IMBALANCED DATASET EFFECT ON CNN-BASED CLASSIFIER PERFORMANCE FOR FACE RECOGNITION

Miftah Asharaf Najeeb and Alhaam Alariyibi

Computer Science Department, University of Benghazi, Benghazi, Libya

ABSTRACT

Facial Recognition is integral to numerous modern applications, such as security systems, social media platforms, and augmented reality apps. The success of these systems heavily depends on the performance of the Face Recognition models they use, specifically Convolutional Neural Networks (CNNs). However, many real-world classification tasks encounter imbalanced datasets, with some classes significantly underrepresented. Face Recognition models that do not address this class imbalance tend to exhibit poor performance, especially in tasks involving a wide range of faces to identify (multi-class problems). This research examines how class imbalance in datasets impacts the creation of neural network classifiers for Facial Recognition. Initially, we crafted a Convolutional Neural Network model for facial recognition, integrating hybrid resampling methods (oversampling and under-sampling) to address dataset imbalances. In addition, augmentation techniques were implemented to enhance generalization capabilities and overall performance. Through comprehensive experimentation, we assess the influence of imbalanced datasets on the performance of the CNN-based classifier. Using Pins face data, we conducted an empirical study, evaluating conclusions based on accuracy, precision, recall, and F1-score measurements. A comparative analysis demonstrates that the performance of the proposed Convolutional Neural Network classifier diminishes in the presence of dataset class imbalances. Conversely, the proposed system, utilizing data resampling techniques, notably enhances classification performance for imbalanced datasets. This study underscores the efficacy of data resampling approaches in augmenting the performance of Face Recognition models, presenting prospects for more dependable and efficient future systems.

KEYWORDS

Face Recognition (FR), Convolutional Neural Network (CNN), Imbalanced Class Data, Resampling Techniques.

1. INTRODUCTION

FR operates within the realm of computer vision, serving as a pivotal method in various biometric authentication applications like electronic payments, smartphone lock screens, and video surveillance. Its primary objectives involve face verification and face identification through a stored repository of facial data. FR systems demonstrate proficiency in accurately recognizing individuals within images and videos, verifying their identity based on facial features, even amidst challenging conditions such as variations in lighting, expression, and the angle of photo capture [1][2]. Compared to conventional Machine Learning (ML) methods, deep learning techniques have exhibited superior performance in terms of accuracy and processing speed, particularly in image recognition. The CNN stands out as a prominent deep learning approach used extensively in image and facial recognition [3]. CNNs are increasingly pivotal across various domains of ML applications, currently advancing the forefront of computer vision tasks like object detection, image classification, and segmentation [4]. Extensive research spanning

DOI:10.5121/ijaia.2024.15102

decades has been dedicated to FR. Certainly, neural network classifiers, notably CNNs, have gained immense popularity in the realm of FR, significantly elevating recognition accuracy in recent times. Nonetheless, numerous challenges persist within FR, including the classification of imbalanced data.

During the ML algorithm's learning phase, certain classes within the dataset might exhibit either overrepresentation or underrepresentation. In simpler terms, the proportions of these classes vary significantly. This challenge is commonly referred to as classification with imbalanced datasets, posing a substantial issue in the domain of Machine Learning. Consequently, this imbalance can impede learning algorithms and prompt biased predictions from deep learning models [5][6]. In practical scenarios, imbalanced datasets are commonly generated, necessitating strategies to manage such data disparities. Two primary approaches exist for handling class imbalance: the algorithm-level and data-level approaches. The algorithm-level approach focuses on enhancing the current classifier by adjusting algorithms to better identify the smaller class. Conversely, the data-level approach aims to achieve a balanced distribution by modifying or altering the number of instances within the majority and minority classes before training a classifier [5]. The technique of rectifying imbalance by adjusting the number of instances in each class is termed resampling. Resampling involves two primary methods: under-sampling the majority class and oversampling the minority class data, both aiming to achieve a balanced data distribution [7]. However, oversampling can potentially trigger overfitting issues by repeatedly utilizing duplicated minority samples, whereas under-sampling might discard valuable information present in the majority of samples. While these methods primarily address imbalanced learning in binary classification, multi-class challenges are prevalent in certain applications. As a result, there is a growing research focus on effectively applying resampling techniques to address imbalanced multi-class data and attain satisfactory classification outcomes [5][7]. Crafting a learning method tailored for imbalanced datasets poses a significant challenge, especially in tasks where the cost of misclassifying a minority class instance is substantial [8]. Standard learning algorithms, optimized for balanced datasets, require supplementary procedures to effectively manage imbalanced datasets [9]. The impact of class imbalance intensifies with the scale of the task, particularly when working with extensive data. Although CNN-based approaches excel in classifying large, balanced datasets, they lack specific algorithmic strategies to address imbalanced data, as they were not initially designed for this purpose [10].

Driven by the aforementioned challenges, this paper introduces a deep learning model aimed at addressing the imbalanced dataset concern within FR tasks using real-world data. The primary contribution of this study lies in presenting a robust recognition model to enhance the effectiveness and accuracy of FR systems when operating with imbalanced datasets. Furthermore, this research expands existing binary class resampling methods to tackle multi-class imbalanced scenarios significantly, thereby improving the performance of minority classes while preserving the performance of majority ones. The proposed system comprises two key components. The development of the FR model and the creation of a balanced dataset. This process entails the construction of an efficient FR model utilizing a CNN-based classifier for effective classification. The CNN is chosen as the foundational technique owing to its exceptional performance in image-related tasks, especially in challenging conditions. To the generation of a balanced dataset, a procedure is implemented based on the random balance framework, integrating random under-sampling and random oversampling techniques. Furthermore, data augmentation techniques are employed to mitigate overfitting towards minority classes and enhance model generalization. The proposed model not only enhances classification accuracy but also effectively mitigates the inherent class imbalance observed in multiclass classification problems. The FR process in this paper comprises three phases: pre-processing, extraction of facial features, and classification based on the extracted feature set.

The subsequent sections of this paper are structured as follows: Section 2 introduces related works, Section 3 introduces CNN architecture, Section 4 describes dataset and delves into methodology, Section 5 presents the experimental results, Section 6 provides a discussion, and finally, Section 7 offers conclusions.

2. RELATED WORKS

In the last few decades, facial recognition technology has evolved significantly in research and practical applications, transitioning from simple geometric approaches to complex techniques utilizing ML. This section explores previous studies involving the development of FR models based on CNNs and other research centered on deep learning methods targeting class imbalance through data-level approaches.

2.1. CNNs-Based FR Models

This section provides an overview of various CNN frameworks proposed for facial recognition tasks. In study [11], Wu et al. proposed a Light CNN framework aimed at learning a concise embedding from extensive face data characterized by substantial noisy labels. They introduced a Max-Feature-Map (MFM) activation within each convolutional layer of the CNN. This activation functions to differentiate between noisy and informative signals while also serving as a feature selector between two feature maps. Many studies and research endeavors in the domain of large-scale FR tasks primarily concentrate on devising effective loss functions for feature learning using deep CNNs. In 2017, Liu et al. [12] introduced SphereFace, a significant advancement in FR. This model reinterprets the linear transformation matrix in the last fully connected layer as class centers within an angular space, effectively penalizing angles between deep features and their corresponding weights multiplicatively. SphereFace's primary contribution lies in optimizing loss functions. Hassan and Abdulazeez [13] commended this achievement, noting substantial improvements in addressing challenges related to occlusion, illumination variations, pose, expression, while highlighting persistent issues like high GPU usage and network depth. In addition, [14] proposed the Vision Transformer (ViT), surpassing many existing face recognition methods and establishing a potent baseline referred to as ViT. They leverage the intrinsic capability of transformers to handle information from irregular grids, leading to the creation of a pipeline resembling part-based face recognition methods. A current trend involves integrating margins into established loss functions to enhance class separability. In line with these advancements, Deng et al. [15] introduced the Additive Angular Margin Loss (ArcFace), providing a geometrically intuitive approach to generating highly discriminative features during CNN training.

2.2. Imbalanced Data Learning

This section provides an overview of studies focusing on data resampling techniques for training CNN classifiers using imbalanced data. In the paper [4], the authors conducted a thorough exploration into how class imbalance affects the classification performance of CNNs. They investigated various prevalent methods to tackle class imbalance, such as oversampling, under-sampling, two-phase training, and adjusting thresholds to account for prior class probabilities. The authors assessed these methods using the MNIST, CIFAR-10, and ImageNet datasets. Their analysis revealed that class imbalance had a negative impact on classification performance, with this effect worsening as imbalance levels and task scales increased. Their findings indicated that ROS (Random Oversampling) surpassed both the baseline and RUS (Random Under-Sampling) methods across most cases, showcasing improved accuracy through thresholding, while RUS exhibited generally poorer performance. The authors' discoveries carry considerable weight in comprehending and managing class imbalance within deep learning applied to facial recognition tasks. In a recent study [10], a novel adaptive sampling approach was proposed: Dynamic Curriculum Learning (DCL). This method devised a dynamic curriculum strategy for data sampling to rebalance classes. The fundamental concept revolves around adjusting the probability of sampling instances from a class as training progresses. Initially, random sampling is employed to acquire general representations. Subsequently, DCL uses a curriculum strategy, increasing the sampling of specific class instances to address the imbalance. In their research, Hensman and Masko [16] investigated the impact of Random Oversampling (ROS) on imbalanced image data derived from the CIFAR10 dataset, employing deep CNNs. The authors crafted ten imbalanced distributions by manipulating class sizes, reaching a maximum imbalance ratio. They designated any class smaller than the largest as a minority class and employed random oversampling on these minority classes until a balanced distribution was attained. The findings indicate that Inf Syst Front models trained with ROS perform nearly as effectively as the baseline models trained with the original balanced distributions. Despite showcasing the effectiveness of ROS, the focus lies on scenarios with exceedingly high levels of class imbalance. Lee et al. [17] introduced a two-phase learning approach incorporating Random Under-Sampling (RUS) with transfer learning to classify highly imbalanced datasets, specifically the WHOI-Plankton dataset. This method involves initially pre-training a deep CNN with thresholded data and subsequently fine-tuning it using the entire dataset. The comparison of the proposed model involved six alternative methods, which integrated transfer learning and augmentation techniques. The assessment used unweighted average F1-scores for result comparison. In contrast to plain RUS, which eliminates potentially valuable information from the training set, the two-phase learning method employed here selectively removes samples only from the majority group during the pre-training phase. This strategic approach allows the minority group to exert a more substantial influence on the gradient during pre-training, while still enabling the model to access all available data during the subsequent fine-tuning phase.

The challenge of class imbalance has been a prominent subject within CNN and facial recognition research. Several studies offer detailed insights into the impacts of class imbalance and diverse mitigation approaches. While there is emphasis on oversampling, under-sampling has not received comparable attention, especially its specific implications in FR, notably within large-scale, real-world multi-class scenarios. Conversely, other studies offer a holistic perspective on FR disparities, particularly within critical sectors like law enforcement. Their discussion extended beyond imbalances in data representation but lacked an in-depth exploration of effective strategies to address these disparities. This gap is what our research endeavors to bridge. We aim to translate fundamental concepts into tangible real-world outcomes by amalgamating theoretical foundations from prior studies with innovative data resampling strategies tailored for FR. This approach aims to revolutionize the treatment of class imbalance, potentially setting new benchmarks for the capabilities of FR technology.

3. CONVENTIONAL NEURAL NETWORK ARCHITECTURE

Indeed, a distinct type of deep learning architecture tailored for handling spatial data is the CNN, also known as ConvNet. Inspired by biological beings' visual perception mechanisms, the CNN is a specialized feedforward neural network explicitly developed to handle multi-dimensional data, such as images [18]. A standard CNN architecture comprises one or multiple blocks featuring convolution and pooling layers, succeeded by one or more fully connected layers, culminating in an output layer. The convolutional layer, responsible for acquiring feature representations from the input, stands as the core component within a CNN. Within this structure, multiple learnable convolution kernels or filters are employed to compute distinct feature maps, each linked to a receptive field in the preceding layer. The resulting feature map is generated by convolving the input with these kernels and applying a non-linear activation function. Subsequently, the pooling layer down samples the output from the convolutional layer to yield a single output. Finally, one or more fully connected layers form the concluding segments of the CNN, producing the model's output [19]. Figure 1 provides a visual representation depicting the CNN structure for image classification.

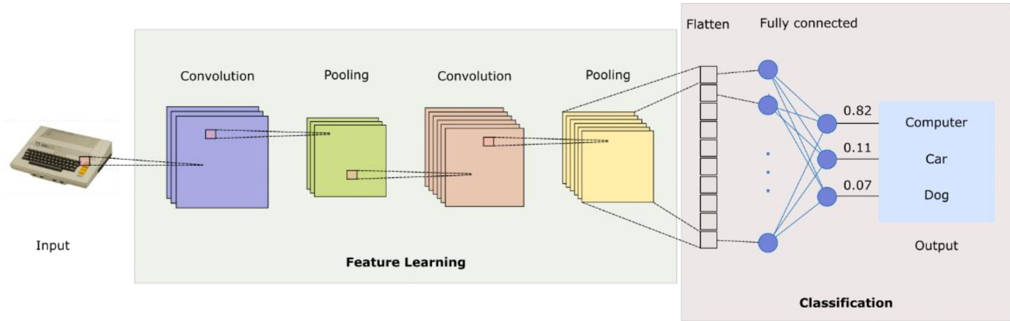

Figure 1. Convolutional Neural Network structure for image classification [20].

For image classification tasks, a CNN operates as a fusion of feature extraction (convolution and pooling layers) and classification (fully connected layers). Various convolutional layers are employed to detect diverse features within an image. Subsequently, fully connected layers are integrated as classifiers atop these extracted features, attributing probabilities to input images. Enhancements to CNNs can be achieved through several means, encompassing activation function choices, normalization techniques, loss function selection, regularization methods, optimization strategies, and enhancements in processing speed [19]. Despite the impressive performance seen in CNN models across various applications, there's ongoing development in our understanding of the reasons behind their effectiveness, necessitating further research into their fundamental principles. CNNs are well-known for their efficiency in image recognition and classification, primarily due to their ability to extract feature representations from images. After a thorough examination of existing architectures, our customized model underwent optimization to excel in handling large-scale datasets containing diverse images of individuals with variations in lighting, age, photography angles, facial expressions, and more. Subsequent sections elaborate on the specifics of this optimized model.

4. Dataset and Methodology

This section commences with a discussion regarding the description of dataset. It subsequently delves into the description of the proposed system, providing comprehensive insights into its stages including pre-processing, data resampling and augmentation strategies, and the proposed CNN model architecture. Finally, it explores the performance metrics employed to evaluate the obtained results.

4.1. Dataset Description

In the realm of FR, assembling and curating a fitting, high-quality dataset from the ground up presents a notable challenge. This dataset must possess ample size to effectively train the CNN model and encompass a broad range of variations in human facial features and expressions. The dataset used in this study was sourced from the Pins Face Recognition dataset [21], comprising over 100 celebrities and approximately 17,000 facial images collected and cropped from Pinterest. The decisive factor in choosing this dataset over others, such as VGGFace2 and LFW, was its abundance of images for each individual and a substantial number of classes, well-suited to address the computational resource challenges we encountered. These images encompass various lighting conditions, capturing diverse facial expressions and orientations, which contribute to the dataset's complexity and resilience. Merely amassing a large volume of facial data is not sufficient; accurate labeling is imperative for data utility. In FR tasks, this entails linking each image to the corresponding individual's identity. Figure 2 provides an illustrative example from the Pins dataset.

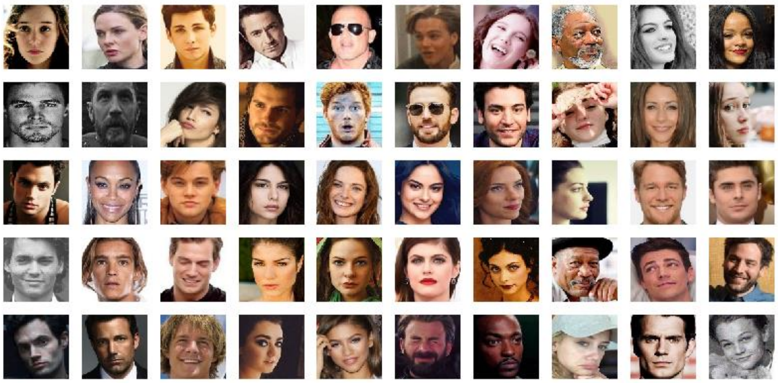

Figure 2. Pins FR dataset example [21].

4.2. The Proposed System

Within this section, we will further explore our proposed system tailored to tackle the complexities of FR tasks utilizing CNNs. This model is specifically engineered to handle large-scale datasets encompassing diverse conditions, such as discrepancies in lighting, age, photography angles, a spectrum of facial expressions, and class imbalances—common challenges encountered in FR scenarios. Figure 3 illustrates the phases comprising the proposed FR system.

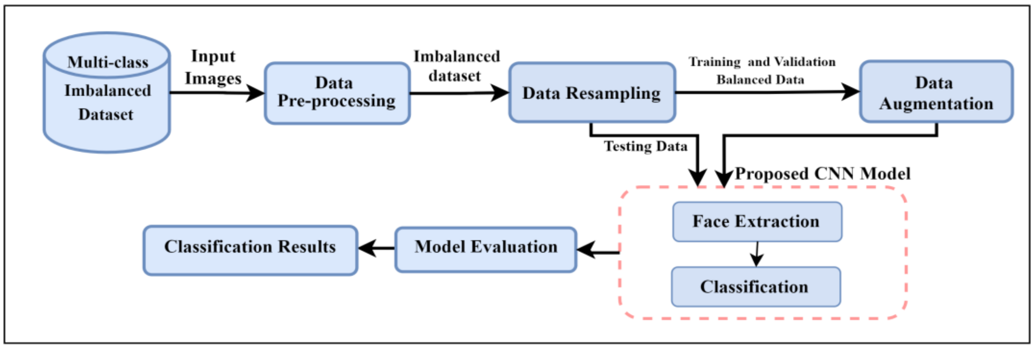

Figure 3. Workflow of the proposed FR system.

4.2.1. Data Pre-processing

The pre-processing phase stands as the foundational framework within our system's architecture, transforming raw images into a structured format conducive to efficient model input. This phase directly influences the model's performance and the reliability of its outputs. During the initial phase of pre-processing, facial detection within each image is executed using the Haar Cascade classifier. This machine learning-based approach is trained utilizing both positive and negative images. Upon detection, individual faces are cropped from the image, isolating the focal features and minimizing the inclusion of extraneous information fed into the model.

Subsequently, the classifier is employed to identify objects within other images. Following facial detection, all non-RGB images undergo conversion to RGB format, ensuring uniformity across all images. Inconsistencies in color formats can introduce conflicts in the input data, potentially resulting in a significant decline in model performance. Simplifying the model's complexity is a crucial stage that contributes to a streamlined and more efficient CNN model. Post-face detection, cropping, and color conversion, the images undergo transformation into grayscale images. This format is easier to process, containing fewer color channels compared to colored images. Each grayscale image undergoes quality assessment using a Histogram of Oriented Gradients (HOG) feature descriptor [22]. This descriptor offers a quantified measure of the image's quality. For instance, in this scenario, any image falling below the 0.05 threshold is excluded from the dataset. Following that, facial alignment is conducted using dlib's shape predictor, identifying 68 (x, y)-coordinates corresponding to facial structures. This pivotal step ensures proper centering and alignment of faces, making the model invariant to scale and rotational variations. Subsequent to facial alignment, all images are resized to a standardized dimension of 124x124 pixels, ensuring consistent input size for the model. Normalization and standardization of the images are carried out, resulting in pixel values within the range of [0, 1] and data characterized by a zero mean and unit variance. These measures are crucial as they diminish the likelihood of getting trapped in local optima and expedite the model training process. Figure 4 shows the stages of image pre-processing.

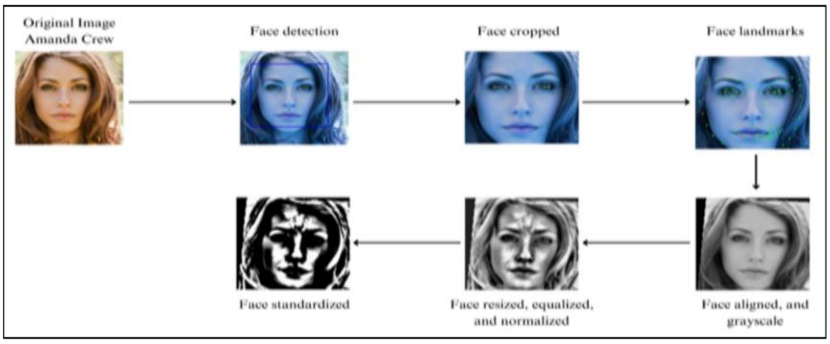

Figure 4. An overview of the pre-processing stages.

4.2.2. Data Resampling and Augmentation

Data-level approaches aim to rectify the class distribution imbalance in a pre-processing capacity. This approach holds appeal as it necessitates modifying solely the training data, eliminating the need to alter the learning algorithm. These methodologies can be categorized into oversampling, under-sampling, and a hybrid data-level approach. In our model, we employed a combined strategy of oversampling and under-sampling to generate a balanced dataset (Hybrid method). The Pins face dataset used in this study exhibited significant imbalance, with some face images having as few as 86 instances, while others had as many as 200.

The hybrid approach initiates with under-sampling the majority classes and subsequently oversampling the minority classes based on a parameter established through iterative processes until achieving an equal number of images across all classes. The oversampling process involves using diverse methods such as duplicating images or extracting small sections from images of other individuals. Indeed, the process of under-sampling classes may result in the loss of valuable images within the class, potentially undermining the model's predictive robustness in diverse real-world situations. Conversely, when oversampling classes to a specific quantity, there is a risk of unintentionally fostering an overly optimistic perception within the model. The recurrent exposure to identical images can prompt the model to overgeneralize from these duplicates, potentially leading to overfitting. Overfitting occurs when the model excessively conforms to the training set, causing diminished performance on unseen, new data [23]. Intensified data augmentation for minority classes can aid in mitigating overfitting. Data augmentation is a strategy employed in both Machine Learning and deep learning models, artificially broadening the diversity of the training dataset through various transformations like rotations, scaling, and flips.

4.2.3. The Structure of The Proposed CNN Model

The primary objective of the proposed model is to execute a multi-class classification, specifically tailored for FR task in our system. At the heart of the proposed CNN model for FR lie the convolutional layers. Operating on grayscale face images as input, the model aims to discern and glean features by employing distinct convolutional filters. In fact, the model's structure commenced with fewer layers and underwent gradual layer increments through experimentation until reaching 22 layers: comprising an input layer, 20 hidden layers, and an output layer. In addition, the proposed model uses pooling, normalization, and dropout layers to reduce spatial dimensions, standardize inputs to each layer, and mitigate overfitting possibilities. Dropout layers, as detailed in reference [24], randomly nullify a fraction of input units to 0 during updates in training. Following this, a flatten layer is employed to flatten the extracted 2D arrays (features), facilitating their transfer to fully connected layers responsible for executing advanced reasoning and classification. The architecture uses four fully connected layers, referred to as Dense layers. Moreover, the model integrates data resampling methodologies to tackle the class imbalance issue within the dataset. The parameters utilized in the proposed model are described in Table 1.

Table 1. The parameters of the proposed CNN-based model.

Layer Type	Output Shape	Parameters	Activation	Regularization
Conv2D	(124,124,32)	320	ReLU	L2(0.01)
MaxPooling2D	(62, 62, 32)	0	-	-
BatchNormalization	(62, 62, 32)	128	-	-
Conv2D	(62, 62, 64)	18496	ReLU	L2(0.01)
MaxPooling2D	(31, 31, 64)	0	-	-
BatchNormalization	(31, 31, 64)	256	-	-
Conv2D	(31, 31, 128)	73856	ReLU	L2(0.01)
MaxPooling2D	(15, 15, 128)	0	-	-
Conv2D	(15, 15, 256)	295168	ReLU	L2(0.01)
MaxPooling2D	(7, 7, 256)	0	-	-
BatchNormalization	(7, 7, 256)	1024	-	-
Dropout	(7, 7, 256)	0	-	0.3
Flatten	(12544)	0	-	-
Dense	(512)	6423040	ReLU	L2(0.01)
Dropout	(512)	0	-	0.3
BatchNormalization	(512)	2048	-	-
Dense	(256)	131328	ReLU	L2(0.01)
BatchNormalization	(256)	1024	-	-
Dense	(128)	32896	ReLU	L2(0.01)
BatchNormalization	(128)	512	-	-
Dense	(60)	7740	Softmax	L2(0.01)

As indicated in Table 1, individual layers are characterized by specific neuron counts; the Conv2D layers encompass 32, 64, 128, and 256 neurons, while the Dense layers consist of 512, 256, and 128 neurons, culminating in the final output layer comprised of 60 neurons. For activation, the Conv2D layers and the initial three Dense layers leverage the Rectified Linear Unit (ReLU) activation function. This selection is based on the advantageous characteristics of ReLU in enhancing the training efficiency and overall performance of deep networks, as referenced in [25]. Nonetheless, in the final Dense layer, the Softmax function is employed to generate a probability distribution across the 60 output classes, aligning with the requirements of multi-class classification tasks. The model's input comprises images sized 124 x 124 with a single-color channel, which undergo processing via a sequence of Conv2D, MaxPooling2D, Batch Normalization, and Dropout layers. This process concludes with a Flatten layer that transforms the tensor to suit the Dense layers, ultimately leading to the 60-neuron output layer, corresponding to the 60 classes within the dataset. Each layer, along with its associated parameters, performs a distinct function in handling the image data, identifying distinctive features, and ultimately categorizing the input into one of the 60 classes. The architecture of the proposed model capitalizes on CNNs to address image classification tasks, offering a robust and efficient tool aligned with our objectives. Figure 5 visually delineates architecture of proposed CNN model.

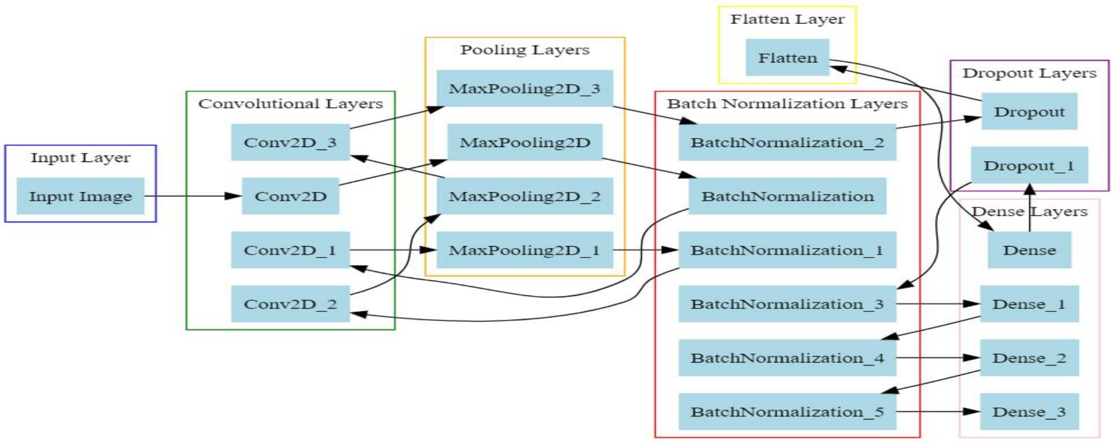

Figure 5. Architecture of proposed CNN model.

4.2.4. The Performance Evaluation

In assessing our proposed model, we conduct a comparative analysis between the classification performances of two models: the proposed CNN model trained and tested on the original dataset, and the proposed model trained and tested on the dataset after resampling. For evaluating the proposed model's FR capabilities, five metrics—accuracy, precision, recall, F1-score, and the error rate—are selected as performance measures, as outlined in reference [26]. It is important to note that the evaluation metrics should ensure equal treatment for each class. Specifically, the model's performance was assessed for each individual class, and subsequently, an average result for the entire subset was computed.

5. EXPERIMENTS AND RESULTS

Two experiments were conducted to explore the effect of data imbalance on the performance of the proposed model. These experiments involved utilizing both the original (imbalanced) dataset and a balanced dataset. The original dataset comprised 10,800 images across 60 classes. For our experiments, this dataset was divided into training and testing sets, with an 80% allocation for training and a 20% allocation for testing. Throughout the training phase, a random selection of

20% from the training set was employed to form the validation set. The purpose of incorporating a validation set was to aid in monitoring the model's performance on unseen data while training and to manage the risk of overfitting. Both sets included samples from the pool of 60 distinct classes. The images underwent resizing to dimensions of (124, 124) pixels. To ascertain the ideal number of epochs, we conducted training iterations, commencing with 50 epochs and incrementally progressing to 100, 200, and beyond up to 500 epochs. Unfortunately, due to limitations in hardware resources, we were unable to proceed with further training beyond 500 epochs as it demanded extensive time and processing power. As a result, the model underwent 500 epochs of training with a batch size of 32, a size previously validated in prior studies. To mitigate overfitting in complex, multi-layered networks, a dropout rate of 0.3 was introduced as a regularization technique. Furthermore, L2 regularization, set at 0.01, was implemented specifically on the convolutional layers to further address overfitting concerns. The non-saturating ReLU activation function, commonly used in CNNs, was employed to expedite training by mitigating the vanishing gradient issue. The choice of the Categorical Cross Entropy as the loss function is based on its suitability for multi-class classification tasks, as referenced in [27]. Consequently, this loss function assesses the alignment between the model's predictions and the true values in the context of the proposed model. For optimization purposes, the Adaptive Moment Estimation (Adam) optimizer was employed, leveraging a learning rate of 0.0001 due to its recognized efficiency. Consistency in the network parameters for the CNN was maintained across both experiments. The proposed system was developed using the Tensorflow, scikit-learn and Keras packages in Jupyter Notebook with Python language. The specific hyperparameters for the model, derived through experimentation, are detailed in Table 2.

Table 2. Hyper-parameters for the proposed model.

Hyperparameters	**Value**
Optimizer	ADAM
Learning Rate	0.0001
Loss	Categorical Cross Entropy
Activation Function	ReLU
Metrics	Accuracy, F1-score, Recall, Precision and Error Rate.
L2 Regularization	0.01
Epochs	500
Batch Size	32
Training Split	6912 Images
Validation Split	1728 Images
Testing Split	2160 Images

5.1. Experiment I: Performance Proposed Model with Imbalanced Dataset

In the first experiment, the proposed CNN model underwent training using the original (imbalanced) dataset. The outcomes highlighted the influence of class imbalance on the training of the CNN model, showcasing a decline in performance when data resampling techniques were not applied. Throughout each epoch, the loss and accuracy for both training and validation sets were closely tracked to observe the model's learning progress. The learning outcomes, encompassing training and validation accuracy alongside their respective losses, are illustrated in Figure 6. Notably, the model attained an overall accuracy of 88.77% on the training data. On the validation set, the accuracy and loss curves display suboptimal behavior. Specifically, the validation accuracy fluctuates between 0.70 and 0.76 across the majority of epochs, while the validation loss exhibits an initial sharp decrease, followed by a gradual decline throughout most epochs. It's important to emphasize that the testing data remained entirely separate and were not incorporated into the training process.

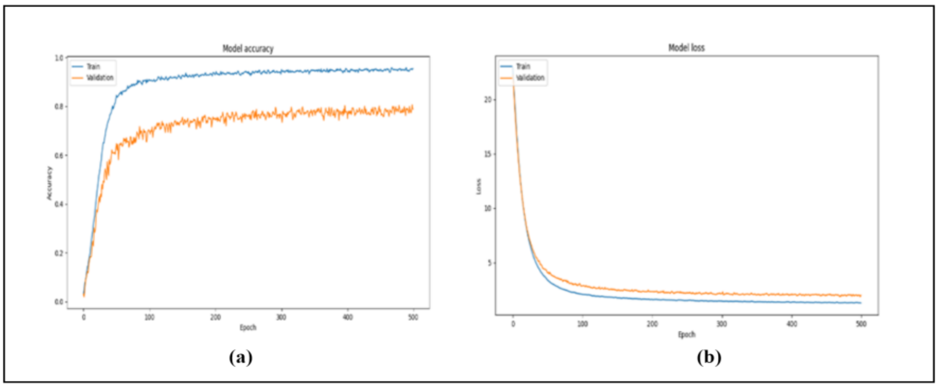

Figure 6. The learning curves of the proposed model before resampling: (a) The accuracy learning curves of the model; (b) The loss learning curves of the model.

Following the training phase, the model underwent evaluation using the testing dataset, resulting in a test accuracy of 81.17% and a loss value of 1.8329. Figure 7 depicts a bar chart illustrating the precision, recall, and F1-score pertaining to each class. The outcomes from experiment I are succinctly summarized in Table 3.

Table 3. The results of experiment I.

Metrics	**Value**
Training Accuracy	88.770%
Training Loss	2.3582
Validation Accuracy	71.321%
Validation Loss	3.0188
Test Accuracy	81.171%
Test Loss	1.8329
Precision	83.411%
Recall	81.712%
F1-score	81.351%
Error Rate	18.828%

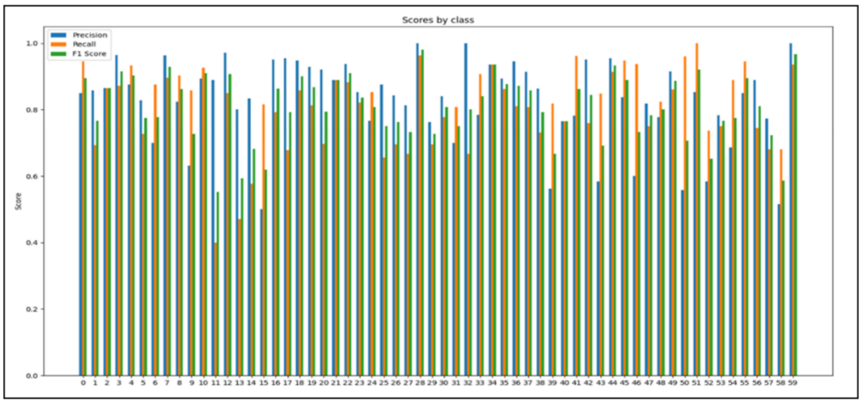

Figure 7. The performance evaluation of the proposed model for experiment I.

This experiment thoroughly evaluated the model's performance when trained without data resampling. These results will act as a baseline for comprehending the impacts of data resampling techniques, which will be further investigated in the subsequent experiment.

5.2. Experiment II: Performance with Balanced Augmented Dataset

In this experiment, the proposed system underwent training and evaluation subsequent to implementing a preprocessing procedure involving resampling techniques. This adjustment aimed to alter the distribution of classes within the dataset, mitigating the existing class imbalances. Following resampling, an augmentation technique was applied to both the training and validation datasets to prevent overfitting. Further details outlining these procedures are expounded upon in the subsequent sections.

5.2.1. Data Resampling

Figure 8(a) illustrates the high imbalance present within the dataset, with 60 classes exhibiting dominance. Conversely, Figure 8(b) displays the outcome of the data resampling process, showcasing a balanced dataset where each class contains an equal number of images per identity. The imbalance concern was tackled by employing a combination of random oversampling and random under-sampling techniques, resulting in the generation of a balanced dataset. This approach equalized the classes by ensuring a specific number of images per class, determined by a user-defined parameter. Throughout this process, various values were experimented with, ultimately leading to the optimal outcome when all classes were equalized with 180 images each. In our system, the hybrid sampling method starts by decreasing the size of the majority classes by 10% (20 images). Subsequently, the minority classes are oversampled until all classes possess an identical size, comprising 180 images each. This guarantees uniform representation of all classes in the dataset, preventing any bias toward the majority classes within the model.

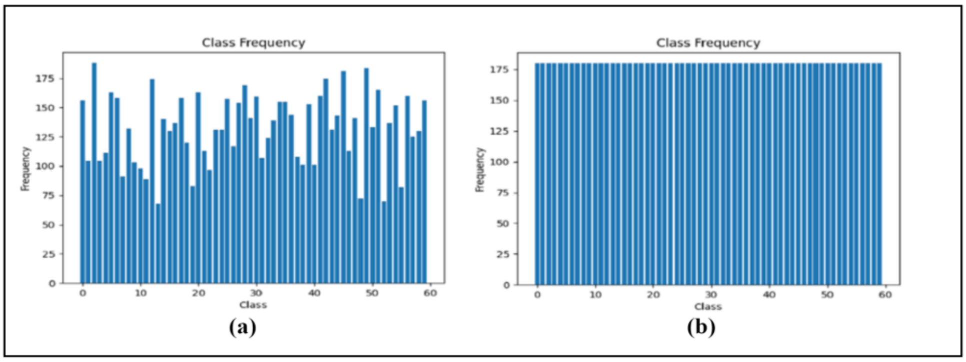

Figure 8. Class frequency: (a) Class frequency before resampling. (b) Class frequency after resampling.

5.2.2. Data Augmentation

In our model, on-the-fly data augmentation was used to combat model overfitting. This technique facilitates improved generalization to unseen data by generating new data in each epoch. Augmenting the training and validation sets with diverse samples helps the model make more robust and generalizable predictions, while the testing dataset remains unchanged. This approach ensures the preservation of the testing dataset's integrity, aimed at replicating real-world, unaltered data as accurately as possible. In addition, maintaining the original state of our testing dataset allows for a fair and unaltered benchmark, enabling an assessment of the model's ability to generalize predictions to unseen data. In this paper, data augmentation involves horizontal flipping and slight rotation of randomly selected face images. The augmentation process uses the Image Data Generator layer in Keras. The specified augmentation parameters in this paper consist of zoom_range=0.1, brightness_range=(0.9, 1.1), rotation_range=10, width_shift_range=0.2, horizontal_flip=True, and height_shift_range=0.2.

5.2.3. Classification

In this experiment, the proposed CNN-based classifier was trained once more, this time utilizing the balanced augmented dataset, and subsequently evaluated on the unprocessed test set. The learning progress of the model with the balanced data was monitored by tracking the training and validation loss and accuracy across all epochs. The training concluded with a training accuracy of 91.108% and a validation accuracy of 85.392%. Figure 9 illustrates the learning curves of the proposed model, highlighting its response to the balanced and augmented data throughout the training process.

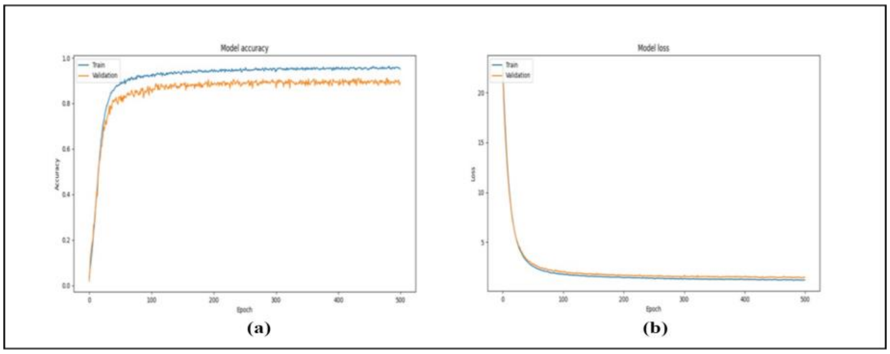

Figure 9. The learning curves of the proposed model after resampling: (a) The accuracy learning curves of the model; (b) The loss learning curves of the model.

The training and validation accuracy exhibited swift initial increments followed by gradual improvements over multiple epochs, while the loss learning curves in this experiment indicated lower values compared to the first experiment. Following the model's training, evaluation was conducted using the testing dataset. Table 4 indicates an increase in test accuracy, reaching 91.296% in comparison to the basic dataset. Moreover, it showcases an enhancement in the classification performance, evident in the overall accuracy, precision, recall, and F1-score (as outlined in Table 4). Figure 10 displays the visualized results of other computed metrics, while Table 4 summarizes the outcomes of experiment II.

Table 4. The results of experiment II.

Metrics	**Value**
Training Accuracy	91.108%
Training Loss	2.0366
Validation Accuracy	85.392%
Validation Loss	2.2409
Test Accuracy	91.296%
Test Loss	1.3913
Precision	92.035%
Recall	91.296%
F1-score	91.304%
Error Rate	8.703%

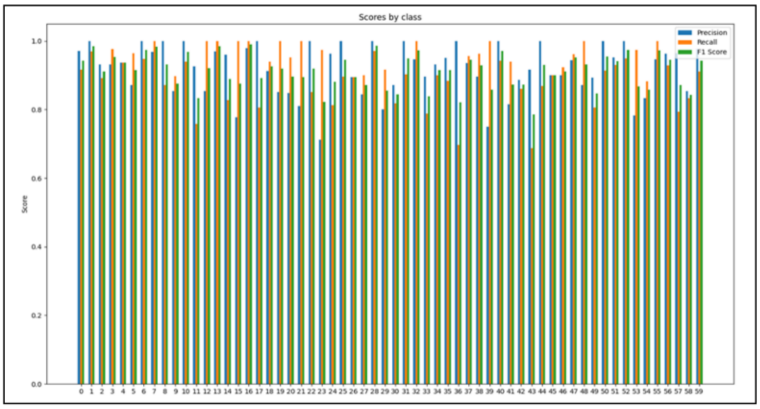

Figure 10. The performance evaluation of the proposed model for experiment II.

This experiment validates that through this dataset transformation, we achieve improved accuracy and reduced loss rates, leading to enhanced test results.

6. Discussion

In this study, two experiments were carried out to investigate the impact of an imbalanced dataset on the performance of CNN techniques for FR, one involving a resampling strategy and the other without. The initial experiment involved training the model using a basic dataset without employing any data resampling techniques. The training and validation losses and accuracies followed a conventional learning curve. However, despite the model achieving an overall test accuracy of 81.17%, it is crucial to note the varied performance across different classes due to the inherent class imbalance in the dataset. Conversely, in the second experiment, the proposed CNN model was trained on a balanced dataset achieved through hybrid oversampling and under-sampling techniques. The objective of this resampling was to rectify the inherent class imbalances within the dataset. Through the same training protocol employed in the initial experiment, the training and validation losses and accuracies indicated an enhanced learning curve. As a result, the model achieved a substantial improvement in overall accuracy, reaching 91.29%. Moreover, the model showcased a more balanced performance across classes, signifying its effective generalization across the entire spectrum of classes. The summary of the model's performance on the testing dataset for both experiments is provided in Table 5.

Table 5. Performance evaluation on testing dataset before and after resampling.

Metrics	Proposed CNN Model + Imbalanced Dataset	Proposed CNN Model + Balanced Dataset
Accuracy	81.171%	91.296%
Loss	1.8329	1.3913
Precision	83.411%	92.035%
Recall	81.712%	91.296%
F1-score	81.351%	91.304%
Error Rate	18.828%	8.703%

Figure 11 displays the performance analysis of both experiments concerning accuracy, precision, recall, and F1-score using the Pins dataset. It's important to note that the average performance of the proposed CNN model trained on imbalanced datasets was compared to the performance achieved through the same training method when the data was balanced. The comparison between the two experiments underscores the substantial and noteworthy impact of class imbalance on the CNN model's performance in FR. The Avg. F1-score is specifically adopted as the primary evaluation metric due to its significance in assessing imbalanced data, representing the trade-offs between precision and recall. These metrics demonstrate that the proposed model not only enhances predictions for individual minority classes but also improves overall performance. The performance analysis for the proposed model is outlined as follows: First, regarding accuracy, the proposed CNN model achieved 91.108% accuracy when trained on balanced data. In contrast, when trained on the original dataset, the proposed model attained an accuracy of 88.770%. Upon testing the dataset, the proposed classifier showcased the highest accuracy, achieving 91.296% and 81.171% on the balanced and imbalanced datasets, respectively. Second, the precision value for the proposed model stood at 83.411% when trained on the basic dataset. In contrast, the best precision result of 92.035% was achieved with the balanced dataset. Third, concerning recall, the proposed model attained 81.712% on the imbalanced dataset and improved to 91.296% after implementing resampling techniques. Finally, in terms of F1-score, the assessment yielded 81.351% and 91.304% without and with a balanced dataset, respectively. The CNN proposed in this paper achieved an error rate of 18.828% on the original data. Table 5 indicates consistent precision, recall, and F1-score across imbalanced classes, accompanied by a notably small error rate of 8.7%. These findings highlight that the performance of the proposed FR system exhibits approximately 10% higher accuracy compared to the performance the proposed CNN model with imbalanced data.

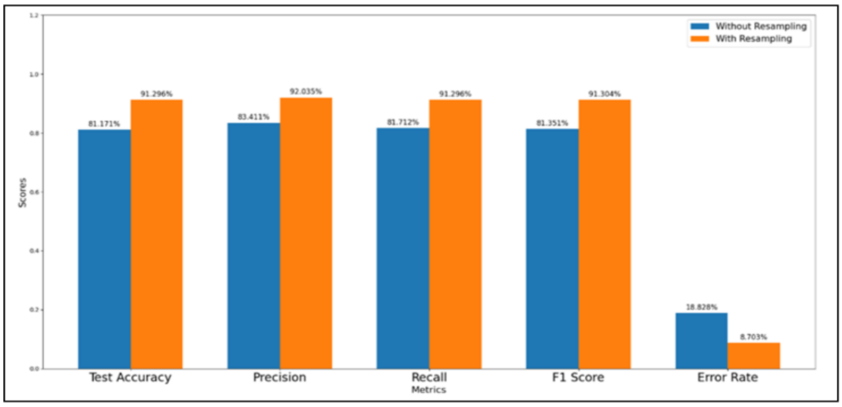

Figure 11. Comparison of model metrics with and without data resampling.

The findings indicate that the proposed system demonstrates effective performance, enhancing the classification accuracy on the Pins Face Recognition dataset. This improvement can be attributed to three key factors: the designed CNN network structure, the utilization of hybrid data resampling techniques, and the implementation of augmentation method. The integration of these three techniques collectively enhances the model's classification performance, especially on imbalanced datasets.

7. CONCLUSION

In this study, we delved into the creation and assessment of a CNN model designed for FR. In addition, we explored the influence of imbalanced datasets, a crucial issue particularly notable for CNNs when dealing with extensive datasets, emphasizing the significant concern regarding accuracy and performance. Our experiments involving the direct utilization of imbalanced datasets revealed notable accuracy levels when compared with the results obtained from the same dataset post-balancing. The application of resampling techniques led to a 10.12% improvement in accuracy outcomes for our model. From our conducted experiments, we highlighted the crucial role of data resampling in mitigating the prevalent class imbalance within FR tasks. These findings suggest the importance of considering data distribution when employing such techniques for image classification, as a pronounced imbalance can detrimentally impact the obtained results. While this study contributed to comprehending the significance of data resampling in a CNN Face Recognition model, it is crucial to acknowledge the multitude of elements influencing the performance of these models. Various factors can impact the effectiveness of such models, prompting the necessity for ongoing research and adaptation. Within ML, especially in Face Recognition tasks, there exists a perpetual need for advancements and further exploration.

REFERENCES

[1] S. Karamizadeh, S. M. Abdullah, and M. Zamani, “An Overview of Holistic Face Recognition,” International Journal of Research in Computer and Communication Technology (IJRCC), vol. 2, no. 9, pp. 738–741, Sep. 2013.
[2] Y. Kim, W. Park, and J. Shin, “BroadFace: Looking at Tens of Thousands of People at once for Face Recognition,” in European Conference on Computer Vision, Cham: Springer International Publishing, Aug. 2020, pp. 536–552.
[3] M. Coşkun, A. Uçar, Ö. Yıldırım, and Y. Demir, “Face recognition based on convolutional neural network,” in 2017 International Conference on Modern Electrical and Energy Systems (MEES), IEEE, Nov. 2017, doi: 10.1109/mees.2017.8248937.
[4] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” Neural Networks, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/j.neunet.2018.07.011.
[5] P. Jeatrakul and K. W. Wong, "Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm," in 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, Jun. 2012, pp. 1-8, doi: 10.1109/IJCNN.2012.6252450.
[6] J. M. Sáez, B. Krawczyk, and M. Woźniak, “Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets,” Pattern Recognition, vol. 57, pp. 164–178, Sep. 2016, doi: 10.1016/j.patcog.2016.03.012.
[7] S. Tyagi and S. Mittal, “Sampling approaches for imbalanced data classification problem in machine learning,” in Lecture notes in electrical engineering, 2019, pp. 209–221. doi: 10.1007/978-3-030-29407-6_17.
[8] S. Nejatian, H. Parvïn, and E. Faraji, “Using sub-sampling and ensemble clustering techniques to improve performance of imbalanced classification,” Neurocomputing, vol. 276, pp. 55–66, Feb. 2018, doi: 10.1016/j.neucom.2017.06.082.
[9] Taherkhani, G. Cosma, and T. McGinnity, “AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning,” Neurocomputing, vol. 404, pp. 351–366, Sep. 2020, doi: 10.1016/j.neucom.2020.03.064.
[10] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y. -H. Lu, S. -C. Chen and M. -L. Shyu, “Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification,” in 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, Apr. 2018, pp. 112-117, doi: 10.1109/MIPR.2018.00027.
[11] X. Wu, R. He, Z. Sun, and T. Tan, “A light CNN for deep face representation with noisy labels,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2884–2896, Nov. 2018, doi: 10.1109/tifs.2018.2833032.
[12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj and L. Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jul. 2017, pp. 6738-6746, doi: 10.1109/CVPR.2017.713.
[13] R. J. Hassan and A. M. Abdulazeez, “Deep Learning Convolutional Neural Network for face Recognition: a review,” International Journal of Science and Business, vol. 5, no. 2, pp. 114–127, Jan. 2021, doi: 10.5281/zenodo.4471013.
[14] Z. Sun and G. Tzimiropoulos, “Part-based Face Recognition with Vision Transformers,” arXiv (Cornell University), Nov. 2022, doi: 10.48550/arxiv.2212.00057.
[15] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 5962-5979, 1 Oct. 2022, doi: 10.1109/TPAMI.2021.3087709.

[16] P. Hensman and D. Masko, “The Impact of Imbalanced Training Data for Convolutional Neural Networks,” Degree Project in Computer Science, KTH Royal Institute of Technology, Jan. 2015, [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A811111.
[17] H. Lee, M. Park and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3713-3717, doi: 10.1109/ICIP.2016.7533053.
[18] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” Journal of Big Data, vol. 6, no. 1, Mar. 2019, pp. 1-54, doi: 10.1186/s40537-019-0192-5.
[19] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, “Fundamental concepts of Convolutional Neural Network,” in Intelligent systems reference library, 2019, pp. 519–567. doi: 10.1007/978-3-030-32644-9_36.
[20] N. Shahriar, “What is Convolutional Neural Network — CNN (Deep Learning),” Medium, Feb. 01, 2023. [Online]. Available: https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5
[21] “Pins face recognition,” Kaggle, Mar. 2020. https://www.kaggle.com/datasets/hereisburak/pins-face-recognition.
[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.
[23] R. Ruizendaal, “Deep Learning #3: More on CNNs & Handling Overfitting,” May. 2017, [Online]. Available: https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d
[24] Keras Team, “Keras documentation: Dropout layer.” [Online]. Available: https://keras.io/api/layers/regularization_layers/dropout/
[25] J. Brownlee, “A gentle introduction to the rectified Linear Unit (RELU),” MachineLearningMastery.com, Aug. 2020, [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.
[26] M. Grandini, E. Bagli and G. Visani, "Metrics for multi-class classification: An overview" in arXiv:2008.05756, 2020.
[27] A. Kumar, “Keras - Categorical Cross Entropy Loss Function - Analytics Yogi,” Analytics Yogi, Oct. 28, 2020. https://vitalflux.com/keras-categorical-cross-entropy-loss-function/.