the known network applications and service calls to or from the mentioned applications and data sources such as hypervisor logs, syslog, and IDS alerts. Another input category is attack fingerprints, which are not automatically generated by the system and developed by security experts from past attacks. Even without this type of data, the system can leverage attack signatures indirectly through its IDS alerts. However, these fingerprints provide an additional layer of information to the model in the form of a database of well-known attacks that can help the identification by better incorporation into the model.

3.2 Graph Definition

To understand how the graph is constructed, we must analyze the features mentioned above to determine distinguishable characteristics. To begin with, it is worth considering how each feature can be used to model the network. As a sample, IP address-domain names relationship can be considered. Graph nodes represent the features, and the edges represent their relationships. However, features such as protocol and port are better utilized as relationships' descriptors and therefore are employed to characterize attributes of either nodes or edges. Generally, the AEN model's nodes are interpreted by any feature for which valuable relationships can be formed, and the edges are constructed based on those relationships and their direction. The rest of the features can be utilized to describe attributes of either nodes or edges. It is important to note that the model is confident in identifying or in the correctness of the extracted information from the features. For instance, when related TCP handshake packets between two nodes are received, the model is confident that the hosts are communicating and in which direction. However, it is worth noting that the IDS alerts and graph operations such as pruning, chaining, decaying, and clustering are inherently imperfect. That adds a layer of uncertainty to the nodes, their relationships, and their respective attributes. Ultimately, as there are continuous changes in the system and a constant demand to track the network entities and their relationships, the model requires to maintain the information on the periods in which each element has existed to identify the vital chronological relationships. In summary, below are the characteristics of the graph model. 1) Processing times/latencies are negligible. 2) Through time, nodes, edges, and their attributes vary and have a lifetime. 3) Nodes have their attributes and are labeled. 4) Nodes can have various relationships at the same time. Consequently, nodes can have various edges connecting them. 5)Relationships have a source and a destination. Therefore edges are directed. 6) Relationships have their properties. Thus edges are labeled. 7) Both relationships and nodes can be uncertain. Thus nodes, edges, and their attributes are weighted by probabilities of existence.

4 Proposed Visualization Model

To address the visualization challenges discussed, many layers have been developed. The implementation is performed via Javascript programming language, and with the help of vis.js library. As previously mentioned, by feeding the raw data collected from the network logs to the system, the graph is constructed. The system before the implementation of any visualization layer is depicted with all nodes represented as grey ovals and all edges as grey arrows without any labels in the zoom-out view versus with the labels in the zoom-in view. To integrate different visualization layers to the model, a distinct tab named node view, is designed in order to provide the capability of viewing the network elements based on the type of nodes, type of edges,

property, node age, probability of compromise and threat horizon. In addition, a distinct layer is designed to show or hide the labels.

4.1 Element Type Layer

The initial visualization layer developed provides the ability to filter network elements based on their types. There exists a checkbox named 'Show/Hide Labels' that enables the end-user to view the network nodes with or without the labels. Besides, when the user selects not to include the label in the graph, all the nodes would be depicted as same-sized circles.This layer functionality consists of the different node/edge types and the node properties. The node type category includes HOST, DOMAIN, ALERT, IP, ORGANIZATION, and LOCATION that provide the relative information. Upon selecting each type, the nodes belonging to that specific category are colored while the rest of the nodes are grey. In addition to node types, different types of edges have been added to this layer, including AUTH_ATTEMPT, TRIGGER,USED, IP_LOCATED_AT, PART_OF, RESOLVED_TO, CONTROLS, LOCATED_AT, OWNS,SESSION, ALERT_TRG_BY_HOST. Upon selection of edges, the user would again observe them in color and the rest of the network elements in grey. Thanks to different layers operating together, selecting different types of nodes and edges is possible. It gives users a better understanding of network elements. Another function of this layer is its ability to demonstrate the malicious nodes in red color. Figure 1 is an example of a malicious node highlighted in red when selecting the malicious property checkbox. In conclusion, this layer provides the ability to show/hide all the labels and all or some of the network elements based on their node type or edge type, or node property in color.

Fig. 1. Malicious Property

4.2 Node Age Layer

One of the key characteristics of the AEN model is its dynamic nature. Nodes and edges are added continuously as more data arrive. Timestamps are associated with the graph elements to convey their age. The functionality provided by this layer mainly concerns the age of each node in the network. It was uncertain how far in time each specific node has been added to the graph or updated in the graph (for instance two days ago or three days ago). The visualization of this layer has been implemented with the aid of the opacity feature. Opacity can have a value between zero and one that means if the node is added or updated more recently, say two days ago, it is of higher opacity than the nodes added or updated four days ago. When the related node and age data come from the back-end of the system to the front-end, the required properties are extracted, the timestamp, in this case. The node's age is computed based on the timestamp.It is noteworthy that the age limit is ten days and the nodes which are added/updated before that time are shown with the same opacity. The opacity is calculated based on below formula.

$$\text{Opacity} = (1\text{-}(\text{age}/10))$$

Figure 2 represents an overview of the network elements when this layer is selected. As part of this layer's functionality, it is possible to choose it together with other layers such as the node's type layer to merge the opacity with the nodes/edges colors.

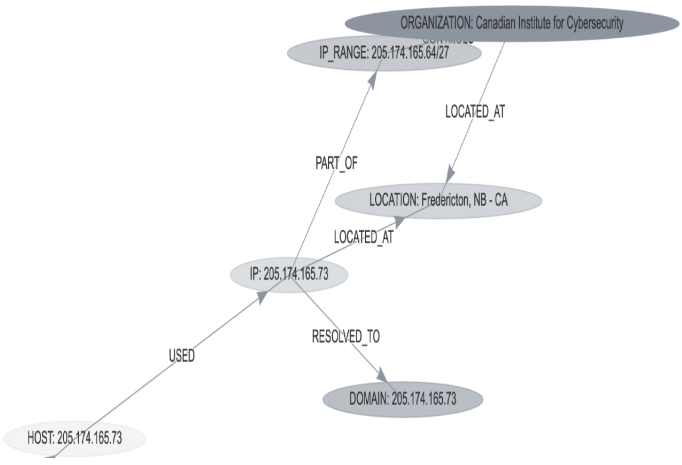


Fig. 2. Node Age visualization layer

4.3 Probability of Compromise Layer

This layer illustrates the nodes based on their probability of being compromised. The graph engine assigns the probability and shows that a specific host is malicious or is compromised, by considering all the detectors together. The probability of being compromised is a value between zero and one fetched from the backend. After the comparison is performed, the size of the nodes of the probability of compromise category is computed based on the below formula.

$$\text{size} = \text{Int}(100 + \text{node.prob\_c} * 30)$$

Where prob_c denotes the probability of compromise. Figure 3 depicts the graph nodes with a higher probability of being compromised as larger objects while the rest of the nodes as smaller objects.

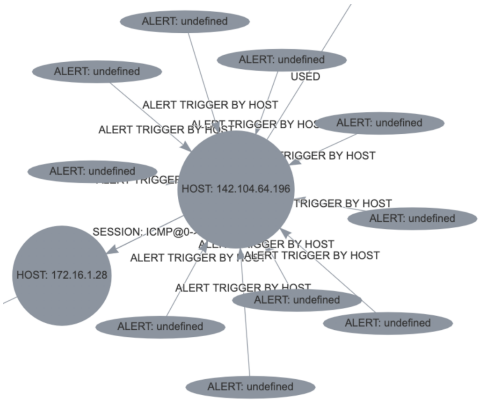


Fig. 3. Probability of being compromised

4.4 Threat Horizon Layer

Threat Horizon is defined as the set of all possible nodes with which a node ‘u‘ could have exchanged data. As a result, the data exchange ultimately affected them, directly or indirectly. In practice, if node ‘u‘ is malicious or compromised, the set of nodes which ‘u‘ could have been in contact with are also compromised, or at least been targeted. More formally, two underlying concepts must be defined [17]: (1) Journey, which is a ”time-respecting walk” through a path over time, that is, a walk through a path between 2 nodes ‘u‘ and ‘v‘ over time such that each edge is traversed in chronological order and such that each edge existed at the point in time it was traversed; (2) reachability of 2 nodes ‘u‘ and ‘v‘, which is defined

as the existence of a Journey between nodes ‘u‘ and ‘v‘.Then we can define the Threat Horizon of node ‘u‘, ‘TH_u‘ in short, as the set of all nodes in a graph ‘G‘ reachable from ‘u‘.Figure 4 depicts the threat horizon, and the node clicked to show the threat horizon is called the focal point. The hosts that become orange upon selecting the focal point are the nodes that had direct or indirect communication with it. If the focal point is malicious (which is not necessarily the case), then there is a chance that the focal point host could have compromised the hosts inside the threat horizon.

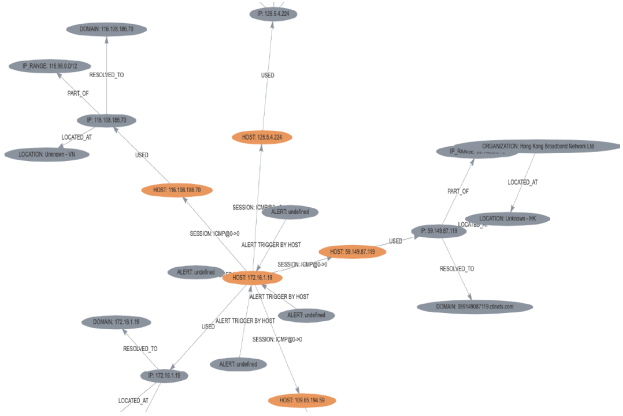


Fig. 4. Threat Horizon layer

5 Performance Evaluation

The response time and CPU utilization are used to evaluate the performance of AEN graph visualization layers. We have used different sizes of snort alert and the netflow files to determine the size of the graph. Snort is one of the IDS used among the data sources. Netflows are aggregate packets, also part of the data sources. The evaluation was run on a Dell 2.6 GHz Core i7 (6 cores) and 16GB of memory. The response time and the CPU utilization are studied to measure the scalability of the developed layers. Each file is input into the system five times, and the average output values are computed for the response time and CPU utilization. The response time is the time duration over which the graph is loaded to the system, and the time the graph appears on the canvas. The average response time is measured in terms of seconds. The CPU utilization indicates a computer’s processing resources or the volume of work handled by a CPU. The actual CPU utilization differs based on the amount and type of managed computing tasks. Specific tasks require heavy CPU time, while others require less because of non-CPU resource

requirements. CPU Utilization is measured in terms of the unit ms. Figures 5 and 6 depict the response time of different sized snort alert and netflow log files fed to the system before the activation of visualization layers. As observed in the graph, the response time increased as the file size enlarged. To handle larger capacity, more computation power would be required, like by using powerful server machines on the cloud. The same experiment was again conducted with the visualization layers to be able to compare the performance before and after the visualization layers activation. As shown in Figures 5 and 6, the response time has changed in terms of 2 milliseconds to 12 seconds depending on the file sizes compared to before, which can be interpreted as the hardware limitations.

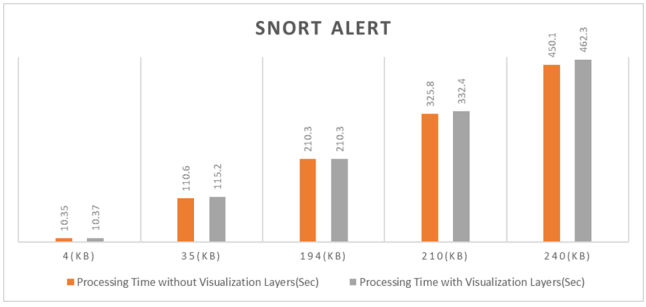


Fig. 5. Response time of the different sizes of snort alert file with vs without visualization layers

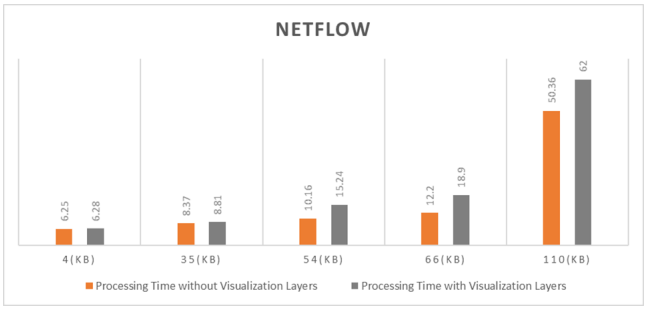


Fig. 6. Response time of the different sizes of Netflow file with vs without visualization layers

Furthermore, the same experiment has been run on networks of different sizes to measure the CPU utilization of the system. As expected, the system utilization increases with the size of the network. For the files less than 100 KB, the CPU utilization is acceptable, but as the size of files grows to 194 KB that contains hundreds of nodes, the system utilization also increases to the more than 100%. As

mentioned before, the system on which the experiment is conducted is a multi-core computer where the CPU performs significantly better than a single-core CPU of the same speed. Multiple cores not only allow PCs to run multiple processes simultaneously with greater ease and increasing the performance when multitasking but also provide the possibility to have more than 100% utilization depending on the number of cores available on the system. Again, as mentioned above, it is recommended to use computers with higher configuration capabilities to be able to load higher data rates and fully utilize the features provided by the AEN system model. In addition, Figures 7 and 8 depict the average CPU utilization after the activation of visualization layers and as expected the CPU utilization increased by a range of 8% to that of 20% comparing visualization version to no visualization version in different input file sizes.

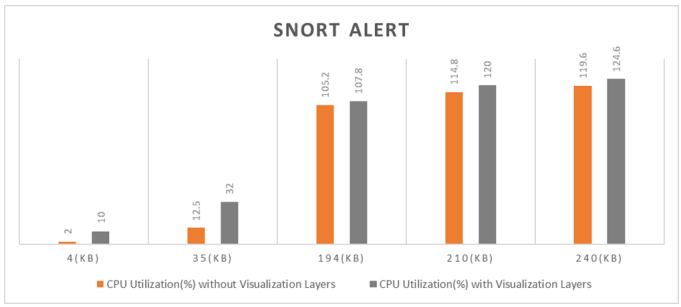


Fig. 7. CPU Utilization of the different sizes of snort alert file with vs without visualization layers

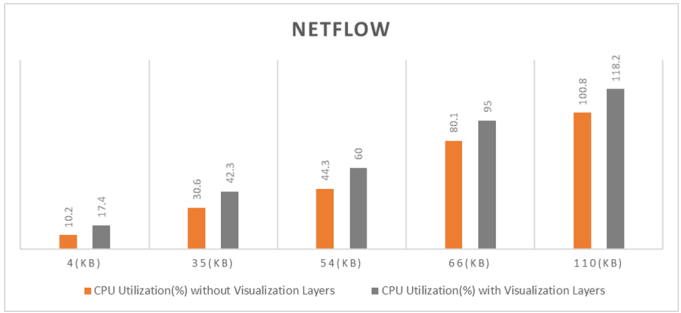


Fig. 8. CPU Utilization of the different sizes of Netflow file with vs without visualization layers

6 Conclusion

In this paper, many visualization layers were developed to enable the security analysts to have a high-level view of the ongoing network activities. Different visualization layers include the following. Show/Hide Labels checkbox allows the end-user to view the network nodes with or without the labels in addition to all the nodes in same-sized circles when the user selects not to include the label in the graph. In the element type layer, the functionality is defined by depicting the specific node/edge types and the node properties in distinct colors upon selecting each checkbox in this layer. The element type and show/hide labels can be chosen together. Node age layer depicts the age of each node (i.e., how far in time each specific node has been added/updated in the network) with different opacity levels. Node age layer can be selected along with the element type layer and the show/hide labels checkbox. Probability of Compromise Layer employs the graph engine to assign the likelihood and show if a specific host is malicious or compromised by enlarging the node. The threat horizon layer lets us see all the nodes that had direct or indirect communication with the focal point. The threat horizon layer functions with the probability of compromise and node age layers. Before implementing the visualization layers, the AEN graph model did not explicitly represent the nodes/edges that might be of interest to the security analysts. With the help of visualization layers, analysts can view the elements of interest in different colors/shapes than the rest of the nodes in the network. It enables the analysts to save time by not investigating raw data, especially as the AEN graph model is designed to be continually growing. In the performance evaluation section, the visualization layers' performance has been measured in terms of processing time. It has been observed that different layers are being loaded in a matter of ms. This time grows as the size of the files gets bigger, which is expected, and since the processing time difference for the same layers of different file sizes is 1 to 3 ms, it can be ignored.Although different visualization layers have been developed, there are still many features that can be added to the system to enhance the system's performance, such as attack progression visualization. The goal of this feature is to make it easier for an analyst to view how an attack progressed through time. In practice, that means that once an alert is generated, either via the IDS or via one of our detectors, the system should identify and highlight an attack path by using data from the alert, fingerprint, threat horizon, etc. and then, by using the graph timeline feature, correlate the attack path with the graph elements in previous and future points in time and display the attack progression from the first element to its current form. The progression here can be understood as a series of subgraphs/attack paths that show the attack as it progresses through time.

References

1. Hadi Shiravi, Ali Shiravi, and Ali A. Ghorbani, "A Survey of Visualization Systems for Network Security", IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 18,NO. 8, AUGUST 2012
2. R. Erbacher, K. Walker, and D. Frincke, "Intrusion and Misuse Detection in Large-Scale Systems," IEEE Computer Graphics and Applications, vol. 22, no. 1, pp. 38-48, Jan./Feb. 2002.
3. R. Erbacher, "Intrusion Behavior Detection through Visualiza- tion," Proc. IEEE Int'l Conf. Systems, Man and Cybernetics, pp. 2507-2513, 2003.
4. T. Takada and H. Koike, "Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs," Proc. Sixth Int'l Conf. Information Visualisation, pp. 570-576, 2002.
5. K. Lakkaraju, W. Yurcik, and A. Lee, "NVisionIP: Netflow Visualizations of System State for Security Situational Aware- ness," Proc. ACM Workshop Visualization and Data Mining for Computer Security, vol. 29, pp. 65-72, 2004.
6. K. Lakkaraju, R. Bearavolu, A. Slagell, W. Yurcik, and S. North, "Closing-the-Loop in Nvisionip: Integrating Discovery and Search in Security Visualizations," Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05), pp. 75-82, 2005.
7. G. Fink, P. Muessig, and C. North, "Visual Correlation of Host Processes and Network Traffic," Proc. IEEE Workshop Visualization for Computer Security (VizSEC 05), pp. 11-19, 2005.
8. R. Ball, G.A. Fink, and C. North, "Home-Centric Visualization of Network Traffic for Security Administration," Proc. ACM Work- shop Visualization and Data Mining for Computer Security, pp. 55-64, 2004.
9. X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "Visflowconnect: Netflow Visualizations of Link Relationships for Security Situational Awareness," Proc. ACM Workshop Visualization and Data Mining for Computer Security, pp. 26-34, 2004.
10. K. Abdullah, C. Lee, G. Conti, and J. Copeland, "Visualizing Network Data for Intrusion Detection," Proc. Sixth Ann. IEEE SMC Information Assurance Workshop (IAW '05), pp. 100-108, 2005.
11. S. Lau, "The Spinning Cube of Potential Doom," Comm. the ACM, vol. 47, no. 6, pp. 25-26, 2004.
12. J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "PortVis: A Tool for Port-Based Detection of Security Events," Proc. the ACM Workshop Visualization and Data Mining for Computer Security, pp. 73-81, 2004.
13. L. Girardin, "An Eye on Network Intruder-Administrator Shoot- outs," Proc. First Conf. Workshop Intrusion Detection and Network Monitoring, vol. 1, pp. 3-13, 1999.
14. K. Nyarko, T. Capers, C. Scott, and K. Ladeji-Osias, "Network Intrusion Visualization with niva, an Intrusion Detection Visual Analyzer with Haptic Integration," Proc. 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '02), pp. 277 -284, 2002.
15. L. Colitti, G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia, "Visualizing Interdomain Routing with BGPlay," J. Graph Algorithms and Applications, vol. 9, pp. 117-148, 2005.
16. T. Wong, V. Jacobson, and C. Alaettinoglu, "Internet Routing Anomaly Detection and Visualization," Proc. Int'l Conf. Dependable Systems and Networks (DSN '05), pp. 172-181, 2005.
17. Issa Traore, Paulo Gustavo Quinan, Waleed Yousef, "The Activity and Event Network (AEN) Model: Graph Elements and Construction", Technical report, ISOT lab, ECE Department, University of Victoria, January 2020.
18. S. F. Nadeem and C. -Y. Huang, "Data Visualization in Cybersecurity," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), 2018, pp. 48-52, doi: 10.1109/CSCI46756.2018.00017.

19. Fan, X., Li, C. Dong, X. A real-time network security visualization system based on incremental learning (ChinaVis 2018). J Vis 22, 215–229 (2019). https:/doiorg.ezproxy.library.uvic.ca/10.1007/s12650-018-0525-z
20. Ilnaz Nikseresht, "Data Visualization of Graph-Based Threat Detection System", A Report Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering In the department of Electrical and Computer Engineering, University of Victoria, 2021.

AN ADAPTIVELY SECURE NIPE SCHEME BASED ON DCR ASSUMPTION

Haiying Gao and Chao Ma

Information Engineering University, Zhengzhou, China

ABSTRACT

Non-zero inner product encryption provides fine-grained access control to private data, but the existing non-zero inner product encryption schemes are mainly constructed based on the problem of bilinear groups and lattices without homomorphism. To meet the needs of users to control private data and cloud servers to directly process ciphertexts in a cloud computing environment, this paper designs a non-zero inner product encryption scheme based on the DCR assumption. Specifically, the access control policy is embedded in the ciphertext by a vector ***y****, and the user attribute vector* ***x*** *is embedded in the secret key. If the inner product of the policy vector* ***y*** *of the encryptor and the attribute vector* ***x*** *of the decryptor is not zero, the decryptor can decrypt correctly. This scheme has additive homomorphism in the plaintext-ciphertext space, and it can be proved to be additive homomorphic and adaptively secure.*

KEYWORDS

Non-Zero Inner Product Encryption, Adaptive secure, Decision Composite Residuosity.

1. INTRODUCTION

With the rapid development of cloud computing and big data technology, the protection of cloud data has attracted more and more attention. ABE (Attribute-based Encryption) is a new type of Function Encryption (FE), which can simultaneously support sensitive data protection and access control [1, 2, 3]. For example, it can be used for fine-grained access control to cloud-encrypted data and support conditional information sharing in the cloud computing environment. In a inner product attribute encryption scheme, the policy vector ***y*** and the attribute vector ***x*** is embedded in the ciphertext or secret key. If the inner product of the decrypted user's attribute vector ***x*** and the policy vector ***y*** is equal to the preset value, the decryption algorithm can output plaintext. A scheme is called Zero Inner Product Encryption (ZIPE) scheme if the preset value is zero, otherwise, it will be called a Non-zero Inner Product Encryption (NIPE) scheme. This paper studies the design of a NIPE scheme.

The existing NIPE schemes are mainly constructed based on the difficult problems of bilinear groups and lattices, unfortunately, they do not have homomorphism. To meet the needs of users to control the private data in the cloud computing environment and the direct processing of ciphertext by the cloud server, this paper proposes a NIPE scheme based on the Decision Composite Residuosity (DCR) assumptio. Specifically, the policy vector ***y*** is embedded in the ciphertext in the form of a vector by modular multiplication and the user attribute vector ***x*** is embedded in the secret key by calculating the inner product of the attribute vector and the master secret key. The NIPE scheme can be used for ciphertext access control. As shown in Figure 1, the goal of the encryptor Alice is: Bob can decrypt correctly if his attribute w not belongs to the set $\Omega = \{w_1, w_2, ..., w_{n-1}\}$. The NIPE scheme can be used for this purpose and the specific description

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 169-183, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120914

is as follows.

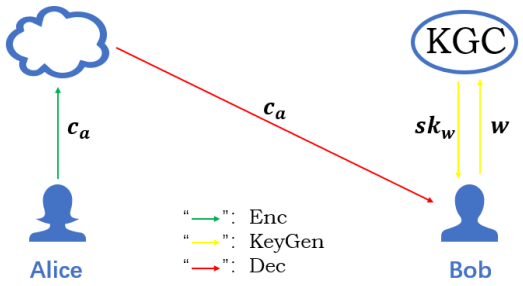


Figure 1. A NIPE scheme for ciphertext access control

Encryption. Alice constructs the polynomial $\phi(x)=(x-w_1)(x-w_2)...(x-w_{n-1})=a_0+a_1x+...+a_{n-1}x^{n-1}$ cording to the above set Ω, then embeds the policy vector $\boldsymbol{a}=(a_0,a_1,...,a_{n-1})$ to the ciphertext c_a.

Key Generation. Bob sends his attribute vector $\boldsymbol{w}=(1,w,...,w^{n-1})$ to Key Generation Center (KGC), then KGC generates a secret key sk_w and sends it to Bob securely.

Decryption. Bob downloads the ciphertext c from the cloud server to the local machine and he can decrypt it correctly if $\langle \boldsymbol{a},\boldsymbol{w}\rangle \neq 0$ (ie $\boldsymbol{w}\notin\Omega$).

Through the above scheme, Alice realizes the encryption and access control of the message at the same time.

Homomorphic Encryption. The homomorphic encryption scheme allows anyone to directly process the ciphertext without knowing the plaintext. And the effect is equivalent to operating on the plaintext first and then encrypting the result. Homomorphic encryption can be widely used in secret voting, bidding and so on [4]. According to the type of homomorphic mapping [5], homomorphic encryption schemes can be divided into additive homomorphism and multiplicative homomorphism. For example, RSA and ElGamal encryption belongs to multiplicative homomorphism, while Paillier encryption belongs to additive homomorphism [6, 7, 8].

This paper studies the NIPE scheme with additive homomorphism, which supports the direct operation of the cloud server on the ciphertext and realizes the access control of the encrypted user to the ciphertext at the same time. To illustrate the practical application of NIPE with additive homomorphism, Figure 2 shows an example of this type of scheme applied to confidential data query and the example is described as follows:

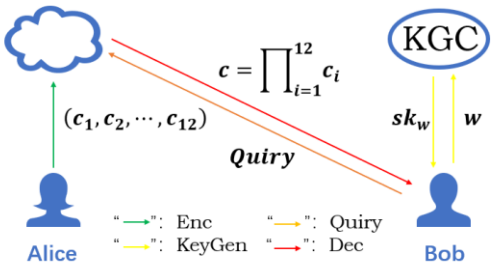


Figure 2. A NIPE scheme for querying average salary.

Encryption. Alice uses the NIPE scheme with additive homomorphism to encrypt the everyone's salary $m_1, m_2, ..., m_{12}$ in a department and obtains the ciphertext sequence $c_1, c_2, ..., c_{12}$, which is embedded with the same policy vector. Then upload the ciphertext sequence to the cloud server.

Key generation. Bob sends his attribute vector $\boldsymbol{w} = \left(1, w, ..., w^{n-1}\right)$ to KGC, then KGC generates a secret key and sends it to Bob securely.

Query. Bob wants to know the average salary in this department, so he sends a request to the cloud server to "Query the average salary".

Operation. The cloud server calculates $c = c_1 + c_2 + ... + c_{12}$ and sends c Bob.

Decryption. Bob can get $ave = \left(m_1 + m_2 + ... + m_{12}\right)/12$ if his attribute vector and the policy vector in ciphertext c meet the conditions $\langle \boldsymbol{a}, \boldsymbol{w} \rangle \neq 0$.

To design the NIPE scheme with additive homomorphism, we need to consider both the IPE schemes and the design method of the FE schemes with additive homomorphism. The following introduces and analyzes the current research status. Katz et al. proposed the first IPE scheme based on composite-order bilinear groups, which only achieves selective security and the length of the ciphertext is linearly related to the dimension of the policy vector [9]. Attrapadung and Libert constructed ZIPE and NIPE schemes with constant-size ciphertexts, but only the zero inner product encryption scheme can be proved to be adaptively secure [10]. Okamoto and Takashima proposed two adaptively secure NIPE schemes based on the DLIN assumption, one of which has a constant-size ciphertext and the other has a constant-size secret key [11]. Later, they first proposed an adaptively secure IPE scheme with constant-size public parameters. The dimension of the policy vector and the number of attributes are not restricted by public parameters [12]. In 2014, Chen and Wee first proposed the NIPE scheme based on the DBDH assumption which can be used for identity revocation, but only has selective security [13]. To resist quantum attacks, designing public-key cryptographic schemes based on difficult problems on the lattice has become a research hotspot. Agrawal et al. first proposed an inner product function encryption scheme based on Learning with Errors (LWE) assumption in a lattice group, which has weak attribute hiding but only selective security [14]. The above schemes are all constructed on bilinear groups or lattices without additive homomorphism and multiplicative homomorphism so that they are not suitable for dense state computing of cloud data. In 2016, Agrawal et al. constructed a FE scheme on the Paillier group based on the DCR assumption, which provided us with ideas for designing an additive homomorphic NIPE scheme [15]. Table 1 shows the characteristics of our scheme and previous schemes.

Our contribution. Considering that the public key encryption scheme constructed based on the DCR assumption has additive homomorphism, we decided to learn from the advantages of the existing IPE schemes to design a NIPE scheme based on this assumption. Katsuma and Yamada presented an adaptively secure NIPE scheme based on lattice and a framework for transforming from FE to NIPE [16]. Using this framework, this paper designs a NIPE scheme with additive homomorphism based on the DCR assumption. It can not only realize the access control to ciphertext data but also is suitable for ciphertext calculation so that it has a wider application prospect. In terms of the security, we prove that the scheme is adaptively secure based on the DCR assumption with a series of indistinguishable game sequences. In short, the scheme given in this article has the following two characteristics.

1. It can be widely used in attribute revocation, blacklist management, secret voting, etc.
2. It has additive homomorphism and can be used for ciphertext retrieval, etc.

Organization. Section 2 provides the basic knowledge and symbols that need to be explained in this article. Section 3 introduces our scheme and its security proof. Section 4 gives the performance analysis.

Table 1. Comparison of previous works about IPE

Paper	Type	Homomorphism	Assumption	Security
[9]	ZIPE	No	Subgroup Decision	Selective
[10]	ZIPE	No	DLIN	Adaptive
	NIPE			Selective
[11]	NIPE	No	DLIN	Adaptive
[12]	ZIPE	No	DLIN	Adaptive
[13]	NIPE	No	DBDH	Selective
[14]	FE	No	LWE	Selective
Our	NIPE	Addition	DCR	Adaptive

2. FORMAT GUIDE

Notation. Let Z denote the set of integers. Let Z_N^* denote the reduced residues system of modulo N. We represent a vector $(x_1,\ldots,x_l)\in \mathrm{Z}_p^l$ with lowercase boldface characters $\boldsymbol{x}$, and represent its infinite norm with $\|\boldsymbol{x}\|_\infty$. Let $[a,b]=\{a,a+1,..,b\}$ for natural numbers a and b if $a<b$. In particular, $[a,b]$ will be written as $[b]$ if $a=1$. Let natural number λ denotes the standard security parameter. Let $negl(\lambda)$ denotes a negligible function which is less than $1/\mathrm{p}(\lambda)$ for a polynomial function $\mathrm{p}(\lambda)$.

Definition 1. (s-DCR Assumption [4, 17]). Given $N=pq$ and p,q are two large prime numbers. Define the s-DCR (with integer $s>0$) assumption as follows: For any Probability Polynomial Time (PPT) adversary A, the advantage of distinguishing the following two distributions is negligible.

$$D_0=\left\{z=z_0^{N^s} \bmod N^{s+1}, z_0\in \mathrm{Z}_N^*\right\},\ D_1=\left\{z\in \mathrm{Z}_{N^{s+1}}^*\right\}.$$

Note. We set $D_0=\left\{z=z_0^N \bmod N^2, z_0\in \mathrm{Z}_N^*\right\}$, $D_1=\left\{z\in \mathrm{Z}_{N^2}^*\right\}$ in 1-DCR problem.

Definition 2. (Formal Definition of the NIPE Scheme).

$\textbf{\textit{Setup}}\left(1^{\lambda},1^{l}\right)\rightarrow mpk,msk$: First input the security parameters $(\lambda,\ l)$, then output the master public key mpk and master secret key msk .

$\textbf{\textit{KeyGen}}\ (msk,\boldsymbol{x})\rightarrow sk$: First input an attribute vector $\boldsymbol{x}$, then compute the secret key sk and retain it.

$\textbf{\textit{Enc}}(mpk,m\cdot\boldsymbol{y})\rightarrow c$: First input the message m and vector $\boldsymbol{y}$, then output the ciphertext c .

$\textbf{\textit{Dec}}\left(mpk,(\boldsymbol{x},sk),(\boldsymbol{y},c)\right)\rightarrow m$: First input the $(\boldsymbol{x},sk)$ and $(\boldsymbol{y},c)$, then output the message m if $\langle\boldsymbol{x},\boldsymbol{y}\rangle\neq 0$.

Definition 3. (Adaptive Security Model of the NIPE Scheme).

This model is described by a series of games between adversary A and challenger Bŀ.

Setup. The challenger Bŀ runs the ***Setup***, then sends mpk to the adversary A .

Phase 1. The adversary A adaptively chooses an attribute vector $\boldsymbol{x}'$ for a secret key query. Then the challenger Bŀ runs ***KeyGen*** and returns $sk_{x'}$.

Challenge Phase. The adversary A submits two equal-length messages (m_1 and m_2) and the challenge vector $\boldsymbol{y}$ to the challenger B (Any challenge attribute vector $\boldsymbol{x}'$ and vector $\boldsymbol{y}$ satisfy $\langle\boldsymbol{x}',\boldsymbol{y}\rangle=0$). Then challenger B samples $b\in\{0,1\}$, runs ***Enc*** and returns c to the adversary A .

Phase 2. The adversary A may do more secret key queries, the specific steps are as Phase 1.

Guess. Finally, the adversary A outputs a guess b' about $b\in\{0,1\}$ and A win the game if $b'=b$. Now we let Adv_i denote the advantage of a PPT adversary A wins in the $Game_i$. If $Adv_{\mathrm{A}}(\lambda)$ is a negligible function for any PPT adversary A , we say it is adaptively secure.

3. ADAPTIVELY SECURE NIPE SCHEME BASED ON DCR ASSUMPTION

In this section, we propose our NIPE scheme based on DCR assumption. Sect.3.1, Sect.3.3 and Sect.3.4 show a specific description of its construction, security and homomorphism proof.

3.1. Construction

We first show our scheme by four PPT algorithms.

Algorithm 1 $\textbf{\textit{Setup}}\left(1^{\lambda},1^{l}\right)$

Input: Security parameter λ and vector dimension l .
Output: Master public key mpk and master secret key msk .

1. Pick two prime numbers p and q of the form $p=2p'+1$ (random prime numbers $p',q'>2^{\mathrm{p}(\lambda)}$).

2. Let $N = pq$.
3. Sample $g' \in \mathrm{Z}_{N^2}^*$ and compute $g = g'^{2N} \bmod N^2$.
4. Sample $s_i \in \left\{-2^{\lambda-1}N^4, -2^{\lambda-1}N^4+1, ..., 2^{\lambda-1}N^4\right\}$ and compute $h_i = g^{s_i} \bmod N^2$.
5. Return $mpk = \left(N, g, \{h_i\}_{i\in[l]}\right)$, $msk = \{s_i\}_{i\in[l]}$.

Algorithm 2 ***KeyGen*** $(msk, \boldsymbol{x})$

Input: Master secret key *msk* and attribute vector $\boldsymbol{x}$.
Output: Secret key *sk*.

1. Select the attribute vector $\boldsymbol{x} = (x_1, x_2, ..., x_l) \in \mathrm{Z}^l$ and $0 \le x_i < N^{1/4}l^{-1/2}$.
2. Compute $sk = \sum_{i=1}^{l} s_i \cdot x_i \in \mathrm{Z}$.
3. Return *sk*.

Algorithm 3 ***Enc*** $(mpk, m\cdot \boldsymbol{y})$

Input: Message $0 < m < N^{1/2}$, vector $\boldsymbol{y} = (y_1, y_2, ..., y_l) \in \mathrm{Z}^l$ and $0 \le y_i < N^{1/4}l^{-1/2}$.
Output: Ciphertext *c*.

1. Pick $r \in \{0, \ldots, \varphi(N)/2\}$ randomly and compute
$$c = \left\{ \begin{matrix} c_0 = g^r \bmod N^2 \\ \left\{c_i = (1 + m\cdot y_i N)\cdot h_i^r \bmod N^2\right\}_{i\in[l]} \end{matrix} \right\}.$$
2. Return the ciphertext *c*.

Algorithm 4 ***Dec*** $(mpk, (\boldsymbol{x}, sk), (\boldsymbol{y}, c))$

Input: $(\boldsymbol{x}, sk)$ and $(\boldsymbol{y}, c)$
Output: message *m*

1. Compute
$$\left.\right) c = \prod_{i=1}^{l} c_i^{x_i} \cdot c_0^{-sk} \bmod N^2, \; z = \begin{cases} \left.\right)(c-1) \bmod N^2 / N, & if \; \langle \boldsymbol{x}, \boldsymbol{y} \rangle > 0 \\ \left(\left.\right)(c-1) \bmod N^2 - N^2\right) / N, & if \; \langle \boldsymbol{x}, \boldsymbol{y} \rangle < 0 \end{cases}.$$
2. Return the message $m = z / \langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

3.2. Correctness

The correctness of the decryption algorithm is given by the following formula.

$$
\begin{aligned}
\overset{)}{c} &= \prod_{i=1}^{l} c_i^{x_i} \cdot c_0^{-sk} \bmod N^2 \\
&= \prod_{i=1}^{l} \left(1 + N \cdot m \cdot y_i\right)^{x_i} \cdot h_i^{r \cdot x_i} \cdot g^{-r \cdot sk} \bmod N^2 \\
&= \prod_{i=1}^{l} \left(1 + N \cdot m \cdot x_i \cdot y_i\right) \cdot g^{r \cdot s_i \cdot x_i} \cdot g^{-r \sum_{i=1}^{l} s_i \cdot x_i} \bmod N^2 \\
&= \prod_{i=1}^{l} \left(1 + N \cdot m \cdot x_i \cdot y_i\right) \bmod N^2 \\
&= 1 + m \cdot N \sum_{i=1}^{l} x_i \cdot y_i \bmod N^2 \\
&= 1 + m \cdot N \cdot \langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod N^2
\end{aligned}
$$

$z = (\overset{)}{c} - 1) \bmod N^2 / N = m \cdot \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle > 0$ and $z = \left((\overset{)}{c} - 1) \bmod N^2 - N^2\right) / N = m \cdot \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ if not, so there must be $m = z / \langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

Note. A few notes about the process of decryption.

(1). Regarding the calculation of $c_0^{-sk} \bmod N^2$. We first need to calculate $c_0^{sk} \bmod N^2$ if $sk > 0$, and then use the extended Euclidean algorithm to calculate $\left(c_0^{sk} \bmod N^2\right)^{-1} \bmod N^2$. The following analysis that $c_0^{sk} \bmod N^2$ must be reversible. And $c_0^{sk} \bmod N^2$ can be written as $(g')^w \bmod N^2$ because of $c_0 = g'^{2Nr} \bmod N^2$ and $g' \in \mathrm{Z}_{N^2}^*$. If $(N\varphi(N)) \mid w$, then the inverse element of $c_0^{sk} \bmod N^2$ is 1. Otherwise, $u^w \bmod N^2 = (g')^{-w} \bmod N^2$ and $u = (g')^{-1} \bmod N^2$.

(2) According to the parameter setting, we have

$$
\left| m \cdot \langle \boldsymbol{x}, \boldsymbol{y} \rangle \right| < \left| N^{1/2} \cdot \sum_{i=1}^{l} x_i y_i \right| < N^{1/2} \cdot l \cdot N^{1/4} l^{-1/2} \cdot N^{1/4} l^{-1/2} = N .
$$

That is, if the attribute vector $\boldsymbol{x}$ satisfies $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \neq 0$, the decryption algorithm can output the message m.

To facilitate the understanding of the above scheme, specific examples are given below to illustrate the specific operation process of the scheme.

Setup: Set the number of attributes $l = 2$, "safe" parameters $\lambda = 2$, $\mathrm{p}(\lambda) = \lambda = 2$, two isometric prime numbers $(p, q) = (11, 13)$. Compute $N = 143$, $N^2 = 20449$, $2 < N^{1/4} l^{-1/2} < 3$. Select $g' = 3$, $g = g'^{2N} = 9441 \bmod N^2$, $(s_1, s_2) = (2, 3)$, and compute $(h_1, h_2) = (15739, 9465)$. Output $mpk = (143, 9441, \{15739, 9465\})$, $msk = \{2, 3\}$。

KeyGen: Let the attribute vector $\boldsymbol{x} = (2, 2)$, output the secret key $sk = s_1 x_1 + s_2 x_2 = 10$.

The following shows the corresponding encryption and decryption operations for embedding two

different vectors $\boldsymbol{y}$ in the ciphertext.

1. If the vector $\boldsymbol{y}$ selected in the encryption algorithm satisfies $\langle \boldsymbol{x}, \boldsymbol{y} \rangle > 0$.

Enc: Input $m = 5,\ \boldsymbol{y} = (1,2),\ r = 2$ and output as follows.

$$\begin{cases} c_0 = 9441^2 \bmod 20449 = 15739 \\ c_1 = (1 + 5 \times 1 \times 143) \times 15739^2 \bmod 20449 = 13952 \\ c_2 = (1 + 5 \times 2 \times 143) \times 9465^2 \bmod 20449 = 19176 \end{cases}$$

Dec: At this point, we have $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 6 > 0$. Firstly, calculate $c_0^{sk} = 19119 \bmod N^2$, c_0^{-sk}. $= 19119^{-1} = 10286 \bmod N^2$ and $\hat{c} = c_1^{x_1} \cdot c_2^{x_2} \cdot c_0^{-sk} = 7723 \bmod N^2$. Compute $z = (\hat{c} - 1)/N = 30$ because of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle > 0$. Finally, output the message $m = z / \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 5$.

2. If the vector $\boldsymbol{y}$ selected in the encryption algorithm satisfies $\langle \boldsymbol{x}, \boldsymbol{y} \rangle < 0$.

Enc: Input $m = 5,\ \boldsymbol{y} = (1,-2),\ r = 2$ and output as follows.

$$\begin{cases} c_0 = 15739,\ c_1 = 13952 \\ c_2 = (1 + 5 \times (-2) \times 143) \times 9465^2 \bmod 20449 = 20034 \end{cases}$$

Dec: At this point, we have $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = -2 < 0$. Firstly, calculate $c_0^{sk} = 19119 \bmod N^2$, c_0^{-sk}. $= 19119^{-1} = 10286 \bmod N^2$ and $\hat{c} = c_1^{x_1} \cdot c_2^{x_2} \cdot c_0^{-sk} = 19020 \bmod N^2$. Then because of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle < 0$, $z = (\hat{c} - 1 - N^2)/N = -10$. Finally, output the message as $m = z / \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 5$.

3.3. Security

Our security proof relies on a series of games which are detailed below. Table 2 shows some parameters in these four games.

Games.

$Game_0$. The original system generates the sk and ciphertext c.

$Game_1$. Same as $Game_0$, but the range of $\{s_i\}_{i \in [l]}$ is changed. Specifically, the challenger B reduces the upper bound of $\{s_i\}_{i \in [l]}$ such as $|s_i| < 2^{\lambda - 1} N^4 - (N^4/2)$.

Note that the upper bound of the value of $\{s_i\}_{i \in [l]}$ is reduced in Game_1, and the upper bound of the value is restored to $2^{\lambda - 1} N^4$ by a special parameter setting method in the proof of $Game_3$. This step is to prepare for finding equivalent parameters in $Game_3$.

$Game_2$. Compare to $Game_1$, modify the challenge ciphertext $c=(c_0,c_1,...,c_l)$. Specifically, the challenger B computes

$$c=\left\{\begin{array}{l} c_0=z^2=\left(z_0{}^N\right)^2 \bmod N^2, z=z_0{}^N \bmod N^2, z_0\in \mathrm{Z}_N^* \\ \left\{c_i=\left(1+m_b\cdot y_i N\right)\cdot c_0{}^{s_i} \bmod N^2\right\}_{i\in[l]} \end{array}\right\}.$$

The modification is to connect the attack scheme with the 1-DCR assumption.

$Game_3$. Modify the first item of the challenge ciphertext

$$c_0=z^2 \bmod N^2, z\in \mathrm{Z}_{N^2}^*.$$

Then compute c_i as $Game_2$

$$c_i=\left(1+m_b\cdot y_i N\right)\cdot c_0{}^{s_i} \bmod N^2, i\in[l].$$

Note that $Game_3$ is designed to make the ciphertext of m_0 under one set of parameters, from the algebraic expression, equivalent to the ciphertext of m_1 under another set of parameters. And the adversary cannot distinguish between the two sets of parameters, so the adversary's attack advantage is almost zero.

Table 2. Master secret key and ciphertext in four games.

Game	$\boldsymbol{Game_0}$	$\boldsymbol{Game_1}$	$\boldsymbol{Game_2}$	$\boldsymbol{Game_3}$
c_0	g^r	g^r	$z_0^{2N}, z_0\in \mathrm{Z}_N^*$	$z^2, z\in \mathrm{Z}_{N^2}^*$
c_i	$\left(1+m\cdot y_i N\right)\cdot h_i{}^r$	$\left(1+m\cdot y_i N\right)\cdot h_i{}^r$	$\left(1+m\cdot y_i N\right)\cdot c_0{}^{s_i}$	$\left(1+m\cdot y_i N\right)\cdot c_0{}^{s_i}$
$\lvert s_i\rvert$	$2^{\lambda-1}N^4$	$2^{\lambda-1}N^4-\left(N^4/2\right)$	$2^{\lambda-1}N^4-\left(N^4/2\right)$	$2^{\lambda-1}N^4-\left(N^4/2\right)$

Lemma 1 ($Game_0\approx Game_1$). The advantage of the adversary As in $Game_0$ and $Game_1$ satisfies $\left|Adv_0-Adv_1\right|\le l/2^{\lambda}$.

Proof. We first analyze the differences about various parameters and focus on $\{h_i\}_{i\in[l]}$. Obviously, relative to $Game_0$, $Game_1$ only changes the bound of $\{s_i\}_{i\in[l]}$. Consequently, $\left|Adv_0-Adv_1\right|\le l\cdot\left(N^4/2^{\lambda}N^4\right)=l/2^{\lambda}$.

Lemma 2 ($Game_1\approx Game_2$). The advantage of the adversary A in $Game_1$ or $Game_2$ satisfies $\left|Adv_1-Adv_2\right|<1/2^{p(\lambda)}$.

Proof. We conclude that $g^r\approx_c z^2 \bmod N^2$ based on that they are both $(2N)$th residuals in $\mathrm{Z}_{N^2}^*$, which means that the ciphertext in $Game_1$ and $Game_2$ have the same distribution. That is, As can't determine whether the ciphertext distribution belongs to $Game_1$ or $Game_2$. Consequently,

$\left|Adv_1 - Adv_2\right| \le 1/2^{p(\lambda)}$.

Lemma 3 ($Game_2 \approx Game_3$). There exists a challenger B_1 who can solve 1-DCR problem with a non-negligible advantage if the adversary A can determine c in $Game_2$ or $Game_3$. That is, $\left|Adv_2 - Adv_3\right| \le Adv_{B_1}^{1-DCR}(\lambda)$.

Proof. From 1-DCR assumption, let

$$D_0 = \left\{ z = z_0^{2N} \bmod N^2, z_0 \in Z_N^* \right\}, \quad D_1 = \left\{ z^2, z \in Z_{N^2}^* \right\}.$$

T is the input of the challenger Bþ. To determine T belongs to D_0 or D_1, Bþ performs as the following algorithm.

Algorithm 3.5

Setup:

1. B_1 runs algorithm ***Setup***.

2. B_1 samples

$$s_i \in \left\{ -2^{\lambda-1}N^4 + \left(N^4/2\right), -2^{\lambda-1}N^4 + 1, ..., 2^{\lambda-1}N^4 - \left(N^4/2\right) \right\}.$$

3. B_1 sends mpk to the adversary A .

Phase 1:

1. A adaptively choose an attribute vector $\boldsymbol{x}' = \left(x_1', x_2', ..., x_l'\right) \in Z^l$.
2. B_1 runs the *KeyGen* and sends $sk_{x'}$ to adversary A .

Challenge:

1. A submits two equal-length messages (m_1 and m_2) and the challenge vector $\boldsymbol{y}$ to B_1 (Any challenge attribute vector $\boldsymbol{x}'$ and vector $\boldsymbol{y}$ satisfy $\langle \boldsymbol{x}', \boldsymbol{y} \rangle = 0$).
2. B_1 samples $b \in \{0,1\}$, runs the algorithm *Enc* and returns c to the adversary A .

$$c = \left\{ c_0 = T \bmod N^2, \left\{ c_i = \left(1 + m_b \cdot y_i N\right) \cdot T^{s_i} \bmod N^2 \right\}_{i \in [l]} \right\}$$

Phase 2: A may do more secret key queries, the specific steps are as Phase 1.

Guess: A outputs a guess $b' \in \{0,1\}$ and A wins the game if $b' = b$.

Note: If $T \in D_0$, we have

$$c = \left\{ c_0 = z_0^{2N} \bmod N^2, \left\{ c_i = \left(1 + m_b \cdot y_i N\right) \cdot z_0^{2Ns_i} \bmod N^2 \right\}_{i \in [l]} \right\},$$

If $T \in D_1$, we have

$$c = \left\{ c_0 = z^2 \bmod N^2, \left\{ c_i = \left(1 + m_b \cdot y_i N\right) \cdot c_0^{s_i} \bmod N^2 \right\}_{i \in [l]} \right\}.$$

That is, B_1 can solve the 1-DCR problem if A can distinguish $Game_2$ from $Game_3$. So we

conclude $\left|Adv_2 - Adv_3\right| \le Adv_{\mathrm{B}_1}^{1-DCR}(\lambda)$.

Lemma 4 $\left|Adv_3 - 1/2\right| < 1/2^{\mathrm{p}(\lambda)}$.

Proof. We sample $\alpha_z \in \mathrm{Z}_N$, $r_z \in \mathrm{Z}_{p'q'}$ and modify $c_0 = (1+\alpha_z N)\cdot g^{r_z} \bmod N^2$. This is the same as the expression $c_0 = z^2 \bmod N^2$ because it is also the square residue in $\mathrm{Z}_{N^2}^*$ (This is due to $c_0 = (1+\alpha_z N)\cdot g^{r_z} = \left(\left(\alpha_z N\cdot(N+1)/2\right)+1\right)\cdot g'^{N\cdot r_z}\right)^2 \bmod N^2$). Then we have

$$\begin{aligned} c_i &= (1+m_b\cdot y_i N)\cdot(1+\alpha_z N)^{s_i}\cdot g^{r_z\cdot s_i} \bmod N^2 \\ &= (1+m_b\cdot y_i N)\cdot(1+s_i\alpha_z N)\cdot g^{r_z\cdot s_i} \bmod N^2 \\ &= (1+m_b\cdot y_i N+\alpha_z s_i N)\cdot g^{r_z\cdot s_i} \bmod N^2,\ i\in[l] \end{aligned}.$$

We observe that the integer $\alpha_z \in \mathrm{Z}_N$ is invertible with the probability $1-(p+q-1)/pq$ which is close to 1. So there must be an integer $\mu \in \mathrm{Z}$, $|\mu| < N$ which causes $\mu\cdot p'q' \equiv 1 \bmod N$ based on the fact $\gcd(p'q', N) = 1$. Besides, define

$$\left\{s_i' = s_i + \left(a_z^{-1} \bmod N\right)\cdot(m_b - m_{1-b})\cdot y_i\cdot(\mu\cdot p'q') \in \mathrm{Z}\right\}_{i\in[l]},$$

which shows

$$\left\{\begin{array}{l} s_i' = s_i + \left(a_z^{-1} \bmod N\right)\cdot(m_b - m_{1-b})\cdot y_i \bmod N \\ s_i' = s_i \bmod p'q' \end{array}\right\}_{i\in[l]},$$

and

$$\begin{aligned} c_i &= (1+m_{1-b}\cdot y_i N)\cdot c_0^{s_i'} \bmod N^2 \\ &= (1+m_{1-b}\cdot y_i N)\cdot\left((1+\alpha_z N)\cdot g^{r_z}\right)^{s_i'} \bmod N^2 \\ &= (1+m_{1-b}\cdot y_i N+\alpha_z s_i' N)\cdot g^{r_z\cdot s_i'} \bmod N^2 \end{aligned}.$$

Besides, we show $s_i' \in \left\{-2^{\lambda-1}N^4, -2^{\lambda-1}N^4+1, ..., 2^{\lambda-1}N^4\right\}$ due to the following inequation:

$$\left|\left(a_z^{-1} \bmod N\right)\cdot(m_b - m_{1-b})\cdot y_i\cdot(\mu\cdot p'q')\right| \le N\cdot 2N^{1/2}\cdot N^{1/4}l^{-1/2}\cdot N\cdot\frac{N}{4} < \frac{N^4}{2}.$$

According to $\langle \boldsymbol{x}', \boldsymbol{y}\rangle = 0$, we get $\sum_{i=1}^{l} s_i\cdot x_i' = \sum_{i=1}^{l} s_i'\cdot x_i'$, which means that the adversary A is blind for these two master secret keys. That is, the adversary A cannot determine what the message is m_0 or m_1 because these two ciphertexts have the same distribution. That is, the advantage of the adversary A satisfies

$$\left|Adv_3 - 1/2\right| \le (p+q-1)/pq < 1/2^{\mathrm{p}(\lambda)}.$$

Theorem 1. From the 1-DCR assumption, our NIPE scheme over Z is adaptively secure.

Proof. We can determine the advantage of the adversary A in $Game_0$ by lemma 1~4.

$$\begin{aligned}&\left|Adv_0 - 1/2\right| \le \left|Adv_0 - Adv_1\right| + \left|Adv_1 - Adv_2\right| + \left|Adv_2 - Adv_3\right| + \left|Adv_3 - 1/2\right| \\ &\le l/2^{\lambda} + 1/2^{\mathrm{p}(\lambda)-1} + Adv_{\mathrm{B}_0}^{1-DCR}(\lambda)\end{aligned}$$

3.4. Homomorphism

The following shows that the scheme in this chapter has additive homomorphism, where $m_1, m_2, ..., m_k$ represents k plaintexts and $m = \sum_{j=1}^{k} m_j < \sqrt{N}$. $E(m_j) = \{c_0^j, c_1^j, ..., c_l^j\}$ represents a ciphertext obtained by encrypting the plaintext m_j and $E(m_j)_i = c_i^j$.

$$\begin{aligned}&\left\{\prod_{j=1}^{k} E(m_j)_0, ..., \prod_{j=1}^{k} E(m_j)_i, ...\right\} \\ &= \left\{\prod_{j=1}^{k} c_0^j, ..., \prod_{j=1}^{k} c_i^j, ...\right\} \\ &= \left\{\prod_{j=1}^{k} g^{r_j} \bmod N^2, ..., \prod_{j=1}^{k} (1 + m_j \cdot y_i N) \cdot g^{s_i \cdot r_j} \bmod N^2, ...\right\} \\ &= \left\{g^{\sum_{j=1}^{k} r_j} \bmod N^2, ..., \prod_{j=1}^{k} (1+N)^{m_j \cdot y_i} \cdot g^{s_i \cdot r_j} \bmod N^2, ...\right\} \\ &= \left\{g^{\sum_{j=1}^{k} r_j} \bmod N^2, ..., (1+N)^{y_i \cdot \sum_{j=1}^{k} m_j} \cdot g^{s_i \cdot \sum_{j=1}^{k} r_j} \bmod N^2, ...\right\} \\ &= \left\{g^{\sum_{j=1}^{k} r_j} \bmod N^2, ..., (1+N)^{y_i \cdot m} \cdot g^{s_i \cdot \sum_{j=1}^{k} r_j} \bmod N^2, ...\right\} \\ &= \left\{E(m)_0, ..., E(m)_i, ...\right\}\end{aligned}$$

To better illustrate the additive homomorphism of the scheme in this chapter, an example is given below, in which the number of plaintexts is $k = 2$.

Setup: Set the number of attributes $l = 2$, "safe" parameters $\lambda = 2$, $\mathrm{p}(\lambda) = \lambda + 1 = 3$, two isometric prime numbers $p = 11$, $q = 13$. So $N = 143$, $N^2 = 20449$, $2 < N^{1/4} l^{-1/2} < 3$. Select $g' = 3$ and compute $g = g'^{2N} = 9441 \bmod N^2$, $(s_1, s_2) = (2,3)$, $(h_1, h_2) = (15739, 9465)$. Finally, output

$$mpk = (143, 9441, \{15739, 9465\}),\ msk = \{2,3\}$$

KeyGen: Let the attribute vector $\boldsymbol{x} = (2,2)$, output the secret key $sk = s_1 x_1 + s_2 x_2 = 10$.

Enc: Input $(m_1, m_2) = (4,5)$, $\boldsymbol{y} = (1,2)$, $(r_1, r_2) = (2,3)$, then compute and output as follows.

$$\left\{c_0^1 = 15739,\ c_1^1 = 2369,\ c_2^1 = 15172\right\},\ \left\{c_0^2 = 9465,\ c_1^2 = 9166,\ c_2^2 = 15965\right\}$$

Query: A user B sends a "query request" for $m = m_1 + m_2$ to the cloud server.

Compute: The cloud server calculates the new ciphertext as follows and send it to B.

$$\left\{c_0 = c_0^1 \cdot c_0^2 = 19119 \bmod N^2,\ c_1 = c_1^1 \cdot c_1^2 = 17865 \bmod N^2,\ c_2 = c_2^1 \cdot c_2^2 = 2575 \bmod N^2\right\}$$

Dec: User B first calculates $c_1^{x_1} \cdot c_2^{x_2} \bmod N^2 = 14257 \bmod N^2$ and $c_0^{sk} = 19119 \bmod N^2$ after receiving the ciphertext, then $c_0^{-sk} = 19119^{-1} = 10286 \bmod N^2$. Secondly, he calculates $\hat{c} = c_1^{x_1} \cdot c_2^{x_2} \cdot c_0^{-sk} = 7723 \bmod N^2$ and $z = (\hat{c} - 1)/N$. Finally outputs the message as follows.

$$m = z/\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 9 (= m_1 + m_2)$$

4. PERFORMANCE ANALYSIS

This paper constructs an adaptively secure NIPE scheme with additive homomorphism based on DCR assumption. Table 3 shows the parameter comparison between our scheme and the existing NIPE scheme, where l represents the number of attributes, $|G|$ and $|G_T|$ represents the order of the group G and G_T. B, E, D represents the time complexity of bilinear pairing, exponential, and division operations in group G.

Table 3. Parameter length and comparison of existing NIPE schemes.

Scheme	Homomo-rphism	Master public-key	Ciphertext	Decryption complexity	Assump-tion	Security
[18]	No	$6l\|G\|$	$5l\|G\|$	$9B$	DLIN	Adaptive
[13]	No	$(l^2+3l+1)\|G\| + \|G_T\|$	$(2l+3)\|G\| + \|G_T\|$	$9B + D$	DBDH	Selective
[15]	No	$(l+1)\|G\|$	$9\|G\| + \|G_T\|$	$3B + D$	DLIN	Selective
Our	Additive	$(l+1)\|G\|$	$(l+1)\|G\|$	$(n+1)E + D$	DCR	Adaptive

Compared with the existing NIPE schemes, the scheme in this paper has additive homomorphism, which facilitates the processing of ciphertext data in the cloud. The scale of the public parameters of the scheme is the same as that of the scheme [15], but this scheme is adaptively secure, and its security intensity is far greater than the selectively secure. Compared with the same security scheme [18], its number of ciphertexts has advantages.

5. CONCLUSIONS

This paper proposes a NIPE scheme with additive homomorphism. Based on DCR assumption, it is proved that the scheme is adaptively secure. In terms of whether it has homomorphism, this scheme selects composite-order residual class rings to replace the bilinear group used in the previous NIPE schemes, which makes the scheme have additive homomorphism and is more suitable for dense state calculation of cloud private data. But it needs to be specially pointed out that the parameter range is expanded. And how to design an adaptively secure NIPE scheme with additive homomorphism in a small parameter range is a new challenge.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (No. 61902428).

REFERENCES

[1] Dan Boneh, Amit Sahai & Brent Waters, (2011) "Functional encryption: Definitions and challenges", TCC 2011: Theory of Cryptography, LNCS 6597, pp 253-273.
[2] Adam O'Neill, (2010) "Definitional Issues in Functional Encryption", Cryptology ePrint Archive, Report 2010/556. https://eprint.iacr.org/2010/556.pdf
[3] Amit Sahai & Brent Waters, (2005) "Fuzzy Identity-Based Encryption", Cryptology – EUROCRYPT 2005, LNCS 3494, pp 457-473.
[4] Ronald L. Rivest, Len Adleman & Michael L. Dertouzos, (1978) "On Data Banks and Privacy Homomorphisms", Foundations of Secure Computation, vol. 4, pp 169-179.
[5] T. W. Hungerford, (1982) "ALGEBRA: (Graduate Texts in Mathematics, 73)", Bulletin of the London Mathematical Society, vol. 14, pp 158-159.
[6] Pascal Paillier, (1999) "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", Advances in Cryptology - EUROCRYPT '99, LNCS 1592, pp 223-238.
[7] R.L. Rivest, A. Shamir & L. Adleman, (1983) "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM 26, pp 96–99.
[8] T. Elgamal, (1984) "A Public-Key Cryptosystems and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472.
[9] J. Katz, A. Sahai & B. Waters. (2013) "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products", Journal of Cryptology 26, pp 191–224.
[10] N. Attrapadung & B. Libert, (2010) "Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation," Public Key Cryptography – PKC 2010, LNCS 6056, pp 384-402.
[11] T. Okamoto & K. Takashima, (2009) "Hierarchical Predicate Encryption for Inner-Products", Advances in Cryptology – ASIACRYPT 2009, LNCS 5912, pp 214-231.
[12] T. Okamoto & K. Takashima, (2012) "Fully Secure Unbounded Inner-Product and Attribute-Based Encryption", Advances in Cryptology – ASIACRYPT 2012, LNCS 7658, pp 349-366.
[13] J. Chen & H. Wee, "Doubly spatial encryption from DBDH", Theoretical Computer Science. 543, pp 79-89.
[14] S. Agrawal, D. M. Freeman & V. Vaikuntanathan, (2011) "Functional Encryption for Inner Product Predicates from Learning with Errors," Advances in Cryptology – ASIACRYPT 2011, LNCS 7073, pp 21-40.
[15] S. Agrawal, B. Libert & D. Stehle, (2016) "Fully Secure Functional Encryption for Inner Products, from Standard Assumptions", Advances in Cryptology – CRYPTO 2016, LNCS 9816, pp 333-362
[16] S. Katsumata & S. Yamada, (2019) "Non-zero Inner Product Encryption Schemes from Various Assumptions: LWE, DDH and DCR", Public-Key Cryptography – PKC 2019, LNCS 11443, pp 158-188.
[17] I. Damgard & M. Jurik, (2001) "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System", Public Key Cryptography- 2001, LNCS 1992, pp 119-136.
[18] T. Okamoto & K. Takashima, (2011) "Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption", Cryptology and Network Security, LNCS 7092, pp 138-159.
[19] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters & H. Gilbert, (2010) "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption", Advances in Cryptology – EUROCRYPT 2010, LNCS 6110, pp 62-91.

AUTHORS

Haiying Gao

Female, born in 1978, from Zhoukou City, Henan Province, China. In 2006, she received a PhD degree from Beijing University of Posts and Telecommunications. Now she is a professor and doctoral supervisor at the Information Engineering University, and her research direction is design and analysis of cryptographic algorithm.

Chao Ma

Male, born in 1995, from Zhengzhou City, Henan Province, China. In 2021, he received a master's degree from the University of Information Engineering. The research direction is the design and analysis of public key cryptographic algorithms.

INTELLIGENT UNIT LEVEL TEST GENERATOR FOR ENHANCED SOFTWARE QUALITY

Ning Luo and Linlin Zhang

Visual Computing Group,
Intel Asia-Pacific Research & Development Ltd, Shanghai, China

ABSTRACT

Unit level test has been widely recognized as an important approach to improving software quality, as it can expose bugs earlier during the development phase. However, manual unit level test development is often tedious and insufficient. Also, it is hard for developers to precisely identify the most error prone code block deserving the best test coverage by themselves. In this paper, we present the automatic Unit level test framework we used for intel media driver development. It can help us identify the most critical code block, provide the test coverage recommendation, and automatically generate >80% ULT code (~400K Lines of test code) as well as ~35% test cases (~7K test cases) for intel media driver. It helps us to greatly shrink the average ULT development effort from ~24 Man hours to ~3 Man hours per 1000 Lines of driver source code.

KEYWORDS

Unit level test, error prone logic, test coverage inference, automatic ULT generation, fuzzing, condition/decision coverage.

1. INTRODUCTION

Unit level test (ULT) has been widely recognized as an important approach to improving software quality, as it can expose bugs earlier during the development phase. For Intel Media driver, our ULT coverage target is 100% for functional coverage and >70% for conditional coverage.

However, the large scale of Intel Media driver (~1.7 millions of lines of code), as well as the tremendous amount of the classes, methods and possible inputs, makes manual ULT development till sufficient test coverage a mission impossible.

Meanwhile, for the same reason, to best ensure the software quality, developers need identify the most error prone logic for better prioritization between different components.

In this paper, we will introduce the automatic Unit level test framework used in intel media driver development for quarters. It can automatically identify the most critical code block, provide the recommended test coverage for each component, and automatically generate >80% ULT code (~400K Lines of test code) as well as ~35% test cases (~7K test cases) for Intel media driver. It helps us to greatly shrink the average ULT development effort from ~24 Man hours to ~3 Man hours per 1000 Lines of driver source code.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 185-191, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120915

2. AUTO UNIT LEVEL TEST FRAMEWORK

The Auto Unit Level Test Framework for Intel Media driver is composed of the Server-side and the client-side facilities. The server-side of Auto ULT Framework is in charge of automatic error Prone logic detection and the per component test coverage recommendation generation. While the client-side will automatically generate the test code and test cases based on the recommendation from server side. The overall architecture of Auto ULT framework can be shown by Figure1 below.

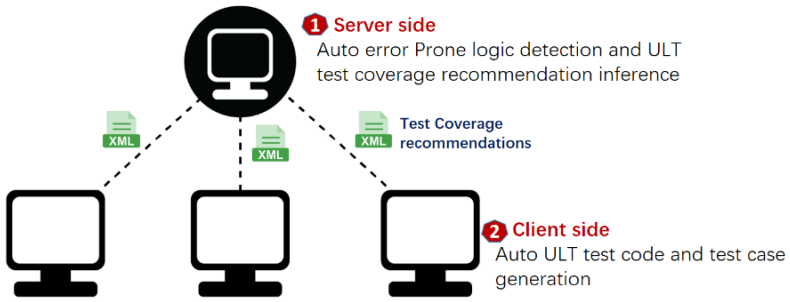


Figure 1. Auto Unit Level Test Framework Overview

2.1. Auto Error Prone Logic Detection and Test Coverage Recommendation Inference

To better expose the potential issues with limited test cases, different components need be treated differently on their ULT test coverage and those error-prone logics requires better test coverage.

At server side of the Unit Level test framework, it includes one machine learning based inference system which can help to identify the error prone logic based on the recent bug trend and then infer the per component ULT test coverage recommendation.

The basic working flow of the Auto ULT framework Server-side can be shown by figure 2.

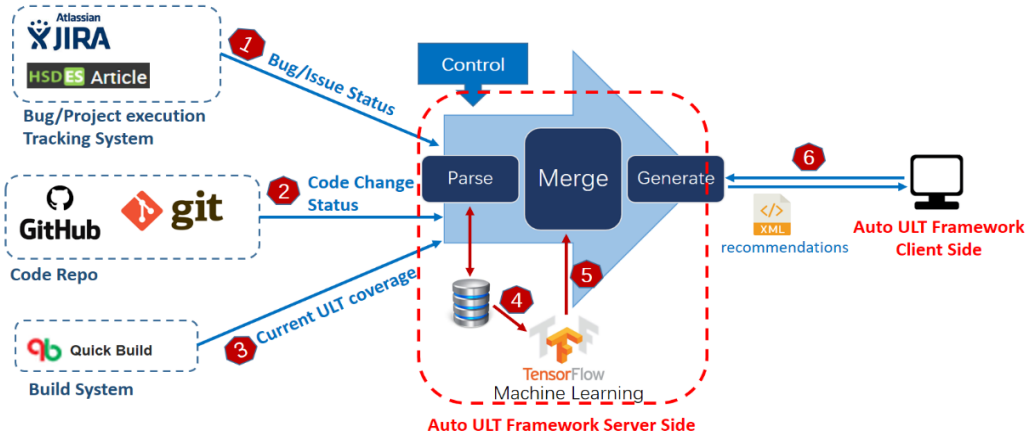


Figure 2. Auto ULT Framework Server-side Working Flow

The detailed steps are as below:

<1> The Automatic ULT framework will regularly (monthly in our case) poll the Bug system for recent bug status. For each new bug, it can get the culprit commit ID from the bug system.

<2> With the culprit commit id listed in the bug, the Automatic ULT framework will query the code repo system for the components / code blocks impacted by the culprit commit.
<3> Meanwhile, from the build the system, the automatic ULT framework can get the current ULT coverage status per each component/code block.
<4> Then by querying the internal database, the Automatic ULT framework can generate the trend for bug V.S. ULT coverage per each component.
<5> At backend of the server-side, we have one pre-trained machine learning system against tensor flow. It can deliver the inference from the trend data got in step 4 and generate the recommendation of the optimal ULT coverage for each component.
<6> The Client side will automatically query for the updated ULT coverage recommendation every time when it launched. If any test coverage improvement required per the new recommendation, it will be highlighted at client side.

The working flow of the machine learning based inference to ULT coverage recommendation mentioned in step 5 can be shown by Figure3 below.

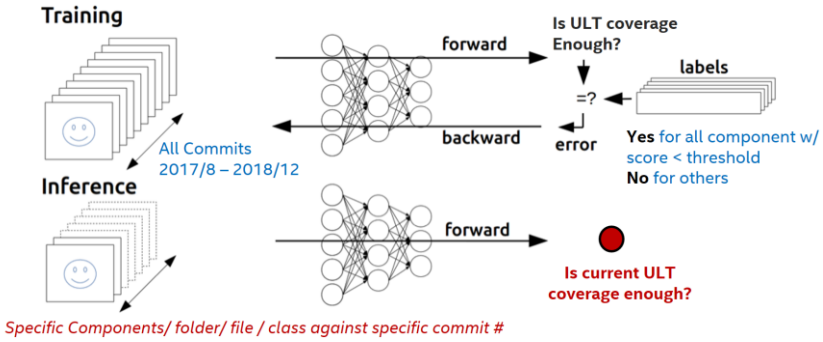


Figure 3. Machine Learning based per component ULT Coverage recommendation

The ULT coverage recommendation can then be used by developers to decide the ULT design and coverage target.

Per the recent survey to all internal media driver developers, >90% interviewees agree upon the accuracy of the recommendation.

2.2. Auto Test Generation

After nailing down the ULT design target, developers can then leverage the auto ULT framework on test code and test case generation.

The basic flow for Auto ULT test generator can be shown by Figure4 below.

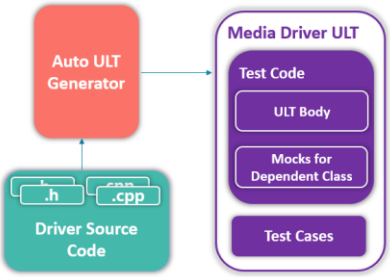


Figure 4. Client side: Auto ULT Test code and Test Case Generator

2.2.1. Auto Test Code Generation

Comparing to Driver under Test, ULT is usually of simple logic and fixed pattern which makes the auto ULT code generation feasible.

Let us see what kind of test code is required in a common ULT.

To apply the finer granularity unit level test onto one class, usually it requires several facility classes: one text fixture class, one or more test classes and one or more mock classes.

Figure5 below shows one typical example. Let's say Class A (with one dependent class: class C) is the class under test. To apply the class level ULT onto Class A, we need one test fixture class A_TestCase, one test class Test_A and one mock class to its dependency Mock_C.

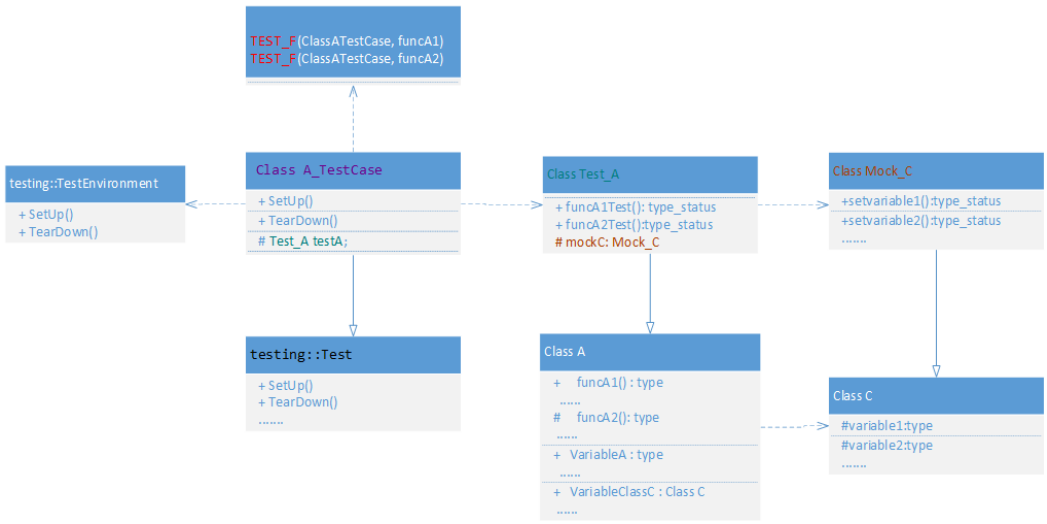


Figure 5. Test Code Required for Finer Granularity ULT

Class under test can be as below,

```
class A
{
     ......
    void func1();
    void func2();
    int variable1;
    int variable2;
    .......
};
```

Then the **test fixture class** *A_TestCase* will look like below

```
class A_TestCase : public testing::Test
{
public:
      virtual void SetUp();
      virtual void TearDown();
      Test_A *testA;
      .......
```

```
};

TEST_F (A_TestCase, function1)
{
      testA->function1Test();
}
TEST_F (A_TestCase, function2)
{
      testA->function2Test();
}
```

In the test fixture class, firstly there are Setup() and TearDown() functions including the preparation & cleanup operations for the unit level test.

Secondly it needs include a set of isolated test cases. Each test case will then call into its correspondent in the **test class** *Test_A*, to deliver the real test.

The **test class** *Test_A* is inherited from the **class under test** class A and will provide the real test implementation, including the parameter & logic check. It will be called by the test fixture.

Meanwhile, to achieve the conditional coverage for Class A, we also need a **mock class** for its dependent class *ClassC* which looks like below:

```
class C
{
   ......
   .......
   int variable1;
   int variable2;
   .......
   .......
};
class MOCK_C : public C
{
   .......
   .......
   void SetVariable1();
   void SetVariable2();
   .......
   .......
};
```

From the above description, we can see most of the above code snippets are similar & repetitious. The only exceptions could be the test fixture methods, Setup() and TearDown() which may require some case by case customizations.

Based on above patterns, ULT test code can be easily auto generated from the source code of driver under test.

The high-level blocking diagram for Auto ULT test code Generation can be shown by Figure6 below.

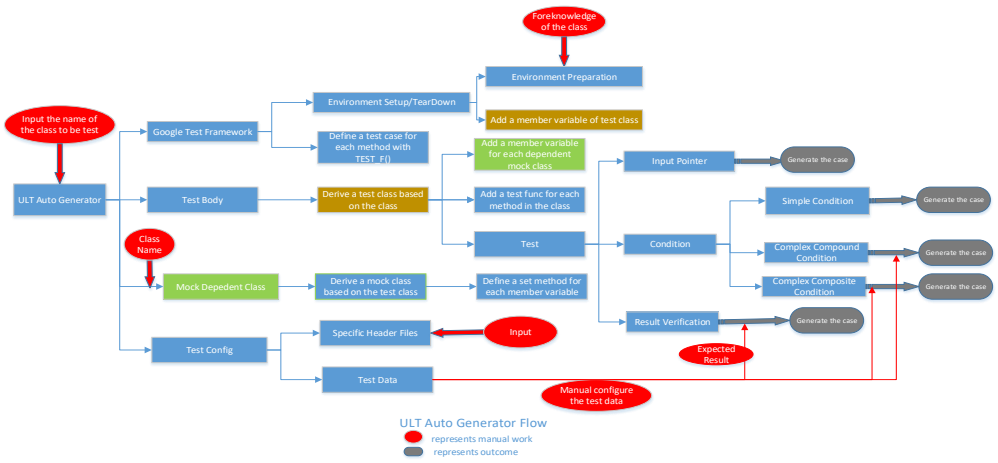


Figure 6. High level blocking diagram for Auto ULT test code Generation

Based on above design, now >80% test code (~400K LOC#) for Intel Media driver ULT can be auto generated by Auto ULT Framework.

2.2.2. Auto Test Case Generation

Test case design is the most important portion in ULT implementation which we believe deserves more developer involvement. Our strategy on media driver test case design is to have developers focusing on the core logic's test case design for functionality assurance, while try to leverage automation on robustness check for those "else" paths.

In Media driver ULT development, functionality test cases for core logic mainly come from the manual effort. Developers is allowed with the explicit control on interesting members of the input parameters through some pre-defined configuration files, and the Auto ULT framework will help on corresponding initialization code generation in test class/mock class based on the settings in configuration files.

Robustness assurance is another important goal for our ULT. Unlike functionality, robustness is more decided by the code quality of so-called "else" paths, and effective robustness check usually requires higher ULT coverage till condition/decision level. Also, usually robustness check just needs ensure no crash or assert been triggered during the test and does not require any reference. All above specialties make robustness test cases more suitable to be auto generated.

In Auto ULT framework, to promise the sufficient condition/decision coverage, Auto ULT Framework will try to add robustness test cases based on the following policies:

<1> For those conditions decided by input parameters, auto ULT framework will leverage the fuzzing to generate the input parameters for corresponding test cases.

<2> For those conditions decided by return value of the nested function call, auto ULT framework will generate the required fake return inside the mock classes.
For Robustness check, currently we use the condition/decision coverage as the metrics to measure the quality for auto-generated test cases. We believe as long as we got sufficient coverage on conditions/decisions, most of the robustness issues should be well exposed.

Now in total we can have ~35% test cases (~7K cases) auto generated by the auto ULT framework requiring almost no extra changes from developers.

SUMMARY

Unit level test is an important approach to reduce the defect and improve the software quality. For Intel Media driver, we have the ULT coverage target of 100% functional coverage and >70% conditional coverage. With the help from ULT, in 2018, Intel Media driver achieved 0 Quality Events and OEM driver escapes is dramatically reduced by 36%.

But manual ULT development can lead to big extra development effort. To better offload developer from the tedious ULT development task and focus on the real test case design/verification, we designed the Auto ULT framework. It can help our developer easily identify the coverage gap based on bug trend and automatically generate >80% ULT code (~400K Lines of test code) as well as ~35% test cases (~7K test cases).

With the help of Auto ULT framework, our average ULT development effort has been greatly shrunk from ~24 Man hours to ~3 Man hours per 1000 lines of driver source code.

ACKNOWLEDGEMENTS

Thanks to all colleagues working on refactoring for continuous software delivery and competency improvement. Appreciate your hard work to turn all our good designs into the reality.

REFERENCES

[1] Kolawa, Adam, Huizinga, Dorota (2007). Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press. p. 75. ISBN 978-0-470-04212-0.
[2] Hamill, Paul (2004). Unit Test Frameworks: Tools for High-Quality Software Development. O'Reilly Media, Inc. ISBN 9780596552817.
[3] Luo Ning, Zhang Livia, Inherent Quality Metrics for Continuous Software Quality Enhancement, International Journal of Software Engineering & Applications (IJSEA)
[4] Hayhurst, Kelly; Veerhusen, Dan; Chilenski, John; Rierson, Leanna (May 2001). "A Practical Tutorial on Modified Condition/ Decision Coverage" (PDF). NASA.

AUTHORS

Ning Luo is the senior software architect at Intel. His research interests include software requirements and architecture, continuous delivery, DevOps, and software product lines. Please contact him at ning.luo@intel.com.

Linlin Zhang is a senior software engineer at Intel. Please contact her at livia.zhang@intel.com

FITCONNECT: AN INTELLIGENT MOBILE APPLICATION TO AUTOMATE THE EXERCISE TRACKING AND PERSONALIZATION USING BIG DATA ANALYSIS

Michael Li[1] and Yu Sun[2]

[1]Northwood High School, 4515 Portola Parkway, Irvine, CA 92620
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

In recent times with the pandemic, many people have been finding exercise as an outlet. However, this situation has made it difficult for people to connect with one another and share their progress with friends and family. This paper designs an application to utilize big data, a social media network, and exercise tracking [1][2]. The program aims to help people connect with others to support one another in their fitness journey. Through various experiments we demonstrated that the application was effective in connecting users with each other and overall improving their fitness experience. Additionally, people of all experience levels in fitness were generally satisfied with the performance of FitConnect, with those of higher experience being less satisfied than those with lesser experience. This application will facilitate getting into fitness through positive means for any person who wants to pursue a healthy lifestyle, whether in the walls of their house, a swimming pool, or a gym [3].

KEYWORDS

Big Data, Social Community, Exercise Tracking.

1. INTRODUCTION

Fitness has consistently remained a prevalent topic because it can better or maintain the health of people. Additionally, exercising serves as an outlet for people to relieve stress and get away from the tough things in their life. With the situation of the 2020 pandemic creating a nerve-racking environment, fitness has become increasingly popular across the world as a way to deal with this stressful situation. However, similar to starting any other kind of hobby, many people may not know where to start in this endeavor. The world of health and fitness is extremely vast, with an overwhelming amount of information that may seem daunting to some, making it difficult to enter this world. In addition, all this information often conflicts with one another over how to carry out the optimal fitness plan through different exercises, training intervals, and more. One of the goals of this application is to improve the accessibility of this knowledge and make it easy for beginners to start. By having a support system of others in the fitness industry, those who are beginners will have an easier time getting started with working out and be more motivated to continue. Additionally, people of all levels will be able to easily access advice and support one another in their fitness journey.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 193-201, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120916

People have used various methods in order to start in their exploration of fitness [4]. Many applications already exist which allow the user to create workout plans and track their previous exercises. They can access charts and data to see their progress of the weight they lifted and the muscle groups used. Additionally, this application allows users to add friends and track their workouts, however, the friend system does not go much further than this, which is one of the first issues. Users cannot communicate with friends or share progress with one another, which makes the entire friend system seem meaningless. Another practical problem is that some users may find it hard to understand certain terms or features upon first using the program, especially for beginners. The database of exercises is so vast, yet it may be difficult for those who are new to fitness. Another issue is that many of the application's useful features are locked and can only be accessed through a subscription. For some this payment may not seem like an issue, but for people who are new to fitness this may act as a barrier to furthering their health. Various experiments were done in order to determine how effective social connections were for fitness. One study designed a mobile game that was based around physical activity to see how users interacted and played in a group [5]. The results discovered that physical activity rates increased by 15% when using the app compared to when the individuals would work out alone. The competitive drive and cooperation between people increased and played a role in increasing exercise.

Our method is a fitness application, Fitconnect, that allows users to connect with other friends and create personalized workout plans. The program uses big data to gather information on the user's friends, workouts, and other information. The data collected by the users is stored inside of a Firebase database. This Firebase features a user authentication system that can allow people to access their accounts safely with email verification. Additionally, the application uses a social community to allow users to view posts and add friends to build a community and help gain support. Ideally, a user would be able to look to their more experienced friends for help by communicating with them to come up with an optimal workout plan or any fitness advice. These features are packed into an easy-to-access system that is extremely navigable. Ideally, if users were having trouble with anything, they could seek help from their friends. While other applications only allow people to collaborate in person, FitConnect aims to allow people to exercise together and gain inspiration remotely as well. Combining these features of different applications into one, it encourages users to exercise more in large groups at a time, as different friends will lift each other up and increase the competitive drive. This app assists users in furthering their fitness journey and keeping them motivated to exercise more, furthering their health and overall well-being, all from the safety of home.

In two applications of FitConnect, we demonstrate how the combination of the features on this program helps to increase the activity levels of users. First, we can show the usefulness of our approach by accessing various data from the Firebase as well as the number of interactions of users on the application. For instance, we can collect data of the number of workouts from individual profiles of users. Additionally, we can utilize the amount of activity on the app, specifically the total time on the app as time progresses. We can also utilize the usage of the social media features, such as the number of friends as the user continues to use the app as well as the number of interactions with these friends. Second, we can use this data and see its progression over time to find various patterns and make a conclusion. For instance, we can analyze if the participants will do more workouts at a higher frequency and for a longer time over a period of a few weeks. Alternatively, we can survey various users based on different exercise experience levels to provide feedback on how effective FitConnect was in facilitating fitness. Finally, we can compare our analysis to other fitness-related studies and applications to see if Fitconnect is the best method.

The remainder of the paper is organized as follows: Section 2 will give the details of the various obstacles and challenges that we faced while designing the experiment and carrying out this research. Section 3 will then talk about the way we solved these problems and overcame the challenges aforementioned in Section 2. Section 4 discusses the experiment and its findings, and after it will analyze the results to place it into the bigger picture of things. Section 5 follows and will present related research and work done in this field, and Section 6 will give the final comments and point out the future of this particular research.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. User doesn’ t use the social media feature

While the social media feature is a large part of the application, some users may not necessarily use that aspect. Instead, they may only use the app to create a workout plan and keep track of their fitness journey by themselves. The question now for this situation is what do we do with the data of this user? Since the experiments are based on how effective the integration of social media is in encouraging people to exercise more, all the data from this user is essentially pointless. The application aims to lessen or prevent this scenario from occurring by making it easier to connect with others and add friends on the app. We intend to allow users to more efficiently use the features of the app to encourage interaction [6].

2.2. Keeping the users motivated

Often, with fitness applications and many other apps in general, one of the standing issues is retaining user interest and keeping them motivated [7]. Some companies try to “gamify” their application, using points and reward systems to keep the users inspired. However, these attempts often fail in keeping interested. The best method in keeping the attention of people is by having change, constantly having users ride the high of something new. In the case of our application, changing the app is difficult because it may affect the outcomes of our data, and it is hard to manage with all the other aspects of the research paper. The social media function aims to keep users’ attention with its constantly changing posts and addition of new friends without drastically changing the app and affecting the experiment.

2.3. Managing the data in an efficient manner

Another challenge lies in dealing with the data in an efficient manner as more users begin to join and create workout plans. Each user has hundreds of different data points stored in the database from login information, workout plan, split days, friends, posts created, the likes/comments on those posts, and more. As the research continues, more and more people will join the app, and the current users will continue to add to the app through their workouts. The database can quickly get flooded with data, and it will be difficult to manage all of this in an optimal way. By organizing the types of data into different classes and even having different sub classes, we can extract the data points necessary for a certain experiment, no matter how many users there are. There may also be an issue with the stability of the firebase, depending on the popularity and usage of the app itself.

3. SOLUTION

FitConnect is a fitness application that allows people to create and track their workouts as well as adding and interacting with friends on the app [8]. This friend building is done through a social media system that allows users to interact with other users through chatting, sharing, and posting [9]. FitConnect has three components that provide users with an efficient performance: the accessible front end design of the application, the Firebase database, and the user authentication system [10]. The front end application is divided into four different parts to make it easily accessible. The home page contains posts from friends, the plan page allows the user to view and edit their custom-made workout plan. Any previous workouts on past days will be displayed and show the various statistics of the workout, such as the weight lifted, time ran, and repetitions. The friends page allows users to search for a certain friend or add new ones. Finally, the profile page allows users to edit their own social media profile and create their own posts. The Firebase writes all the data that users input such as their workout plans, user information, and more straight into the database. The information is separated into different classes and maps to allow for easy access. Firebase also contains a user authentication system which verifies the identity of the person logging in, protecting the user's account information. The following sections will go into detail on the various components of FitConnect and how they transfer data with one another.

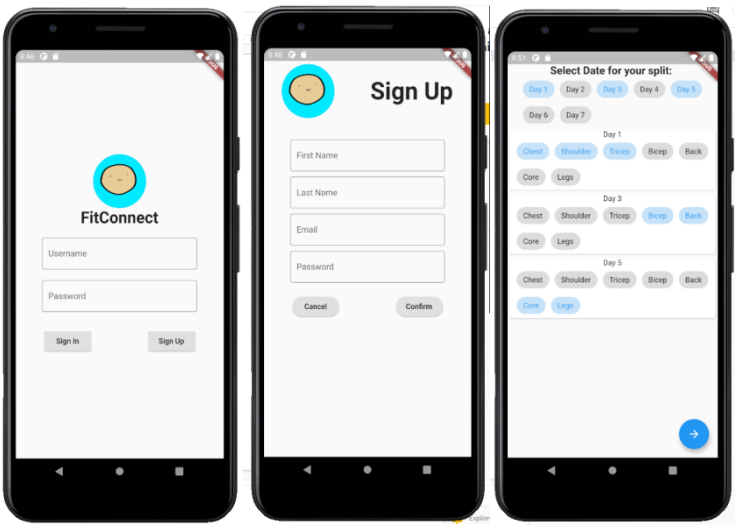


Figure 1. Account Creation

Upon first opening the app, users will be given the option to log in or create a new account. When creating an account, they will be used to provide an email for the authentication system, input various information about themselves, and create an initial workout plan, which can later be edited.

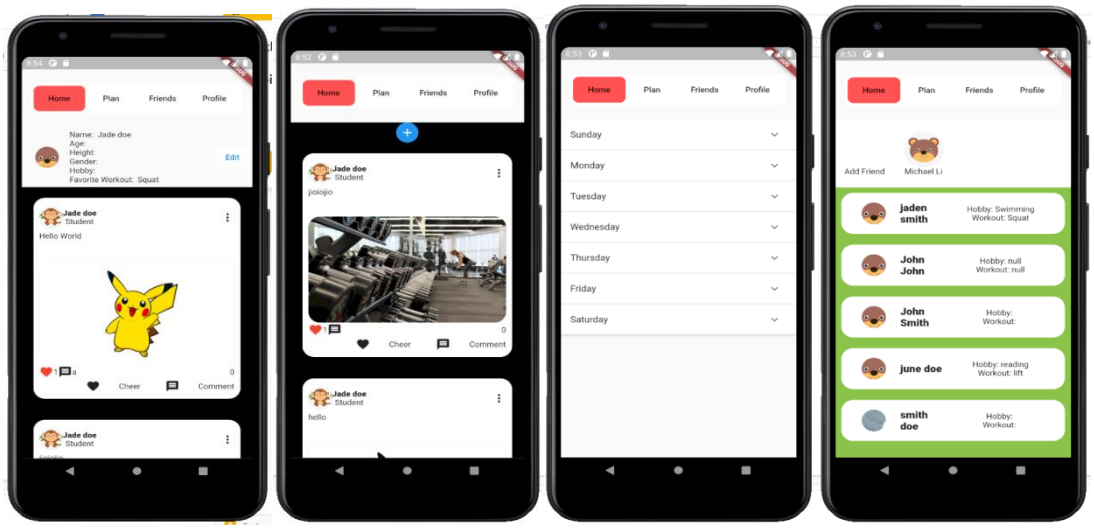


Figure 2. Application Pages

Once logged in, the home page will first appear, with a navigation bar at the top separated into four different pages: Home, Plan, Friends, and Profile. The Home plan contains all posts from friends added, the plan displays the users' previously created workout plan and allows users to start an exercise session, users can interact with and add friends on the Friends page, and the Profile page contains the users' own information and own posts. All of the data shown in these pages are stored in the Firebase as well.

```
return ExpansionPanelList(
  expansionCallback: (int index, bool isExpanded){
    setState(() {
      dailyExercises[index].isExpanded = !isExpanded;
    });
  },
  children: dailyExercises.map<ExpansionPanel>((DailyExercise exercise){
    return ExpansionPanel(
      headerBuilder: (BuildContext context, bool isExpanded){
        return ListTile(title: Text(exercise.day));
      },
```

Figure 3. Capture Device Screenshot

FitConnect was created with the software development program Flutter, which uses the programming language Dart. Flutter was used to develop the front end and the back end of the application. Dart has a surplus of tools that allowed us to format the app how we wanted to, down to the specific margins of a button. One of the difficulties we faced when making the front end of the application was designing the plan page, as it was difficult to manage all the different data required for the workouts. However, we found the ExpansionPanelList, which is a useful class that allowed us to format the Plan page in an efficient manner with a drop down menu, as shown in Figure 3.

```
onPressed: () {
  FirebaseAuth.instance
      .createUserWithEmailAndPassword(
          email: emailController.text,
          password: passwordController.text)
      .then((value) {
    FirebaseFirestore.instance
        .collection('userInformation')
        .doc(emailController.text)
        .set({
      'email': emailController.text,
      'firstName': firstNameController.text,
      'lastName': lastNameController.text,
      'password': passwordController.text,
```

Figure 4. Backend #1 - Application Code

The backend of this application consists of the code itself and the Firebase. The code behind different pages or features were separated into different Flutter documents. The code takes the information input by the user and transfers it directly to the database for later use. It can also go through the database, and if given a DocID, can take information from the database for different uses as well. Figure 4 shows the code behind the sign up system of the code.

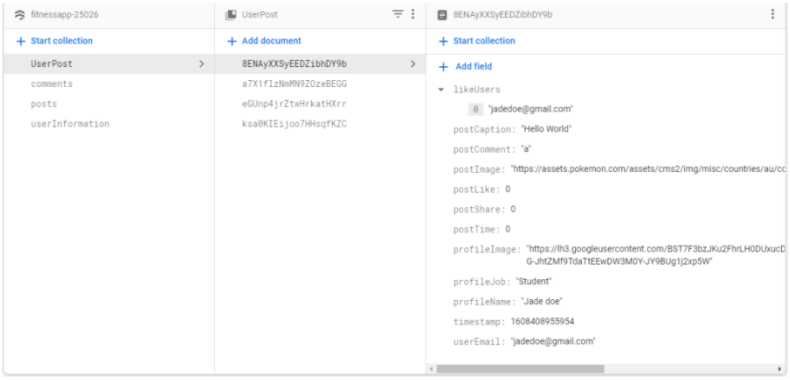


Figure 5. Backend #2- Firebase Organization

The Firebase is separated into 3 parts: Collections, Documents, Fields/New Collection. This particular database is organized into 4 different classes: UserPost, comments, posts, and userInformation. These four classes have different sub categories with various data, all organized meaningfully to provide for efficient access. For instance, the class shown, userInformation, contains all the emails of the users. Each email has a collection containing their workout plan and cumulative workout sessions, while the other field contains all of their information such as their friends, age, name, password, and more.

4. EXPERIMENT

In the following experiments, we will aim to discover if the implementation of a social media feature into a Fitness app will increase a user's activity and exercise levels based on their various interactions upon using the application. Data will be gathered from the user information in the Firebase for various purposes. For instance, the total number of workouts can be used to see if other factors have increased the exercise time of the users. Additionally, the interactions with the social media system such as the number of posts, number of friends, and the time of a post can be used to determine certain results.

4.1. Experiment 1

In the following two experiments, a total of ten people were used to determine certain results about the effectiveness of the app. The first experiment conducted was to find how long it took for any given user to create a new account and interact with another user or make a friend. Upon first opening the app, each user was timed by the same person to provide minimal variations and human error. The intention of this test is to see if people are encouraged to interact with users on the app to gain advice or make new friends.

Over twelve trials, the average time it took for a friend to be added was approximately 2:12 minutes, which was much lower than expected. While some participants were much more precarious in selecting which friends to add, others were more accepting and were more eager to connect with others immediately. These results are positive as they show that the social media system of the app is a vital part of the application and allows users to connect with others easily.

4.2. Experiment 2

After the ten subjects created their account and added friends, they were allowed to use the app for two weeks. At the end of the two weeks, each person would give a feedback score on the app on a scale of 1-10 on the app's effectiveness and were surveyed on their overall experience. The group was also separated into three groups, divided relatively evenly (one group having an extra person): those who did not exercise, those who sometimes exercised, and those who consistently exercised.

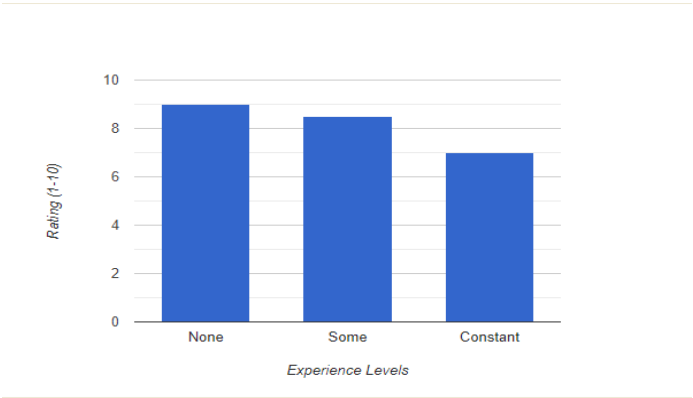


Figure 6. Experiment 1.2 Bar Graph Results

The graph shows that users who had some to no experience in fitness found great satisfaction in FitConnect ' s features, while those who had constant experience were not as satisfied. This chart shows an inversely proportional relationship between the fulfillment of users with their exercise levels and experience. The first two categories reported that the social media feature and workout plan creation served as great tools for building a foundation, as they were able to find a surplus of advice. Those who exercised more reported that various friends and posts sometimes provided small tips, they overall found little use for the social media feature. However, this group was still pleased with the plan creation system.

Overall, it was found that the application' s social media system was effective in encouraging users' exercise levels, although not as much as those who were already more experienced. The social media system allowed people to connect with friends and gain advice. This feature, in pair with an accessible workout plan creator, made the participants' involvement with fitness much

more beneficial. These results line up with our expectations, as we argued that having access to friends in fitness creates a positive environment that can boost motivation. The more experienced subject group already had their own workout plans and a basis in fitness, which gave them less incentive to connect with others seeking advice.

5. RELATED WORK

Wijnand IJsselsteijn utilized a virtual coach in pair with a stationary bike to encourage more exercise at home [11]. The results reveal that there was a positive correlation between immersion with the virtual coach and the motivation/activity levels of the users. The experiments featured a two-by-two design by gathering data with and without the coach as well as a high/low immersion. The two-by-two experiment design is similar to the one we used with either using the social media feature or not using it at all. A difference between Ijsselsteijn's method is by conducting the experiment in a laboratory simulated to be like a home environment, rather than the experiment taking place at the participant's actual home. Additionally, the virtual coach differs from this experiment as it is not an actual person one can interact with. This paper demonstrates the connection between an outside source of aid and the exercise levels of people.

Fletcher Lu et. al aimed to reduce obesity in adolescent populations by creating a mobile fitness application in the form of a game [12]. The application used software to track the workouts completed and also added features to allow users to compete and collaborate with other friends. The experiment classified people into underweight, normal, and overweight, using their BMI as means of measurement. Some similarities between this experiment and our experiment was that they tracked the workouts completed for their data and implemented a social feature. A major difference was that the mobile application was a game, which may have motivated the subjects. Additionally, while the game did include a feature to interact with others, it was relatively uniform and not as complex as the social media feature implemented into our application. The results showed that after six weeks, those who were overweight had a decrease in BMI, healthy subjects maintained their BMI levels, and all but one underweight case saw an increase in BMI. Overall, this paper proved that a mobile fitness application that was easy to access helped encourage exercise, but it did not delve specifically into how social interactions between others affected these results.

Sallis, James F. et al. explored how the exercise habits of others and their support could influence one's own exercise habits [13]. By surveying different people about their exercise habits compared to their family/friends, the results showed that exercise habits of friends are associated with the exercise habits of the individuals. Additionally, it showed that high levels of social support were also associated with the exercise levels of the person receiving the support. This experiment was very different from the other related works and our experiment, however, it shows that there is a positive correlation between the social support of others and the exercise habits of individual people.

6. CONCLUSIONS

Fitness is an important aspect of life and can improve someone's overall physical and mental health [14]. The global pandemic has made it difficult for people who enjoy fitness to workout again and has also inspired new people to start exercising at home. FitConnect provides a solution for those who want to remain safe while still having the option to interact with others. By giving more people motivation to workout, people can greatly benefit in many aspects in life. Not only will they be able to have a more healthy lifestyle, they will also find new friends and establish meaningful relationships along the way [15].

The application is still in its developing stages and has numerous limitations. One limitation is that there are not many intricate features within the social media system. Additionally, the data used for experiments was not on a large scale, so in future experiments, I would like to have more participants to further improve the application.

If given more time to work on this project, I would like to further develop the social media system and make it more complex. For instance, adding a chatting option between friends would significantly improve interactions between people on the app. Additionally, a machine learning system that would recommend friends based off of current ones would also bring people with common interests together.

REFERENCES

[1] Sagiroglu, Seref, and Duygu Sinanc. "Big data: A review." 2013 international conference on collaboration technologies and systems (CTS). IEEE, 2013.
[2] Wakefield, Robin, and Kirk Wakefield. "Social media network behavior: A study of user passion and affect." The Journal of Strategic Information Systems 25.2 (2016): 140-156.
[3] Thoday, John M. "Components of fitness." Symposia of the society for experimental biology. Vol. 7. No. 9. New York: Academic Press, 1953.
[4] Ariew, André, and Richard C. Lewontin. "The confusions of fitness." British Journal for the Philosophy of Science 55.2 (2004).
[5] Blair, Steven N., et al. "How much physical activity is good for health?." Annual review of public health 13.1 (1992): 99-126.
[6] Pike, William A., et al. "The science of interaction." Information visualization 8.4 (2009): 263-274.
[7] Dörnyei, Zoltán, and Ema Ushioda. Teaching and researching: Motivation. Routledge, 2013.
[8] Calder, Bobby J., Lynn W. Phillips, and Alice M. Tybout. "Designing research for application." Journal of consumer research 8.2 (1981): 197-207.
[9] Gundecha, Pritam, and Huan Liu. "Mining social media: a brief introduction." New directions in informatics, optimization, logistics, and production (2012): 1-17.
[10] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 51-71.
[11] IJsselsteijn, W. A., et al. "Virtual Cycling: Effects of immersion and a virtual coach on motivation and presence in a home fitness application." Proceedings Virtual Reality Design and Evaluation Workshop. 2004.
[12] Lu, Fletcher, Kei Turner, and Bernadette Murphy. "Reducing adolescent obesity with a mobile fitness application: study results of youth age 15 to 17." 2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013). IEEE, 2013.
[13] Sallis, James F., et al. "The development of scales to measure social support for diet and exercise behaviors." Preventive medicine 16.6 (1987): 825-836.
[14] Ohrnberger, Julius, Eleonora Fichera, and Matt Sutton. "The relationship between physical and mental health: A mediation analysis." Social science & medicine 195 (2017): 42-49.
[15] Stipek, Deborah. "Relationships." Educational leadership 64.1 (2006): 46-49.

AN INTELLIGENT ALARM CLOCK SYSTEM BASED ON BIG DATA AND ARTIFICIAL INTELLIGENCE

Leon He[1] and Ang Li[2]

[1]La Canada High School, 4463 Oak Grove Drive, Irvine, CA 92604
[2]California State University,
Long Beach, 1250 Bellflower Blvd, Long Beach, CA 90840

ABSTRACT

Sleep is a crucial part of a person's daily routine [1]. However, oversleeping is often a hindrance to many people's daily life. This paper develops an application to prevent people from oversleeping or falling back to sleep after snoozing the alarm. We applied our application to fellow students and conducted a qualitative evaluation of the approach. The results show that the application improves the chances of waking up to a significant degree.

KEYWORDS

Machine Learning, Recommendation System, Data Mining.

1. INTRODUCTION

In the current society, high expectations and workloads are held towards youngsters [2]. Students and others with a cluttered regular schedule suffer from pressure and deadlines [3][4]. Luckily, sleep can relieve such stress, improve people's mood, reduce the risk of diseases, plus other benefits like improving memory and productivity. After an exhausting day, a good night's sleep would be perfect for recovery. But sometimes, when people resort to sleeping, they snooze for a bit too long [5]. This phenomenon occurs in almost every single person's life, whether they forgot to set an alarm before falling asleep, if they are simply too tired after a long day of work and remain in deep sleep instead of waking up, or if they are just reluctant to wake-up even if the alarm rings off [6][7]. They oversleep, Oversleeping is also likely to cause depression and nausea.

Some existing methods are physical alarm clocks or applications on phones. Other methods include lighting or even smell. These methods rely on sound to stimulate people's senses. The main issue is the lack of sophisticated interactions that engages users as a process of waking up. Sure most of the time a distorted sound or emitted fragrance could wake people up but it is inevitable that someone just falls back asleep immediately right after responding to the device. Sometimes, people even forget to set up alarms before they decide to take a snooze.

When the alarm goes off, we are able to subconsciously hit the snooze button without even being fully conscious. To solve these issues, Interactive Alarm implements a system to require users to complete customisable tasks before being able to turn off the alarm while mimicking a calendar format. While setting up the alarm, users are able to randomize or customize trivial questions

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 203-212, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120917

that require a certain degree of thinking to answer. These questions will appear after the user hits the snooze button. But if they are not answered, meaning the user has gone back to sleep, the alarm will go off shortly after the questions are not answered correctly or left blank. This process will repeat until all the questions are answered correctly. To ensure that the user is fully awake after answering the questions, they have the choice to adjust the difficulty and the quantity of the questions. It could range from basic questions like "what times is it?" or thought provoking ones like "what am I waking up for?" Interactive Alarm also features a calendar-like interface where an alarm could be set for multiple days of the week [8]. For example if the user needs to wake-up early on all the weekdays. All alarms can be seen in each section underneath the column for the day of the week. This way, users don't have to remember to turn on the alarm beforehand for regular scheduled alarms.

To prove that the program is stable, we relied on the application as an alarm for 3 weeks. Every single day, the alarm went off at the correct time and we altered between answering the questions and not. For the occasions that we answered the questions, the alarm successfully turned off. For those of which we did not answer the questions, the alarm was able to sound again after a short time. We also recommended this application to several classmates for them to try out. Most of them reported that it was easier for them to wake-up using Interactive Alarm compared to the preexisting alarm applications on their phones.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges thatwe met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. New programming language

To develop this application, we learned a new programming language [9]. In order to use the flutter app we learned about Dart, It has its similarity with a few of the languages we already know such as C++, Java, and other languages used for object oriented programming [10]. But before this, we rarely had any knowledge about graphic user interfaces. This includes learning various emulators, and many new syntax for designing the pages of the application.

2.2. Connecting the user data to online database

Another challenge is connecting the users data to the online database and grouping the user's information under the username. We wanted to incorporate an account system into the application so that users could access all of their alarms on other devices. We had to learn how to Firebase to store user information like the properties of the alarms, the username and password; additionally, figuring out how to access and update pre-existing data when users want to change or delete alarms or if they log in on another device.

2.3. Sitting for hours to debug

The hardest challenge was having to sit down for hours debugging the program for tons of errors.

While syntax errors were fairly obvious, logical errors required additional conditional testing to locate and fix. The biggest issue was the inconsistency in the errors. For example, the application would occasionally not switch to the screen containing the questions after pressing the snooze button. It all came down to patience.

3. SOLUTION

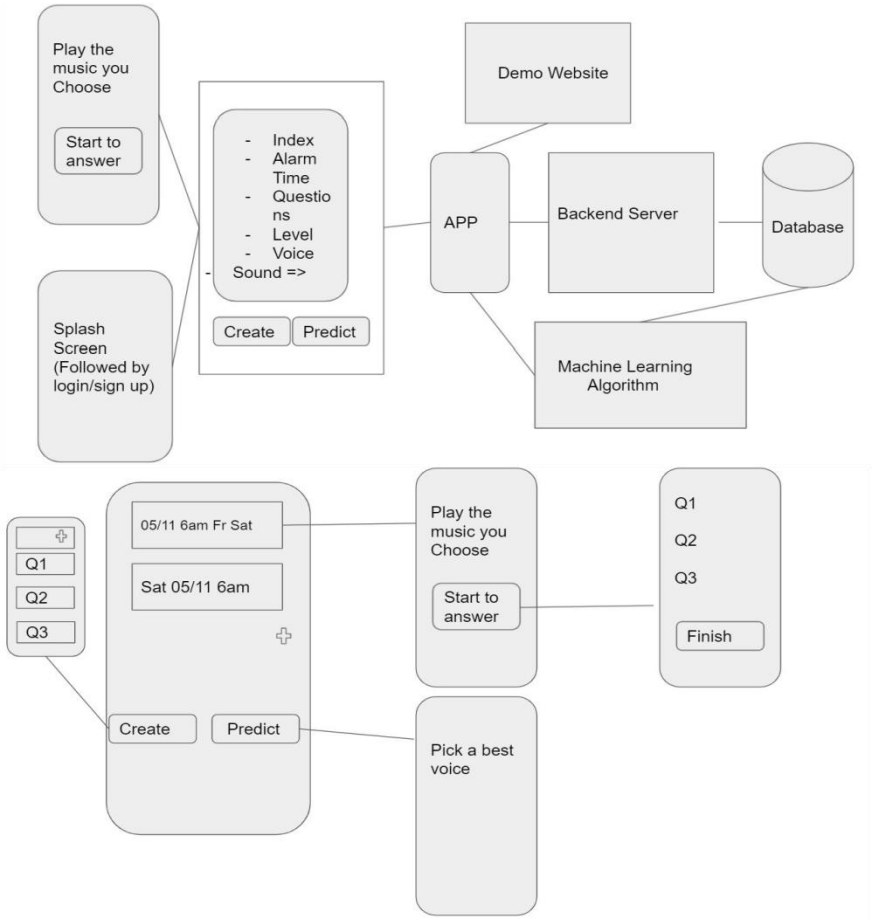
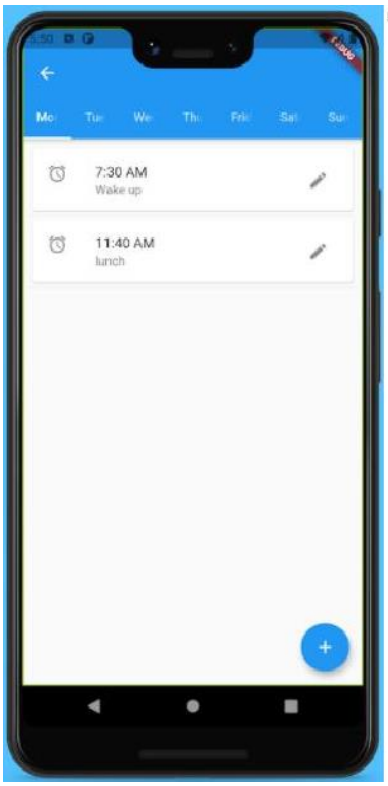


Figure 1. Overview solution of Interactive alarm app

Interactive Alarm is an alarm app with an account system that stores the alarms of users in an online database. After logging in or creating an account, users can create change or delete alarms. Properties of an alarm include ringtone, index/name, time, corresponding question(s), preset volume, and days of the week. Each alarm could be set to go off on selected days of the week. The home page offers tabs for the days of the week so that the user can get a view of all the alarms on a certain day in chronological order. There will also be a demo website for interested people. We also plan on implementing machine learning to predict the ideal volume/questions difficulty needed to wake the user up. Those predictions based on pass data will also be conveniently stored in the online database under each user's information.

The interactive alarm app contains diverse screens. A home screen that contains the list of alarms, an set alarm screen to set up the alarms and the awake/answer screen to answer mathematical or customized questions.

As shown in Figure 2, the home screen contains a tab menu for the day of the week. Each tab contains the list of alarms that is set up for the day of the week. This list shows the time and the description/title of the alarm and the option to edit the alarm. To develop this feature, we use the TabBarView widget that is provided by the material flutter package, an instance of a list for the week day and the alarm information that is saved in the firebase database if it exists. Since there is some day of the week that does not have any alarm set up, we implemented an if-else condition to show the list of the alarms if it exits in the database otherwise the tab shows the message that the "no alarm is set".




```
body: TabBarView(
  children: myTabs.map((Tab tab) {
    final String label = tab.text.toString();
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: Center(
          child: alarms.containsKey(label)
              ? ListView.builder(
                  itemCount: alarms[label].length,
                  itemBuilder: (BuildContext context, int index) {
                    String key = alarms[label].keys.elementAt(index);
                    return new Column(
                      children: <Widget>[
                        Card(
                          child: new ListTile(
                            trailing: IconButton(
                              icon: Icon(Icons.edit),
                              onPressed: () {
                                Navigator.push(
                                    context,
                                    MaterialPageRoute(
                                        builder: (context) =>
                                            NewAlarmPage(
                                              title: 'set alarm',
                                              time: key,
                                              alarm: alarms,
                                            ))).then((value) {
                                .
                                .
                                .
      },
    )
  : Text('no alarm is set')),
```

Figure 2. Home Screen User Interface and implementation

In the Set alarm screen, users add/select the title/description of the alarm, time, a single or multiple day of week and the ringtone. (see Figure 3) Also, they can use the mathematical questions that are provided by the app or use their own questions. The user can select between 1 to 5 questions, so if a user uses an alarm to wake-up and always oversleep then the user can select more than one question to help him/her to wake-up.

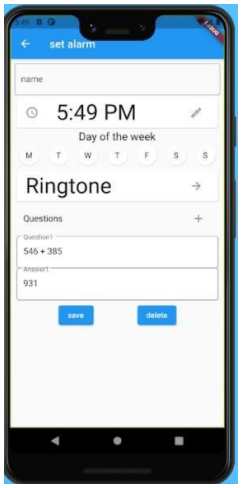


Figure 3. Set up alarm Screen

To automatically generate mathematical questions, we use the random function to select 2 numbers between 1-1000, and add them together to create the addition equation.(see Figure 4) After users select and add the alarm information, the information is sent to the Firebase. We use the Firebase Realtime Database to store the information as a Map structure. (see Figure 5) To develop this feature, we created a Map that contains the name, volume, ringtone, list of questions and answers and time as keys with their respective values. Then the Map is sent to the database and stored in the corresponding user.

```
randomQuestion() {
  int a = Random().nextInt(900) + 100;
  int b = Random().nextInt(900) + 100;
  return [a, b, a + b];
}
Future<Map<String, dynamic>> setalarm() {
  setState(() {
    save = true;
  });

  Map<String, dynamic> alarms = {};
  print(ringtone + volume.toString());
  for (int i = 0; i < values.length; i++) {
    if (values[i] == true) {
      alarms[days[i]] = {
        time: {
          'name': name.text,
          'sound': ringtone,
          'volume': volume,
          'questions': getQuestion().join(','),
          'answers': getAnswer().join(','),
          'id': (DateTime.now().millisecondsSinceEpoch/1000).round()
        }
      };
      database.update(
          'Users/${user.info['username']}/alarms/${days[i]}/$time/',
          alarms[days[i]][time]);
    }
  }

  return Future.delayed(const Duration(seconds: 2), () => alarms);
}
```

Figure 4. Set alarm screen Code

To play the ringtone in the background and schedule alarms, we use flutter_local_notifications and android_alarm_manager packages respectively. As shown in Figure 5, to set up the notification, we set the priority and importance fields to be higher, so the notification shows immediately at the time that is set up. The notification shows the time and the title/description of the alarm, and sound setting as "payload". When users click the alarm notification, the mobile screen is redirected to the screen to answer the questions with the corresponding alarm information. To set up and schedule alarms, we use the android_alarm_manager package to fetch the data every 24 hours from the Firebase and schedule the alarm for the corresponding day of week.

```
static Future showNotification({
  int id = 0,
  String? title,
  String? body,
  String? payload,
}) async =>
    _notification.show(id, title, body, await notificationDetails(),
        payload: payload);

static notificationDetails() {
  return NotificationDetails(
    android: AndroidNotificationDetails(
      'channel id',
      'channel name',
      //'channel description',
      importance: Importance.max,
      priority: Priority.max,
    );
```

```
static Future<void> _setAlarmPeriodic() async {
  await AndroidAlarmManager.periodic(
    Duration(hours: 24),
    // Ensure we have a unique alarm ID.
    0,
    callbackFirebase,
    exact: true,
    wakeup: true,
  );
  callbackFirebase();
}

void listenNotification() {
  print('listen');
  // NotificationService.onNotification.stream.listen(onClickedNotification);
  NotificationApi.onNotification.stream.listen(onClickedNotification);
}

void onClickedNotification(String? payload) {
  print('on clicked');
  List payloadList = payload!.split(",");
  player.stop();
  Navigator.of(context).push(MaterialPageRoute(
      builder: (context) => alarmRing(
            title: 'set alarm',
            time: payloadList[2],
            weekday: payloadList[1],
            username: user.info['username'],
            fileName: payloadList[3],
            volume: double.parse(payloadList[4]),
          )));
}
```

Figure 5. Set notification code

4. EXPERIMENT

In order to verify that our solution can effectively solve problems at different levels and have good user feedback, we decided to select multiple experimental groups and comparison groups for several experiments. For the first experiment, we want to prove that our solution works stable and continuously, so we choose a group size of 40 different trials in 4 different math problems. The 4 different types of problems are add operations, sub operations, multiple operations, divide operations. The goal of the first experiment is to verify if the Calculation System works good for different types of operations of math problems. Through sampling 4 groups of operations of math problems ask the same person to finish all these tasks with the schedule of our app. Result is collected by statistics if the app releases the clock once the user answers the problem correctly. Experiments have shown that all operations in different types unlock the screen and stop the clock. Add operations has the most correct rates, which means our user are works more better in Add operations. This experiment could explain that the operation types do have a obvious impact on the arrange results. The average using time (in minutes) of 4 different types of the operations shows below:

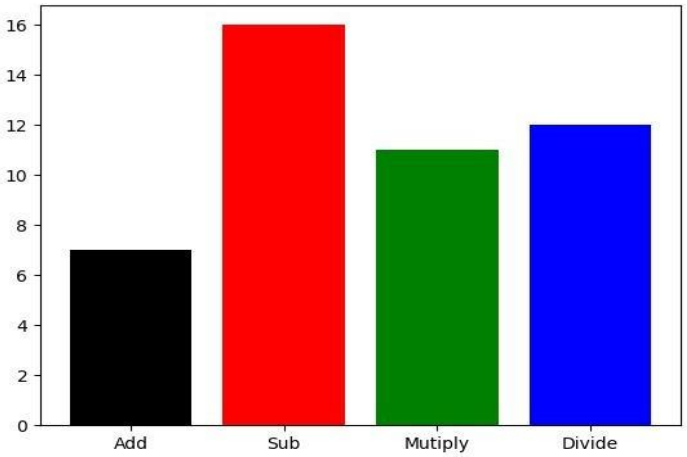


Figure 6. Result of experiment 1 (1)

A good user experience is as important as a good product. So a perfect solution should have excellent user experience feedback.In order to prove that our solution has the best user feedback, we specially designed a user experience questionnaire base on the US system usability questionnaire rules. We statistics the feedback result from 100 users, Track the user's data for 5 days, let them explore freely on the functionality. we divide those users into Five different groups. The first group of users ages from 10 - 20, the second group of users ages from 20 - 30, the third group of users ages from 30 - 40, the fourth group of users ages from 40 - 50, the fifth group of users ages from 50 - 60The goal of the first experiment is to verify high feedback scores shows high performance We collect the feedback scores form these 5 different group of users and analyze it.Experiments have shown that users who ages from 10 - 20 give the highest result feedback to our app. Which may because of the age between those range are more likely to have trouble with get up from bed. The experiment graph shows below:

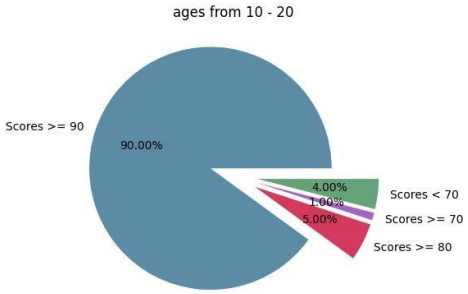


Figure 7. Result of age 10-20

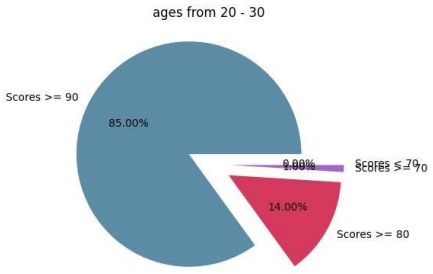


Figure 8. Result of age 20-30

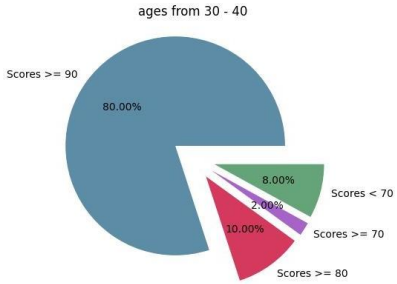


Figure 9. Result of age 30-40

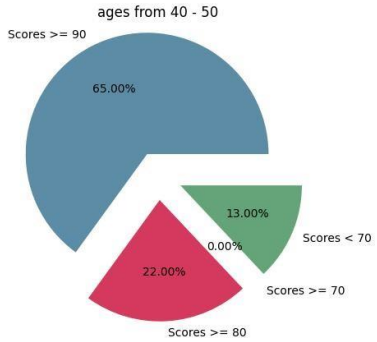


Figure 10. Result of age 40-50

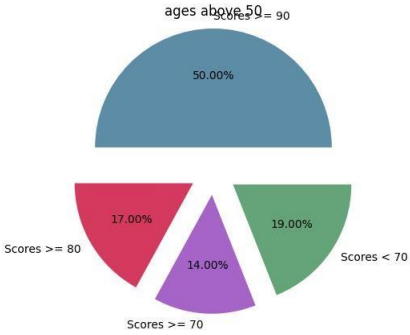


Figure 11. Result of age above 50

5. RELATED WORK

Oh, et al.perform and analysis of a Wake-Up Task-Based Mobile Alarm App [11]. In this analysis, they compare different features: touching a button, taking a picture, shaking the device, and solving math problems at the time that the alarm fires. The analysis shows that users who set harder tasks, math or picture tasks, do not fall asleep again as compared to the normal or easy tasks that alarms snooze multiple times. Similar to this mobile app, we use math equations to help users to be aware and not fall asleep after the alarm is fired.

Fagerjord presented a study of mobile applications [12]. The author says that we are moving to a sensor era and the mobile app would not diminish the access to the Web since mobile apps need access to the web as part of their important feature. One of the mobile apps that the author describes is the Sleep cycle that utilizes touch screen and the accelerometers to calculate the sleep cycle and matching sleep cycle to wake-up time. As different from our app, we do not compare the desired wake-up time with the average sleep cycle to wake-up users; we use the

questions at the time that the alarm fires, so the users would be more aware and not fall asleep again. Ilhan, et al. presented a study of using Gamification in mobile applications related to subjective well-being and sleep/wake activities [13]. The study shows that Gamification can motivate people to begin their day and improve their sleep-wake behaviors.

6. CONCLUSIONS

In this project, we proposed an Android mobile application that helps people to wake-up and not fall asleep again [14]. The interactive alarm app provides a wake-up task that involves mental efforts. Users must solve mathematical equations or answer their customized problems and/or questions in order to stop the ringtone otherwise the alarm would fire again until all the questions are responded. To develop the mobile app, we use Firebase realtime database to store alarms for each user and flutter_local_notifications and android_alarm_manager packages to play the ringtone in the background and schedule alarms.

Some limitations that we encountered during this study is that we could only develop the interactive alarm app for only Android devices since some of the flutter packages can be only set up in Android devices. Moreover, we had a time limitation,so we could not develop the feature in which users can include their own ringtone for their alarm [15].

As for future work, we desired to build and deploy the interactive alarm in the IOS platform since some of the features that we use in our app are only for the Android platform. Also, we would like to add a feature where users can input their own ringtone instead of using the ringtone provided for the app since we believe that adding the customized ringtone can help users to wake-up easier.

REFERENCES

[1] Ferrara, Michele, and Luigi De Gennaro. "How much sleep do we need?." Sleep medicine reviews 5.2 (2001): 155-179.
[2] Murphy, Joseph F., et al. "Academic press: Translating high expectations into school policies and classroom practices." Educational Leadership 40.3 (1982): 22-26.
[3] Dignum, Frank, et al. "Meeting the deadline: Why, when and how." International Workshop on Formal Approaches to Agent-Based Systems. Springer, Berlin, Heidelberg, 2004.
[4] Robin, Pierre-Yves F. "Note on effective pressure." Journal of Geophysical Research 78.14 (1973): 2434-2437.
[5] Bryant, Penelope A., and Nigel Curtis. "Sleep and infection: no snooze, you lose?." The Pediatric infectious disease journal 32.10 (2013): 1135-1137.
[6] Izadi, Iman, et al. "An introduction to alarm analysis and design." IFAC Proceedings Volumes 42.8 (2009): 645-650.
[7] Meisner, David, and Thomas F. Wenisch. "Dreamweaver: architectural support for deep sleep." ACM SIGPLAN Notices 47.4 (2012): 313-324.
[8] Lines, Lorna, and Kate S. Hone. "Eliciting user requirements with older adults: lessons from the design of an interactive domestic alarm system." Universal Access in the Information Society 3.2 (2004): 141-148.
[9] Iverson, Kenneth E. "A programming language." Proceedings of the May 1-3, 1962, spring joint computer conference. 1962.
[10] Godefroid, Patrice, Nils Klarlund, and Koushik Sen. "DART: Directed automated random testing." Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation. 2005.
[11] Oh, Kyue Taek, et al. "Analysis of a Wake-Up Task-Based Mobile Alarm App." Applied Sciences 10.11 (2020): 3993.
[12] Dieter, Michael, et al. "Multi-situated app studies: Methods and propositions." Social Media+ Society 5.2 (2019): 2056305119846486.

[13] Ilhan, Ayse Ezgi, Bahar Sener, and Huseyin Hacihabiboglu. "Improving Sleep-Wake Behaviors Using Mobile App Gamification." Entertainment Computing 40 (2022): 100454.
[14] Ma, Li, Lei Gu, and Jin Wang. "Research and development of mobile application for android platform." International journal of multimedia and ubiquitous engineering 9.4 (2014): 187-198.
[15] Doz, Yves, and Keeley Wilson. Ringtone: Exploring the rise and fall of Nokia in mobile phones. Oxford University Press, 2017.

INDEXED PARALLEL SPHERE PACKING FOR ARBITRARY DOMAINS

Cuba Lajo Rubén Adrián and Loaiza Fernández Manuel Eduardo

Department of Computer Science,
Universidad Católica San Pablo, Arequipa, Perú

ABSTRACT

Particle packings are used to simulate granular matter, which has various uses in industry. The most outstanding characteristics of these are their density and their construction time, the density refers to the percentage of the space of the object filled with particles, this is also known as compaction or solid fraction. Particle packing seeks to be as dense as possible, work on any object, and have a low build time. Currently there are proposals that have significantly reduced the construction time of a packing and have also managed to increase the density of these, however, they have certain restrictions, such as working on a single type of object and being widely affected by the characteristics of the object. The objective of this work is to present the improvement of a parallel sphere packing for arbitrary domains. The packing to improve was directly affected in time by the number of triangles in the mesh of object. This enhancement focuses on creating a parallel data structure to reduce build time. The proposed method reduces execution time with a high number of triangles, but it takes up a significant amount of memory for the data structure. However, to obtain high densities, that is, densities between 60% and 70%, the sphere packing construction does not overwhelm the memory.

KEYWORDS

Sphere Packing, Geometric Algorithm, Parallel.

1. INTRODUCTION

Things that surround us usually are mostly granular matter, this is a set of solid particles that is widely used in industry and its behaviour is widely studied in physics [1][2]. Granular matter is virtually represented by particle packings. Particle packings are represented by an object filled with particles, where the object is a 3D mesh, this is called a container, and the particles are 3D objects of known volumes, for example, spheres, cubes, tetrahedrons, etc. Within the particles used, the most used are the spheres due to their simple representation of four real numbers, the first three, the centre, referring to its position and the last, the radius, referring to the space it occupies. That is why in this research we work with a sphere packing.

A particle packing seeks to be as dense as possible, work on any object, and have a low build time. The density of a packing is the percentage of the container space occupied by particles, this is also called compaction or solid fraction. Particle packings have two approaches, dynamic and geometric packings, the former focused on physical simulations and the latter on geometric constructions. There are also some particle packings that we will call overlapping packings, these use overlapping particles, that is, they do not take into account their collisions, what they do is accommodate the particles in some containers covering the largest possible space, they are mostly used to detect collisions between objects, these packings are not taken into account in this

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 213-226, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120918

research since they move away from its purpose by not having collisions between particles of the same container.

The shortcomings of particle packing are mostly the use of a single container and the density achieved. The objective of this work is to improve a method that already solves the problems mentioned above, the method on which we worked is the method proposed by Cuba and Loaiza [3], which is a method that works in parallel, but does not count with a wide analysis of said parallelism, in addition to having a high delay with 3D meshes that have many triangles. To solve these problems, a data structure will be used which is also parallelized, and the method will be tested on different graphics cards.

This article is organized as follows: Section 2 presents the best particle packing methods today along with their limitations. In Section 3 the proposed sphere packing method is presented. Section 4 shows the tests performed, as well as their results. Finally, we present the conclusions in Section 5.

2. RELATED WORK

Particle packings are varied and seek to be dense. These can be classified into dynamic and geometric. Dynamic packings use the Discrete Element Method (DEM), they are stable and provide the necessary information to perform physical simulations, however, they take a long time, which restricts the number of particles in the packing. Geometric packings do not take as long as dynamic packings and provide more control of particle distribution in the container.

2.1. Dynamic Packings

These methods seek to carry out physical simulations, for which they work with different forms of particles, in this way they are more similar to reality, encompassing a broader study. By vibrating a container full of particles the density increases, with this in mind it was proposed to use composite cubes with superimposed spheres subjected to mechanical vibration to increase the density of the packing, resulting in a maximum density of approximately 70% in a cylindrical container [4]; tetrahedrons composed of superimposed spheres subjected to vibration were also used, but unlike the previous one, a study of the effects of vibration conditions was made, in this study a maximum density of 74.02% was obtained in a cylindrical container [5].

Cylinders composed of superimposed spheres were also used as particles, in this packing it must be fulfilled that the diameter of the container must be larger than the size of the particle, several filling methods were carried out, where in one of them, specifically drop filling, it was observed that the density of the packing is sensitive to height, this packing reached an approximate maximum density of 55% [6]. As you can see, particles composed of spheres are used, this time it was decided to use ellipsoids that are similar to spheres, in this study by using a horizontal orientation in the particles, that is, the elongated part is horizontal, and by increasing the size of the particles increases the density, something interesting, since it is mostly by reducing the size of a particle that greater density is achieved, the maximum density reached in this study was a density of 70% in a cylindrical container [7].

In particle packings, modifying the size of the particles to achieve a higher density is common, taking it further, it was that Zhao et al. [8] proposed to modify the shape of the particles to achieve a higher density, the particles used in this packing are ellipsoids that change shape, this research had a maximum density of about 80% in a rectangular container, which is very good, however, the shape of the ellipsoid that achieved such compaction is similar to a cube. If a cube

is filled with cubes, it can be intuited that it will be a compact packing. As the particles take different shapes to get closer to reality, Rakotonirina et al. [9] carried out research on this, this proposal joins convex particles to build non-convex particles, the distribution and detection of collisions of the particles are based on the convex particles that form the non-convex particles to speed up the calculation, however, the computational cost is increased as well.

The spheres are widely seen in particle packing, Campello and Casares [10] use the spheres as particles, proposing a layered filling together with a compaction system for the spheres used in rectangular containers, the maximum density was 60%. Dynamic packings when focusing on physical simulations is that they only work with rectangular or cylindrical containers and sometimes both, that is, dynamic packings use at most two domains, this restricts the approach that is desired in this research, which is to cover any domain. Currently there are more dynamic packing, but their results are similar to those mentioned previously, for example, the proposal by Wang et al. [11] that uses octahedrons for vibration experiments.

2.2. Geometric Packings

These methods leave aside physical simulations to focus on the distribution of particles and thus build a dense packing. The most used particles in this type of packing are the spheres. The spheres are widely used, since for all the calculations that involve them, only four numbers must be known, three are their position and the other is their size. These types of packings seek to work in any container. A common and widely used method for geometric packing is the advancing front approach, this generates an initial set of spheres and new spheres are inserted with a strategy based on the previously inserted spheres.

Among the geometric methods, the proposal of Wang et al. [12] stands out, since it is currently a geometric method that plans to be dynamic, this method works in a cubic container, and uses various geometric methods in the construction of a package of non-spherical particles with controllable shapes, this to have realistic particles.

Lozano et al. [13] proposed a method to fill arbitrary containers, this method is based on a 2D advancing front approach using a distance field to achieve contact spheres tangent to the triangles of the container mesh. This method reaches a maximum density of approximately 60% in arbitrary containers.

Li and Ji [14] proposed a method also based on the advancing front approach for arbitrary containers. It is proposed to change the size of the particles while building the packing, the new particles adjust to the previously inserted neighbouring particles according to the trilateration equations. It uses a spatial grid to optimize the detection of particles and accelerate their positioning, reaching a maximum density of 73% in a cubic container.

Weller and Zachmann [15] proposed a parallel method using GPU to rapidly pack spheres, successively inserting spheres of the largest possible size to fit into the empty spaces. However, this method does not display geometric data such as the radii of the spheres or the densities of the packings. Also due to its behaviour, relevant data cannot be input apart from the container mesh. This method was used for object collisions [16].

Cuba and Loaiza [3] proposed a sequential and parallel method using GPU, unlike the previous method, taking into account geometric data such as the radii and densities of the generated packing, this study shows a significant number of results that support their proposal, however, these results have shortcomings such as the time it takes to build the packing when working with

meshes of high number of triangles, and the shortage of tests of the parallel method, so in this research this method is improved and more tests of the parallel methods are carried out.

3. PACKING GENERATION

The proposed method is the improvement of an existing method, this method is the method proposed by Cuba and Loaiza [3] called Parallel Sphere Packing for Arbitrary Domains, for which this method will be explained in a simplified way, and then the improvements made will be explained.

3.1. Parallel Sphere Packing for Arbitrary Domains

This method is the method of Cuba and Loaiza [3], which works with four radii, these are called r_{max}, r_{min}, r_{med}, r_{mid}. The positions of the spheres are given by a hexagon of spheres, which can be seen in Figure 1. The input data for this method is the radius r_{max} and the mesh of the arbitrary container. This parallel packing of spheres leaves randomness aside, therefore, it eliminates the need to detect collisions between spheres, it works in two phases, in the first phase a box is filled, the box is a rectangular container that surrounds the 3D object to be filled, in the second phase a verification is performed on all the spheres to see if they are inside the arbitrary container or not, this verification is performed in parallel.

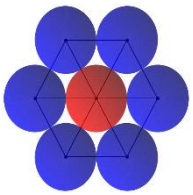


Figure 1. Spheres hexagon [3].

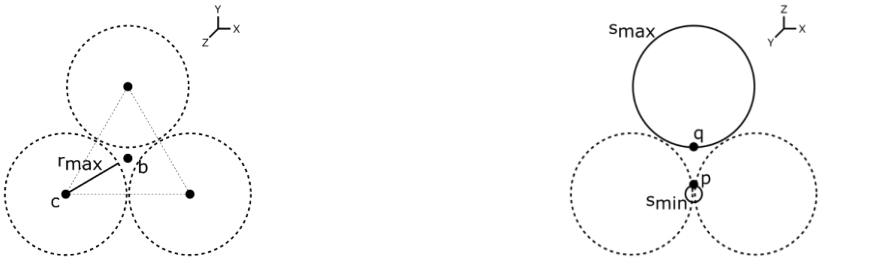
3.1.1. First Phase

First the 3D mesh of the arbitrary container is stored, then the radii r_{min} and r_{med} are calculated based on the input radius r_{max}. To calculate r_{min}, Equation 1 is used, where b is the barycentre of the upper right triangle and c is the centre of the central sphere, this can also be seen in Figure2a. To calculate r_{med} two points are used, r_{max} and r_{min}, the two points are $p = (s_{min}.x, s_{min}.y, s_{min}.z + r_{min})$ and $q = (s_{max}.x, s_{max}.y, s_{max}.z - r_{max})$, where s_{min}represents the sphere with radius r_{min} that must be added and s_{max} represents the sphere with radius r_{max} that only varies its position z in relation to the sphere with radius r_{min}, this can be seen in Equation 2 and Figure 2b. The position of the sphere with radius r_{min} is the barycentre of the upper right triangle. The position of the sphere with radius r_{med} only varies its position on the z axis, therefore, its position on the x axis is the same as that of the x axis of s_{max} and its position on the y axis is also the same as the position on the y axis of s_{max}; its position in z can be seen in Equation 3.

$$\boldsymbol{r_{min} = distance(b, c) - r_{max}} \tag{1}$$

$$r_{med} = \frac{distance(p, q)}{2} \tag{2}$$

$$c_{med}.z = \frac{(s_{max}.z - r_{max}) + (s_{min}.z - r_{min})}{2} \tag{3}$$



(a) Data for r_{min} [3] (b) Data for r_{med} and c_{med} [3]

Figure 2. Data for radii.

In this phase, a rectangular container is filled that surrounds the arbitrary container, the filling of the rectangular container is done in layers starting at the minimum point of this container until reaching its maximum point. The filling is done with a small structure called hepta-sphere, that seen in the xy plane moves in the direction of the positive x axis until reaching the limit and then moves in the direction of the positive y axis until reaching the limit, in this way one layer would be filled, then it moves in the direction of the positive z axis to fill the next layer. In the filling of the enveloping rectangular container, only the centres of the spheres are taken into account for the collisions with the container, this is done to cover a greater possible number of spheres. The hepta-sphere can be seen in Figure 3, the value of their positions is shown below:

$$p1 = (p0.x + 2r_{max}, p0.y, p0.z)$$
$$p2 = (p0.x + r_{max}, p0.y + (2r_{max}sin60°), p0.z)$$
$$p3 = (p0.x, p0.y - (r_{max} + r_{min}), p0.z)$$
$$p4 = \left(p0.x + r_{max}, p0.y - \left((r_{max} + r_{min})cos60°\right), p0.z\right)$$
$$p5 = \left(p0.x + r_{max}, p0.y + \left((r_{max} + r_{min})cos60°\right), p0.z\right)$$
$$p6 = (p0.x, p0.y + (r_{max} + r_{min}), p0.z)$$

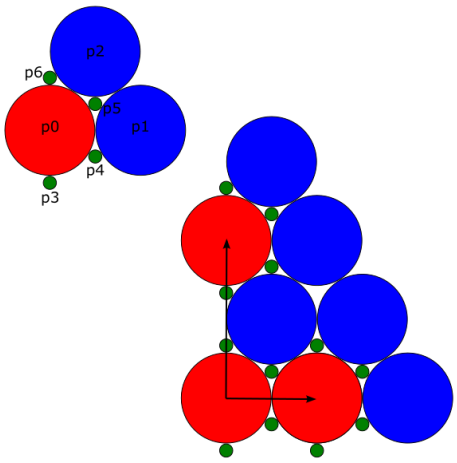


Figure 3. In the left part, a hepta-sphere and in the right, possible movements [3].

The procedure for filling a layer can be seen in Figure 4a, between every three spheres with radius r_{max} there is a sphere with radius r_{min} and two spheres with radius r_{med} as seen in Figure 4b.

(a) One xy layer filling. (b) One min sphere and two med spheres [3].

Figure 4. Data for layers.

3.1.2. Second Phase

In this phase it is verified if a sphere from the previous phase is inside the arbitrary container. For each sphere, it is first verified that the centre of the sphere is inside the arbitrary container, for this purpose raycast is used using the Jimenez et al. [17] algorithm for the intersection of a segment with a triangle, then it is verified if the sphere collides with any triangle of the mesh, for this it is necessary to find the closest point of a triangle, for this the Eberly [18] algorithm is used. If a sphere of radius r_{max} intersects some triangle, it is replaced by 61 spheres of radius r_{mid}, the positions of these spheres and their radius are obtained by recalculating the positions and radius of the dense packing of 61 spheres of Pfoertner [19]. If a sphere of radius r_{med} intersects any triangle of the container's mesh, this sphere is replaced by a sphere of radius r_{min}. Replacements are also checked.

This phase is the part that is parallelized, its form of parallelization is dividing the number of spheres of the first phase, this parallelization was done using CPU threads and GPU threads.

3.2. Method Improvement

As can be inferred from the previous method, the number of spheres is not the only thing to take into account, but also the triangles, since despite being parallel, each sphere is compared with all the triangles, being an operation expensive, then it is planned to reduce the cost of said operation using indexes in a data structure, the data structures to choose were the octree and the uniform grid, finally the last one was chosen as the basis for the data structure to be used due to the amount of memory used, indexes are used so that each sphere is compared only with the triangles that must be compared, the creation of this structure is parallelized to reduce the time as much as possible.

3.2.1. Data Structure

As mentioned above, the data structure used is based on a uniform grid. A uniform grid is a set of boxes of the same size that have things inside. The proposed structure is a matrix in which each value represents the type of intersection between the information belonging to the row and column indices. Boxes will be represented taking their size as data for their creation, their positions will come out of their indices using Equation 4, where $bpos$ is the position of the box on an axis, $cmin$ is the minimum point of the grid on the same axis, idx is the index of the box on the same axis and be is half the distance of the box on the same axis. The number of boxes is given by dividing the size of the enveloping rectangular container by axis by the size of the boxes. In this case we must place both triangles and spheres, so it was decided to have one structure for the triangles and another for the spheres. In the first data structure the rows are the

indices of the boxes and the columns are the indices of the triangles, in the second data structure the rows are the indices of the spheres and the columns are the indices of the boxes.

$$bpos = cmin + \big((2idx + 1)be\big) \tag{4}$$

The data structure for triangles has values 0 and 1, where 0 is that the triangle does not intersect the box and 1 is that the triangle intersects the box. The data structure for spheres has values 0, 1, and 2, where 0 is that the sphere does not intersect the box, 1 is that the sphere intersects the box, and 2 is that the centre of the sphere is in the box, this last value is to avoid calculations, since to use the raycast only the centre of the spheres is used. These values are to reduce time, however, having an array consumes significant memory. For the intersection of a triangle in a box, the Separating Axis Theorem (SAT) algorithm will be used, since there would only be thirteen quick checks [20], for the intersection of a sphere with a box the distances per axis are calculated [21], and for the algorithm of a point in a box a simple check per axis is used. The method for box-sphere intersection has some slight modifications, so it is shown in Figure 5, where bc is the centre of the box, be is half the size of the box, c is the centre of the spheres, and $radius$ is the radius of the sphere.

Data: $bc, be, c, radius$
$bxmin \leftarrow bc[0] - be$
$bymin \leftarrow bc[1] - be$
$bzmin \leftarrow bc[2] - be$
$bxmax \leftarrow bc[0] + be$
$bymax \leftarrow bc[1] + be$
$bzmax \leftarrow bc[2] + be$
$dmin \leftarrow 0$
if $c[0] < bxmin$ **then**
  $dmin \leftarrow dmin + (c[0] - bxmin)^2$
else if $c[0] > bxmax$ **then**
  $dmin \leftarrow dmin + (c[0] - bxmax)^2$
end if
if $c[1] < bymin$ **then**
  $dmin \leftarrow dmin + (c[1] - bymin)^2$
else if $c[1] > bymax$ **then**
  $dmin \leftarrow dmin + (c[1] - bymax)^2$
end if
if $c[2] < bzmin$ **then**
  $dmin \leftarrow dmin + (c[2] - bzmin)^2$
else if $c[2] > bzmax$ **then**
  $dmin \leftarrow dmin + (c[2] - bzmax)^2$
end if
return $dmin <= radius^2$

Figure 5. Box-sphere intersection algorithm.

3.2.2. Sphere Inside the Mesh

That changes with respect to the previous method is the use of data structures, both triangles and spheres, to reduce time in knowing if a sphere is inside the 3D mesh of the arbitrary container. As the initial datum is the index of the sphere, the boxes linked to it are searched for, and with the boxes linked to it, the triangles linked to the boxes are searched, that is, the triangles linked to the sphere are searched for, in this way it is not necessary to perform calculations with all the triangles, however, there is a problem with this, in the raycast algorithm a counter is used per triangle segment intersection and if any triangle is found in more than one box there will be problems since will count more than once, this problem is solved with a revision of the next one, that is, it will look in the next box to check if the index of the compared triangle is also there, if so, the current possible intersection will not be performed, of this way only one calculation will be performed per triangle.

3.2.3. Parallelization

The method parallelization works with the previous method parallelization, and adds a parallelization to the generation of the data structures. The parallelization of the method of Cuba and Loaiza [3], worked by dividing the number of spheres for the calculation of the verification of spheres, since it was what took more time, however, due to the creation of the data structures, this time is reduced. considerably, being the generation of the data structures what takes more time now, for which the generation of the data structures is also parallelized. As the data structures are matrices, they can be treated as arrays that are divided to then find the row and column indices, then with these indices make the intersections with boxes, either triangles or spheres. In this way we have two parallel parts, the generation of the data structures and the verification of the spheres inside the mesh of the arbitrary 3D container. The new parallelization of the method considerably reduces the packing construction time, although depending on the number of boxes the memory can become too high, therefore, the amount of this can be decided in the creation of the packing, that is, as input data we will have not only the radius r_{max}, but also the size of all boxes. Implementations of parallelizations are shown in Figure 6, Figure 7a and Figure 7b, where $tgrid$ is the grid of triangles, $sgrid$ is the grid of spheres, $bsizes$ are the sizes of the grid for each axis, $cmins$ are the values of the minimum point of the grid, $triangles$ are the triangles, $spheres$ are the centres of the spheres, $radius$ is the radius used in $spheres$, bxs_{num} is the number of boxes, trs_{num} is the number of triangles, $sphrs_{num}$ is the number of spheres and $valids$ are the positions of the spheres inside the arbitrary domain.

Data: $valids$, $bsizes$, $cmins$, $spheres$, $radius$, $triangles$, $tgrid$, $sgrid$, trs_{num}, bxs_{num}, $sphrs_{num}$

1: $i \leftarrow threadIdx.x + blockIdx.x * blockDim.x$
2: **if** $i < sphrs_{num}$ **then**
3: $valids[i] \leftarrow SphereInsideMesh(i, bsizes, cmins, spheres, radius, triangles, tgrid, sgrid, trs, bxs)$
4: **end if**

Figure 6. Spheres validation algorithm.

Data: $tgrid$, be, $bsizes$, $cmins$, $triangles$, trs_{num}, bxs_{num}

$bc[3] \leftarrow null$
$tr[9] \leftarrow null$
$tg_{id} \leftarrow threadIdx.x + blockIdx.x * blockDim.x$
$tr_{id} \leftarrow tg_{id}$ mod trs
$bx_{id} \leftarrow tg_{id}/trs$
$xpos \leftarrow bx_{id}/(bsizes[1] * bsizes[2])$
$ypos \leftarrow (bx_{id}/bsizes[2])$ mod $bsizes[1])$
$zpos \leftarrow bx_{id}$ mod $bsizes[2]$
if $bx_{id} < bxs_{num}$ **and** $tr_{id} < trs_{num}$ **then**
  $bc[0] \leftarrow cmins[0] + ((2 * xpos + 1) * be)$
  $bc[1] \leftarrow cmins[1] + ((2 * ypos + 1) * be)$
  $bc[2] \leftarrow cmins[2] + ((2 * zpos + 1) * be)$
  $tr[0] \leftarrow triangles[tr_{id} * 9]$
  $tr[1] \leftarrow triangles[tr_{id} * 9 + 1]$
  $tr[2] \leftarrow triangles[tr_{id} * 9 + 2]$
  $tr[3] \leftarrow triangles[tr_{id} * 9 + 3]$
  $tr[4] \leftarrow triangles[tr_{id} * 9 + 4]$
  $tr[5] \leftarrow triangles[tr_{id} * 9 + 5]$
  $tr[6] \leftarrow triangles[tr_{id} * 9 + 6]$
  $tr[7] \leftarrow triangles[tr_{id} * 9 + 7]$
  $tr[8] \leftarrow triangles[tr_{id} * 9 + 8]$
  $tgrid[bx_{id} * trs_{num} + tr_{id}] \leftarrow$ '0'
  if $IntersectBxTr(bc, be, tr) is$ **true then**
    $tgrid[bx_{id} * trs_{num} + tr_{id}] \leftarrow$ '1'
  end if
end if

(a) Triangles grid generation algorithm.

Data: $sgrid$, be, $bsizes$, $cmins$, $spheres$, $radius$, $sphrs_{num}$, bxs_{num}

$bc[3] \leftarrow null$
$c[3] \leftarrow null$
$sg_{id} \leftarrow threadIdx.x + blockIdx.x * blockDim.x$
$bx_{id} \leftarrow sg_{id}$ mod bxs_{num}
$sphr_{id} \leftarrow sg_{id}/bxs_{num}$
$xpos \leftarrow bx_{id}/(bsizes[1] * bsizes[2])$
$ypos \leftarrow (bx_{id}/bsizes[2])$ mod $bsizes[1])$
$zpos \leftarrow bx_{id}$ mod $bsizes[2]$
if $sphr_{id} < sphrs_{num}$ **and** $bx_{id} < bxs_{num}$ **then**
  $c[0] \leftarrow spheres[sphr_{id} * 3]$
  $c[0] \leftarrow spheres[sphr_{id} * 3 + 1]$
  $c[0] \leftarrow spheres[sphr_{id} * 3 + 2]$
  $bc[0] \leftarrow cmins[0] + ((2 * xpos + 1) * be)$
  $bc[1] \leftarrow cmins[1] + ((2 * ypos + 1) * be)$
  $bc[2] \leftarrow cmins[2] + ((2 * zpos + 1) * be)$
  $sgrid[sphr_{id} * bxs_{num} + bx_{id}] \leftarrow$ '0'
  if $IntersectSphrBx(bc, be, c, radius)$ is **true then**
    $sgrid[sphr_{id} * bxs_{num} + bx_{id}] \leftarrow$ '1'
    if $PointInBox(bc, be, c)$ is **true then**
      $sgrid[sphr_{id} * bxs_{num} + bx_{id}] \leftarrow$ '2'
    end if
  end if
end if

(b) Spheres grid generation algorithm.

Figure 7. Grid generation algorithms.

4. RESULTS

This section presents the tests performed as well as their results, the implementation was done in C++ OpenGL using different hardware to provide a further study of the method. The range of densities to be obtained for each container will be between 60% and 70% based on the method of Cuba and Loaiza [3]. First, it will be shown how the proposed method works compared to the method of Cuba and Loaiza [3], then comparisons of both methods will be made using containers with a high number of triangles, finally the comparison of the parallel methods will be made using different graphics cards. The size used for the box is four times the size of the sphere, this is because less memory is used without significantly affecting time.

4.1. Comparison of Methods Using Containers from Cuba and Loaiza [3]

The method of Cuba and Loaiza [3] is compared with the proposed method in different containers, the data of these containers are found in Table 1. The results of the comparisons are found in Table 2. For this comparison, the algorithms are running on an Intel Core i7-8550U @1.80GHz with 12GB of RAM, 8 threads and an NVIDIA GeForce MX130 graphics card under Windows 11 64bits. The radius r_{max} chosen for these tests is $0.2u$, however, in the Torus and the Stanford Dragon, a density between 60% and 70% was not reached due to the shape of their meshes, for which, the radius was reduced until reaching this density range. In the case of the Torus, the density reached with radius r_{max} $0.2u$ was 59.78%. In the case of the Stanford Dragon, since it is smaller, it was decided to start with a radius r_{max} of $0.1u$ where its density was 53.03%.

Table 1. Containers.

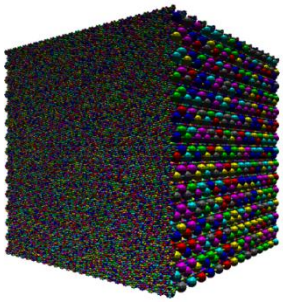
Container	**Measures (u)**	**Triangles**	**Volume (u^3)**
Cube	Width (10), height (10), depth (10)	12	1000.0
Cone	Radius (5), height (10)	62	261.80
Cylinder	Radius (5), height (10)	124	785.40
Capsule	Spherical radius (5), cylindrical height (10)	832	245.44
Sphere	Radius (5)	960	523.60
Torus	Max radius (5), min radius (1.25)	1152	154.21
Stanford Bunny	Domain size ($8.07 \times 8.11 \times 6.17$)	7202	109.90
Stanford Dragon	Domain size ($2.24 \times 3.52 \times 5.00$)	21782	6.95

Table 2. Comparison of times.

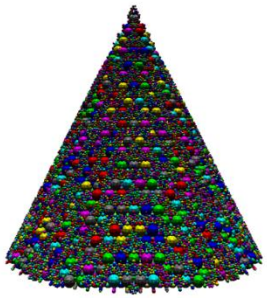
Container	**Radius (u)**	**Density (%)**	**Cuba and Loaiza [3]**		**Proposed Method**	
			Sequential Time (s)	**Parallel Time (s)**	**Sequential Time (s)**	**Parallel Time (s)**
Cube	0.2	71.96	0.07	0.47	2.28	0.81
Cone	0.2	69.34	0.30	0.58	1.56	0.76
Cylinder	0.2	71.11	0.76	0.55	2.42	1.04
Capsule	0.2	72.95	3.06	1.11	0.78	0.82
Sphere	0.2	70.65	8.43	2.01	3.67	1.36
Torus	0.15	63.27	11.44	2.31	4.82	1.89
Stanford Bunny	0.2	64.69	42.97	5.54	4.47	2.24
Stanford Dragon	0.05	64.00	910.12	75.31	78.82	25.35

The results of Table 2 show the advantage of the proposed method compared to the method of Cuba and Loaiza [3] in the cases of the Stanford Bunny and the Stanford Dragon, however, in the other containers, the time differences are small, due to their low number of triangles, so tests are then performed using containers with a high number of triangles.

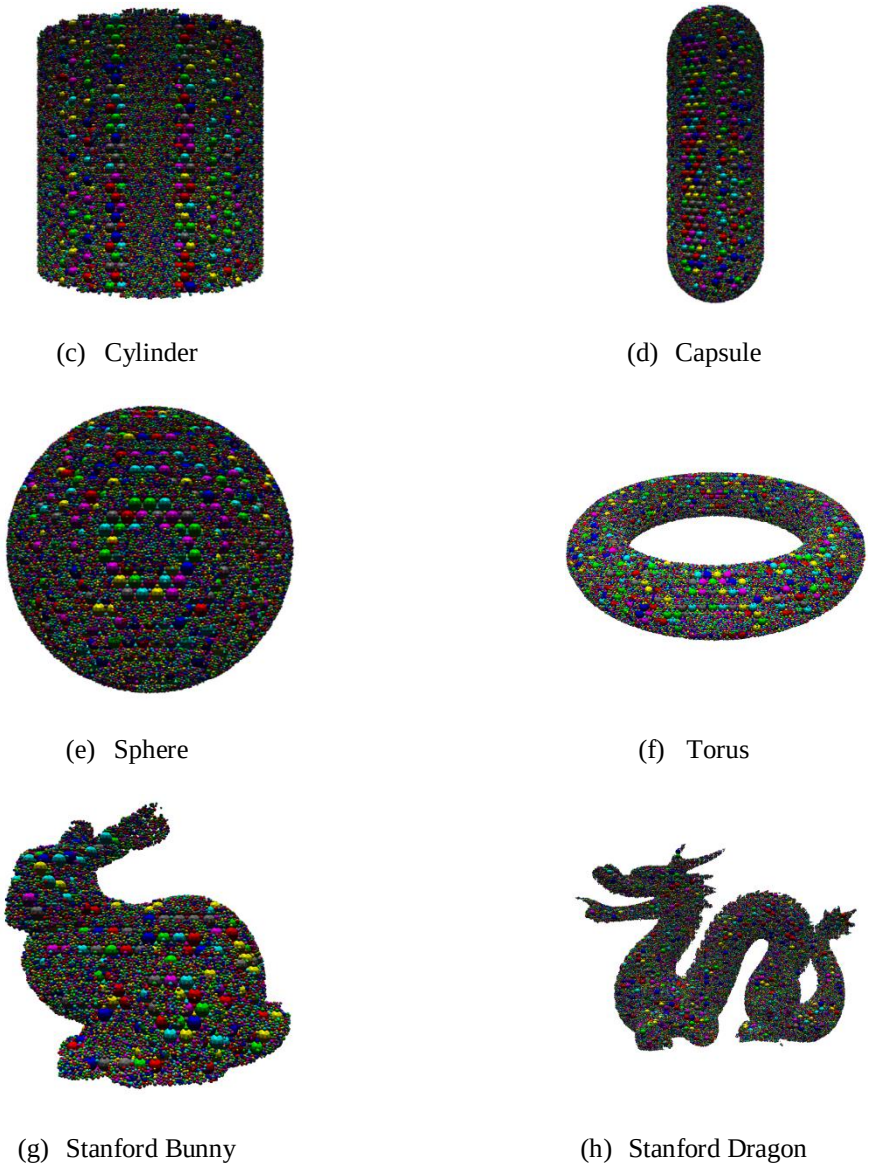
The visual results of the packing made in Table 2 are shown in Figure 8, in these images it is observed that with the densities reached in this table the objects correctly show their shape, including details such as the ears of the Stanford Bunny or the tongue of the Stanford Dragon.



(a) Cube



(b) Cone



(c) Cylinder (d) Capsule

(e) Sphere (f) Torus

(g) Stanford Bunny (h) Stanford Dragon

Figure 8. The proposed method in different containers.

4.2. Comparison of Methods Using Containers with a High Amount of triangles

The method of Cuba and Loaiza [3] is compared with the method proposed in the Stanford Bunny and the Stanford Dragon used previously, but with more triangles, these two models were modified by McGuire [22]. The Stanford Bunny used for this comparison has 144046 triangles and is called Stanford Bunny HT. The Stanford Dragon used for this comparison has 871306 triangles and is called Stanford Dragon HT. This comparison occurs in more powerful hardware since the triangles of each mesh are many. The comparison is made on an Intel Core i7-8700 @3.70GHz with 64GB of RAM, 12 threads and an NVIDIA GeForce RTX 2060 graphics card under Windows 10 64 bits. The results of this comparison are shown in Table 3.

Table 3. Comparison of times in container with high number of triangles.

Container	Radius (u)	Density (%)	Cuba and Loaiza [3]		Proposed Method	
			Sequential Time (s)	Parallel Time (s)	Sequential Time (s)	Parallel Time (s)
Stanford Bunny HT	0.2	64.78	411.39	21.56	47.56	4.70
Stanford Dragon HT	0.05	64.07	17639.96	454.52	1144.77	50.24

The results of Table 3 show a clear advantage of the proposed method compared to the method of Cuba and Loaiza [3] both in its sequential forms and in its parallel forms. To have a deeper analysis of their parallel forms, a comparison of both methods on three different graphics cards is made below.

4.3. Comparison of Parallel Methods Using Different Hardware

A comparison of the parallel method of Cuba and Loaiza [3] and the proposed parallel method is made using the containers of Table 1 and Table 3, these results are shown in Table 4.

Table 4. Comparison of times in parallel methods.

Container	Radius (s)	Density (%)	Times (s)					
			Cuba and Loaiza [3]			Proposed Method		
			Intel Core i7-8550U @1.80GHz, 12GB RAM, NVIDIA MX130	Intel Core i7-8700 @3.70GHz, 64GB RAM, NVIDIA RTX 2060	Intel Core i5-10400F @2.90GHz, 16GB RAM, NVIDIA RTX 3060	Intel Core i7-8550U @1.80GHz, 12GB RAM, NVIDIA MX130	Intel Core i7-8700 @3.70GHz, 64GB RAM, NVIDIA RTX 2060	Intel Core i5-10400F @2.90GHz, 16GB RAM, NVIDIA RTX 3060
Cube	0.2	71.96	0.47	0.13	0.20	0.81	0.21	0.22
Cone	0.2	69.34	0.58	0.15	0.17	0.76	0.21	0.21
Cylinder	0.2	71.11	0.55	0.19	0.19	1.04	0.22	0.24
Capsule	0.2	72.95	1.11	0.28	0.22	0.82	0.17	0.18
Sphere	0.2	70.65	2.01	0.44	0.40	1.36	0.27	0.27
Torus	0.15	63.27	2.31	0.52	0.50	1.89	0.33	0.31
Stanford Bunny	0.2	64.69	5.54	1.28	1.22	2.24	0.45	0.39
Stanford Dragon	0.05	64.00	75.31	11.44	12.14	25.35	1.95	1.65
Stanford Bunny HT	0.2	64.78	97.90	21.56	21.50	38.26	4.70	4.32
Stanford Dragon HT	0.05	64.07	4854.48	454.52	472.89	1602.52	50.24	47.28

The results of Table 4 show that the proposed method obtains densities between 60% and 70% in less than a minute in all the tests carried out, except for the Stanford Dragon HT using the NVIDIA GeForce MX130 graphics card, where this container has 871306 triangles. This indicates that the method works well, since its execution time is very low, except in cases with a very high number of triangles on moderately powerful hardware.

5. CONCLUSIONS

The limitations of the proposed method are aimed at the number of triangles contained in the model and the size of the radius chosen as input, since reducing this radius increases the number of spheres. An excessive number of triangles would saturate the GPU memory and consume more time, the same happens when reducing the input radius too much. This excessive number of triangles and spheres is in the millions. This indicates that the method would fail when constructing a packing of spheres with a density close to or greater than 60% in a highly detailed 3D model.

The proposed method does not affect the use in arbitrary domains or the densities reached by the method of Cuba and Loaiza [3], these densities are between 60% and 70%. In the tests carried out, it is observed that the improvement in time is high both in sequential and in parallel, for which it is considered a successful improvement of the algorithm. The times achieved by the method are relatively low even with a high number of triangles and the memory is not saturated for densities between 60% and 70%. This indicates the effectiveness of the method for arbitrary containers.

6. FUTURE WORKS

As future works, it is planned to create a software for the industry with indications and restrictions of the method so that it can be used in 3D printing. This software will be used in the material reduction of a 3D print, the spheres will be holes in the 3D model that allow to print a 3D model with internal supports. When printing a model, an amount of material of approximately the density percentage of the packaging is saved, that is, when using the proposed method in 3D printing, between 60% and 70% of material will be saved in a model in the internal part. For external supports, it is planned to modify the method by reducing the thickness of the possible supports generated, in this way they can be broken without damaging the quality of the 3D printing. This software will save a large amount of material in a 3D print.

ACKNOWLEDGEMENTS

M. E. LOAIZA acknowledges the financial support of the CONCYTEC – BANCO MUNDIAL Project “Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica” 8682-PE, through its executing unit PROCIENCIA, within the framework of the call E041-01, Contract No. 002-2020-FONDECYT.

REFERENCES

[1] H. M. Jaeger, S. R. Nagel, and R. P. Behringer, “Granular solids, liquids, and gases,” *Reviews of modern physics*, vol. 68, no. 4, p. 1259, 1996.
[2] J. Duran, *Sands, powders, and grains: an introduction to the physics of granular materials*. Springer Science & Business Media, 2012.
[3] R. A. Cuba Lajo and M. E. Loaiza Fernandez, “Parallel sphere packing for arbitrary domains,” in *International Symposium on Visual Computing*. Springer, 2021, pp. 447–460.
[4] Y. Wu, X. An, and A. Yu, “Dem simulation of cubical particle packing under mechanical vibration,” *Powder technology*, vol. 314, pp. 89–101, 2017.
[5] B. Zhao, X. An, Y. Wang, Q. Qian, X. Yang, and X. Sun, “Dem dynamic simulation of tetrahedral particle packing under 3d mechanical vibration,” *Powder technology*, vol. 317, pp. 171–180, 2017.
[6] H. Tangri, Y. Guo, and J. S. Curtis, “Packing of cylindrical particles: Dem simulations and experimental measurements,” *Powder technology*, vol. 317, pp. 72–82, 2017.

[7] J. Gan and A. Yu, "Dem study on the packing density and randomness for packing of ellipsoids," *Powder Technology*, vol. 361, pp. 424–434, 2020.
[8] S. Zhao, N. Zhang, X. Zhou, and L. Zhang, "Particle shape effects on fabric of granular random packing," *Powder technology*, vol. 310, pp. 175–186, 2017.
[9] A. D. Rakotonirina, J.-Y. Delenne, F. Radjai, and A. Wachs, "Grains3d, a flexible dem approach for particles of arbitrary convex shape—part iii: extension to non-convex particles modelled as glued convex particles," *Computational Particle Mechanics*, vol. 6, no. 1, pp. 55–84, 2019.
[10] E. Campello and K. R. Cassares, "Rapid generation of particle packs at high packing ratios for dem simulations of granular compacts," *Latin American Journal of Solids and Structures*, vol. 13, no. 1, pp. 23–50, 2016.
[11] L. Wang, X. An, Y. Wu, Q. Qian, R. Zou, and K. Dong, "Dem simulation of vibrated packing densification of mono-sized regular octahedral particles," *Powder Technology*, vol. 384, pp. 29–35, 2021.
[12] X. Wang, Z.-Y. Yin, D. Su, X. Wu, and J. Zhao, "A novel approach of random packing generation of complex-shaped 3d particles with controllable sizes and shapes," *Acta Geotechnica*, pp. 1–22, 2021.
[13] E. Lozano, D. Roehl, W. Celes, and M. Gattass, "An efficient algorithm to generate random sphere packs in arbitrary domains," *Computers & Mathematics with Applications*, vol. 71, no. 8, pp. 1586–1601, 2016.
[14] Y. Li and S. Ji, "A geometric algorithm based on the advancing front approach for sequential sphere packing," *Granular Matter*, vol. 20, no. 4, pp. 1–12, 2018.
[15] R. Weller and G. Zachmann, "Protosphere: A gpu-assisted prototype guided sphere packing algorithm for arbitrary objects," in *ACM SIGGRAPH ASIA 2010 Sketches*. Association for Computing Machinery, 2010, pp. 1–2.
[16] R. Weller, U. Frese, and G. Zachmann, "Parallel collision detection in constant time," in *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2013.
[17] J. J. Jimenez, R. J. Segura, and F. R. Feito, "A robust segment/triangle intersection algorithm for interference tests. efficiency study," *Computational Geometry*, vol. 43, no. 5, pp. 474–492, 2010.
[18] D. Eberly, "Distance between point and triangle in 3d," *Geometric Tools*, 2020.
[19] H. Pfoertner, "Numerical results for densest packing of n equal spheres in a larger sphere with radius=1," 2013, Accessed on: Aug. 2, 2021. [Online]. Available: http://oeis.org/A084827/a084827.txt
[20] C. Ericson, *Real-time collision detection*. CRC Press, 2004.
[21] J. Arvo, "A simple method for box-sphere intersection testing," in *Graphics gems*, 1990, pp. 335–339.
[22] M. McGuire, "Computer graphics archive," July 2017, Accessed on: Dec. 1, 2021. [Online]. Available: https://casual-effects.com/data

AUTHORS

Cuba Lajo Rubén Adrián is currently a Bachelor of Computer Science from Universidad Católica San Pablo, Arequipa, Perú (2022). His current research interest focus on Computer Graphics and Videogames.

Loaiza Fernández Manuel Eduardo holds a PhD in Computer Science from Pontifícia Universidade Católica Do Rio De Janeiro, Rio de Janeiro, Brasil (2009). His current research interest focus on Computer Graphics and Computer Vision.

A SURVEY OF DEEP FAKE DETECTION FOR TRIAL COURTS

Naciye Celebi[1], Qingzhong Liu[1] and Muhammed Karatoprak[2]

[1]Department of Computer Science,
Sam Houston State University, Huntsville, TX, USA
[2]Department LLM in US Law, The University. Of Houston, Houston, TX, USA

ABSTRACT

Recently, image manipulation has achieved rapid growth due to the advancement of sophisticated image editing tools. A recent surge of generated fake imagery and videos using neural networks is DeepFake. DeepFake algorithms can create fake images and videos that humans cannot distinguish from authentic ones. (GANs) have been extensively used for creating realistic images without accessing the original images. Therefore, it is become essential to detect fake videos to avoid spreading false information. This paper presents a survey of methods used to detect DeepFakes and datasets available for detecting DeepFakes in the literature to date. We present extensive discussions and research trends related to DeepFake technologies.

KEYWORDS

DeepFake, Digital Forensics, Law.

1. INTRODUCTION

In the last two decades, image manipulation has achieved rapid growth due to the advancement of sophisticated image editing tools. The antiquated phrase "seeing is believing" is still the main perspective of validating such information. Hence, manipulated images convey misinformation and can be used to discredit people. Therefore, detecting manipulated images is increasingly essential and needs to be addressed.

A recent surge of generated fake imagery and videos using neural networks, so-called DeepFakes, are of great public concern and can now be easily created from scratch without leaving obvious perceptual traces. Open-source image editing tools lead to a large amount of generated Deepfake images and videos presented on social media, appearing a vital challenge for its' detection[1].

Recently, a new vein of fake image and video creation/generation methods known as DeepFake has been popular and attracted much attention. Many people can easily generate fake images and videos using smart phone and desktop applications such as FaceApp [2]. These applications have become much more prominent and advanced to generate extensive and realistic fake images, and videos can reveal the difficulty in determining their authenticity.Table1 shows the available DeepFake Generation tools. The step-by-step instructions and tutorials are easily available to create these fake images on the Internet. Therefore, these generation methods can be used for defamation, imitation, and misuse of facts. Furthermore, these DeepFakes can be widely and quickly disseminated on the Internet through social media. DeepFake takes as input an image or image of a specific person (’target’) and outputs another image or video with the target’s faces replaced with those of another person (’source’). DeepFake automatically maps the facial

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 227-238, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120919

expressions of the source to the target. With decent post-processing, the output image has achieved a high level of realism.

Generative Adversarial Networks (GANs) have been extensively used for creating realistic images[3]. The adversarial framework of GANs can also be used in conditional scenarios for image swapping [4],[5],[6], which diversifies media synthesis. The goal of GANs is to generate similar-looking samples to those in the training set. More importantly, GANs generate these samples without access to the original images. As an example, we can complete an actor's movie who had recently passed away with the usage of GANs.

Table 1. DeepFake Generation Tools

Tools	**Repositories**
FaceApp	https://github.com/topics/faceapp
Faceswap-GAN	https://github.com/shaoanlu/faceswap-GAN
DFaker	https://github.com/dfaker/df
DeepFaceLab	https://github.com/iperov/DeepFaceLab
DeepFake-tf	https://github.com/StromWine/DeepFake-tf

GANs are a combination of two neural networks (generator and discriminator) and interact only with the discriminative deep neural network to learn the data distribution. It competes to provide high-quality outputs similar to original inputs. GANs have been used to create new realistic images and videos and enhance those images. However, these machine learning algorithms, including GANs, can be misused to generate fake information to deceive people.

At the same time, however, the advancement of GANs has raised challenges to digital forensics. There is extensive concern about the impact of this technology when used maliciously. This issue has also received increasing public attention regarding disruptive outcomes to laws, politics, visual security, and society in general. Therefore, it is critical to look into practical visual forensics against threats from GANs.

Nowadays, social media has an active role in conducting an investigation. Many police officers can easily find a guilty person from a picture that can be found on his/her social media accounts. If a terrorist posts a video on YouTube expressing how he conducts a crime, investigators can easily find his identity. But how trustworthy are those founded images and videos? What if someone swapped a person's face with others on these images or videos? It is feasible that DeepFake challenges investigators and detecting these DeepFake images and videos are crucial.

2. RELATED WORKS

People are more and more interested in distinguishing GAN-generated media from real media. Many researchers have proposed various image forensics algorithms and tools to detect fake images, audio, and video. Many existing methods use the attributes of the image format and metadata information to determine the authenticity of the image. It is a challenging problem because attackers are also employing the latest image processing techniques to bypass wellknown forgery detection. Ding et al. [11] used deep transfer learning to extract a set of compact features and fed it into various classifiers such as SVM, random forest, and multi-layer perceptrons (MLP) for discriminating swapped face images from the genuine[11]. They are also different from existing methods that only provide detection accuracy. This study provides uncertainty for each

prediction, which is critical for trust in the deployment of such faceswapping detection systems. They ranked their images according to their fakeness. McCloskey et al.[8] mentioned how much Image forensics is an increasingly relevant problem and they analyzed the structure of the generating network of a GAN implementation. They also showed that the network's treatment of color is markedly different from a real camera in two ways. They further showed that these two signs can be used to identify GAN-generated imagery from camera imagery, showing effective separation between GAN imagery and real camera images used[8] for training purposes. They used two different datasets produced in conjunction with the US National Institute of Standards and Technology's Media Forensics Challenge 2018[8]. These datasets addressed two different sets of GAN imagery: 1. GAN Crop images represent smaller image regions. 2. GAN Full images are mostly camera images, but some faces have been replaced by a GAN-generated face, similar to deep fakes. For both datasets, they computed the features over the entire image. Zhang et al. [12] generated swapped faces using identified faces in the LFW dataset [13]. They used Bag of Words and sped up robust features to create image features instead of using pixels. They tested different machine learning models on the created swap images such as Random Forests and SVM's. They achieved 93 percent accuracy but did not examine beyond their proprietary swapping techniques. Moreover, their dataset only has 5,000 real, 5,000 swapped images, which is relatively small compared to other works. Khodabakhsh et al. [14] examined the previously published methods. They created a new dataset, which is a combination of 53,000 images from 150 videos. The generated faces in their data set were collected using different techniques. They evaluated texture-based and CNN-based fake face detections. They also used smoothing and blending to make the generated faces more photo-realistic.

3. DEEP FAKE AUTHENTICITY

Authenticity, in Federal Courts, is governed by Rule 901(a) which provides "To satisfy the requirement of authenticating or identifying an item of evidence, the proponent must produce evidence sufficient to support a finding that the item is what the proponent claims it is, FRCP 34(b), which means all that is needed is evidence sufficient to convince a reasonable juror that a particular fact or event was more likely than not to have occurred. Before settling the authenticity of the evidence, the court must determine the relevancy of the evidence offered. Federal Rule of Evidence 401 provides that "Evidence is relevant if: (a) it has any tendency to make a fact more or less probable than it would be without the evidence; and (b) the fact is of consequence in determining the action." (1-1). After relevance of evidence offered was determined by courts, then the authenticity of it must be determined. The threshold for Rule 901 is not high "the court need not find that the evidence is necessarily what the proponent claims, but only that there is sufficient evidence that the jury ultimately might do so." Fed.R.Evid. 901(a). Federal Rule of Evidence 901(b) provides a list of 10 authentication methods satisfying the standard of proof for establishing authenticity, which is non-exhaustive and not limited to these methods. Some of the methods the list contains are testimony of a witness with knowledge, Fed.R.Evid. 401(a) and (b), opinion about a voice of United States v. Safavian, 435 F. Supp. 2d 36, 38 (D.D.C. 2006), evidence about a telephone conversation United States v. Vayner, 769 F.3d 125, 130 (2d Cir. 2014), and evidence about a process or system showing that it produces an accurate result. Evidence authentication methods listed under Rule 901(a) and (b) are also applicable to all kinds of digital evidence. Once the threshold requirement of authentication is satisfied, "the ultimate determination as to whether the evidence is, in fact, what its proponent claims is thereafter a matter for the jury. As an illustration, emails have been increasingly offered as evidence at trial. For authenticity purposes, a testimony of a witness(7a) could be used to determine if the email offered as evidence is authentic. The witness can testify that s/he saw the email in question created by the author or received by the person whom the proponent claims authored/received it. (7b)

4. INVESTIGATION OF DEEP FAKE

Due to advancements in technology, social media are currently used for additional evidence in court to establish support alibis and provide important information relevant to court cases. Social data gathering plays an increasingly valuable role in the evidence collection process.

Nowadays, evidence collected from social media could provide beneficial information for locating people in asset tracing investigations or establishing jurisdiction for legal proceedings.
A review of US case law shows limits on when the data that gathered from social media can be presented in a court of law as evidence. A murderer was captured on an image, and the image is introduced as evidence of the crime. The court directed that the evidence was allowed as long as the reliability of the digital evidence could be established. Image, audio, and video evidence could be presented under the silent witness theory that holds that the digital evidence(image, audio, or video) may be submitted as evidence without the requirement of a witness to verify its authenticity if it has been confirmed that the manner in which it was produced was reliable. The benefits of social media evidence are undeniable, ranging from primary and corroborative evidence to risk assessment and continuing evidence of activities. What about if the collected evidence is manipulated, so-called DeepFake? The challenge will arise for investigators. How their conducted investigation are accurate? In addition to reliability, courts held that social media evidence are admissible as long as their integrity, accuracy, and authenticity can be established. The question that follows is: how are these established?

Photographs have been manipulated and swapped images were popular and modified long before the digital image was invented. Investigators must compare the original photograph with the negative to verify the authenticity of an image. Digital image forensics has mostly focused on identifying low-level modifications in images, such as dropping or duplicating a frame or frames and regions swapping and copy-pasting a part of the primary image and placing them in other areas. Because of the existence of image, audio, and video manipulation, illegal convictions of perpetrators based on those social data will be at risk if an investigator does not consider the vulnerability of the social data manipulation. Investigators should be cognizant of how easily manipulated images, audio, and videos could present false evidence and lead to wrongful convictions. Some audio recordings present unique authentication issues, and their unaltered audio versions may or may not be available. Because social data that have been verified but are not what they imply could be wrongly admitted as evidence, investigators should utilize caution when introducing this type of evidence in the court of law. Corroborating evidence is required to show that the social data were not manipulated.

Historically, social data has been included to support eyewitness testimony. Currently, eyewitness testimony will be mandated to corroborate social data. Nevertheless, as machine learning and AI technology advances, the investigators may not be sufficient to authenticate evidence, because even expert witnesses may not be able to discern the manipulation made to social data.

Deepfakes are relative newcomers to digital media forensics. Images, audios, and videos must be authenticated before they are presented as evidence in a court of law; nevertheless, this method is difficult by the presence of Generative Adversarial Networks. Despite operating at their full potential, GANs are designed to improve their performance continually. As such, it is only a matter of moment before DeepFakes are so convincing that they are challenging to recognize as fakes.

4.1. Investigation Process

Digital forensics is essential for incident response strategy and provides an adequate response in a forensic manner [33]. These investigative steps are Examination, Identification, Collection, and Documentation. In [34], Tina et al. propose a new forensic model that allows the investigator to carry out a full forensic investigation by using the combination of cyber forensic and incident response models. The forensic process given in [34] consists of the following phases:

- Phase 1- Identification and Preparation: This is the initial phase of the proposed forensic process, and its purpose is to understand the social data which belong to the suspect.
- Phase 2- Identifying data sources: This phase is one of the most important phases of the process because it deals with identifying controllers of the system, the type of data that can be collected, and where the data can be collected. Data sources need to be identified when any type of data gathering is performed. Needless to say, documentation of the actions taken during this phase is critical and essential for a forensically sound investigation.
- Phase 3- Preservation, Prioritizing, and Collection: In this phase, the identified social data is collected from the known locations, and it is preserved and prioritized for the purpose of repeatability and presentation. In this phase, it is also critical to collect volatile data as it might be destroyed easily. For instance, data can be collected from the suspect's Facebook account, which may be deleted right after the investigation started.
- Phase 4- Examination: The purpose of this phase is the forensic examination of the collected evidence. In this phase, possible data filtering techniques can be used to reduce unrelated data. In this phase, the evidence data is simply surfaced using recovery techniques and tools for forensic analysis. Later on, the evidence's authenticity needs to be tested with the usage of the DeepFake Detection Model/Tool. If that evidence is not manipulated, then the evidence is ready for analysis.
- Phase 5- Analysis: This phase includes recovered forensic artifacts and collected evidential data in order to develop a timeline of the events/incidents. The actual analysis of the data is performed in this phase.
- Phase 6- Reporting and Presentation: This phase is the collection of findings during the examination and analysis phases. It should include the chain of custody documents to protect the admissibility and reliability of the evidence.
- Phase 7- Review Results In this phase, all the investigative process is reviewed for a comprehensive look to identify inculpatory or exculpatory data. The investigator may prove or disprove certain explanations made earlier.

4.2. Digital Evidence and Authentication

The rapid development of technology since the late 20th century and the use of technological devices such as phones, computers, and digital storage tools have made a significant transition in people’s daily lives and changed their habits. Digital devices are everywhere in today’s world, helping them in their everyday life, from communication to education, health, productivity, and so on. For example, small devices such as laptops, tablets, and smartphones took shelves full of encyclopedias’ places in searching for information. The courtrooms have also been affected by these developments in collecting and handling evidence. The 2006 amendments to the Federal Rules of Civil Procedure (FRCP) included a new term of Electronically Stored Information (ESI) which refers to any documents or information stored in electronic form. Most common examples of ESI include word processing documents, digital photographs, videos, emails, text messages, and social media postings. After an amendment that included ESI into the Federal Rules of Civil Procedure, digital evidence has been offered more commonly at trial and has dramatically increased in volume as parallel to the increasing use of technological devices. Digital evidence

includes but is not limited to emails, photographs, videos, texts, Facebook posts, info derived from websites, etc.

In terms of authenticity, accepting digital evidence also brought some challenges to courts since the authenticity of digital evidence is usually a central battleground to determine its admissibility. Authenticity in Federal Courts is governed by Rule 901(a) which indicates "to satisfy the requirement of authenticating or identifying an item of evidence, the proponent must produce evidence sufficient to support a finding that the item is what the proponent claims it is" which means all that needed is evidence sufficient to convince a reasonable juror that a particular fact or event was more likely than not to have occurred. Before ruling about the authenticity of the evidence, courts must determine the relevancy of the evidence brought before the court. Federal Rule of Evidence 401 provides that "Evidence is relevant if: (a) it has any tendency to make a fact more or less probable than it would be without the evidence, and (b) the fact is of consequence in determining the action.". After the relevancy of evidence was determined by courts, the authenticity of it must be concluded. The threshold for Rule 901 is not high which the court in the United States v. Safavian case held that "the court need not find that the evidence is necessarily what the proponent claims, but only that there is sufficient evidence that the jury ultimately might do so.". Once the threshold requirement of authentication is satisfied, "the ultimate determination as to whether the evidence is, in fact, what its proponent claims is thereafter a matter for the jury. Federal Rule of Evidence 901(b) provides a nonexhaustive list of authentication methods satisfying the standard of proof for establishing authenticity. Some of the methods the list contains are testimony of a witness with knowledge, opinion about a voice, evidence about a telephone conversation, and evidence about a process or system showing that it produces an accurate result[9]. There are no substantial changes to the Federal Rules of Evidence in terms of authenticating digital evidence and evidence authentication methods listed under Rule 901(b) also applies to all kinds of digital evidence. As an illustration, emails have been increasingly offered as evidence at trial. For authenticity purposes, a testimony of a witness[10] could be used to determine if the email offered as evidence is authentic. The witness can testify that s/he saw the email in question created by the author or received by the person whom the proponent claims authored/received it.

Rule 902 allows certain types of evidence to be self-authenticating which means no need for any extrinsic evidence for the purpose of authenticity, unlike witness testimony given as an example above. However, the opponent of the evidence may always challenge the authenticity. Unlike rule 901(b)'s non-exhaustive list of authentication methods, self-authentication methods listed under 902 are limited. In 2017, sections FRE 902 and 902 were added to the Federal Rules of Evidence as categories of self-authenticating digital records. Added FRE 90213 is related to Certified Records Generated by an Electronic Process or System and provides that "A record generated by an electronic process or system that produces an accurate result, as shown by a certification of a qualified person that complies with the certification requirements of Rule 902 (11) or (12). The proponent must also meet the notice requirements of 902(11).". Also FRE 902(14) regulates Certified Data Copied from an Electronic Device, Storage Medium, or File which provides "Data copied from an electronic device, storage medium or file if authenticated by a process of digital identification, as shown by a certification of a qualified person that complies with the certification requirements of Rule 902(11) or (12). The proponent also must meet the notice requirements of 902(11).". These newly added sections will allow both sides to authenticate certain types of Electronically Stored Information(ESI) without the need to offer any extrinsic authentication method provided in 901(b). For example, a party could introduce GPS data as evidence if s/he could introduce an authentication certificate pursuant to FRE 902(13), or a party could introduce text messages certifying these text messages are the same as the originals. With these amendments, no live witnesses are required to authenticate certain machine-generated data in some cases. However, the ability to fabricate digital data has made the authenticity of digital

evidence a vital issue. With the ongoing advances in technology, manipulation of digital evidence has become easier for perpetrators. Now, it is possible as well as easy to create realistic-looking videos, so-called Deepfakes, that show people saying and doing things that they never said or did. A deepfake video utilizes machine learning to transform people's image, audio, and video so these data resemble someone else and can manipulate people's words and actions. The more video and audio footage of real people that can be fed into the generated model, the more convincing the result will be[35]. The advent of so-called "DeepFake" social data has recently been used widely in political disinformation campaigns to injure political opponents. In 2018, a deepfake video showing Barack Obama saying words he never said went viral on Youtube, also, in another video, the audio of Nancy Pelosi was altered by slowing down the audio to show she seems to be slurring. Most recently, a digitally altered, so-called deepfake, video of Queen Elizabeth delivering an alternative Christmas speech showing her dancing during the speech left too many people behind believing the video is real.

All these examples made people think if the antiquated phrase "seeing is believing" still be the main perspective to validate the information.

4.3. Deepfakes in Courtrooms

As an inevitable consequence of the widespread use of technology, it is not hard to anticipate that we will see more deepfakes in pre-trial and trial stages of courtrooms in near future. In a recent child custody case in the United Kingdom, the voice of a child's father was manipulated by the child's mother showing the father was heard making violent threats towards his wife. After expert examination, the recording was found as manipulated to include words not used by the father. The mother herself used a software program and online tutorials to put together a plausible audio file[36]. The case illustrates a good example of what kind of problems are laying ahead of courts. As the case illustrates, there is no need to be an expert to manipulate digital evidence because even a layperson is able to create DeepFakes by watching a few tutorials or utilizing ready-to-use programs. Deepfakes will have undesirable effects on courts that "The foreseeable effects will be both direct and indirect"[37] says Riana Pfefferkorn, professor of expertise in the area of cyber security. As a direct effect, DeepFakes can cause some additional caseload to courts by giving rise to new tort claims on the basis of this deepfake evidence brought to courts. In the above-mentioned child custody case, for example, the father may bring a tort claim against the mother for the deepfake evidence produced by her. As an indirect effect, the alleged deepfake will not be the main reason for the lawsuit but rather the deepfake will be just another piece of evidence in the course of litigation, where video evidence is already common[38]. Determining if the introduced video and other digital media evidence are either admissible or not will cause courts to spend too much time on this particular piece of evidence as well as will be costly when expert opinion is needed. Video is a widely used digital evidence in courts but the trustworthiness of videos should be questioned very often no matter which source the videos come from. Protecting the integrity of evidence, such as a video, is as important as getting the evidence itself, and in some cases, this integrity could be harmed by people who purposefully attack this evidence. A security firm consultant, Josh Mitchell, analyzed five body camera models from five different companies which all companies sell their devices to law enforcement groups around the US. The result of the study showed that four of the five body camera devices are remotely accessible and have security issues that allow an attacker to download footage off a camera, add modifications, or erase some footage from the video that the attackers don't want law enforcement to see, and then re-upload it, leaving no trace of the change[37]. Even though courts can authenticate these video evidence by one of the non-exhaustive methods listed under FRE 901(b) such as a witness with knowledge, not necessarily the person who took the video, providing when, where and under what circumstances the video was taken, bigger problems give rise under the jurisdictions that accept so-called silent witness theory which means photo or video speaks for itself. Silent witness

theory focuses on the automatic operation of the video and allows photographic evidence to be admitted without verification of accuracy from eyewitnesses. Under jurisdictions that accept silent witness theory a photo or video, even it is a deepfake, may get into evidence more easily even confronted by the other side. Courts are already familiar with forgery and have rules of evidence to deal with the authenticity of evidence[38]. But, technology's widespread accessibility and the capacity to produce deepfakes by manipulating videos is growing at a meteoric rate compared to the ability as well as in terms of available tools used to distinguish deepfakes from genuine evidence. Soon it will be a big hurdle for courts to distinguish genuine evidence from deepfakes as the software programs and instructive materials to make deepfakes continue to improve at that rate. On the other side, there are available remedies to keep the negative effects of deepfakes at a minimum in courtrooms. For example, a total ban of deepfakes should be thought on but a total injunction of creating deepfakes most probably will face the First Amendment's freedom of speech clause and will be hard to prevail[39]. Still, some states enacted laws focusing on particular issues caused by deepfakes[21]. As a more effective remedy, more sophisticated tools that use Artificial Intelligence(AI) may be utilized by courts to detect manipulated evidence, so-called DeepFakes, from genuine evidence. In 2018, the Defense Department has produced the first tools for catching deepfakes[40]. Also, big tech companies, such as Microsoft, Facebook, Google, Amazon have been investing and developing AI-based detection methods[41].

Consequently, the importance of digital forensics and AI tools to detect deepfakes from genuine evidence has been becoming more important than ever. Courts need to use AI against AI to get better results in terms of authenticating evidence. Even though it is still not a big hurdle to distinguish authentic evidence from deepfakes, experts warn that deepfakes will be indistinguishable from real evidence at that pace. "In January 2019, deep fakes were buggy and flickery," said Hany Farid, a UC Berkeley professor and deepfake expert. "Nine months later, I've never seen anything like how fast they're going. This is the tip of the iceberg." "We are outgunned," said Farid. "The number of people working on the video-synthesis side, as opposed to the detector side, is 100 to 1."[42]

5. AVAILABLE DATASETS

Deepfakes are potential counter-productive to society, security, and privacy. Many researchers have proposed methods for detecting DeepFakes. Early attempts were based on handcrafted features collected from artifacts and deviations of the fake image combination process. On the other hand, recent methods mainly focused on automatic detection system, which consists of Deep Learning models. The detection methods are focused on a binary classification situation where classifiers are used to classify between real videos and tampered ones. Real and fake video and images dataset are required in order to apply the deep learning models to train classification models. The number of fake videos and images is available, but it is still limited in terms of setting a benchmark for validating various detection methods.

- Celeb-DF[15]: This dataset is comprised of 590 real videos and 5, 639 fake videos

- FaceForensics++[16]: his dataset includes a subset of DeepFakes videos, which has 1, 000 real YouTube videos and the same number of synthetic videos generated using faceswap.

- DeepFake Detection[17]: The Google/Jigsaw DeepFake detection dataset is a subset of 3, 068 DeepFake videos generated based on 363 original videos of 28 consented individuals of various genders, ages, and ethnic groups. The details of the generation algorithm are not disclosed.DeepFake Detection Preview: The preview of the Facebook DeepFake detection challenge dataset is a subset of the DeepFake detection challenge, which has 4, 113 DeepFake videos created based on 1, 131 original videos of 66 consented individuals of various genders,

ages, and ethnic groups. This dataset is created using two different synthesis algorithms, but the details of the synthesis algorithm are not disclosed.

- HOHA Dataset[18]: This dataset is a combination of 300 real videos.
- UADFV[19]: This dataset contains 49 real YouTube and 49 fake videos. The DeepFake videos are generated using the FakeAPP.
- DF-TIMIT[20]: This dataset includes 640 fake videos generated with faceswap-GAN and is a subset of the Vid-TIMIT dataset. The videos are divided into DF-TIMIT-LQ and DF-TIMITHQ datasets, with synthesized faces of size 64 × 64 and 128 × 128 pixels, respectively.
- VTD Dataset[21]: This dataset includes 320 real videos collected from YouTube.

6. AVAILABLE DETECTION MODELS

Guera et al.[7] proposed a system that detects fake videos. They used the convolutional neural network (CNN) for extraction purposes and used these extracted futures to train a recurrent neural network (RNN) that learns to classify if a video has been subject to manipulation or not. They collected 300 deepfake videos from some video-hosting websites. Then, they randomly selected 300 videos from HOHA dataset. Their system had achieved more than 95 percent accuracy. They reached high accuracy because the trained dataset size is not big enough. Typically, CNN and RNN models require more datasets to train. [9] Korshunov et al. introduced a publicly available dataset for Deepfake and named as VidTIMIT database. In their paper, they introduce open-source software based on GANs to produce the Deepfakes, and they showed that training and combining parameters could significantly affect the quality of the produced videos. To demonstrate this result, they generated low and high-quality images and videos using separately tuned parameter sets. They pointed out that recently, accessible face identification systems are vulnerable to DeepFake images and videos, with 85.62 percent and 95.00 percent false acceptance rates. They also stated that it is necessary to identify DeepFake images and videos. They found out that the audio-visual method based on lipsync inconsistency detection could not identify Deepfake videos. Their experiments prove that GAN-generated DeepFake images and videos are challenging. They also mentioned that the further development of faceswapping technology would make it even more challenging. [10] Agarwal et al. described a forensic method that shows facial expressions and actions that symbolize an individual's speaking pattern. They also discussed how DeepFakes are designed and can, therefore, be used for authentication. They used the open-source facial expression analysis toolkit OpenFace2 to extract facial and head movements in a video. They also used t-SNE method for visualization of the 190-dimensional characteristics for Hillary Clinton, Barack Obama, Bernie Sanders, Donald Trump, Elizabeth Warren. They compare CNN and FaceForensics++ models. They detected that FaceForensics++ operates reasonably well on face-swapping but did not conclude lip-sync deep fakes.

Li et al. [22] proposed a new deep learning-based model that can efficiently identify DeepFakes. Their proposed method is focused on the new DeepFake algorithm that can only produce images and videos of limited resolutions, which require to be further distorted to meet the real faces in the source image and video. In this study, the authors pointed out that DeepFakes could be effectively obtained by convolutional neural networks (CNNs). They presented that their proposed model does not need DeepFake generated images as false positives since they target the artifacts in affine face-swapping as the distinctive feature to distinguish real and fake images[22]. They also stated their success in this study, (1) Such artifacts can be simulated instantly using simple image processing models on an image to make it a negative example. (2) Since such

artifacts generally exist in DeepFake videos from various sources, their method is more robust than others.

Fernandes et al [23] mentioned that it is become essential to detect fake videos to avoid the spread of false information. In this paper, they used the heart rate of fake videos to distinguish between original and fake videos. They obtained the heart rate of original videos and trained the state-of-the-art Neural Ordinary Differential Equations (Neural-ODE) model. Then they created DeepFake videos using DeepFake generation tools. The average loss obtained for ten original videos is 0.010927, and ten donor videos are 0.010041[23]. The trained Neural-ODE was able to predict the heart rate of our 10 DeepFake videos generated using DeepFake generation tools and 320 DeepFake videos of the deepfakeTIMIT database. This is the first attempt to train a Neural-ODE on original videos to predict the heart rate of fake videos.

Amerini et al. [24] proposed a new forensic technique that can be discerned within fake and real image and video sequences. They compared state-of-the-art techniques that resort to every image frame, and their proposed selection of visual movement fields employs reasonable interframe variations. They additionally used the feature which is learned by CNN classifiers. Preliminary results were collected on the FaceForensics++ dataset and received quite promising performances. FaceForensics++ has been used with three automated image and video manipulation methods: Deepfakes, Face2Face, and FaceSwap. 720 of the image and videos are used for training purposes, 120 for validation, and another 120 for testing. They received VGG16: 81.61percent ResNet50: 75.46 percent on the Face2Face dataset.

In this paper, Afchar et al.[25] introduced a method to automatically and effectively distinguish DeepFakes, and they essentially concentrated on two contemporary methods used to generate manipulated images and videos: Face2Face and DeepFake. They stated that recent image forensics methods are normally not well suitable for image videos due to the concentration that completely discredits the data. Therefore, in this paper, the authors developed a deep learning approach and performed two networks, both with a low number of layers, to focus on images' mesoscopic properties. They also evaluated those fast networks on both Face2Face datasets. The experiments show a quite well detection rate with more than 98 percent for Deepfake and 95 percent for Face2Face.

Jeon et al.[26] proposed FakeTalkerDetect, which is based on siamese networks to detect the talking head with few-shot learning. Unlike conventional methods, they also proposed to use pre-trained models with only a few real image datasets for fine-tuning in siamese networks to efficiently detect the fake images in a highly unbalanced data setting. The FakeTalkerDetect achieves an overall accuracy of 98.81 percent accuracy in detecting DeepFakes generated from the latest neural talking head models. In particular, their preliminary work also demonstrates the effectiveness of the highly unbalanced dataset.

Xuan et al.[27] mentioned that GAN-generated images and videos are getting more practical with the high-quality generation, and it is infeasible for individual eyes to detect. On the other hand, researchers generate methods to identify these DeepFakes and guarantee social data's reliability. They also stated that the detection of DeepFake images and videos is a fundamental research problem. This paper explored this problem and proposed using preprocessed images and videos to embed a forensic CNN model. By performing image-level preprocessing to both real and fake images and video training sets, the CNN model is forced to learn to distinguish the fake and real images and videos.

Sabir et al.[28] extract the most suitable strategy for combining varieties in the RNN model and domain-specific face preprocessing methods through experimentation to obtain a state-of-the-art

review on the FaceForensics++ dataset. The authors endeavored to detect Deepfake, Face2Face, and FaceSwap tampered faces in video streams in this study. Evaluation is performed on the recently introduced FaceForensics++ dataset, and they developed the previous state-of-the-art with up to 4.55 percent of accuracy.

Yang et al.[29] introduced a new method to detect DeepFakes. Their approach is mainly focused on the observations that Deep Fakes are created by splicing combined face area into the primary image/video. Hence, they suggested that errors may be exposed when 3D head postures are determined from the images and videos. They also conducted analyses to interpret the DeepFakes and, moreover, develop a classification method based on this suggestion. An SVM classifier is evaluated with the usage of real face images and videos.

Koopman et al. [30] declared that DeepFake poses forensic challenges concerning the authenticity of image and video evidence. As a result, photo response non-uniformity (PRNU) analysis is examined by authors for its effectiveness at identifying DeepFakes. The PRNU analysis shows an important variation in mean normalized cross-correlation scores between authentic image/videos and Deepfakes.

7. CONCLUSION

Recently, many people have begun to apprehend the existence of DeepFakes. They are aware that DeepFakes convey misinformation and can be used to misled people due to the dissemination of DeepFakes are increasingly approachable, and social media platforms can spread the DeepFakes quickly. Can awareness of DeepFakes helps people to validate such information, or it may distress, and adverse effects to targeted, heighten disinformation? The answer is really easy; it consumes the trust of people in media content, as seeing them is no longer believing in them. More often, DeepFakes do not need to be expanded to the massive audience to cause harmful effects.

REFERENCES

[1] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In Computer Vision and Pattern Recognition Workshops, volume 1, pages 38–45, 2019

[2] Faceapp.com.(2019).FaceApp.[online]Availableat:https://www.faceapp.com/ [Accessed 21 Oct. 2019].

[3] Bi, Lei, et al. ”Synthesis of positron emission tomography (PET) images via multi-channel generative adversarial networks (GANs).” Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment. Springer, Cham, 2017. 43-51.

[4] Isola, Phillip, et al. ”Image-to-image translation with conditional adversarial networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[5] Zhu, Jun-Yan, et al. ”Unpaired image-to-image translation using cycle consistent adversarial networks.” Proceedings of the IEEE international conference on computer vision. 2017.

[6] Battenberg, Eric, and Jitong Chen. ”Rewon Child, Adam Coates, Yashesh Gaur, Yi Li, Hairong Liu, Sanjeev Satheesh, David Seetapun, Anuroop Sriram, and Zhenyao Zhu. 2017. Exploring Neural Transducers for End-to-end Speech Recognition.” IEEE Automatic Speech Recognition and Understanding Workshop.

[7] G ¨uera, David, and Edward J. Delp. ”Deepfake video detection using recurrent neural networks.” 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2018.

[8] McCloskey, Scott, and Michael Albright. ”Detecting GAN-generated Imagery using Color Cues.” arXiv preprint arXiv:1812.08247 (2018).

[9] Korshunov, Pavel, and S’ebastien Marcel. ”Deepfakes: a new threat to face recognition? assessment and detection.” arXiv preprint arXiv:1812.08685(2018).

[10] Agarwal, Shruti, et al. ”Protecting World Leaders Against Deep Fakes.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019.
[11] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. IEEE Transactions on Multimedia, 17(11):2049–2058, 2015.
[12] Y. Zhang, L. Zheng, and V. L. Thing. Automated face swapping and its detection. In 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), pages 15–19. IEEE, 2017
[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007
[14] Li, Yuezun, et al. ”Celeb-DF: A New Dataset for DeepFake Forensics.” arXiv preprint arXiv:1909.12962 (2019).
[15] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In ICCV, 2019.
[16] Nicholas Dufour, Andrew Gully, Per Karlsson, Alexey Victor Vorbyov, Thomas Leung, Jeremiah Childs, and Christoph Bregler. Deepfakes detection dataset by google jigsaw
[17] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (DFDC) preview dataset.arXiv preprint arXiv:1910.08854, 2019.
[18] Di.ens.fr. (2019). Ivan Laptev ¿ Projects ¿ Human Action Classification. [online] Available at: https://www.di.ens.fr/ laptev/actions/ [Accessed 7 Dec. 2019].
[19] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
[20] Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685, 2018

FACIAL EMOTION RECOGNITION IN IMBALANCED DATASETS

Sarvenaz Ghafourian, Ramin Sharifi and Amirali Baniasadi

Department of Electrical and Computer Engineering,
University of Victoria, Victoria, Canada

ABSTRACT

The wide usage of computer vision has become popular in the recent years. One of the areas of computer vision that has been studied is facial emotion recognition, which plays a crucial role in the interpersonal communication. This paper tackles the problem of intraclass variances in the face images of emotion recognition datasets. We test the system on augmented datasets including CK+, EMOTIC, and KDEF dataset samples. After modifying our dataset, using SMOTETomek approach, we observe improvement over the default method.

KEYWORDS

Emotion Recognition, Residual Network, VGG.

1. INTRODUCTION

Since the human face plays an integral part in expressing a person's mental state, facial expression analysis is a significant research focus with numerous potential uses. Scientists from many areas like psychology, finance, marketing, and engineering have been greatly interested in this subject due to the practical benefits.

Artificial intelligence is becoming more prevalent in many aspects of human life. The technologies are adapted to the needs of human beings, and artificial intelligence is what makes this adaptation between technology and humans possible. While it may come easy for most humans to process emotions without any extra effort, computers have struggled with the idea of recognizing them automatically for decades. This challenge is due to face appearance changes caused by pose variations, illumination variations, camera quality, and angle changes. Research from different disciplines such as computer vision and machine learning focus on utilizing computers to categorize emotions exhibited by humans properly. In this work we focus on analyzing facial expressions. Specifically, we study the task of facial emotion recognition based on two deep learning models using our suggested dataset. We use various face images for seven emotions and improve the efficiency of emotion detection.

The face is defined as the front portion of the human head, from above by the scalp border, below by the corners and bottom edge of the lower jaw, and on the sides by the margins of the lower jaw branches and the base of the auricles [1].

Facial emotion recognition (FER) is a method to identify human expressions, which is one of the factors involved in emotion recognition. Emotions are inherent characteristics of people and play a significant part in social communication [2][3]. Humans show emotion in a variety of ways, including facial expressions [4], gestures, vocalizations, body language [5], and speech [6]. The

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 239-251, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120920

six basic emotions described by Eckman [7] are: happiness, sadness, fear, disgust, anger, and surprise.

Overfitting occurs when the Facial Emotion Recognition (FER) model is trained on imbalanced datasets, making the model less capable of performing FER tasks in real-world scenarios. As a result, overfitting due to lack of sufficient data remains a problem for most FER systems. Thus, we create an augmented dataset in this work to mitigate the overfitting problem and improving generalization.

Our goal is to identify an individual's emotion from observing their facial expressions. First, cropped headshots are extracted using the FaceNet architecture. Second, the extracted face images from three different datasets are used as a single dataset for the transfer-learning task on VGG-16 and ResNet-50.

In summary the contributions of this paper are:

- A comprehensive analysis of popular emotion recognition datasets, such as CK+, EMOTIC, and KDEF. We describe how images are categorized in each dataset.
- Creating a custom dataset consisting of the three above-mentioned datasets to cover a wide range of variations in face images. We explain how different images of our dataset are cropped to fit our criteria.
- Improving class imbalance problem in the custom emotion recognition dataset over VGG-16 and ResNet-50. We show how SMOTETomek technique improves the distinction accuracy over VGG-16 and ResNet-50 models.

This paper is organized as follows. Section 2 describes related works. Section 3 demonstrates the background. Section 4 represents experiments and results. Section 5 offers concluding remarks.

2. RELATED WORKS

The traditional approach to detecting emotions consists of a two-stage machine learning process. The first phase involves collecting characteristics from the pictures, and the second phase involves using a classifier, such as an SVM, neural network, or random forest, to determine the emotions.

The histogram of oriented gradients (HOG) [8], local binary patterns (LBP) [9], Gabor wavelets[10], and Haar features [11] are some of the prominent hand-crafted features utilized for face emotion identification. The appropriate emotion is then assigned to the image using a classifier.

While these methods work for small datasets, they start showing their limits when applied to more complex datasets, with higher intraclass variances. Moreover, there are some issues with face images when the face is partially visible [12].

The majority of contemporary computer vision research into recognizing people's emotional states is based on facial expression analysis. Psychologists, Ekman and Friesen, identified six fundamental emotions and multiple methods for recognizing them. The Facial Action Coding System is used in several of these approaches. Action Units (AU) are a collection of unique localized movements of the face that encode facial emotion. This approach uses a set of distinct localized facial movements known as Action Units to represent facial emotion [13].

Convolutional Neural Networks (CNNs) have been used in recent studies for emotion detection

based on facial expression to recognize emotions and Action Units [14].

In response to the great success of deep learning and, in particular, CNNs for image classification and other vision challenges, a number of organisations have built deep learning-based facial expression recognition (FER) models [15]. Mollahosseini et al. showed that CNNs could recognize emotions accurately and achieve state-of-the-art results. The results are based on a zero-biased CNN on the expanded Cohn-Kanade dataset (CK+) and the Toronto Face Dataset (TFD). Mollahosseini also, suggested an FER neural network with two convolution layers, one max-pooling layer, and four inception layers, in each layer [16].

Aneja et al. in [17] created a model of facial expressions for stylized animated characters using deep learning. Their training included a network that represented human expressions, and a network that represented animated faces. The loopy network was first proposed by Liu in [18], noting the importance of feedback of the weak classifiers. Instead of using a strong classifier, a loop of weaker classifiers are used for emotion detection. They used their Boosted Deep Belief Network (BDBN) over CK+ and JAFFE datasets to achieve a higher accuracy.

In addition to determining the face characteristics, some studies [19] detect fundamental emotions using the position of shoulders. Schindler et al.[20] used a limited dataset of non-spontaneous postures obtained under controlled conditions to detect the six primary emotions.

Rather than identifying emotion categories, some more recent research on facial expression [21] employs the Valence, Arousal, Dominance (VAD) Emotional State Model continuous dimensions to describe emotions [22].

It should be noted that the majority of the past research is based on widely used facial expression recognition datasets, such as FER2013, the extended Cohn-Kanade (CK+), and the Japanese Female Facial Expression dataset (JAFFE). These datasets consist of frontal face images, and the photos lack any contextualized backgrounds and have fewer differences, such as spectacles or face masks. This makes the facial action units detection easier. However, we expect our model to perform on more challenging images as well. Images consisting of illumination, pose, occlusion, and low resolution ones are considered challenging images.

3. BACKGROUND

3.1. Face Recognition

The challenge of recognizing and validating people in an image by their faces is known as face recognition. Face recognition is sometimes defined as a four-step process that begins with face detection, then moves on to face alignment and feature extraction respectively, and ultimately face identification (Fig. 1) [23].

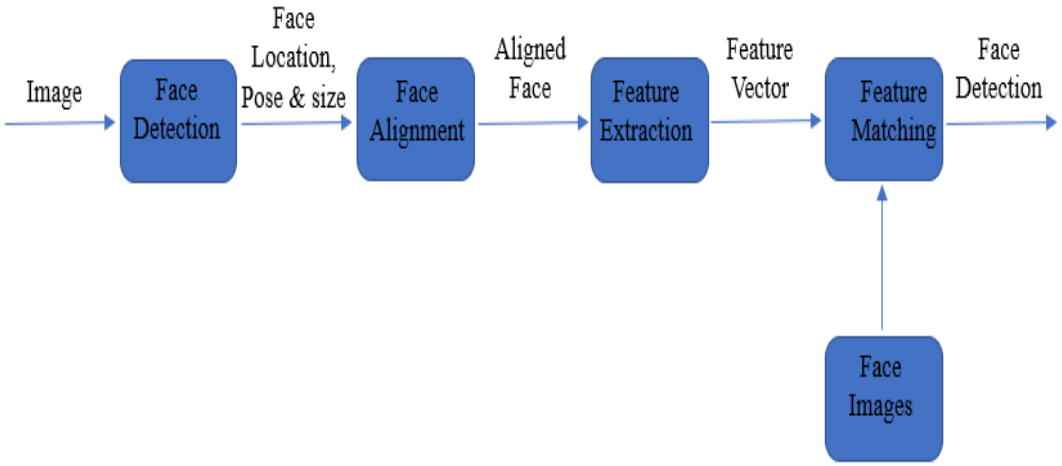


Figure 1. Face recognition processing flow [23].

Some of the face recognition task challenges include illumination that is caused by light fluctuation, pose which is the result of head movement and viewing angle changes, occlusion that is caused by blocking of one or more portions of a face, and low resolution.

3.2. Datasets

3.2.1. EMOTIC Dataset

Emotion in context (EMOTIC) [22] is a dataset of images with people in real environments, annotated with their apparent emotions. Since the photos are collected from real-life settings, the images' variations are higher than other common datasets, such as CK+ and KDEF. This dataset includes facial occlusion (usually with a hand), partial faces, low-contrast images, and eyeglasses.

An extended list of 26 emotion categories is defined in this dataset to annotate the images, combined with three standard continuous dimensions: Valence, Arousal, and Dominance. Rather than recognizing emotion categories, several new studies on facial expression employ the aspects of the VAD Emotional State Model to depict emotions. The VAD model describes emotions using three numerical dimensions:

- Valence (V): A scale evaluates how pleasant or pleasant a feeling is, from negative to positive.
- Arousal (A): A scale that assesses a person's level of agitation, ranging from nonactive/calm to agitated/ready to act.
- Dominance (D): A scale that evaluates a person's amount of control over a situation, ranging from submissive/non-control to dominant/in-control [22].

The pictures in the EMOTIC dataset are mostly from well-known datasets such as MSCOCOC [24] and ADE20K [25]. The EMOTIC dataset consists of 18316 images with 23788 people annotated.

3.2.2. KDEF

The Karolinska Directed Emotional Faces (KDEF) [5] is one of the most widely used human facial expressions databases. KDEF is a collection of 4900 photographs depicting human face

emotions. There are 70 people in the photo collection, each with a different emotional expression. Each emotion is examined from five distinct perspectives.

3.2.3. CK+ Dataset

Cohn-Kanade (CK) plus is the extended version of regular CK, which covers the shortcomings of its previous versions. CK+ has 593 sequences and 123 subjects, which is 22% more sequences and 27% more subjects than the original CK. Participants of CK+ dataset range between 18 and 50 years old. They were told to show 23 facial expressions consisting of single and multiple action units. The results of these sequences and subjects are distributed over seven different emotion categories that we are trying to detect.

3.3. Convolutional Neural Network

In this paper, we use Convolutional Neural Networks to detect emotions on our dataset. CNNs are the most popular architecture for image classifications. Pre-trained VGG-16 and Resnet-50 are customized to classify ten different categories of emotions. Customization of the VGG-16 and Resnet-50 are done by altering the classification part of the network. We added two fully-connected layers to produce ten outputs. Each output represents the probability of the image belonging to a specific category. No changes have been made to the feature-extraction architecture. In the next section, more detail is given about VGG and ResNet.

3.3.1. Residual Networks

After the first CNN-based architecture (AlexNet), which won the ImageNet 2012 competition, subsequent winning architectures use more layers in a deep neural network to minimize the error rate. The Residual Network [26] has a large number of layers. As the number of layers grows, the gradient vanishing problem arises. This issue changes the gradient value to either 0 or too large, which prevent the system to learn. Thus, as the number of layers increases, the training and test error rate also increases. ResNet resolves the vanishing/exploding gradient problem by adding the input features to the output. Fig. 2 demonstrates the residual block in the ResNet architecture.

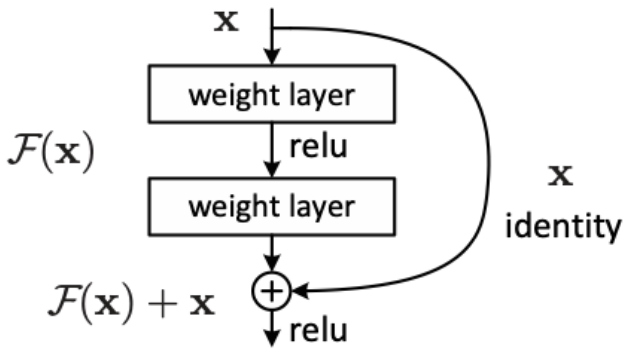


Figure 2. Residual Block in ResNet-50 architecture [27]

3.3.2. VGG

VGG-16 outperforms AlexNet by replacing large kernel filters sequentially (11 and 5 in the first and second convolutional layers, respectively) with numerous 3x3 kernel filters [27].

Rather than having a large number of parameters, VGG-16 employs 3x3 convolution filter layers with a stride 1. Also, the padding and maximum pooling layer with 2x2 filters and stride 2 remain the same. The convolution and max pool layers are placed in the same way throughout the design. At the end of its architecture, it has two fully-connected layers. The output is then followed by a softmax. The conv1 layer receives a 224 by 224 coloured image as input [28].

The features are extracted using convolutional layers with the smallest feasible dimensions: 3x3 to capture left/right, up/down, and centre of images. In one of the VGG variances, an extra 11 convolution filters are added, which may be regarded as a linear change to the input channels followed by non-linearity. The convolution spatial padding is set to 0 and the convolution stride is set to 1 pixel. After convolution, the spatial resolution of the layer input is preserved, i.e. the padding is 1-pixel for 33% of the convolutional layers. Spatial pooling is done via five max-pooling layers that follow part of the convolutional layers (not all the convolutional layers are followed by max-pooling). Max-pooling is done with stride 2 across a 2x2 pixel frame [29].

Following a stack of convolutional layers of varying depth in various designs, three Fully-Connected (FC) layers are added. 4096 channels are included in the first two FC layers. The last FC layer has 1000 channels since ImageNet dataset contains 1000 classes. The last layer performs as a softmax layer.

Rectified Linear Unit (ReLU) non-linearity is present in all hidden layers. Local Response Normalization (LRN), which does not enhance performance on the ILSVRC dataset but increases memory usage and computation time, is also included in none of the networks.

4. EXPERIMENT AND RESULTS

The dataset used in this work is a combination of three different datasets (CK+, EMOTIC, and KDEF), each of them having their unique features. Firstly, the augmented dataset is trained on a deep neural network, called FaceNet, to extract features from images of a person's face and detect the face. After the face images are detected, they should be augmented by image transformation to be fed to the input of emotion recognition networks. The final dataset consists of cropped, rotated, and horizontally-flipped images of the original dataset. After splitting the total images into training and testing 70% and 30%, respectively, the distribution of seven emotions are shown in Fig. 3 and Fig. 4.

Figure 3. Distribution of training images in seven emotion categories.

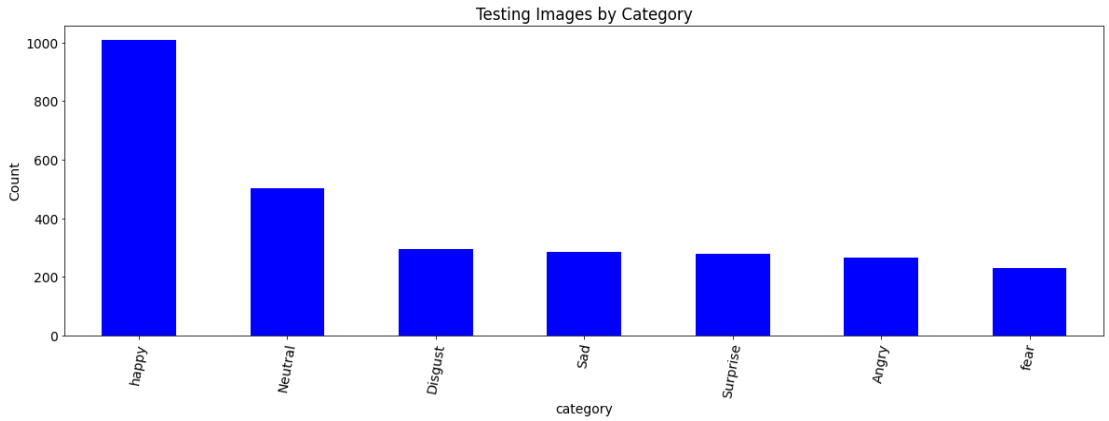


Figure 4. Distribution of testing images in seven emotion categories.

FaceNet, which is a Google-developed facial recognition system that obtains state-of-the-art results on various face identification benchmark datasets in 2015 [30], is used in this work to extract the faces. After extracting the face images, our pre-processed dataset is trained on two different convolutional neural network architectures: VGG-16 and ResNet-50.

As shown in Fig. 5, the accuracy of 52.49% is reached on the VGG-16 architecture. Moreover, the related confusion matrix is show in Fig. 6. Confusion matrix is a performance evaluation metric for machine learning classification problem. The actual target values and predicted values are compared using a confusion matrix. Labels are shown from 0 to 6, which maps to Anger, Disgust, Neutrality, Sadness, Surprise, Fear and Happiness. As illustrated, "Happiness" images are significantly higher in volume compared with other emotion categories. This explains the more accurate prediction of the architecture for this category shown in the confusion matrix as in Fig. 6.

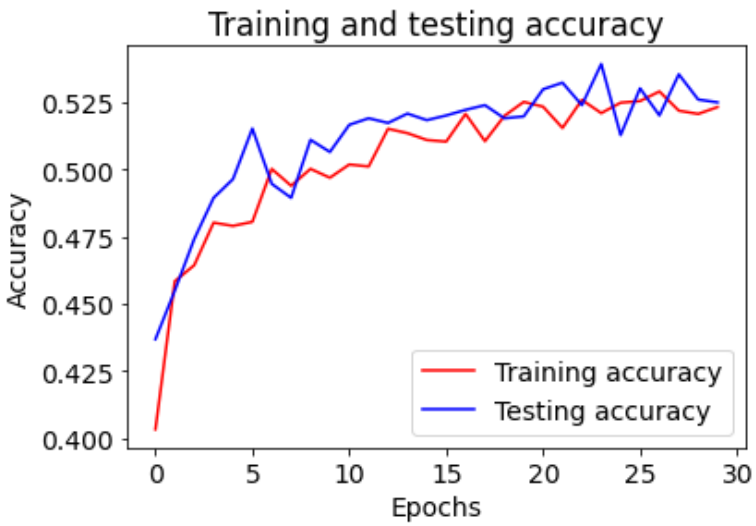


Figure 5. VGG-16- Training and testing accuracy with 30 epochs.

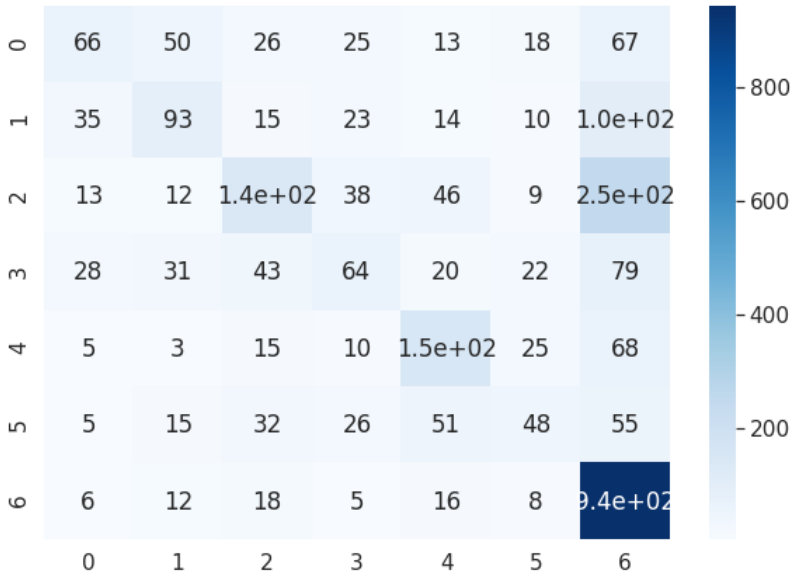


Figure 6. VGG-16- Confusion matrix.

As shown in Fig. 7, the accuracy of 55.63% is reached on the ResNet-50 architecture. The related confusion matrix is show in Fig. 8. As illustrated, "Happiness" images are significantly higher in volume compared with other emotion categories. This explains the more accurate prediction of the architecture for this category shown in Fig. 8.

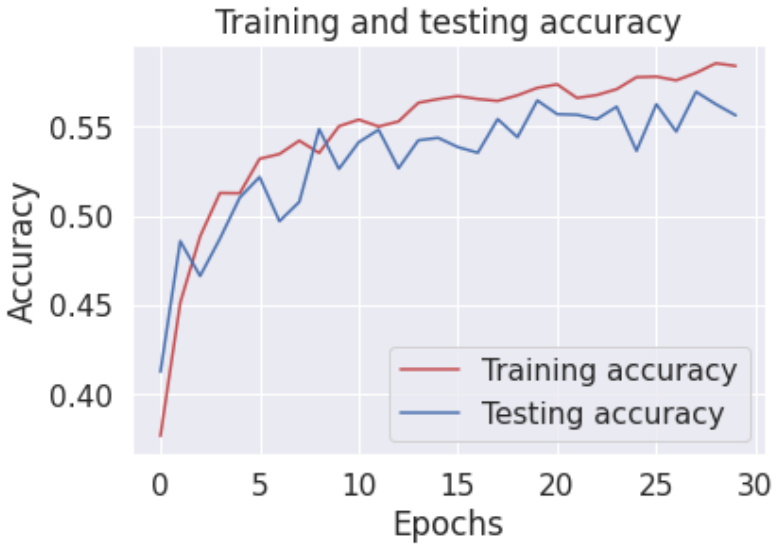


Figure 7. ResNet-50- Training and testing accuracy with 30 epochs.

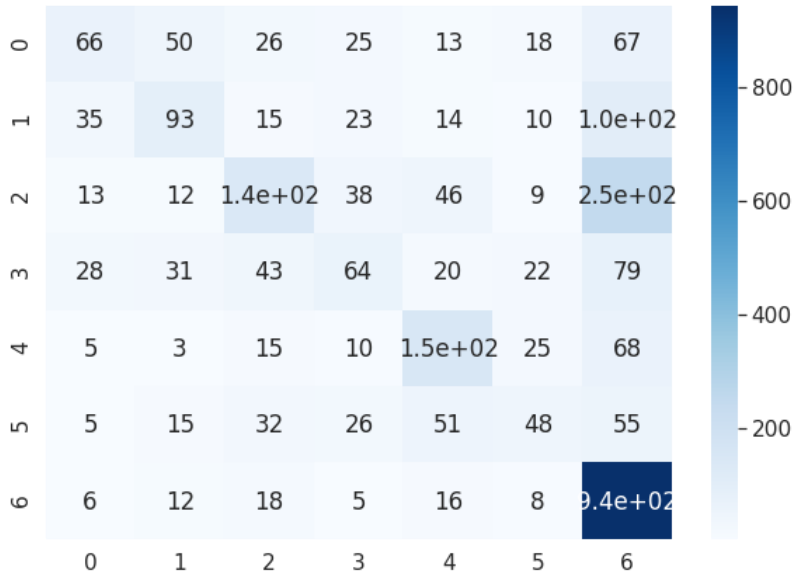


Figure 8. ResNet-50- Confusion matrix.

The above confusion matrices show high off diagonal values, which is due to class imbalance. To overcome this issue, a combination of alternative approaches called Synthetic Minority Oversampling Technique (SMOTE) and [31] is used.

In SMOTE, the minority class is over-sampled by creating “synthetic” examples instead of replacing the over-sampled examples [31]. The new samples are duplicated based on the Euclidean distance of each data and the minority class nearest neighbours. Therefore, the generated examples are different from the original minority class and provide additional information. This is useful for the system to learn the model.

In Tomek Link approach observations from the majority class are removed. This is also considered as an enhancement of Nearest-Neighbor Rule (NNR) [32]. This method uses NNR to select the pair of examples that fulfill specific properties. One of the advantages of this method is that it removes the data from the majority class that has the lowest Euclidean distance with the minority class data, therefore make it less ambiguous to detect the emotion.

For better comparison, we have shown how VGG-16 and ResNet-50 improved in Fig.9 and Fig. 10. The Y-axis shows the accuracy of the architecture over a certain emotion category. The X-axis represents the emotion categories from 0 to 6 mentioned previously. The enhanced dataset gives priorities to the emotion categories with less data.

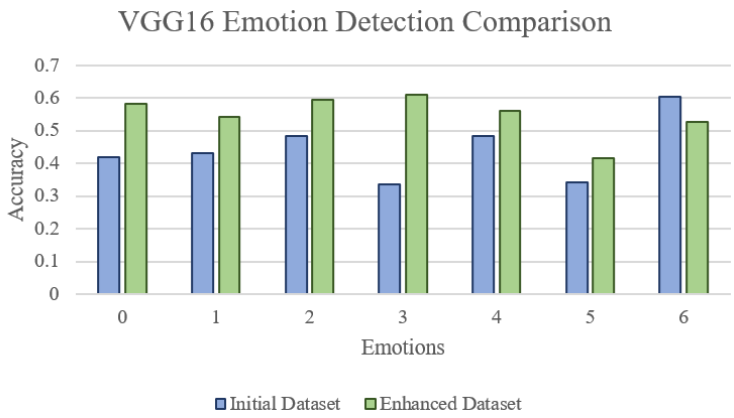


Figure 9. VGG-16 Emotion Detection Comparison.

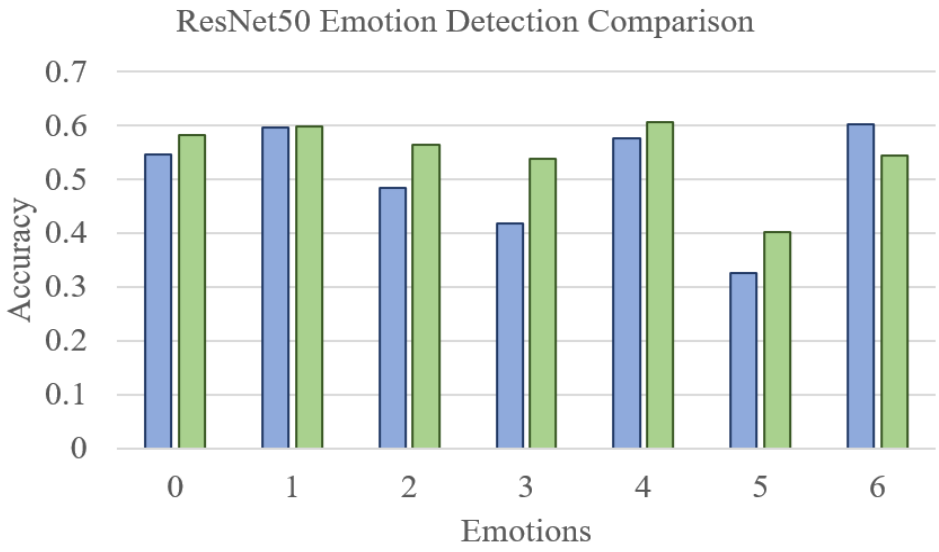


Figure 10. ResNet-50 Emotion Detection Comparison.

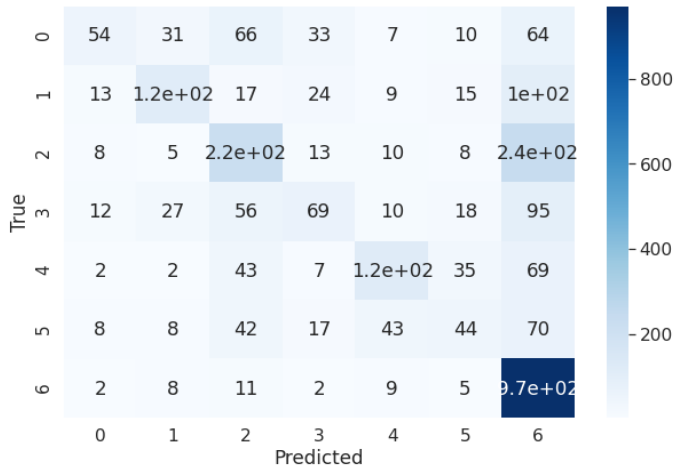


Figure 11. VGG-16- Confusion matrix for enhanced dataset.

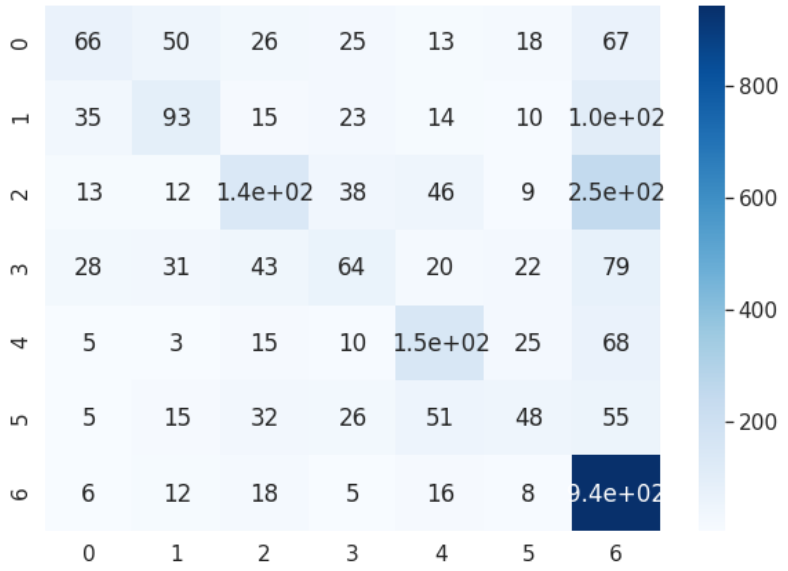


Figure 12. ResNet-50- Confusion matrix for enhanced dataset.

After modifying our dataset, using SMOTETomek approach, improvement over the initial method is observed. These improvements are reported over both architectures in Fig. 11 and Fig. 12. Comparing the enhanced confusion matrices with the initial confusion matrices, we can observe more consistency on seven emotion categories. Still, "Happiness" emotion distinction dominates other emotion detection.

Comparing the two confusion matrices (Fig.6 and Fig. 11), shows that our architecture produces more accurate results. This confirms the effectiveness of the proposed method compared with the initial one.

5. CONCLUSIONS

This work provides the implementation of facial emotion recognition based on two deep learning algorithms, VGG-16 and ResNet-50. From a technical point of view, this work has served to clearly demonstrate the advantage of using a balanced and enhanced dataset including almost the same number of examples in each class. The class imbalance data problem is also tackled using a combination of oversampling and undersampling techniques, called SMOTETomek. We have shown that VGG-16 and ResNet-50 can improve from about 50% up to 60.16% and 60.71% respectively. Future research can consider different architectures and fine-tuning hyper parameters.

REFERENCES

[1] L. Tereikovska, I. Tereikovskyi, S. Mussiraliyeva, G. Akhmed, A. Beketova, and A. Sambetbayeva, "Recognition of emotions by facial Geometry using a capsule neural network," *Int. J. Civ. Eng. Technol.*, vol. 10, no. 03, pp. 1424–1434, 2019.

[2] P. Ekman, "FACIAL EXPRESSION Edited by An imprint of The Institute for the Study of Human Knowledge," 1973.

[3] A. Kelly, "Facial expression," *Talkabout*. pp. 61–70, 2019, doi: 10.4324/9780429427251-6.

[4] H. Facial, "Book Reviews," no. 1994, pp. 1187–1194, 1996.

[5] F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera, and G. Anbarjafari, "Audio-Visual Emotion Recognition in Video Clips," *IEEE Trans. Affect. Comput.*, vol. 10, no. 1, pp. 60–75, 2019, doi:

10.1109/TAFFC.2017.2713783.
[6] M. Soleymani, M. Pantic, and T. Pun, "Multimodal emotion recognition in response to videos," *IEEE Trans. Affect. Comput.*, vol. 3, no. 2, pp. 211–223, 2012, doi: 10.1109/T-AFFC.2011.37.
[7] Harappa, "Types Of Emotions." 2020.
[8] J. Chen, Z. Chen, Z. Chi, and H. Fu, "Facial Expression Recognition Based on Facial Components Detection and HOG Features," 2014.
[9] C. Shan, S. Gong, and P. W. McOwan, "Robust facial expression recognition using local binary patterns," in *IEEE International Conference on Image Processing 2005*, 2005, vol. 2, pp. II–370, doi: 10.1109/ICIP.2005.1530069.
[10] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 2, pp. 568–573 vol. 2, doi: 10.1109/CVPR.2005.297.
[11] J. Whitehill and C. Omlin, "Haar features for FACS AU recognition," *7th Int. Conf. Autom. Face Gesture Recognit.*, pp. 5 pp. – 101, 2006.
[12] S. Minaee, M. Minaei, and A. Abdolrashidi, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network," *Sensors*, vol. 21, no. 9, 2021, doi: 10.3390/s21093046.
[13] and W. V. F. Ekman, Paul, "Facial action coding system," *Environ. Psychol. Nonverbal Behav.*, 1978.
[14] H. C. William and W. T. M. Liao, "Facial emotion recognition with transition detection for students with high-functioning autism in adaptive e-learning," *Soft Comput.*, no. 1, 2017, doi: 10.1007/s00500-017-2549-z.
[15] S. Minaee, A. Abdolrashidi, and Y. Wang, "An Experimental Study of Deep Convolutional Features For Iris Recognition Electrical Engineering Department , New York University , Computer Science and Engineering Department , University of California at Riverside," *Signal Process. Med. Biol. Symp.*, 2016.
[16] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," 2016, doi: 10.1109/WACV.2016.7477450.
[17] D. Aneja, A. Colburn, G. Faigin, L. Shapiro, and B. Mones, "Modeling Stylized Character Expressions via Deep Learning," in *Computer Vision -- ACCV 2016*, 2017, pp. 136–153.
[18] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial Expression Recognition via a Boosted Deep Belief Network," *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1805–1812, 2014.
[19] M. A. Nicolaou, H. Gunes, and M. Pantic, "Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space," *IEEE Trans. Affect. Comput.*, vol. 2, no. 2, pp. 92–105, 2011, doi: 10.1109/T-AFFC.2011.9.
[20] K. Schindler, L. Van Gool, and B. de Gelder, "Recognizing emotions expressed by body pose: A biologically inspired neural model," *Neural Networks*, vol. 21, no. 9, pp. 1238–1246, 2008, doi: 10.1016/j.neunet.2008.05.003.
[21] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, "Emotion recognition in context," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1960–1968, 2017, doi: 10.1109/CVPR.2017.212.
[22] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, "EMOTIC: Emotions in Context Dataset," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 2309–2317, 2017, doi: 10.1109/CVPRW.2017.285.
[23] S. Ouyang, T. Hospedales, Y.-Z. Song, and X. Li, "A Survey on Heterogeneous Face Recognition: Sketch, Infra-red, 3D and Low-resolution." 2014.
[24] G. T. U. A. Colleges *et al.*, "Microsoft COCO," *Eccv*, no. June, pp. 740–755, 2014.
[25] B. Zhou *et al.*, "Semantic Understanding of Scenes Through the ADE20K Dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019, doi: 10.1007/s11263-018-1140-0.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." 2015.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." 2015.
[29] A. Savoiu and J. Wong, "Recognizing Facial Expressions Using Deep Learning."
[30] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and

clustering," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 815–823, 2015, doi: 10.1109/CVPR.2015.7298682.

[31] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[32] A. Elhassan and Al-Mohanna, "Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method," 2017.

MONOCULAR CAMERA CALIBRATION USING PROJECTIVE INVARIANTS

Vilca Vargas Jose R[1] and Quio Añauro Paúl A[2] and Loaiza Fernández Manuel E[3]

Universidad Católica San Pablo, Arequipa Perú

ABSTRACT

Camera calibration is a crucial step to improve the accuracy of the images captured by optical devices. In this paper, we take advantage of projective geometry properties to select frames with quality control points in the data acquisition stage and, further on, perform an accurate camera calibration. The proposed method consists of four steps. Firstly, we select acceptable frames based on the position of the control points, later on we use projective invariants properties to find the optimal control points to perform an initial camera calibration using the camera calibration algorithm implemented in OpenCV. Finally, we perform an iterative process of control point refinement, projective invariants properties check and recalibration; until the results of the calibrations converge to a minimum defined threshold.

KEYWORDS

Camera calibration, Pattern recognition, Optimization & Frontoparallel projection.

1. INTRODUCTION

The main objective of computer vision area can be defined as the creation of systems that, through the analysis and biological inspired image processing, are able to "understand" a visual input [1]. To achieve this objective, object detection and tracking is one of the most important processes, because it lets the computer get a better model of the real world. Optical technology is widely used because of its low costs and availability. Also, optical technology has the great advantage that the tracking object does not have to be overposed by cables or other components [2].

Camera calibration is an essential step in computer vision, image processing and optical measuring. Since the accuracy at computing the 3D position of the object is directly related to a precise camera calibration [2,3].

Camera calibration faces some error sources such as the non-linear distortion caused by a non-frontoparallel input image, the imperfection of the calibration object, lens distortion or the lack of certainty when locating the control points directly from the geometric calibration patterns [4,5]. To face these problems, most modern algorithms include an iterative frontoparallel reprojection and pattern refinement step [5].

We propose an optimized camera calibration method based on the use of projective invariants properties to detect invariant projective patterns that grant sturdiness and a better accuracy in the data acquisition stage and take advantage of them in the next steps, alongside the use of the algorithm proposed by Zhang [6] implemented in OpenCV [7] to find the best values of the intrinsic and extrinsic parameters of the camera.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 253-272, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120921

The rest of this paper is organized as follows. In Section 2, we present a brief collection of related works. In Section 3 we detail our proposed method. In Section 4 we present the test environments and the results of the performed experiments. Later in Section 5 we present the accuracy comparison expressed in centimetres, and finally in Section 6 we present our conclusions and future work.

2. RELATED WORK

2.1. Camera Calibration

Camera calibration problem has been extensively studied. According to Song et al. [8] it can be divided in three main categories. Traditional camera calibration, camera auto-calibration and camera calibration based on active vision, although we consider hybrid methods as well. Thus, we consider the hybrid methods as a fourth category.

- **Traditional camera calibration** consists of the calculation of the intrinsic and extrinsic parameters of the camera through mathematical transformations after the processing of the images. It acquires an advantage when the shape and size of the calibration object is known. One of the most visited and well-known methods was proposed by Zhang [6], consisting on capturing several angles of a plain calibration pattern with the camera and restrict the internal and external camera parameters analysing the relation between each control point in the plain pattern and the corresponding control point in the image, and finally, performing a non-linear optimization of the calculated results by a maximum similarity criteria. In 2009, Datta et al. [4] recognized the deficiency on the control points localization as an error source and revisited Zhang's method proposing an iterative control point refinement strategy. We can find many novelty calibration techniques in the last decade, replacing the plain calibration pattern with a hanging chain curve [9], spheres [10] or using the motions of a wand [11]. Liu et al. [12] propose a method that presents a new calibration target which uses projective invariants properties to find feature points and feature lines to reduce the image distortion. Sarmadi et al. [13] propose a method that calculates the extrinsic parameters of multiple cameras and the relative position between the cameras and a rigid set of planar markers at each frame.

- **Camera auto-calibration** does not depend on the reference calibration object. The calibration is performed through the comparison of the relationship between the environment images through the camera movement. Yao et al. [14] proposes a camera auto-calibration method that consists of the construction of a trajectory matrix through the tracing of characteristic points, low range decomposition of the trajectory matrix and sparse restriction, and a sturdy homography matrix estimation. Chen [15] proposes a method of camera auto-calibration based on a geometrical analysis from four coplanar corresponding points and a fifth non-coplanar.

- **Camera calibration based on active vision** consists of acquiring several images after controlling the camera to perform a defined special movement. Afterwards, the intrinsic and extrinsic parameters can be obtained linearly using both the acquired images and the known movement trajectory. De Ma Sang [16] proposes a method of camera calibration based on triorthogonal translation movement that can calculate both intrinsic and extrinsic parameters through three to six translational movements.

- **Hybrid methods** combine more than one camera of the aforementioned categories. Loaiza et al. [2] proposes a hybrid method for stereo camera calibration using both a traditional photogrammetric method to calibrate each camera, and an auto-calibration method to compute the extrinsic parameters of the cameras related to the position between them.

2.2. Projective invariants

The transformation of 3D coordinates in the real world to 2D coordinates in the image is known as camera projection. Anyway, there is a different need to calculate the rules to transform the coordinates from a point in an image to the coordinates in another one, as well known as 2D homography [17].

This transformation must be able to calculate the coordinates on an image to an infinitely distant point; for which, Euclidean geometry is not powerful enough. However, those cases can be handled by projective geometry.

In projective geometry, parallelism and orthogonality are not necessarily present, and the distance between two points can be affected in the coordinates transformation, as seen in Figure 1. Nevertheless, there are certain properties that are not affected in projective transformations, as listed by Clemens [18], called projective invariants.

1. **Collinearity and coplanarity:** A defined set of points is considered collinear if every point rest on a same line (see Figure 1). In this way, any pair of point in R^2 will be always collinear.

   The collinearity of three points on a two-dimensional space $v_1, v_2, v_3 \in R^2$ can be defined as

   $$\begin{vmatrix} v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{vmatrix} = 0, \quad (1)$$

   where, if the determinant is different to zero, the points are not collinear. Similarly, each set of n points that belong to R^m are collinear if the distance from points $v_3, v_4, \ldots, v_n$ to the line defined by points v_1 *and* v_2 is, for each point, equal to zero.

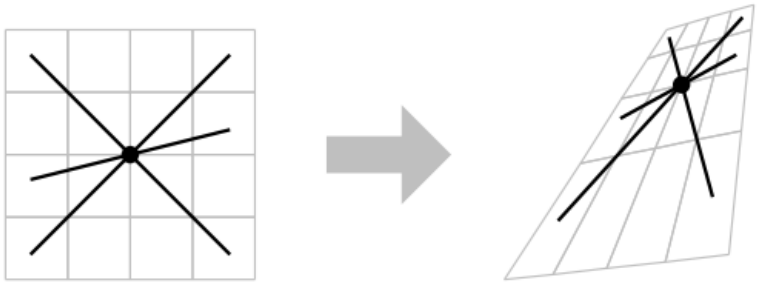
   A set of points is considered coplanar if there is a plane that contains all the set of points. In this way, any set of three points in R^3 is always coplanar.

   The coplanarity of four points in a three-dimensional space $w_1, w_2, w_3, w_4 \in R^3$ can be defined as

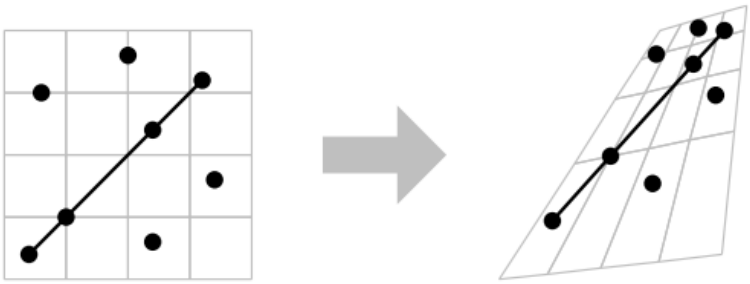
   $$\begin{vmatrix} w_1 & w_2 & w_3 & w_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0, \quad (2)$$

   where, if the determinant is different to zero, the points are not coplanar. In this way, a set of n points that belong to R^m are coplanar if the distance of the points $w_4, w_5, \ldots, w_n$ to the plane defined by the points w_1, w_2 *and* w_3 is, for each point, equal to zero.

   After any projective transformation, the collinear points stay collinear, and coplanar points stay coplanar.



(a) Projective invariants of incident lines.



(b) Projective transformation example with collinear and coplanar points.

Figure 1. Projective invariants transformations, extracted from Clemens [18].

2. **Incidence:** The incidence criteria means that three or more lines are concurrent, that is to say, they intersect each other in the same point (see Figure 1a).

   Thus, for three lines described as the equations

$$\begin{aligned} a_1x + b_1y + c_1 &= 0 \\ a_2x + b_2y + c_2 &= 0 \\ a_3x + b_3y + c_3 &= 0 \end{aligned}, \tag{3}$$

   are concurrent if

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = 0, \tag{4}$$

   where, if the determinant is not equal to zero, the lines are not concurrent.

   In the same way, a set of n lines in R^m, each one defined by two points $(v_{11}, v_{12}), \ldots, (v_{n1}, v_{n2})$ are concurrent if every possible combination of lines has the same intersection point.

3. **Cross relations:** A cross relation is the product of two relations. Hence, each cross relation needs four different values. Cross relations are invariant after any projective transformation.

   - Collinear points:

     The cross relation of the distance of four collinear points is defined as:

$$CR(A,B,C,D) = \frac{\overline{AC}}{\overline{CD}} \cdot \frac{\overline{BD}}{\overline{AD}}, \tag{5}$$

     where $\overline{XY}$ represents the Euclidian distance between point *X* and point *Y*. The points are ordered alphabetically from *A* to *D* as seen in Figure 2a.

     In this way, the cross relation of collinear points is given when we can trace a straight line that connects four points *A*, *B*, *C* and *D* so the distance between points can be replaced in Equation 5.

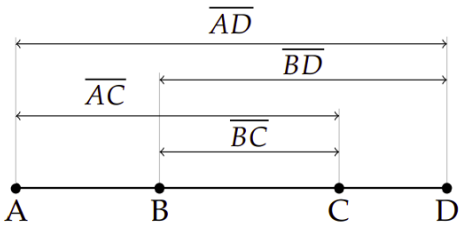
   - Concurrent lines:

     The cross relation of the angles of four concurrent lines is defined as:

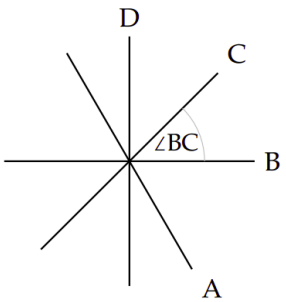
$$CR(A,B,C,D) = \frac{\sin(\measuredangle AC)}{\sin(\measuredangle CD)} \cdot \frac{\sin(\measuredangle BD)}{\sin(\measuredangle AD)}, \tag{6}$$

     where $\measuredangle XY$ represents the angle formed by the line *X* and the line *Y*. Lines are ordered alphabetically from *A* to *D* counterclockwise as seen in Figure 2b.

     In this way, the cross relation of collinear points is given when a set of four lines *A*, *B*, *C* and *D* collide in the same point. Therefore, the Equation 6 must be replaced with the angles formed between each pair of lines.



(a) Cross relation of collinear points.



(b) Cross relation of concurrent lines.

Figure 2. Cross relationships, extracted from Clemens [18].

3. PROPOSED METHOD

Our proposed method is based on an iterative control points refinement process through a frontoparallel projection and reprojection, in addition to a well distributed location of the pattern calibration control points across the size of the frame and the usage of projective invariants properties to assure the data acquisition of quality calibration control points.

We use a calibration pattern that consists of 20 concentric rings divided in 5 columns and 4 rows. The concentric rings pattern show a great performance due to the presence of two centroids that, in an ideal situation, are the same point [4]. The number of columns and rows correspond as the minimum rings quantity to recognize and validate the projective invariants properties of collinearity and angle cross relations, as well as the minimum quantity to achieve a rectangular pattern, so its orientation can be defined by the position of rows and columns in the captured image.

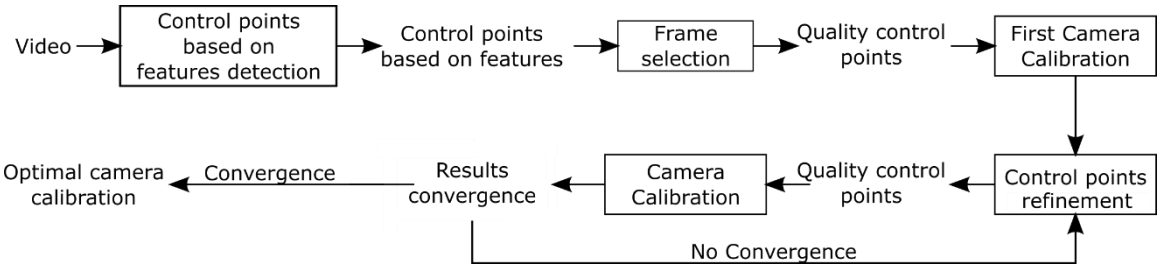


Figure 3. Pipeline of the processes performed in the proposed method.

As seen in our pipeline (Figure 3), our method is based in four principal steps:

1. **Control points based on feature detection** where we locate the position of control points in the input image based on the features of the calibration pattern used.

2. **Frame selection** where we use a grid to distribute the calibration patterns through the frame size and we evaluate the projective invariants properties in order to get a set of well distributed frames which control points satisfy invariant projective properties of collinearity and cross-angle relationship.

3. **First camera calibration** where we perform a first estimation of the camera matrix using OpenCV [7] algorithm.

4. **Control points refinement** where we look to improve the quality of the collected control points performing a frontoparallel projection and a reprojection to avoid distortions and update the correspondent control points.

5. **Camera calibration and results convergence** where we perform the camera calibration to achieve the next estimation of the camera matrix until a convergence point is reached.

3.1. Control points detection based on features detection

The control points detection step consists in four sub steps.

First, we mask the input image to cover just the minimum area containing all the calibration pattern.

Then, we apply an adaptive threshold to divide the background (the white surface of the pattern) from the foreground (the rings).

After acquiring our segmented input, we search for all the isolated ellipses using the OpenCV function *findContours*. We use a contour hierarchy where we must find an exterior contour (father) and an interior contour (son).

Finally, once we have all the ellipses in the input, we define our pattern rings and their centers. To achieve this, we must have some restraints for each ellipse.

- Each ellipse must have at least two near ellipses in a radius not longer than five times the radius of the ellipse from which the comparison is being generated.
- If we have more than 20 ellipses that fulfil the aforementioned father-son hierarchy defined in the previous sub step, we should discard ellipses with different fathers.

To define the centre of the ellipses we use the OpenCV method *fitEllipse* to get an approximation of each ring centre and then we calculate the ellipse getting the average centre of both rings.

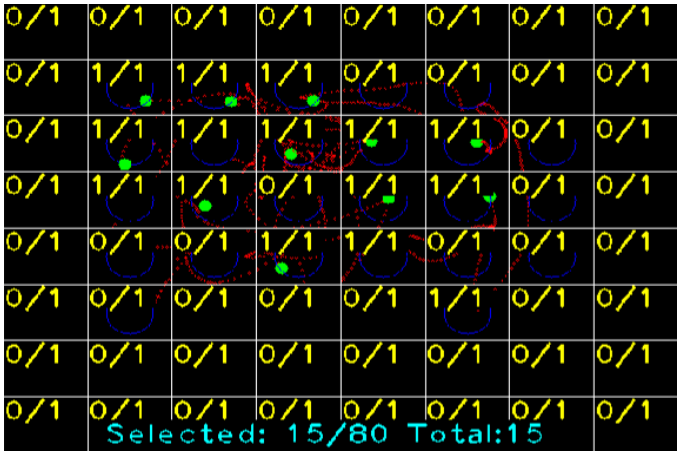
After performing this sub steps, we must have a total of 20 control points defined by the average centre of the concentric rings. In case the total of control points is less than 20, the frame is not considered as a candidate for the next steps.

3.2. Frame selection

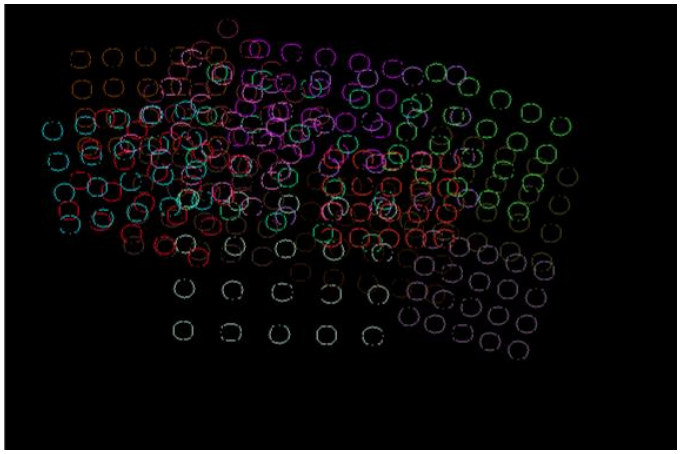
To get a precise calibration we must find frames where the control points are as well distributed as possible around the frame size. To achieve that, we define a grid which cells must contain the centre of the calibration pattern inside each acceptation area to define the frame as a potential frame.

In Figure 4 we can see an example of an *8 x 8* grid where 15 frames had been already selected. As the grid has been defined, each cell can contain a maximum of one potential frame. Thanks to this, the 15 selected potential frames are distributed across a wider section of the frame size. Otherwise, two potential problems would arise. In first instance, the selected potential frames would be agglomerated in small areas of the frame size. The second problem is that we would have to request a high quantity of potential frames, since we would accept every potential frame from the aforementioned agglomerations.

Once we define a potential frame based on its position along the frame size, we evaluate the projective invariants properties of its calibration pattern. We evaluate the collinearity and the cross-angle relationship of the control points of the pattern.



(a) Grid of 8 × 8 quadrants with some calibration patterns accepted.



(b) Control points distribution from the calibration pattern in the selected frames.

Figure 4. Frame selection process.

Collinearity: We evaluate the collinearity of the points that belong to each line as seen in Figure 5, where we can see the set of collinear points matched by red, yellow, blue and green lines for the vertical, horizontal and both diagonals respectively. Also, we are able to see black lines that represent the set of points that will not be evaluated because they will always be collinear.

In this way, we define 17 sets of collinear points distributed in four possible orientations:

- **Horizontal collinear points:**
    1. *point [1], point [2], point [3], point [4] & point [5]*
    2. *point [6], point [7], point [8], point [9] & point [10]*
    3. *point [11], point [12], point [13], point [14] & point [15]*
    4. *point [16], point [17], point [18], point [19] & point [20]*
- **Vertical collinear points:**
    5. *point [1], point [6], point [11] & point [16]*
    6. *point [2], point [7], point [12] & point [17]*
    7. *point [3], point [8], point [13] & point [18]*
    8. *point [4], point [9], point [14] & point [19]*
    9. *point [5], point [10], point [15] & point [20]*

- **Top left to bottom right collinear points:**
    10. *point [11], point [7] & point [3]*
    11. *point [16], point [12], point [8] & point [4]*
    12. *point [17], point [14], point [9] & point [5]*
    13. *point [18], point [14] & point [10]*
- **Bottom left to top right collinear points:**
    14. *point [6], point [12] & point [18]*
    15. *point [1], point [7], point [13] & point [19]*
    16. *point [2], point [8], point [14] & point [20]*
    17. *point [3], point [9] & point [15]*

We trace the best line approach between all the supposedly collinear set of points. Then, we calculate the Euclidean distance between each of these points and the traced line. If this distance is less or equal to our threshold then it is accepted as a collinear point. In case one or more of the set of points are not collinear, the frame is discarded as a potential calibration frame.

In Figure 7a we can see an accepted frame with collinear control points represented by blue lines that match all the defined sets of collinear control points.

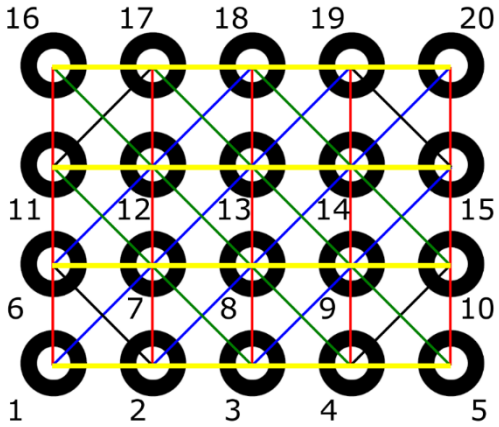


Figure 5. Collinear points in the calibration pattern.

Cross-angle relationship: Once the collinearity is successfully evaluated, we proceed to evaluate the cross-angle relationship of the inner quadrant of the pattern (*point[6], point[7], point[8], point[11], point[12], point[13]*).

We calculate the cross-angle relationship of each point as

$$\frac{\sin(90°)}{\sin(45°)} \cdot \frac{\sin(90°)}{\sin(135°)} \cong \frac{\sin(\measuredangle AC_{ij})}{\sin(\measuredangle CD_{ij})} \cdot \frac{\sin(\measuredangle BD_{ij})}{\sin(\measuredangle AD_{ij})} \tag{7}$$

where *i* is the frame being evaluated and *j* is the evaluated point. The angles are formed as seen in Figure 6, where *A*, *B*, *C* and *D* are enumerated counterclockwise represented by a red, yellow, green and blue line respectively. If the difference of the calculated angle relationship is greater than our threshold, then the frame is discarded as a potential calibration frame.

In Figure 7b we can see an accepted frame with the respective cross-angle relationship for each point of the inner quadrant.

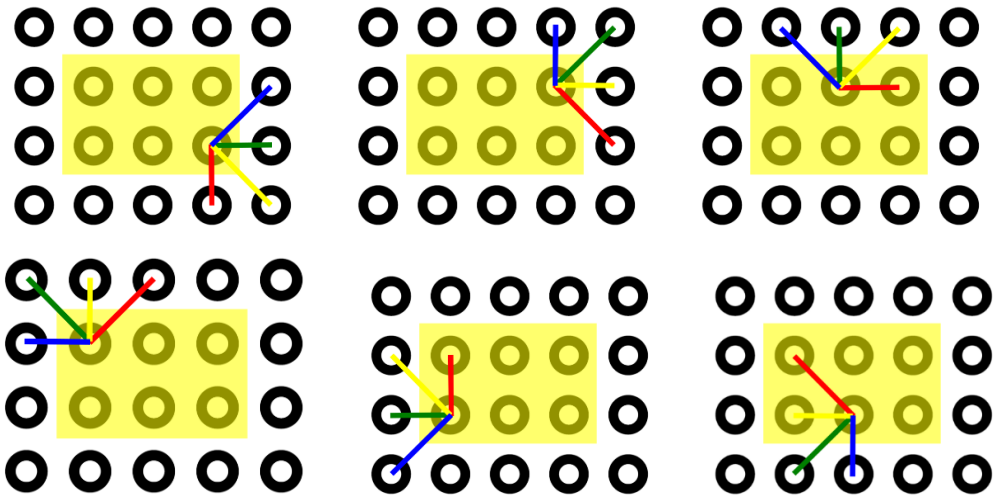
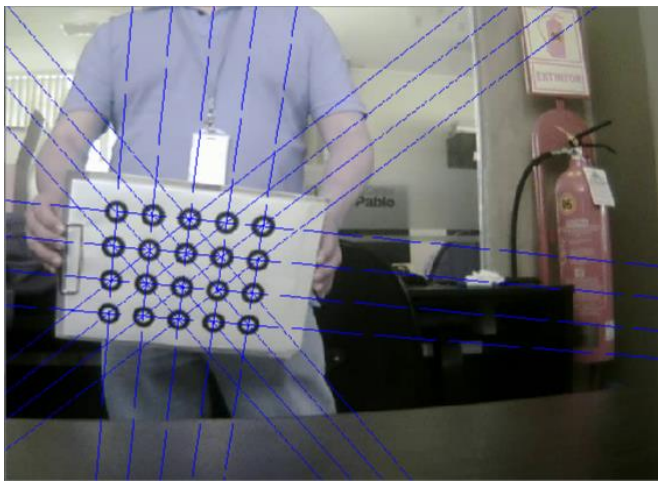
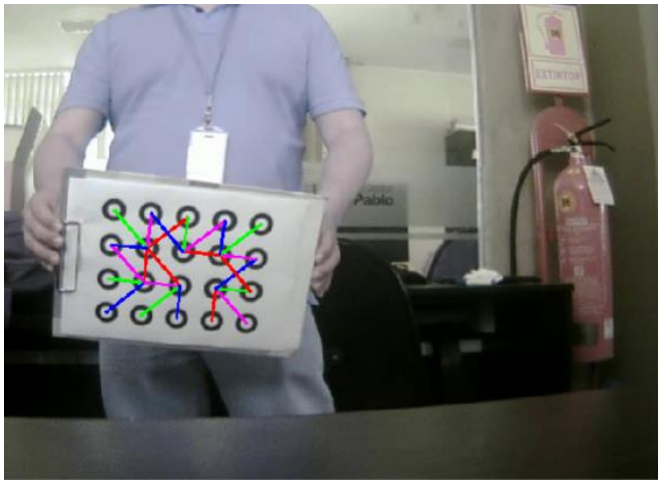


Figure 6. Cross-angle relationship detail of the calibration pattern inner quadrant.



(a) Collinearity of control points found in a frame



(b) Cross-angle relationship of control points found in a frame.

Figure 7. Evaluation of collinearity and cross-angle relationship in a frame.

3.3. First camera calibration

We perform a first camera calibration using all the calibration frames that were not discarded in the previous step. We use the camera calibration algorithm implemented in OpenCV to get an initial estimation of the camera matrix.

This step is crucial and must always be done because the first estimation of the camera matrix is needed to perform the next step. As well, the RMS of this calibration is the base line we look forward to improve in the subsequent steps, as this step performs a non-optimized calibration. In other words, it performs the calibration implemented in OpenCV as detailed by Zhang [6] and it does not take advantage of the control points refinement and their projective invariants properties evaluation as detailed further in this section.

3.4. Control points refinement

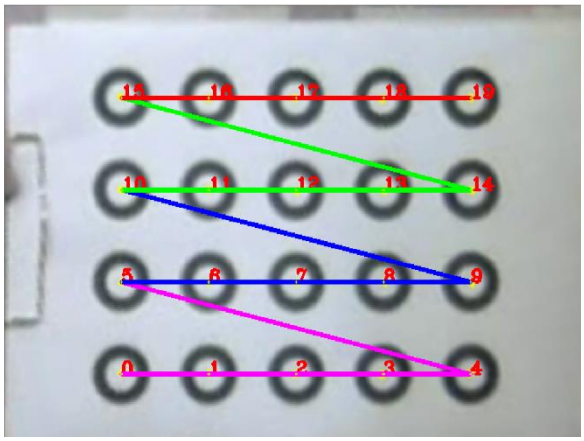
We use the calculated parameters of the first camera calibration to execute a refinement process of the control points. First, we undistort the input image and reproject it in a frontoparallel plane in the world coordinate system as seen in Figure 8a. In this projection we proceed to do a search for the control points. This search must be faster since the image has been undistorted and unprojected, therefore the location and orientation of the calibration pattern grants us a better distinction of where the control points must be located.

Once we find all the control points in the frontoparallel projection, we proceed to reproject them in the original image using the first camera calibration calculated parameters into the camera coordinate system. We use the set of control points found in the original image (old points) as well as the reprojected set of control points (new points) to update the new set of points:

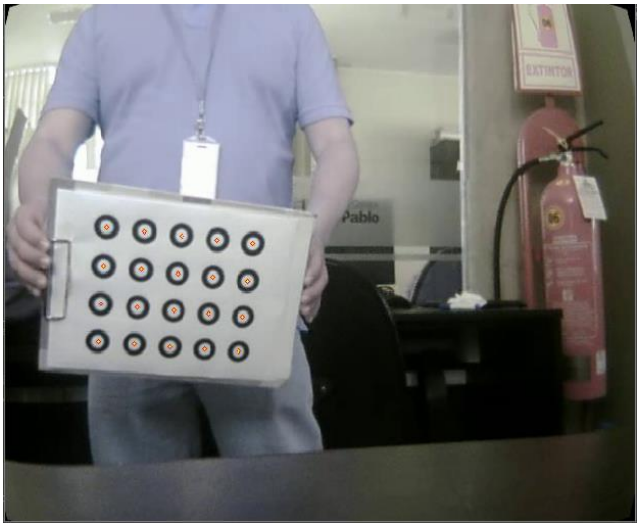
1. For each row we fit a line using the respective points from the new set of points.
2. For each point in the row
    a. Calculate the distance of the original control point to the line.
    b. Calculate the distance from the reprojected point to the line.
    c. Calculate the blend factor as the proportion of each distance to the line giving a priority to the closest one, where the factor is less or equal than 1.
    d. Update the control point using the blend factor

$$newPoint = newPoint * factor + oldPoint * (1 - factor) \quad (8)$$

Once we update our set of control points as seen in Figure 8b, we proceed with the projective invariants properties evaluation, as defined in the second step of the process. If the updated control points do not satisfy the projective invariants properties, then the frame is discarded.



(a) Frontoparallel projection of the calibration pattern.



(b) Control points reprojection in original image.

Figure 8. Control points refinement process

3.5. Results convergence

Once the refinement process is completed, we perform another camera calibration using the same algorithm implemented in OpenCV.

The obtained RMS is then compared to the previous one (the RMS from the first iteration is compared with the RMS from the first camera calibration). If the absolute value of our RMS difference is greater than our convergence threshold, then the obtained camera matrix and RMS becomes the new data to perform the control points refinement process into the next iteration.

Otherwise, if the absolute value of the RMS difference is less or equal to the defined threshold, then we successfully achieved an optimal camera calibration. We can see an example of this process in Table 1 and the graphic results on Figure 9, where our convergence threshold is defined as 0.00005. We can see how, as we iterate, the absolute value of the RMS difference keeps being greater than our threshold. However, when we reach iteration number eight, we get an absolute value of the RMS difference minor to our threshold, consequently we take the calibration of this iteration as optimal.

Table 1. Iterative process and results convergence on PS3 Eye Camera.

# Iteration	Fx	Fy	Uo	Vo	RMS	Convergence Value
First Calibration	716.319	722.284	325.828	244.346	0.33587	-
1	696.936	703.097	330.429	246.309	0.262235	0.073635
2	673.073	680.266	331.41	251.325	0.251114	0.011121
3	670.564	677.487	331.131	257.299	0.23994	0.011174
4	692.207	698.407	334.173	262.209	0.232683	0.007257
5	698.262	704.455	335.011	262.948	0.227108	0.005575
6	699.578	705.77	335.239	263.433	0.227299	0.000191
7	698.871	705.075	335.145	263.302	0.227081	0.000218
8	**699.119**	**705.294**	**335.176**	**263.252**	**0.227034**	**0.000047**

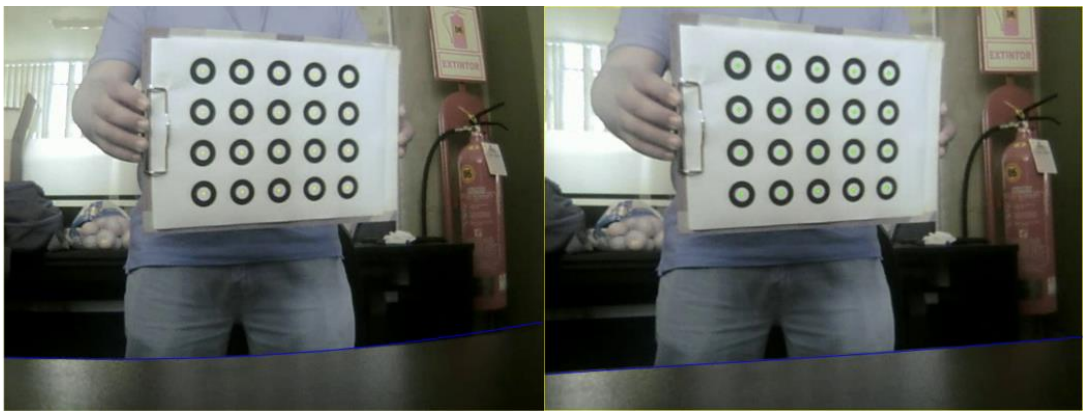


Figure 9. Captured frame from PS3 Eye camera before performing calibration (on the left) and after performing the proposed method (on the right).

4. TEST ENVIRONMENT AND RESULTS

The purpose of the performed experiments was to achieve the best possible configuration of the proposed method as well as demonstrate the improvement granted by the use of projective invariants properties in our method in comparison with a widely used method like OpenCV camera calibration.

For the acquisition of results, we ran multiple experiments in multiple devices. We specify the four camera devices used as well as the three sets of experiments performed with the proposed method in the following sections.

4.1. Test equipment

To perform the tests, four webcams were used:

- *HP Truevision* integrated camera of a *HP Envy 15*, with 640 x 480 px resolution and 30 FPS image rendering.
- *Genius FaceCam 1000X* analog camera, with 640 x 480 px resolution and 30 FPS image rendering.
- *PS3 Eye Camera*, with 640 x 480 px resolution and 60 FPS image rendering.
- *Logitech Brio 4k Pro Webcam*, with 640 x 480 px resolution and 24 FPS image rendering.

4.2. Test methodology

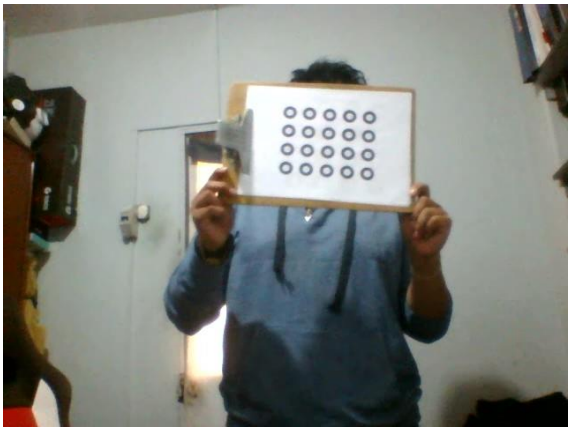
We performed three experiment sets consisting in the algorithm selection, the relative distance of the calibration pattern to the pinhole camera and the performance of the best selected algorithm.

To perform all the experiments, we used a threshold of 0.6 px for the collinearity error, 0.06 for the cross-angle relationship error and we defined a 0.00005 convergence threshold to stop the iterative process considering the quality of the cameras used to perform the experiments which would not allow us to constraint the errors to a lesser value. In the same way, we define the initial size of the grid as *8 x 8* due to the size of the frame, given that the grid cells are small enough to contain the centre of the calibration point at a long distance as far as the calibration pattern is still recognizable for the algorithm considering the quality of the cameras.

1. **Algorithm selection:** We performed an OpenCV calibration compared to four versions of the proposed algorithm to define which one grants better results.
    - **Iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern as proposed by Datta et al. [4] and Prakash et al. [19].
    - **Collinear iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points collinearity.
    - **Cross-angle relationship iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points cross-angle relationship.
    - **Iterative with projective invariants properties**: OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points collinearity and cross-angle relationship.

    To perform these experiments, we require a minimum set of 50 frames. We perform this experiment on three cameras (*HP Truevision*, *PS3 Eye Camera*, *Genius FaceCam 1000x)* with a limit of 10 iterations. Since the number of iterations is low, the minimum set of frames does not need to be high considering that not many frames are going to be discarded in the iterative process.
2. **Relative distance of the calibration pattern:** We perform the proposed method on four different recording of the same camera (*HP Truevision*) where we positioned the calibration pattern at approximately half a meter (Figure 10a), a meter (Figure 10b), meter and a half (Figure 10c) and varied distances identified as close, medium, far and varied respectively to determine which is the best relative distance of the calibration pattern to the camera to perform a camera calibration. To perform this experiment, we require a minimum set of 80 frames since the number of discarded frames is higher due to the verification of projective invariants properties.

(a) Calibration pattern located at approximately half a meter of distance from the camera.



(b) Calibration pattern located at approximately a meter of distance from the camera.

(c) Calibration pattern located at approximately a meter and a half of distance from the camera.

Figure 10. Relative distances of the calibration pattern to the camera

3. **Performance of the best selected algorithm:** We test the performance of the best algorithm from the first experiment combined with the distance of the second experiment to get the general performance of the best selected algorithm. To perform this experiment, we require a minimum set of 80 frames, due to the frames discarded by the projective invariants properties check.

4.3. Results

After performing the explained experiments, we achieve the results detailed in Table 2, Table 3 and Table 4. The results of the first and second experiment (algorithm selection and relative distance of the calibration pattern) affect directly in the last experiment, where we perform the tests with the selected algorithm of the first experiment, and we position the calibration pattern according to the results of the second experiment.

1. **Algorithm selection.** As seen in Figure 11, each version of the proposed method represents a significant improvement in comparison with the OpenCV method which is widely used nowadays. Also, we can see in Table 2 that the improvement percentage is always better when projective invariants properties are applied.

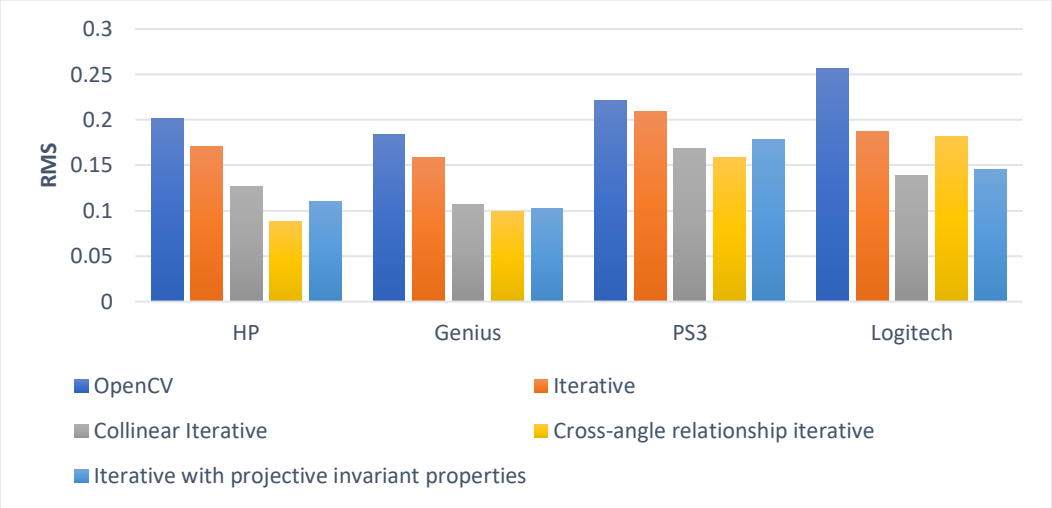


Figure 11. Minimum RMS by algorithm by camera

Table 2. Best improvement percentage by algorithm by camera.

		HP Truevision	Genius Facecam 1000x	PS3 Eye Camera	Logitech Brio 4k Pro Cam
OpenCV	RMS	0.201465	0.183917	0.221083	0.256287
Iterative [4,19]	RMS	0.170335	0.158278	0.209313	0.187268
	% Improvement	15.45	13.94	5.32	26.93
Collinear Iterative	RMS	0.12677	0.106606	0.168134	**0.13837**
	% Improvement	37.08	42.04	23.95	**46.01**
Cross-Angel Relationship Iterative	RMS	**0.088209**	**0.0987823**	**0.159093**	0.181949
	% Improvement	**56.22**	**46.29**	**28.04**	29.01
Iterative with projective invariants properties	RMS	0.110443	0.102717	0.178917	0.145617
	% Improvement	45.18	44.15	19.07	43.18

2. **Relative distance of the calibration pattern.** We can see in Figure 12 how the camera calibration has a better performance when the calibration pattern gets further away from the camera. However, the best performance is reached when the distance is varied containing frames captured with the calibration pattern at diverse distances to the camera. In Table 3 we can notice that the number of iterations grows as we get the calibration pattern further away from the camera. Nevertheless, we get an average quantity when the distance is varied.

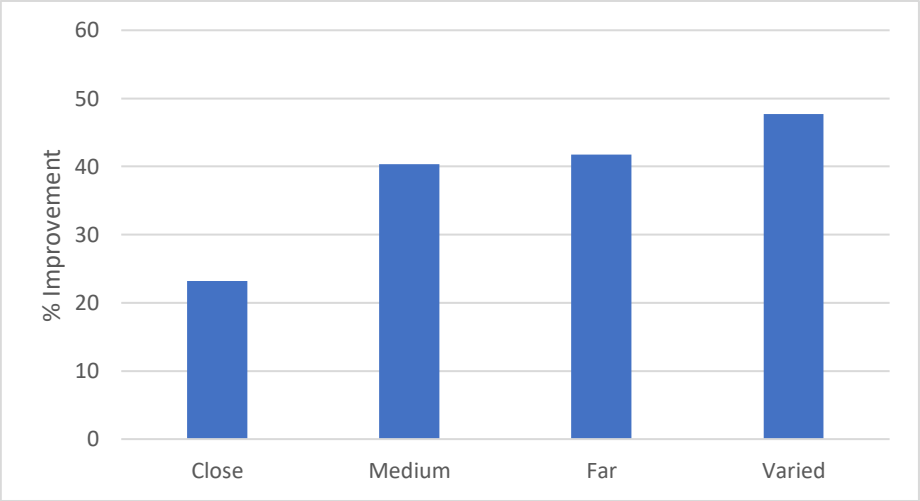


Figure 11. Minimum RMS by algorithm by camera

Table 3. Number of iterations and improvement percentage by the relative distance of the calibration pattern to the camera.

	Close (Half a meter)	**Medium (A Meter)**	**Far (A meter and a half)**	**Varied**
# It	75	**29**	87	65
RMS – OpenCV	0.310345	0.205439	0.173331	**0.170441**
RMS – Our Method	0.238371	0.122536	0.100963	**0.0890842**
% Improvement	23.19	40.35	41.75	**47.73**

3. **Performance of the best selected algorithm.** In Table 4 we observe the performance of the method proposed in all the aforementioned devices. We include the four recordings of *HP Truevision* cameras as well as the recordings of each other camera at varied distances.

Table 4. Improvement percentage and number of iterations of the best selected algorithm.

	HP Truevision				**Genius Facecam 1000x**	**Logitech Brio**	**PS3 Eye Camera**
	Close	Medium	Far	Varied			
# It	33	111	157	55	159	252	9
RMS – OpenCV	0.313	0.2131	0.1736	0.1717	0.1589	0.2317	0.2139
RMS – Our Method	0.2197	0.1274	0.0962	0.103	0.1083	0.144	0.1787
% Improvement	**29.81**	**40.22**	**44.59**	**40.01**	**31.84**	**37.85**	**16.46**

5. ACCURACY COMPARISON

Finally, this section analyzes the accuracy of our method and compares it with other available methods. To perform the comparisson, we use the same setup in each set of results.

We use AprilTag [20] to detect the distance between camera and the pattern. To perform the tests, we use a setup that consists on the pattern tied to the camera with a nylon thread, to reduce obstruction to the minimum, with a radius of 30.7 cm. Also, the position of the pattern was placed throughout all the frame to detect possible distortions in the borders.

We perform these tests with four different variations:

1. **OpenCV Distorted:** The basic camera calibration using OpenCV without performing an undistortion of the evaluated frames.
2. **OpenCV Undistorted:** The basic camera calibration using OpenCV after performing the undistortion of the evaluated frames with the distortion coefficients provided by the calibration method.
3. **Our Method Distorted:** The camera calibration using our method without performing an undistortion of the evaluated frames.
4. **Our Method Undistorted:** The camera calibration using our method after performing the undistortion of the evaluated frames with the distortion coefficients provided by the calibration method.

We compare our method with the calibration parameters calculated by the calibration toolbox provided by Bouguet [21] which is used as ground trouth in the literature [9,10,11,12,15] in both versions, distorted and undistorted.

As we can see on Table 5, our proposed method is the most accurate, gaining an improvement of 16.32% compared to the OpenCV method and reaching up to an improvement of 18.75% when a frame undistortion is performed.

Table 5. Accuracy comparison of the proposed method.

Method	**Average distance detected**	**Average Error**	**% Improvement**
OpenCV Distorted	30.32345931	0.544874483	0
OpenCV Undistorted	30.38143345	0.497539655	8.69
Our Method Distorted	30.67633564	0.455933455	16.32
Our Method Undistorted	**30.80322836**	**0.442726545**	**18.75**
Bouguet Distorted [21]	31.55786665	0.955424898	-75.35
Bouguet Undistorted [21]	32.07657612	1.45828784	-167.64

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the modelling and implementation of a new camera calibration approach. The main novelty is the use of projective invariants properties to grant the quality of control points in data collection. Tests demonstrate that the proposed method brings a great improvement in camera calibration quality compared with one of the most used methods (OpenCV). Also, the sector segmentation brings a robust calibration thanks to the well distributed control points.

The proposed method brings its best performance when the calibration pattern is allocated in varied distances during data collection. In addition, the use of a convergence threshold grants us the certainty to achieve an optimum camera calibration. Although, the number of iterations remains variable in each case.

Additionally, the use of projective invariants properties allows us to discard false positive potential frames in frame selection caused by distortion in the frontoparallel projection and reprojection processes.

In the future, we propose to study the relation between average collinearity and average cross-angle relationship errors and camera distortion so that the static distortion parameters in camera calibration could be selected efficiently and automatically. In addition, we propose to model a new cost function to validate the reprojection error of the pattern centroids. Also, we propose the implementation of techniques that automatically improve the acquisition of quality control points to increase the robustness of the proposed method based on distance of the calibration pattern from the camera or the angle of the calibration pattern.

ACKNOWLEDGEMENTS

We acknowledge the financial support of the "Proyecto Concytec - Banco Mundial", through its executing unit "Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica (Fondecyt)", for his research work entitled "Reconstrucción y modelado 3D de las superficies de componentes y piezas de maquinaria pesada usada en Minería, con nivel de precisión milimétrica, para su aplicación en un nuevo proceso optimizado de manutención especializada".

REFERENCES

[1] Walters, D. (2003) Computer Vision, p. 431–435. John Wiley and Sons Ltd., GBR.

[2] Loaiza, M.E., Raposo, A.B., Gattass, M. (2011) Multi-camera calibration based on an invariant pattern. Computers & Graphics 35(2), 198 – 207.

[3] Gai, S., Da, F., Fang, X. (2016) A novel camera calibration method based on polar coordinate. PLOS ONE 11(10), 1–18.

[4] Datta, A., Kim, J., Kanade, T. (2009) Accurate camera calibration using iterative refinement of control points. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. pp. 1201–1208.

[5] Vo, M., Wang, Z., Luu, L., Ma, J. (2011) Advanced geometric camera calibration for machine vision. Optical Engineering 50(11), 110503.

[6] Zhang, Z. (2000) A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 1330 – 1334.

[7] Bradski, G., (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[8] Song, L., Wu, W., Guo, J., Li, X. (2013) Survey on camera calibration technique. In: 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics. vol. 2, pp. 389–392.

[9] Arık O. and Yuksel S.E. (2022) Camera Calibration Using Catenary. IEEE Sensors Journal, vol. 22, no. 6, pp. 5962-5968.

[10] Wong K. -Y. K., Zhang G. and Chen Z. (2011) A Stratified Approach for Camera Calibration Using Spheres. IEEE Transactions on Image Processing, vol. 20, no. 2, pp. 305-316.

[11] Fu, Q., Quan, Q. and Cai, K.-Y. (2015), Calibration of multiple fish-eye cameras using a wand. IET Comput. Vis., 9: 378-389.

[12] Liu W., Wu S., Wu X. and Zhao H. (2019) Calibration Method Based on the Image of the Absolute Quadratic Curve. IEEE Access, vol. 7, pp. 29856-29868.

[13] Sarmadi H., Muñoz-Salinas R., Berbís M. A. and Medina-Carnicer R., (2019) Simultaneous Multi-View Camera Pose Estimation and Object Tracking With Squared Planar Markers. IEEE Access, vol. 7, pp. 22927-22940.

[14] Yao, Q., Nonaka, K., Sankoh, H., Naito, S. (2016) Robust moving camera calibration for synthesizing free viewpoint soccer video. In: 2016 IEEE International Conference on Image processing (ICIP). pp. 1185–1189.

[15] Chen H. (2017) Geometry-Based Camera Calibration Using Five-Point Correspondences From a Single Image. IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 12, pp. 2555-2566.

[16] Sang De Ma (1996) A self-calibration technique for active vision systems. IEEE Transactions on Robotics and Automation 12(1), 114–120.

[17] Hartley, R.I., Zisserman, A. (2004) *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[18] Clemens, K. (2016) Object Tracking using Projective Invariants. Ph.D. thesis.

[19] Prakash, C.D., Karam, L.J. (2012) Camera calibration using adaptive segmentation and ellipse fitting for localizing control points. In: 2012 19th IEEE International Conference on Image processing. pp. 341–344

[20] J. Wang and E. Olson. (2016) “AprilTag 2: Efficient and robust fiducial detection” in Proceedings of the IEEE/RSJ International Conferenceon Intelligent Robots and Systems (IROS),

[21] J. Y. Bouguet J. Y. (2004) Camera calibration toolbox for matlab. Available: http://www.vision.caltech.edu/bouguetj/calib doc/index.html.

AUTONOMOUS VEHICLES LATERAL CONTROL UNDER VARIOUS SCENARIOS

Mohamed Ali Jemmali and Hussein T. Mouftah

School of Electrical Engineering and Computer Science,
University of Ottawa, Ontario, Ottawa

ABSTRACT

In this paper, the autonomous vehicle presented as a discrete-time Takagi-Sugeno fuzzy (T-S) model. We used the discrete-time T-S model since it is ready for the implementation unlike the continuous T-S fuzzy model. The main goal is to keep the autonomous vehicle in the centreline of the lane regardless the external disturbances. These disturbances are the wind force and the unknown curvature; they are applied to test if the autonomous vehicle moves from the centreline. To ensure that the autonomous vehicle remain on the centreline we propose a discrete-time fuzzy lateral controller called also steering controller.

KEYWORDS

Takagi-Sugeno model, Steering control, lane keeping, observers.

1. INTRODUCTION

Nowadays, vehicles have become an integral part of our daily lives. Most people worldwide need vehicles to move around, such as cars, buses, bicycles, taxis, etc. Also, those who are physically challenged use special vehicles. Due to the constant use of vehicles globally, so many people die every day because of vehicle collisions. Many studies declare that the main causes of these accidents are inattention, drowsiness, and illness [1] [2]. The purpose of autonomous cars lateral control is to maintain the vehicle in the lane under varied limits, and it is one of the most significant safety solutions. Over the last two decades, lateral stability control for autonomous vehicles has gotten a lot of attention from scientists and engineers, and numerous findings have been published [3]–[4]. Therefore, fruitful results with regard to stability and stabilization have been developed by researchers. By merging fuzzy logic and PID control, several authors propose a novel lateral control system design [5]. Others worked on controlling saturation through robust yaw control and stabilising lateral dynamics with concerns of parameter uncertainty. [6] A robust yaw-moment controller architecture for increasing vehicle handling and stability has been designed, taking into account parameter uncertainty and control saturation. However, the state vector only has two controllable parameters: the yaw rate and the sideslip angle; in this instance, there is insufficient information about the condition of the vehicles to allow for robust control. According to Sun et al. [7], the proposed proposal lacks a mathematical model, implying that no mathematical stability and stabilisation criteria exist to get access to system control. On the other hand, several researchers used deep learning and reinforcement learning approaches to examine vision-based autonomous driving [8]. Despite this, the model has no constraints, such as lateral wind force applied to the cars, unknown road curvature, or steering physical saturation. This means that the proposed solution is still far from being applicable in real-world driving conditions.Jiang and Astolfi [9] investigated an asymptotic stabilisation issue for a class of

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL, CYBI - 2022
pp. 273-286, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120922

nonlinear under actuated systems. Its solution is used in the control of a vehicle's nonlinear lateral dynamics, together with back stepping and forward control design approaches. This approach demonstrated that a vehicle may track any conceivable reference at a constant speed using the established controller, and the lateral deviation converges to zero. The challenges in this scenario are that the longitudinal vehicle speed is constant, there is only one controllable internal variable, and the stability is local. Based on the aforementioned rationale, lateral control of autonomous cars requires a system that provides access to several vehicle states and swipes into a wide range of longitudinal speed. As a result, the Takagi–Sugeno (T–S) fuzzy models [10] have been widely acknowledged and used. The T-S is well-known as a valuable and popular paradigm for approximating complicated nonlinear systems. The nonlinear systems were approximated by fuzzy "blending" local linear models using a set of "IF-THEN" rules, which has piqued the control community's curiosity. In this paper, we will focus on discrete-time T-S fuzzy control. This controller is essentially based on feedback control, more specifically the parallel distributed control (PDC) law. In addition, the state vector will contain six internal variables to allow more accessibility for the autonomous vehicle control. In most cases, the state vector could be unreadable, noisy, or completely inaccessible. Based on this motivation, in our work we developed an observer called Luenberger multiobservers to ensure the reconstruction of the system state vector for accurate automatic steering control. This state vector will indeed be used in the control law equation for the fuzzy controller design. The stability and stabilization conditions will be based on the quadratic Lyapunov function. The Linear Matrix Inequality (LMI) approach is used in the optimization.

This paper is presented as follows. The vehicle modelling which is the part who is consecrated for the vehicle parameters and models. The third part presents the control design for the autonomous vehicle, which focuses on the stabilization of the autonomous vehicle. The final part is consecrated to show the results.

2. VEHICLE PARAMETERS AND MODELS

In this part, we show the different steps for the vehicle modelling. We start by introducing the vehicle parameters given in Table 1:

Table 1. Definition of parameters

Parameters	**Description**	**Value**
B_s	Steering system damping	5.73
C_f	Front cornering stiffness	57000 N/rad
C_r	Rear cornering stiffness	59000 N/rad
I_s	Steering system moment of inertia	0.02 kgm2
I_z	Vehicle yaw moment of inertia	2800 kgm2
K_p	Manual steering column coefficients	0.5
l_f	Distance from the CG to the front axle	1.3 m
l_r	Distance from the CG to the rear axle	1.6 m
l_s	Look-ahead distance	5 m
l_w	Distance from the CG to the impactcenter of the wind force	0.4 m
M	Mass of the vehicle	2025 kg

R_s	Steering gear ratio	16
σ_t	Tire length contact	0.13 m

2.1. Road-vehicle model

Using the well-known bicycle model described in [11]. The acquired model of the vehicle lateral dynamics is expressed as:

$$\begin{bmatrix}\dot{\beta}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}\beta\\ r\end{bmatrix}\delta + \begin{bmatrix}e_1\\ e_2\end{bmatrix} f_w. \quad (1)$$

Where β is the sideslip angle at the center of gravity (CG), and the yaw rate is r shown in Figure 1. In (1), the lateral wind force is fw, and the elements of the system matrices are written as follows:

$$a_{11} = -2\frac{(C_r + C_f)}{m\vartheta_x}\,;\; a_{12} = 2\frac{(l_r C_r - l_f C_f)}{m\vartheta_x^2} - 1\,;$$
$$a_{21} = 2\frac{(l_r C_r - l_f C_f)}{I_z}\,;\; a_{22} = -2\frac{\left(l_r^2 C_r + l_f^2 C_f\right)}{I_z \vartheta_x}.$$
$$b_1 = \frac{2C_f}{m\vartheta_x}\,;\; b_2 = \frac{2l_f C_f}{I_z}\,;\; e_1 = \frac{1}{m\vartheta_x}\,;\; e_2 = \frac{l_w}{I_z}\,.$$

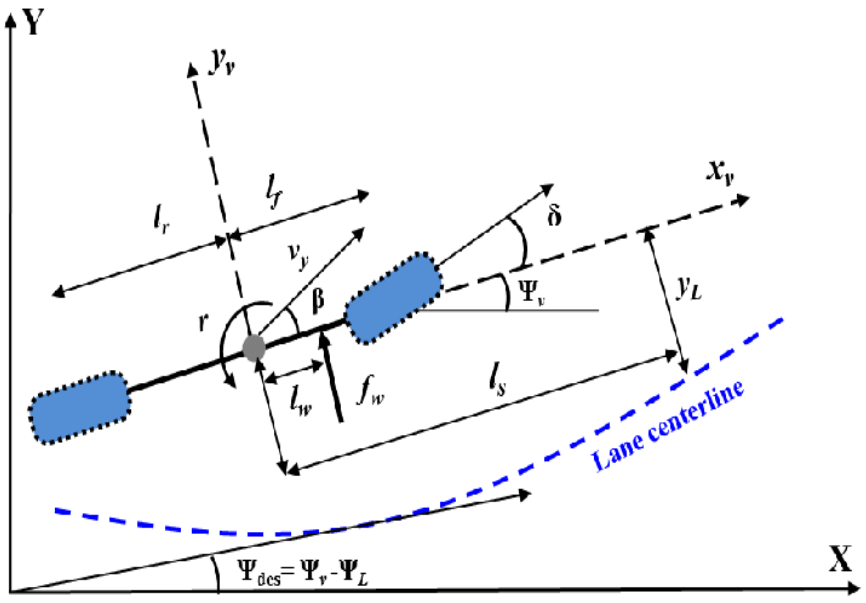


Figure 1. Lateral vehicle behavior modeling.

2.2. Vehicle position model on the road

The vehicle positioning dynamics on the road is described by [11]:

$$\begin{cases}\dot{y}_L = \vartheta_x\beta + \dot{l}_s r + \vartheta_x\psi_L.\\ \dot{\psi}_L = r - \vartheta_x\rho_r.\end{cases} \quad (2)$$

Where y_L is the lateral deviation error from the centerline of the lane projected forward a look ahead distance l_s and ψ_L is the heading error between the tangent to the road and the vehicle orientation. The road curvature is indicated by ρ_r.

2.3. The vehicle steering model

The electronic power steering system is presented as [3]:

$$\ddot{\delta} = 2\frac{K_p C_f \sigma_t}{R_s^2 I_s}\beta + 2\frac{K_p C_f \sigma_t}{R_s^2 I_s}\frac{l_f}{\vartheta_x} r \cdots - 2\frac{K_p C_f \sigma_t}{R_s^2 I_s}\delta - \frac{B_s}{I_s}\dot{\delta} + \frac{1}{R_s I_s} T_s. \quad (3)$$

Where T_s is the steering torque, δ is the steering angle, l_s is the inertia moment of the steering column, B_s is the damping factor of the column, R_s is the reduction ratio of the column, σ_t is the width of the tire contact finally, the manual steering column coefficient is K_p.

2.4. The autonomous vehicle model

Based on (1), (2) and (3) the autonomous vehicles model is:

$$\dot{x}(t) = A_v x(t) + B_{vu} u(t) + B_{vw} w(t). \quad (4)$$

Where $x = \left[\beta r \psi_L y_L \delta \dot{\delta}\right]^T$ is the vehicle vector state, $w = [f_w \rho_r]^T$ is the disturbance vector, and $\mathrm{u} = \mathrm{T_s}$ is the input vector. The control-based system matrices in (4) are expressed as:

$$A_v = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & b_1 & 0 \\ a_{21} & a_{22} & 0 & 0 & b_2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ v_x & l_s & v_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_{61} & a_{62} & 0 & 0 & a_{65} & a_{66} \end{bmatrix}; B_{vlw} = \begin{bmatrix} e_1 & 0 \\ e_2 & 0 \\ 0 & -v_x \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \mathrm{B_v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\mathrm{R_s I_s}} \end{bmatrix}.$$

We have

$$a_{61} = 2\frac{K_p C_f \sigma_t}{R_s^2 I_s},\ a_{62} = 2\frac{K_p C_f \sigma_t}{R_s^2 I_s}\frac{l_f}{v_x},\ a_{65} = -2\frac{K_p C_f \sigma_t}{R_s^2 I_s},\ a_{66} = -\frac{B_s}{I_s}.$$

As a result, the control analysis and development in this work will be based on the discrete-time system described below (5).

$$\mathrm{x(k+1) = A\,x(k) + B_d\,u(k) + B_{lw} w(k)} \quad (5)$$

$\mathrm{Te} = 0.01$ second is the discretization time. A is the discrete-time matrix of $\mathrm{A_v}$, $\mathrm{B_d}$ is the discrete-time matrix of $\mathrm{B_v}$ and $\mathrm{B_{lw}}$ is the discrete-time matrix of B_{vlw}.

3. AUTONOMOUS VEHICLE STABILIZATION ARCHITECTURE

In this section, we present the control design steps for the aim purpose is to control the autonomous vehicle presented in (5).

3.1. The autonomous vehicle T-S model

The discrete-time (T-S) system is presented by fuzzy IF-THEN rules.

The Rule i is of the form:

IF $\mathrm{E_1(k)}$ is $Q_{1\mathrm{i}}$ and ... $\mathrm{E_n(k)}$is $\mathrm{Q_{ni}}$ THEN

$$\begin{cases} \mathrm{x(k+1) = A_i x(k) + B_i u(k).} \\ \mathrm{y(k) = C_i x(k).} \end{cases} \quad (6)$$

Where $E_1, \dots, E_n$ are linear functions, $Q_{1\mathrm{i}}, \dots . Q_{ni}$ are the fuzzy sets, and n is the number of the rules. In addition, $x_k \in \mathbb{R}^n$ is the state vector; $u_k \in \mathbb{R}^p$ is the measurable output vector; A_i, B_i and C_i are the system matrices with appropriate dimension, In addition, the premise variables are represented by the vector $\Gamma(k) = [\Gamma_1(k) \dots . \quad \Gamma_q(k)]$. The autonomous vehicle model is:

$$\mathrm{x(k+1) = \sum_{i=1}^{n} \omega_i \big(\Gamma(k)\big)\big(A_i x(k) + B_i u(k) + B_i^w w(k)\big).} \quad (7)$$

$$\begin{cases} \sum_{\mathrm{i=1}}^{\mathrm{n}} \omega_{\mathrm{i}} \big(\Gamma(\mathrm{k})\big) = 1 \\ \omega_{\mathrm{i}}(\Gamma(\mathrm{k})) \geq 0. \end{cases} \quad (8)$$

With

$$\omega_{\mathrm{i}}(\Gamma) = \frac{\mathrm{w_i}(\Gamma(\mathrm{t}))}{\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{w_i}(\Gamma(\mathrm{t}))} \; ; \; \mathrm{w_i}\big(\Gamma(\mathrm{t})\big) = \prod_{\mathrm{j=1}}^{\mathrm{q}} \mathrm{M_{ji}}(\Gamma(\mathrm{t})).$$

The proposed Lyapunov function is:

$$\mathrm{V\big(x(k)\big) = x_k^T S\, x_k.} \quad (9)$$

The discrete-time T-S fuzzy system described in equation (7) is asymptotically stable if there exists a common symmetric matrix $\mathrm{S = S^T > 0}$ such that the following LMI are feasible [12] [13]:

$$\begin{cases} \mathrm{S > 0.} \\ \mathrm{A_i^T S\, A_i - S < 0.} \end{cases} \quad (10)$$

3.2. Autonomous vehicle stabilization

The control law is:

$$\mathrm{u(k) = -\sum_{i=1}^{n} \omega_i\big(\Gamma(k)\big)[G_i x(k)].} \quad (11)$$

Using the discrete-time T-S system previously described in (7), a closed loop control given by the new PDC law:

$$\begin{cases} \mathrm{x(k+1) = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i\big(\Gamma(k)\big)\omega_j\big(\Gamma(k)\big)\vartheta_{ij} x(k).} \\ \mathrm{\vartheta_{ij} = \big(A_i - B_i G_j\big).} \end{cases} \quad (12)$$

Hypothesis 1: The T-S system is locally controllable as described in [14].

The discrete time T-S models stabilization conditions for a closed-loop PDC controller are that there exists a symmetric matrix $S > 0$ as well as gains $G_i, \forall i \in I_n$ satisfying:

$$\begin{cases} C_{\text{discrete}}(\vartheta_{ii}, S) < 0, \forall i \in I_n, \\ C_{\text{discrete}}(\vartheta_{ij}, S) \leq 0, \forall i, j \in I_n^2 \\ \omega_i(\Gamma(k))\omega_j(\Gamma(k)) \neq 0. \end{cases} \quad (13)$$

And,

$$C_{\text{discrete}}(\vartheta_{ij}, S) = \left[\frac{\vartheta_{ij}+\vartheta_{ji}}{2}\right]^T S \left[\frac{\vartheta_{ij}+\vartheta_{ji}}{2}\right] - S. \quad (14)$$

The conditions are:

$$(A_i - B_i G_i)^T S (A_i - B_i G_i) - S < 0. \quad (15)$$

Multiplying (15) in pre and post by S^{-1}, we come by the following inequality:

$$S^{-1}(A_i S^{-1} - B_i G_i S^{-1})^T S (A_i S^{-1} - B_i G_i S^{-1}) > 0. \quad (16)$$

It's supposed that $X = S^{-1}$ and $H_i = G_i S^{-1}$, thus we get the following:

$$X(A_i X - B_i H_i)^T S\,(A_i X - B_i H_i) > 0. \quad (17)$$

The inequality (17) can be presented in LMI form by the application of the Schur complement as follows:

$$\begin{bmatrix} X & * \\ A_i X - B_i H_i & X \end{bmatrix} > 0 \; \forall \; i \in I_n. \quad (18)$$

By application of the same approach and the same steps, the condition $C_{\text{discrete}}(\vartheta_{ij}, S) \leq 0$ is given by:

$$\begin{bmatrix} X & * \\ \frac{A_i + A_j}{2} X - \frac{1}{2}(B_j H_i + B_i H_j) & X \end{bmatrix} \geq 0 \; \forall (i, j) \in I_n^2, i < j. \quad (19)$$

The discrete-time T-S system described in equation (7) are globally asymptotically stable via the novel PDC control law, if there are symmetric matrices such as $S = S^T > 0$ and $M = M^T \geq 0$ which verifies [15]:

$$\begin{cases} C_{discrete}(\vartheta_{ii}, S) + (r-1)M < 0, \forall i \in I_n, \\ C_{discrete}(\vartheta_{ij}, S) - M \leq 0, \forall i, j \in I_n^2, i < j, \\ \omega_i(\Gamma(k))\omega_j(\Gamma(k)) \neq 0. \end{cases} \quad (20)$$

$$C_{discrete}(\vartheta_{ij}, S) = \left[\frac{\vartheta_{ij}+\vartheta_{ji}}{2}\right]^T S \left[\frac{\vartheta_{ij}+\vartheta_{ji}}{2}\right] - S. \quad (21)$$

Using theorem announced in [13] helps to decrease the conservatism. **Theorem 1:** if there exist matrices $S = S^T > 0$, $M_{ij} = M_{ij}^T$ and matrices G_i which verifies:

$$\begin{cases} C_{discrete}(\vartheta_{ii}, S) + (r-1)M < 0, \forall i \in I_n. \\ C_{discrete}(\vartheta_{ij}, S) - M \leq 0, \forall i,j \in I_n^2, i < j. \\ \omega_i(\Gamma(k))\omega_j(\Gamma(k)) \neq 0. \end{cases} \quad (22)$$

$$M = \begin{bmatrix} M_{11} \ldots . & M_{1n} \\ M_{1n} \ldots . & M_{nn} \end{bmatrix}. \quad (23)$$

Applying the previous equations, the autonomous vehicle model presented in (7) is globally asymptotically stable.

$$\begin{cases} X = S^{-1}. \\ Y_{ii} = XM_{ij}X. \\ G_i = H_iX^{-1}. \\ \forall i \in I_n. \end{cases} \quad (24)$$

3.3. Multiobservers design

To perform the control of the autonomous vehicle we need the entire state vector $x(k)$. To achieve this goal we applied the following observer's equation:

$$\begin{cases} \hat{X}(k+1) = \sum_{i=1}^{n} \omega_i(\Gamma(k)) \left((A_i\hat{x}(k) + B_iu(k)) + N_i(y(k) - \hat{y}(k)) \right). \\ \hat{y}(k) = \sum_{i=1}^{n} \omega_i (\Gamma(k)) C_i\hat{x}(k). \end{cases} \quad (25)$$

The error state vector is written as:

$$\tilde{x}(k) = x(k) - \hat{x}(k). \quad (26)$$

Knowing that, the dynamics of the state vector error is given by:

$$\begin{cases} \hat{x}(k+1) = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i (\Gamma(k)) \omega_j(\Gamma(k)) \alpha_{ij}\tilde{x}(k). \\ \alpha_{ij} = A_i - H_iC_j, \forall (i,j) \in I_n^2. \end{cases} \quad (27)$$

In fact, the design of Luenberger observers requests the computing of local gains $N_i \in I_n$ to guarantee the convergence to 0 of the state vector error dynamics. In addition, we should guarantee that $S = S^T > 0$ and matrices $N_i \in I_n$ valid the following conditions:

$$\begin{cases} C_{discrete}(\alpha_{ii}, S) < 0, \forall i \in I_n. \\ C_{discrete}(\alpha_{ij}, S) \leq 0, \forall i,j \in I_n^2. \\ \omega_i(\Gamma(k))\omega_j(\Gamma(k)) \neq 0. \end{cases} \quad (28)$$

$$C_{discrete}(\alpha_{ij}, S) = \left[\frac{\alpha_{ij}+\alpha_{ji}}{2}\right]^T S \left[\frac{\alpha_{ij}+\alpha_{ji}}{2}\right] - S. \quad (29)$$

The equations (28) and (29) can be written as LMIs applying the Schur complement:

$$\begin{bmatrix} S & * \\ S\frac{A_i+A_j}{2} - \frac{1}{2}\left(H_jC_j + H_jC_i\right) & S \end{bmatrix} \geq 0 \quad (30)$$

When $i < j$.

We improve the observers by using the following theorem. **Theorem 2:** the multiple Luenberger observers are globally asymptotically stable if there exist symmetric matrices $S > 0$, M_{ij} and $N_i \in I_n$ that satisfy:

$$\begin{cases} C_{discrete}(\alpha_{ii}, S) + M_{ii} < 0, \forall i \in I_n. \\ C_{discrete}(\alpha_{ij}, S) + M_{ij} \leq 0, \forall i, j \in I_n^2. \\ \begin{bmatrix} M_{11} & M_{1n} \\ M_{1n} & M_{nn} \end{bmatrix}. \\ \omega_i(r(k))\omega_j(r(k)) \neq 0. \\ \alpha_{ij} = A_i - N_iC_j, \forall (i,j) \in I_n^2. \end{cases} \quad (31)$$

The LMI's are given by:

$$\begin{cases} S > 0. \\ \begin{bmatrix} S - M_{ii} & * \\ SA_i - H_iC_i & S \end{bmatrix} > 0 \ \forall\, i \in I_n. \\ \begin{bmatrix} S - M_{ij} & * \\ S\frac{A_i+A_j}{2} - \frac{1}{2}(H_iC_j + H_jC_i) & S \end{bmatrix} \geq 0 \ \forall\, i < j. i, j \in I_n^2. \\ M = \begin{bmatrix} M_{11} \ldots. & M_{1n} \\ M_{1n} \ldots. & M_{nn} \end{bmatrix}. \\ H_i = SN_i. \end{cases} \quad (32)$$

4. RESULTS

We applied the enhanced PDC control law design called classic PDC [16] to the discrete-Time T-S fuzzy model representing the autonomous vehicle system to ensure the lateral control purpose under certain constraints. Equations (25) and (32) give:

$$\begin{cases} S = X^{-1}. \\ G_i = H_iS. \\ H_i = SN_i. \end{cases} \quad (33)$$

Based on (33), we get the following gains and matrices:

$$S = e^{-03}\begin{bmatrix} 0.0513 & 0.0139 & 0.0693 & 0.0120 & 0.1275 & 0.0070 \\ 0.0139 & 0.0062 & 0.0235 & 0.0041 & 0.0400 & 0.0025 \\ 0.0693 & 0.0235 & 0.1267 & 0.0196 & 0.21158 & 0.0121 \\ 0.0120 & 0.0041 & 0.0196 & 0.0051 & 0.0368 & 0.0019 \\ 0.1275 & 0.0400 & 0.2158 & 0.0368 & 0.4979 & 0.0266 \\ 0.0070 & 0.0025 & 0.0121 & 0.0019 & 0.0266 & 0.0029 \end{bmatrix}$$

$$G_1 = [\,538.7210 \quad 88.1555 \quad 137.5593 \quad 21.1442 \quad -146.3325 \quad -56.7097].$$

$$G_2 = [545.2576 \quad 58.9360 \quad 139.5243 \quad 21.1255 \quad -148.8607 \quad -56.7512].$$

For multi-observer gains are:

$$\mathrm{N1} = e^{03}\begin{bmatrix} 0.0116 & 0.0007 \\ 0.0828 & -0.0020 \\ 0.0285 & -0.0003 \\ 0.2673 & -0.0020 \\ -0.0196 & -0.0273 \\ 9.3447 & 7.3696 \end{bmatrix}, \mathrm{N2} = e^{04}\begin{bmatrix} 0.0026 & 0.0000 \\ 0.0152 & -0.0001 \\ 0.0076 & -0.0000 \\ 0.0624 & -0.0001 \\ 0.0074 & -0.0023 \\ -1.2496 & 0.6264 \end{bmatrix}.$$

We tested our controller with different scenarios. The first scenario is to assume that the autonomous vehicle system will start far from the origin with different orientation. The initial state vector $x_0 = [0; 0.02; 0.04; 0; 0; 0.9]$, which is not the system equilibrium point [0;0;0;0;0;0], example, the lane centreline. We can certainly observe in figure 4-8 that our stabilization control law converges all the state variables to zero, which means that our autonomous vehicle reaches the centreline of the lane. These results prove the robustness of our controller. The second scenario is to apply a disturbance mitigation. We subjected the autonomous vehicle to a lateral wind force of 1500 Newton. Figure 2 and Figure 3 show a remarkable robust stabilization. Our model showed robustness and effectiveness against the disturbances.

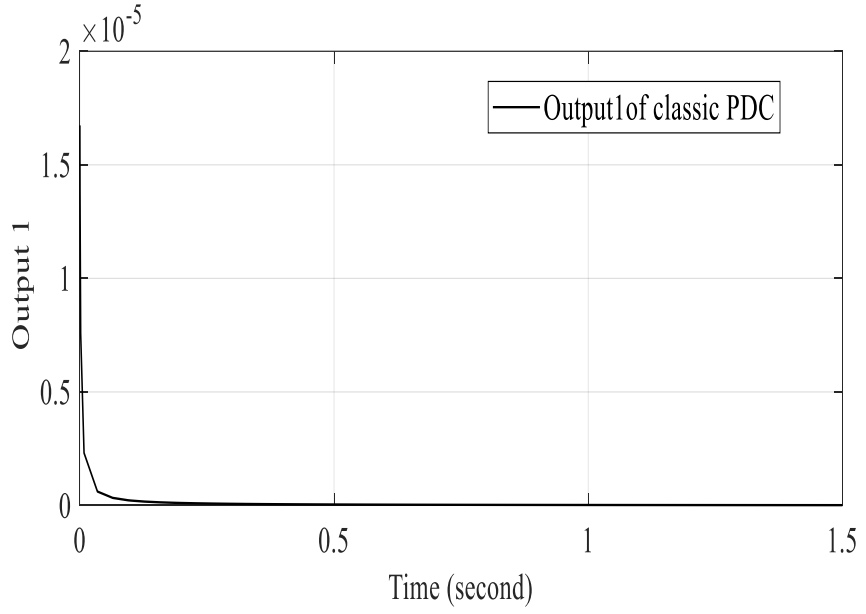


Figure 2. Stabilized output1

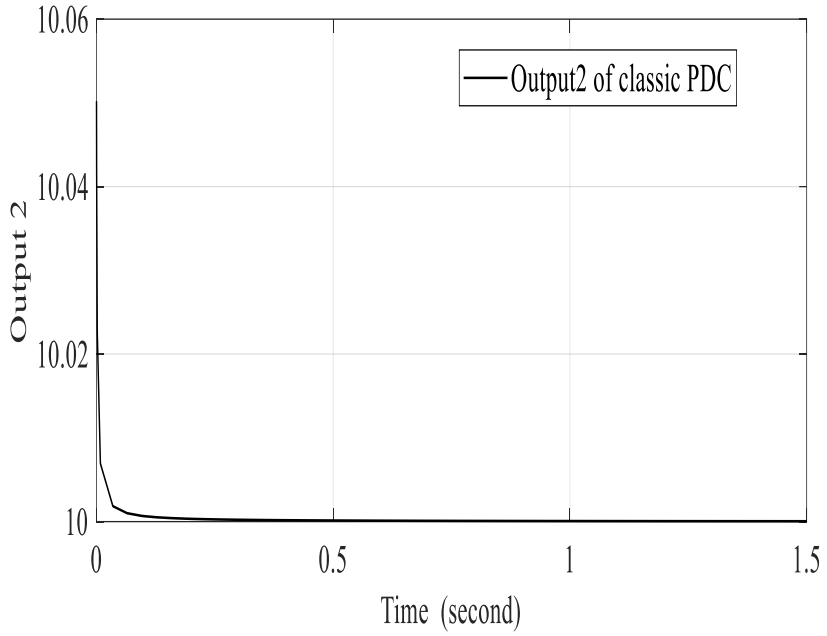


Figure 3. Stabilized output2.

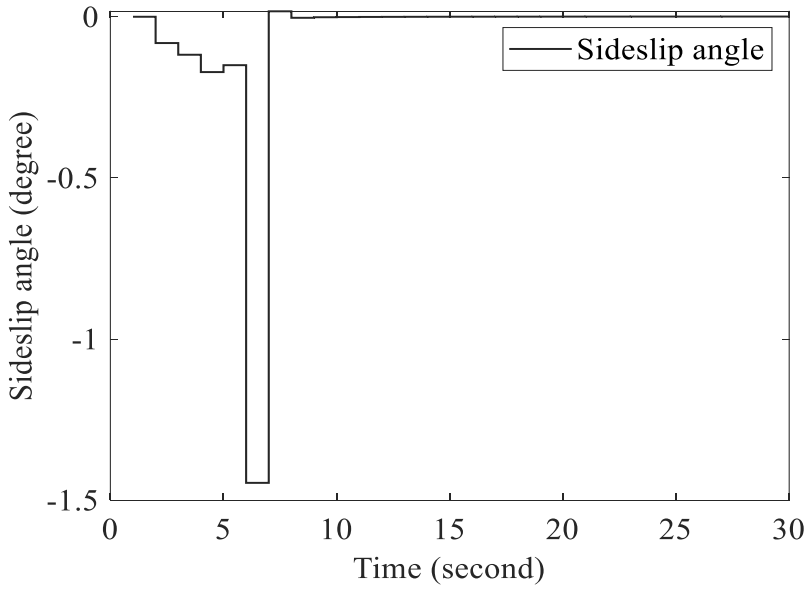


Figure 4. Sideslip angle.

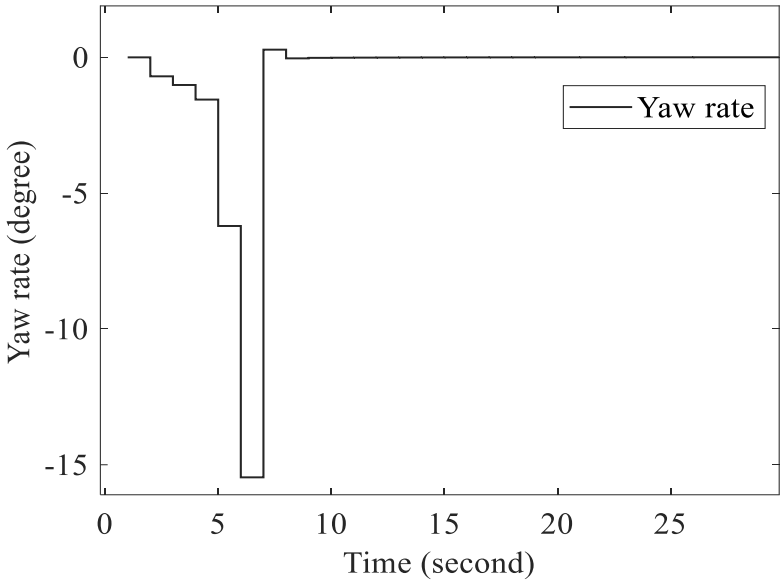


Figure 5. Yaw rate.

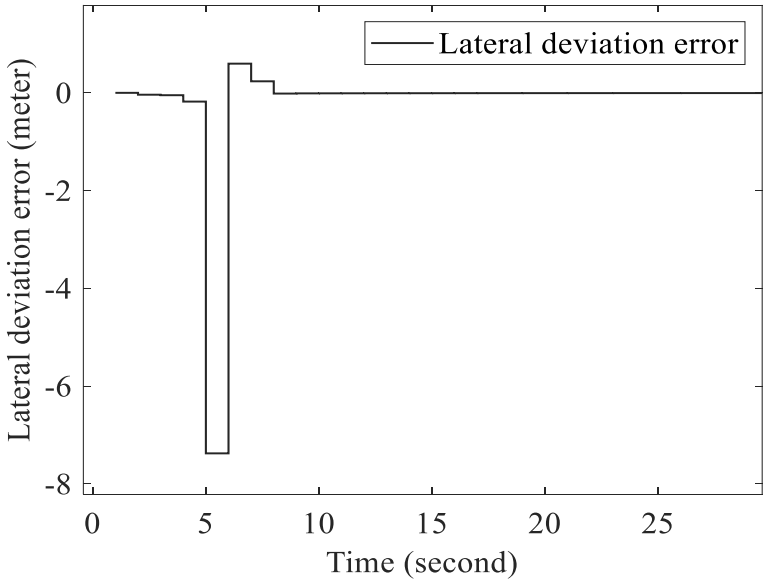


Figure 6. Lateral deviation error.

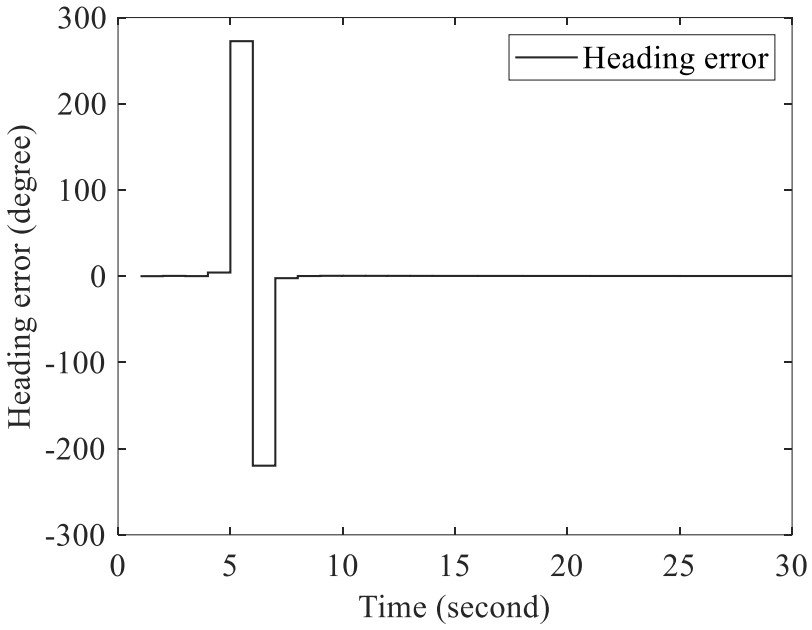


Figure 7. Heading error.

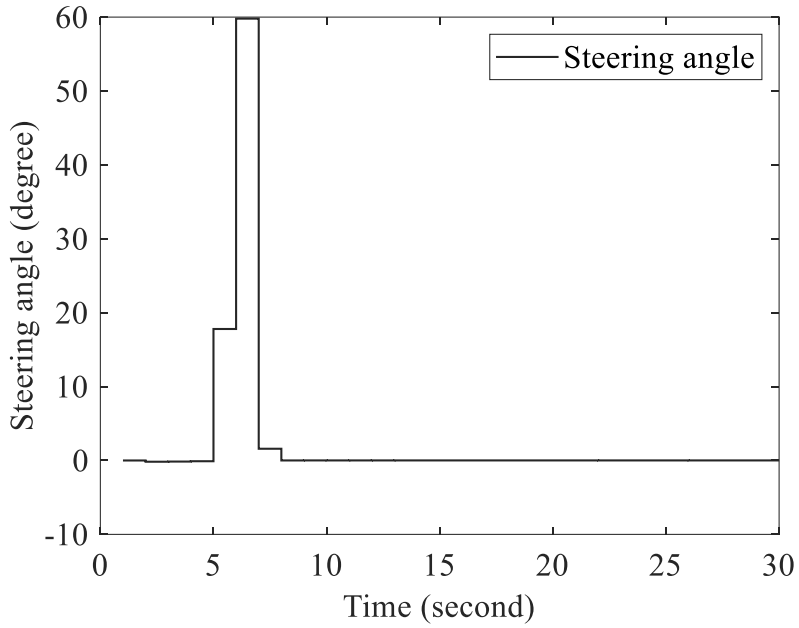


Figure 8. Steering angle.

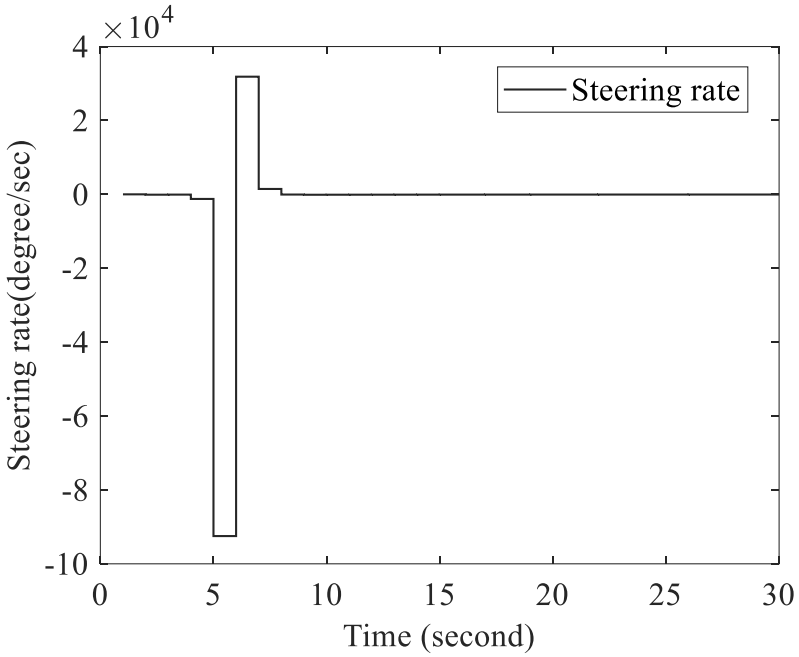


Figure 9. Steering rate

Figure 4 presents the sideslip angle convergence to equilibrium point in 10 seconds. Figure 5 shows that the yaw rate converges in 12 seconds. Figures 6-9 show that the lateral deviation error, the heading error and the steering angle converges in 8 seconds. These results show the effectiveness of our control design.

5. CONCLUSIONS

In this paper, we propose a control design for an autonomous vehicle. The discrete-time T-S system represents the autonomous vehicle model. The state vector is computed by using the Luenberger observer. Furthermore, we applied a PDC control law to the T-S fuzzy model. Feasible LMI conditions have been developed to guarantee lane keeping under certain limits. The results show that the lateral control of the autonomous vehicle to keep it on the centreline of the lane is well done under various constraints and scenarios. In future work, we will improve the heading error percentage to ensure more safety when we want to go back to the centreline of the lane.

ACKNOWLEDGEMENTS

The research work was supported by the Canada Research Chairs Fund Program and Natural Sciences and Engineering Research Council of Canada under Discovery Grant Project RGPIN /1056-2017.

REFERENCES

[1] J. Jiang and A. Astolfi, "Shared-Control for the Lateral Motion of Vehicles,"European Control Conference (ECC), Limassol, pp. 225-230, 2018.

[2] N. Enache, M. Netto, S. Mammar, and B. Lusetti, “Driver steering assistance for lane departure avoidance,” Control Eng. Pract., vol. 17, no. 6, pp. 642–651, Jun. 2009.

[3] Nguyen, A.T.; Coutinho, P.; Guerra, T.M.; Palhares, R.; Pan, J. Constrained Output-Feedback Control for Discrete-Time Fuzzy Systems with Local Nonlinear Models Subject to State and Input Constraints. IEEE Trans. Cybern. 2020, 51, 4673–4683.
[4] Ling, S.; Wang, H.; Liu, P.X. Adaptive Fuzzy Tracking Control of Flexible-Joint Robots Based on Command Filtering. IEEE Trans. Ind. Electron. 2020, 67, 4046–4055.
[5] Naranjo, J.E.; González, C.; García, R.; De Pedro, T.; Haber, R.E. Power-steering control architecture for automatic driving. IEEE Trans. Intell. Transp. Syst. 2005, 6, 406–415.
[6] Du, H.; Zhang, N.; Dong, G. Stabilizing vehicle lateral dynamics with considerations of parameter uncertainties and control saturation through robust yaw control. IEEE Trans. Veh. Technol. 2010, 59, 2593–2597.
[7] Sun, W.; Wang, X.; Zhang, C. A model-free control strategy for vehicle lateral stability with adaptive dynamic programming. IEEE Trans. Ind. Electron. 2019, 67, 10693–10701.
[8] Li, D.; Zhao, D.; Zhang, Q.; Chen, Y. Reinforcement learning and deep learning based lateral control for autonomous driving [application notes]. IEEE Comput. Intell. Mag. 2019, 14, 83–98.
[9] Jiang, J.; Astolfi, A. Lateral control of an autonomous vehicle. IEEE Trans. Intell. Veh. 2018, 3, 228–237.
[10] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
[11] R. Rajamani, "Vehicle Dynamics and Control". Boston, Springer, 2012.
[12] S. Boyd, E. Feron, L. El Ghaoui and V. Balakrishnan, "Linear matrix inequalities in system and control theory" SIAM studies in applied and numerical mathematics; vol. 15. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994.
[13] K. Tanaka, M. Nishimura and H. O. Wang, "Multi-objective fuzzy control of high rise/high speed elevators using LMIs," Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207), Philadelphia, PA, USA, pp. 3450-3454 vol.6, 1998.
[14] Euntai Kim and Heejin Lee, "New approaches to relaxed quadratic stability condition of fuzzy control systems," in IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 523-534, Oct. 2000.
[15] M. Chadli, D. Maquin and J. Ragot, "An LMI formulation for output feedback stabilization in multiple model approach," Proceedings of the 41st IEEE Conference on Decision and Control, 2002., Las Vegas, NV, USA, pp. 311-316 vol.1,2002.
[16] M. A. Jemmali, M. Otis and M. Ellouze. “Robust stabilization for discrete-time Takagi-Sugeno fuzzy system based on N4SID models”. Engineering Computations, May 2019.

DEMPSTER-SHAFER AND ALPHA STABLE DISTANCE FOR MULTI-FOCUS IMAGE FUSION

Rachid Sabre[1] and Ias Sri Wahyuni[2]

[1]Laboratory Biogéosciences CNRS,
University of Burgundy/Agrosup Dijon, France
[2]Faculty of Computer Science and Information Technology,
Universitas Gunadarma, Indonesia

ABSTRACT

The aim of multi-focus image fusion is to integrate images with different objects in focus so that obtained a single image with all objects in focus. In this paper, we present a novel multi-focus image fusion method based using Dempster-Shafer Theory and alpha stable distance. This method takes into consideration the information in the surrounding region of pixels. Indeed, at each pixel, the method exploits the local variability that is calculated from quadratic difference between the value of pixel I(x,y) and the value of all pixels that belong to its neighbourhood. Local variability is used to determine the mass function. In this work, two classes in Dempster-Shafer Theory are considered: blurred part and focus part. We show that our method give the significant result.

KEYWORDS

Multi-focus-images, Dempster-Shafer, distance, Alpha-Stable.

1. INTRODUCTION

Image fusion is the technique of combining relevant information from multiple images to produce a single image that contains more information than the input images. The goal of image fusion is to reduce uncertainty and minimize redundancy on the output as well as maximize relevant information specific to an application or task. In this article, we deal with merging multifocus images. Due to the limited depth of field of optical senses in cameras, it is often not possible to obtain an image containing all relevant objects "in focus". So that a scene image can be taken from a set of images with different focus. Image fusion method is used to get all focus objects.

There are different approaches of multifocal image fusion techniques that have been performed in the literature. These approaches can be divided into two types, the spatial domain method and the multi-scale fusion method. The spatial domain fusion method is performed directly on the source images. In spatial domain techniques, we work directly on the pixels of the image. Fusion methods such as averaging, principal component analysis (PCA) [1], maximum selection rule, two-sided gradient based methods [2], guided image filter based method (GIF) [3] and the maximum selection rule fall under spatial domain approaches. The disadvantage of spatial domain approaches is that they produce spatial distortion in the merged image. Spatial distortion can be very well managed by multi-scale approaches on image fusion. In multi-scale blending methods, the blending process is performed on the source images after decomposing them into multiple scales. Discrete wavelet transform (DWT) [4]-[7], Fusion of Laplacian pyramidal

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL, CYBI - 2022
pp. 287-299, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120923

images [8]-[14], Discrete cosine transform with variance calculation (DCT+var) [15], method based on the Salience detection (SD) [16] are examples of domain transformed image fusion techniques.

As stated in [17], from the point of view of proof, fusion degrades imprecision and uncertainty by using redundancy and complementary information from the source image. This means that the weak evidence of the inputs is used to give the best estimate. The proof theory was first proposed by Shafer in the 1970s, based on Dempster's research. The advantage of the Dempster-Shafer theory (DST) is that it allows to deal with the lack of preference, due to the limitations of available information, which leads to indeterminacy, as in [18] and [19]. This theory has been successful in many applications, including image segmentation [20], [12], pattern classification [22]-[24], object recognition [25], imaging technology [26], sensor fusion [27], [28].

In this article, we use Dempseter Shafer's theory, which has been successful in various image-processing methods. It is based on the plausibility and the weight of dependence of each pixel from a well-chosen distance. We propose the fusion of multi-focal images using the Dempste-Shafer theory, which derives from information: the variability of each pixel with its neighbourhood. This variability is calculated from the stable distance alpha between the value of the pixel I(x,y) and the value of all the pixels belonging to its neighbourhood. The stable alpha distance (alpha is between 0 and 2) generalizes the quadratic distance. This distance was introduced at the time of the discovery of stable alpha stochastic variables and processes [29], [30]. Several works have shown that working in a stable alpha space can improve the estimation and visibility of certain phenomena whose variability increases significantly [31]-[33]. The alpha-stable distribution is widely used in the processing of impulsive or spiky signals. It also has been applied in image processing field. [34] Models the sea clutter in SAR images using alpha stable distribution for ship detection while [35] removes speckle noise using alpha stable based Bayesian algorithm in the wavelet domain. Furthermore, alpha stable distribution is also used in image segmentation [36] and compressive image fusion [37] and alpha stable filter in fusion image [38]. Both [35], [36], and [36] and Wan employ alpha stable in wavelet domain. This section provides a brief of the alpha-stable distribution.

Thus, the stable alpha distance measures local variability around a pixel. This distance taken as an activity measure can detect the abrupt intensity of the image such as the edge. This method also takes into consideration the information in the surrounding region of the pixels and preserves the edge.

The originality of this work lies in the fact of combining Dempster Shafer method with the alpha stable distance adapted to large variations. Thus, we propose a new method that we compare to other existing methods in the literature and we show that it gives better fusion results.

This article is organized as follows: in section 2, we detail the main elements of the Dempster-Shafer proof theory. The definition of the stable distance alpha and the proof are presented in section 3. Section 4 provides the details of the proposed method. Section 5 defines the evaluation measures used in this article. Experiments are performed on different types of images and the results are compared with other works are provided in section 6. Section 7 gives the conclusion of this work.

2. DEMPSTER-SHAFER EVIDENCE THEORY

Let Θ represent a finite set of hypotheses for a problem domain, called frame of discernment.

Define a function m from 2^{Θ} to $[0,1]$ where 2^{Θ} be the set of all subsets of Θ

$$2^{\Theta} = \{A | A \subseteq \Theta\}. \quad (1)$$

The function m is called a basic probability assignment whenever

$$\mathrm{m}(\varnothing) = 0 \text{and} \sum_{\mathrm{A}\subset\Theta} \mathrm{m}(\mathrm{A}) = 1. \quad (2)$$

$m(A)$ is the measure of the belief that is committed exactly to A. According to [39], $m(A)$ is the degree of evidence supporting the claim that a specific element of Θ belongs to the set A, but not to any special subset of A. Each A of Θ such that $m(A) > 0$ are called the focal element of m. By applying the basic assignment function, several evidential functions can be created. A belief measure is given by the function $Bel: 2^{\Theta} \mapsto [0,1]$:

$$Bel(A) = \sum_{B\subset A} m(B). \quad (3)$$

The plausibility measure $Pl: 2^{\Theta} \mapsto [0,1]$ is defined by [28] as follows:

$$Pl(A) = \sum_{A\cap B\neq\varnothing} m(B) = 1 - Bel(\bar{A}). \quad (4)$$

$Bel(A)$ measures the degree of evidence that the element in question belongs to the set A as well as to the various special subsets of A. In stated in [17], an important aspect of DST concerns the aggregation of evidence given by different sources. If two mass function m_1 and m_2 induced by distinct items of evidence are such that $m_1(B) > 0$ and $m_1(C) > 0$ for some non disjoint subsets B and C of Θ, then they are combinable by means of Dempster's rule. [41], [42] followed by [40] suggested a rule of combination which allows that the basic assignments are combined. The combination (joint mass) of two sets of masses m_1 and m_2 is defined as follows

$$m_1 \oplus m_2(\varnothing) = 0 \quad (5)$$

$$m_1 \oplus m_2(A) = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B\cap C=\varnothing} m_1(B)m_2(C)} \quad (6)$$

The numerator represents the accumulated evidence for the sets B and C, which supports the hypothesis A and the denominator sum quantifies the amount of conflict between the two sets. Equation (6) can be written as

$$m_1 \oplus m_2(A) = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{\sum_{B\cap C\neq\varnothing} m_1(B)m_2(C)}. \quad (7)$$

As stated in [43], having a zero mass on a subset A does not mean that the set is impossible, simply that we are not capable of assigning a level precisely to A, since we could have non-zero masses on subsets of A, which would lead us to $Bel(A) \neq 0$.

3. ALPHA STABLE DISTANCE

Our method takes into consideration the information in the surrounding region of pixels. Indeed, at each pixel $I(x,y)$, the method exploits the local variability calculated from alpha stable distance between the value of pixel $I(x,y)$ and the value of all pixels that belong to its neighborhood. The idea comes from the fact that the variability value in blurred region is smaller

than the variability value in focused region, the proof is provided in this section. We use in this work the neighbor, with the size `"a", of a pixel (x, y) defined as follows:

($x + i$, $y + j$) where i = -a, -a + 1, …, a – 1, a and j = -a, -a + 1, …, a – 1, a. For example the neighbor with the small size (a = 1) contains: $(x-1, y-1)$, $(x-1, y)$, $(x-1, y+1)$, $(x, y-1)$, $(x, y+1)$, $(x+1, y-1)$, $(x+1, y)$, $(x+1, y+1)$ as we can see in Fig. 1.

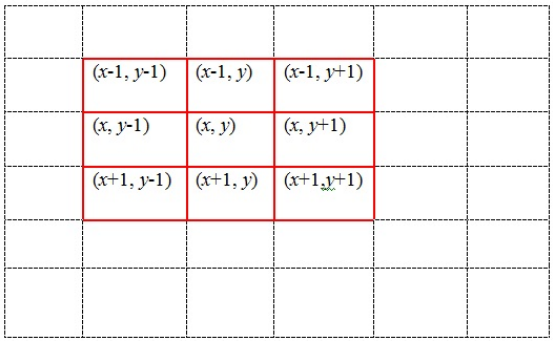


Figure 1. Pixel at (x, y) within its neighborhood, a = 1

We consider p source images $(I_1, I_2, \cdots, I_p)$ where each image has size $(R \times C)$. Alpha stable distance of every source image at pixel (x, y):

$$v_{a,k}(x,y) = \left(\frac{1}{T}\sum_{m=-a}^{a}\sum_{n=-a}^{a} |I_k(x,y) - I_k(x+m, y+n)|^{\alpha}\right)^{1/\alpha} \quad (8)$$

where k is the index of k^{th} source image $(k = 1,2,\cdots,p)$ is the number of source images.

$$I_k(x+m, y+n) = \begin{cases} I_k(x+m, y+n), & \text{if } 1 \le x+m \le R \text{ and } 1 \le y+n \le C, \\ I_k(x,y), & \text{otherwise} \end{cases}$$

$$T = (2a+1)^2 - card(S)$$

$$S = \{(m,n) \in ([-a,a]^2 - \{0,0\}) \; such \; that \; I_k(x+m, y+n) = I_k(x,y)\}$$

We show in the following that this local variability is small enough where the location is on the blurred area (B_1 or B_2). Indeed, we consider, without loss the generality, that we have a focus pixel (x, y) in image I_1 and blurred in image I_2, $((x, y) \in B_2)$

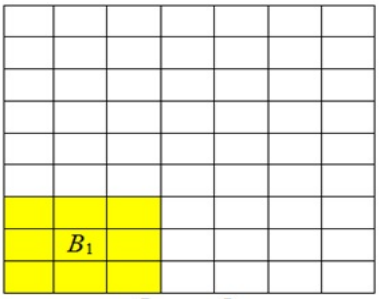


Image I_1

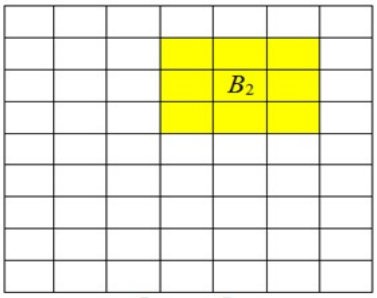


Image I_2

Figure 2. Two multi focus images, the yellow part is blurred area and the white part is clear(focused) area.

The local variability of image I_1 and image I_2 are respectively: $((\frac{1}{T} r_1(x,y))^{1/\alpha}$ and $(\frac{1}{T} r_2(x,y))^{1/\alpha}$, where $r_1(x,y)$ and $r_2(x,y)$ can be written as follow:

$$r_1(x,y) = \sum_{m=0}^{2a} \sum_{n=0}^{2a} |I_1(x,y) - I_1(x+(m-a), y+(n-a))|^{\alpha} \quad (9)$$

$$r_2(x,y) = \sum_{m=0}^{2a} \sum_{n=0}^{2a} |I_2(x,y) - I_2(x+(m-a), y+(n-a))|^{\alpha} \quad (10)$$

Proposition

Let (x,y) a pixel belongs to blurred area of the image I_2 ($(x,y) \in B_2$), then the local variability of (x,y) in image I_2, is smaller that the local variability of (x,y) in image I_1, ($r_2(x,y) < r_1(x,y)$).

Proof

The proof of this is given by using same arguments in [46].

The variability of image expresses the behavior of pixel relative to all pixels belong to their neighborhood. The precision of this fusion is depending on the size of the neighborhood, "a". For each image we try with different values of "a" in the set $\{1,2,\ldots,10\}$ and we get the value of "a" that corresponds to the minimum of root mean square error (RMSE). This operation is repeated for set of 150 multi-focus images [35].

4. THE PROPOSED METHOD

One of the essential problems of image fusion using Dempster-Shafer Theory is to construct the evidential representation of images. In this paper, we use one information as the evidential representation images: local variability. We consider two classes in the Dempster-Shafer theory. Either a pixel belongs to blurred part ω or it belongs to the focus part $\overline{\omega}$. There is also uncertainty θ inherent in the theory of evidence. All this constitute the frame of discernment in Θ our case [20].

$$\Theta = \{\omega, \overline{\omega}, \theta\} \quad (24)$$

For each pixel one value of evidence for information will be obtained, m.

$$\{m(\omega), m(\overline{\omega}), m(\theta)\} \quad (25)$$

with the condition $m(\omega) + m(\overline{w}) + m(\theta) = 1$.

The steps of image fusion in this work as follows. Suppose there are p original source images, $I_1, I_2, \cdots, I_p$, where each image has size $(R \times C)$ with different focus to be fused.

Step 1:

1. To calculate mass function:

for each image where we use different values of size of neighborhood, $a \in \{1,2,\cdots,10\}$, we define: $d'_{a,k}(x,y)$

$$d'_{a,k}(x,y) = 1 - \frac{v_{a,k}(x,y) - \min_{(x',y')}\left(v_{a,k}(x',y')\right)}{\max_{(x',y')}\left(v_{a,k}(x',y')\right) - \min_{(x',y')}\left(v_{a,k}(x',y')\right)} \quad (26)$$

where k is the k^{th} source image, $k \in \{1,2,\cdots,p\}$ and a is size of neighborhood of local variability. We set the standard deviation of $d'_{a,k}(x,y) = \sigma_{a,k}(x,y)$,

for (x,y) belongs to ω, we calculate:

$$m_{a,k}(\omega) = \left(1 - \sigma_{a,k}(x,y)\right) d'_{a,k}(x,y) \quad (27)$$

for (x,y) belongs to θ, we calculate:

$$m_{a,k}(\theta) = \sigma_{a,k}(x,y) \quad (28)$$

for (x,y) belongs to $\overline{\omega}$, we calculate:

$$\begin{aligned} m_{a,k}(\overline{\omega}) &= 1 - \left(1 - d'_{a,k}(x,y)\right)\sigma_{a,k}(x,y) - \sigma_{a,k}(x,y) \\ &= \left(1 - d'_{a,k}(x,y)\right)\left(1 - \sigma_{a,k}(x,y)\right) \end{aligned} \quad (29)$$

The final result of this method is obtained by showing which pixels belong to focus area or which do not, we use concept plausibility. In our case the plausibility of ω is the sum of the masses of the evidence for ω and the uncertainty θ:

$$Pl_{a,k}(\omega) = m_{a,k}(\omega) + m_{a,k}(\theta)$$

And for fusion image of the pixel (x,y), due to ω is a set of pixel on blurred area, we take pixel (x,y) from image k_0 that assigned to minimum $Pl_k(\omega)$, $k = 1,2,\cdots,p$.

Step 2.

For (x,y), we take F_a as fused image with size of neighborhood = a

$$\begin{aligned} \mathrm{F_a(x,y)} &= \mathrm{I_{k_0}(x,y)}, \text{where } \mathrm{k_0} \in \{1,2,\cdots,\mathrm{p}\} \text{ and } \mathrm{Pl_{a,k_0}(\omega)(x,y)} \\ &= \min_{\mathrm{k}\in\{1,2,\cdots,\mathrm{p}\}} \left(\mathrm{Pl_{a,k}(\omega)(x,y)}\right). \end{aligned}$$

Step 3.

For the proposed method, we use different values of size of neighborhood, $a \in \{1,2,\cdots,10\}$, and choose the value of a that corresponds to the minimum value of RMSE, such that our final fused image

$$\mathrm{F} = \mathrm{F_{a_0}} \text{where} \mathrm{a_0} \in \{1,2,\cdots,10\} \text{ and } \mathrm{RMSE(F_{a_0})} = \min_{\mathrm{a}\in\{1,2,\cdots,10\}} (\mathrm{RMSE(F_a)})$$

5. EXPERIMENTAL RESULT

The images used in this section are taken from the database of the webpage [47]. We have blurred an area of each image using the convolution of Gaussian filter applied on the reference image. The choice of Gaussian is approved in the works [44]-[45]. Blurred areas are chosen to hide an object from the photographed scene when there are multiple objects. Thus, the size of blurred areas varies according to the size of the objects hidden in the images. We applied the method on 150 sets of multi focus images on a datasets of images [47]. In this paper, as the number of pages is limited, we present only 3 sets of multi focus images. Figures 4, 5, 7, 8,10 and 11 show the multi focus images obtained by the convolution of Gaussian filter. Figures 6, 9 and 12 show the fused image by proposed method. Visually the image obtained by the proposed method gives a very satisfactory fusion.

Fig. 4 in focus on the right

Fig. 5 in focus on the left

Fig. 6 Fused image by proposed method

Fig.7 in focus on the left

Fig.8 in focus on the right

Fig.9 Fused image by proposed method

Fig.10 in focus on the left

Fig.11 in focus on the right

Fig.12 Fused image by proposed method

For comparison purposes, we perform fusion using methods: PCA method [1], Discrete Wavelet Transform (DWT) method [6], Laplacian Pyramid LP_PCA [13] , LP_DWT [14] and Bilateral gradient (BG) [2].

To objectively evaluate these fusion methods, quantitative measures of the fusion results are needed. According to the evaluation measure RMSE, the Table 1. gives the mean and standard deviation of RMSE for the given methods.

Table 1. Statistic parameters of the sample (150 images)

Method	LP_AV	PCA	BG	LP.PCA	DWT	LP.DWT	Proposed_method
Mean	6.351	6.245	7.7375	1.7456	3.0738	1.7841	0.3360
Standard deviation	2.81099	2.76977	3.77837	0.62897	1.06387	0.638727	0.0338

The results show that the proposed method has a smaller mean of the RMSE. The histograms of RMSE for 150 images by different methods show for almost method that the values of RMSE are almost symmetrically centered around the mean value. In order not to clutter this paper, we present below only the histogram of the proposed method.

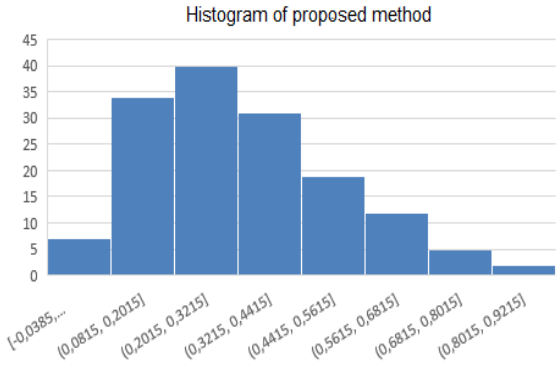


Figure 18. The histogram of proposed method

To compare analytically the proposed method to other methods we use the Analysis of variance (ANOVA) with dependent samples (dependence by image). The software R gives the following Anova table:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
method	6	7310	1218.4	268	<2e-16	***
Residuals	1043	4741	4.5			

As Pr(>F) is smaller than 1% the methods are significantly different. We use now the Newman Keuls test to compare the methods two-by-two and make groups having significantly the same mean. The software R gives the results below of the test.

	A$RMSE	std	r	Min	Max	Q25	Q50	Q75
BG	7.7374833	3.7783706	150	0.7996	22.7874	5.433375	6.92975	9.827225
DWT	3.0737927	1.0638730	150	0.5012	6.0300	2.320575	2.97350	3.851425
LP_AV	6.3513993	2.8109920	150	1.1543	17.3144	4.616225	5.96120	7.538725
LP_DWT	1.7841140	0.6387217	150	0.2923	3.6641	1.338800	1.75805	2.166900
LP_PCA	1.7456267	0.6289769	150	0.2872	3.6378	1.322750	1.72590	2.130525
PCA	6.2446747	2.7697664	150	1.1535	17.1552	4.586575	5.92430	7.342825
proposed_method	0.3360487	0.1839982	150	-0.0385	0.8728	0.194850	0.30595	0.449750

$comparison
NULL

$groups

	A$RMSE	groups
BG	7.7374833	a
LP_AV	6.3513993	b
PCA	6.2446747	b
DWT	3.0737927	c
LP_DWT	1.7841140	d
LP_PCA	1.7456267	d
proposed_method	0.3360487	e

Four different groups: Group “a” contains only method BG has the bigger mean of RMSE (7.737). Group “b” contains 2 methods LP_AV and PCA that have significantly the same average. Group “c” contains only the method DWT which better than group “a” and “b”. Group “d” contains 2 methods LP_DWT and LP_PCA which better than group “a”, “b” and “c”. The last group “e” containing the proposed method that the best method because his mean is the smallest by comparing with other means.

6. CONCLUSION

In this paper, we present the multi-focus image fusion method based using Dempster-Shafer Theory based on local variability. The method calculates the local variability for each pixel of each image and determines the mass function from local variability. The decision of fusion is obtained by pixels that correspond to minimum plausibility. The result of experiment shows that the proposed method gives significant improvement result in both visually and quantitatively. This method can be extended to image fusion for more than two blurred images. Our proposed method can be used in many applications, such as

1. Drone is a new technology in digital imaging, it has opened up unlimited possibilities for

enhancing photography. Drone can capture images on the same scene that zooms in on different objects, and at various altitudes. It will produces several images on the same scene but with different objects in-focus. The proposed method is used to obtain an image with all objects in-focus.

2. In medical imaging, the proposed can be used to detect an anomaly object or cell using local variability where the behavior of each pixel with its neighborhood is given.

3. For quality control in of food industry, cameras are used that take pictures. Each camera targets one of several objects to detect an anomaly. The objects are on a conveyor belt. To have a photo containing all the objects in-focus, we can use our proposed method which gives more details information.

There are several perspectives of this work:

1. As many work on image fusion have implemented on grayscale images. In this paper, the proposed method is performed on the grayscale image. However, the proposed method can be extended to color images as color image conveys significant information.

2. We are also encouraged to fuse more than two images by taking into account the local variability in each image (intra variability) and variability between image (inter variability). This inter variability can detect the 'abnormal pixels' among the images.

3. We are motivated to extend the proposed method to fuse images with different objects from different sensors (multimodal).

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Naidu V.P.S. and. Raol, J.R. (2008) "Pixel-level Image Fusion using Wavelets and Principal Component Analysis", Defence Science Journal, Vol. 58, No. 3, pp. 338-352.

[2] Tian, J., Chen, L., Ma, L., and Yu, W. (2011) "Multi-focus image fusion using a bilateral gradient-based sharpness criterion". Optic Communications, 284, pp 80-87.

[3] Zhan, K., Teng, J., Li, Q., and Shi, J. (2015) "A novel explicit multi-focus image fusion method". Journal of Information Hiding and Multimedia Signal Processing, vol. 6, no. 3, pp.600-612.

[4] Mallat, S.G. A (1989) "Theory for multiresolution signal decomposition: The wavelet representation". IEEE Trans. Pattern Anal. Mach. Intel., 11(7), 674-93.

[5] Pajares, G., Cruz, J.M.(2004) "A Wavelet-Based Image Fusion Tutorial". Pattern Recognition 37. Science Direct. 2004.

[6] Guihong, Q., Dali, Z., and Pingfan, Y. (2001) " Medical image fusion by wavelet transform modulus maxima". Opt. Express 9, pp. 184-190.

[7] Indhumadhi, N., Padmavathi, G.,(2011) "Enhanced Image Fusion Algorithm Using Laplacian Pyramid and Spatial Frequency Based Wavelet Algorithm". International Journal of Soft Computing and Engineering (IJSCE). ISSN: 2231-2307, Vol. 1, Issue 5, November 2011.

[8] Burt, P.J., Adelson, E.H. (1983) "The Laplacian Pyramid as a Compact Image Code". IEEE Transactions on communication, Vol.Com-31, No 40, April 1983.

[9] Burt, P.J. (1984) "The Pyramid as a Structure for Efficient Computation. Multiresolution Image Processing and Analysis", A. Rosenfeld, Ed., Springer-Verlag. New York 1984.

[10] Burt, P.J., Kolezynski, R.J. (1993) "Enhanced Image Capture Through Fusion". in: International Conference on Computer Vision, pp. 173-182.

[11] Wang, W., Chang, F.(2011) "A Multi-focus Image Fusion Method Based on Laplacian Pyramid". Journal of Computers, Vol.6, No 12, December 2011.

[12] Zhao, P., Liu, G., Hu, C., Hu, and Huang, H. (2013) "Medical image fusion algorithm on the Laplace-PCA". Proc. 2013 Chinese Intelligent Automation Conference, pp. 787-794.

[13] Verma, S. K., Kaur, M., and Kumar, R.(2016) "Hybrid image fusion algorithm using Laplacian

Pyramid and PCA method". Proceeding of the Second International Conference on Information and Communication Technology for Competitive Strategies. 2016.
[14] Wahyuni, I. S. and Sabre, R. (2016) "Wavelet Decomposition in Laplacian Pyramid for Image Fusion". International Journal of Signal Processing Systems Vol. 4, No. 1. pp. 37-44.
[15] Haghighat, M. B. A., Aghagolzadeh, A., and Seyedarabi, H. (2010) "Real-time fusion of multifocus images for visual sensor networks". Machine vision and image processing (MVIP), 2010 6th Iranian. 2010.
[16] Bavirisetti, D. P. and Dhuli, R. (2016) "Multi-focus image fusion using multi-scale image decomposition and saliency detection". Ain Shams Eng. J., to be published. [Online]. Available: http://dx.doi.org/10.1016/j.asej.2016.06.011. 2016
[17] Yuan, X., Zhang, J., Yuan, X., and Buckles, B.P.(2003) "Low level fusion of imagery based on Dempster-Shafer theory". Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference, pp: 475-479. 2003.
[18] Denoeux, T. (1999) "Reasoning with imprecise belief structures". International Journal of Approximate Reasoning 20(1): 79-111.
[19] Walley, P.(2003) "Statistical reasoning with imprecise probabilities". London: Chapman and Hall. 1991.
[20] Mena, J.B. and Malpica, J.A. (2003) "Color image segmentation using the Dempster-Shafer theory of evidence for the fusion of texture". Proceeding ISPRS, Vol. XXXIV-3/W8, pp. 139-144.
[21] Rombaut,M. and Zhu, Y.M.(2002) "Study of Dempster-Shafer for image segmentation applications". Image vision Computing, Vol. 20,pp. 15-23.
[22] Kowsalya, M. and Yamini, C. (2015) "A survey on pattern classification with missing data using Dempster-Shafer theory". International conference on information engineering, management, and security: 134-138. 2015.
[23] Zhu, H., Basir, O., and Karray, F.(2002) "Data fusion for pattern classification via the Dempster-Shafer evidence theory". Proceeding IEEE Conf. Syst,. Man, Cybern., vol. 7, pp. 109-114.
[24] Tong Z., Xu P. Denœux T. (2021), "An evidential classifier based on Dempster-Shafer theory and deep learning". Neurocomputin, vol.450, 25, pp. 275-293
[25] Hassan, M. H. (1989) "Object recognition based on Dempster-Shafer reasoning". Proc. SPIE 1002, Intelligent robots and computer vision VII. 1989.
[26] Bloch, I (1996) "Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account". Pattern Recognition Letters 17, pp. 905-919.
[27] Wu, H., Siegel, M., Stiefelhagen, R, and Yang, J. (2002) "Sensor fusion using Dempster-Shafer theory". IEEE Instrumentation and Measurement Technology Conference Anchorage, AK, USA, 21-23 May 2002.
[28] Nazmuzzaman K. and Sohel A. (2019) "Improved Dempster-Shafer Sensor Fusion using Distance Function and Evidence Weighted Penalty: Application in Object Detection" 16th International Conference on Informatics in Control, Automation and Robotics pp. 664 671.
[29] Masry E., Cambanis S., (1984), "Spectral density estimation for stationary stables processes" Stochastic Processes and Their, vol.18, pp. 1-31.
[30] Samorodnitsky G. Taqqu M., (1994), "Stable non gaussian random processes, Stochastic Modeling" Chapman and Hall, New York, London.
[31] Sabre R.and Wahyuni I, (2020); "Wavlet decomposition and Alpha Stable", Signal & Image Processing: An International Journal (SIPIJ), Vol.11, No.1, pp. 11-24.
[32] Sabre R. (2020) The Choice of the Smoothing Parameter for Alpha Stable Signals, International Journal of Signal Processing Systems Vol. 8, No. 2, pp. 49-53.
[33] Sabre R. (2021) "Mixed Spectra for Stable Signals from Discrete Observations" Signal & Image Processing: An International Journal (SIPIJ), Vol.12, No.5, pp. 22-44.
[34] Wang C., Liao M., Li X. (2008), "Ship Detection in SAR Image Based on the Alpha-stable Distribution", Sensors vol. 8 pp. 4948– 4960.
[35] Achim A. , Bezerianos A. and Tsakalides P.,(2001), "An Alpha-Stable Based Bayesian Algorithm For Speckle Noise Removal In The Wavelet Domain", in 5th IEEE-EURASIP Biennial International Workshop on Nonlinear Signal and Image Processing, Baltimore, USA, 2001.
[36] Wan T., Canagarajah N. and Achim A.,(2007) "A Statistical Mul-tiscale Image Segmentation via Alpha-Stable Modeling", IEEE International Conference on Image Processing. pp.357– 360.
[37] Wan T, Canagarajah N., Achim A., (2008) "Compressive Image Fusion", IEEE International

Conference on Image Processing, pp.1308–1311.
[38] Sabre R., Wahyuni I. (2019), “Multifocous Image Fusion Using Laplacian Pyramid Technique Based on Alpha Stable Filter” CRPASE, Vol. 5, 2, pp. 58-62.
[39] Klir, G.J and Folger, T. A. (1988) “Fuzzy sets, uncertainty and information”. Englewood Cliffs. Prentice-Hall. 1988.
[40] Shafer, G.(1967) “ A mathemaical theory of evidence”. Pricenton University Press.1976
[41] Dempster, A. P. (1967) “Upper and lower probabilities induced by a multivalued mapping”. The Annals of Mathematical Statistics 38: 325-339. 1967.
[42] Dempster, A. P (1968). “A genralization of Bayesian inference”. Journal of the Royal Statistical Society, Series B (methodological) 30: 205-247.
[43] Bloch, I. (2008) Information fusion in signal and image processing. John Wiley and Sons, Inc
[44] Nayar, S. K.. (1992) “Shape from Focus System”. Proc. of IEEE Conf. Computer Vision and Pattern Recognition, pp. 302-308.
[45] Petland, A. (1987) “A new sense for depth of field”. IEEE Transactions on Pattern Analysis and Machine Intelligent, Vol. 9, No. 4, pp. 523-531.
[46] Sabre R., Whayuni IS.,(2019) “Alpha Stable Filter and Distance for Multifocus Image Fusion”, International Journal of Signal Processing Systems Vol. 7 , No. 2, pp. 67-72.
[47] www.rawsamples.ch. Accesed: 30 March 2018.

AUTHOR INDEX