IMAGE SEGMENTATION AND CLASSIFICATION USING NEURAL NETWORK

Fatema Tuj Zohra, Rifa Tasfia Ratri, Shaheena Sultana, and Humayara Binte Rashid

Department of Computer Science and Engineering
Notre Dame University Bangladesh

Abstract. Image segmentation and classification tasks in computer vision have proven to be highly effective using neural networks, specifically Convolutional Neural Networks (CNNs). These tasks have numerous practical applications, such as in medical imaging, autonomous driving, and surveillance. CNNs are capable of learning complex features directly from images and achieving outstanding performance across several datasets. In this work, we have utilized three different datasets to investigate the efficacy of various pre-processing and classification techniques in accurssedately segmenting and classifying different structures within the MRI and natural images. We have utilized both sample gradient and Canny Edge Detection methods for pre-processing, and K-means clustering have been applied to segment the images. Image augmentation improves the size and diversity of datasets for training the models for image classification. This work highlights transfer learning's effectiveness in image classification using CNNs and VGG 16 that provides insights into the selection of pre-trained models and hyper parameters for optimal performance. We have proposed a comprehensive approach for image segmentation and classification, incorporating pre-processing techniques, the K-means algorithm for segmentation, and employing deep learning models such as CNN and VGG 16 for classification.

Keywords: Convolutional Neural Network, VGG 16, Image Segmentation, K-means, Image Classification.

1 INTRODUCTION

In the world of artificial intelligence (AI) and computer science, computer vision is a branch that focuses on giving computers the ability to interpret, process, and comprehend visual data from the outside environment. It involves developing algorithms and techniques for processing and analyzing images and videos, and extracting meaningful insights and information from them. A Convolutional Neural Network (CNN) is a type of deep neural network made for processing and evaluating that has a data grid-like structure, together with pictures or movies. Regarding computer vision, image identification is a key task and CNNs have emerged as the most advanced technique for it. Convolutional layers are used to extract information from images and fully connected layers are used to produce predictions in CNNs [30]. CNN is a sort of neural network developed primarily for image recognition tasks. Deep learning is another approach in machine learning that focuses on teaching neural networks. This carries out tasks that require intelligence that is comparable to that of a person, such as speech recognition, object recognition in pictures, and language translation. Recent developments in deep learning have helped the medical imaging industry detect many diseases [4]. To obtain high accuracy, the main purpose of medical image classification is to identify the parts of the human body that are harmful to health [5]. In the pre-processing stage, MRI images are preprocessed to remove noise and enhance contrast using a combination of histogram equalization and median filtering techniques.

In the segmentation stage K-means clustering algorithm divides the images into homogeneous segments based on color, texture, or intensity [29]. K-means is an unsupervised learning algorithm that partitions data into K clusters, initially selecting random points

DOI: 10.5121/ijcsit.2024.16102

as cluster centers and iteratively refining them. The algorithm assigns data points to the nearest cluster center and updates centers based on the mean of points in each cluster. The process continues until cluster centers stabilize or a specified iteration limit is reached. Subtractive clustering is then employed to enhance segmentation by eliminating noise and merging similar clusters, demonstrating the effectiveness through experimental results. In the classification stage, pre-processed images are fed into a CNN model that is based on the VGG 16 architecture [6]. Nath *et.al.* provided a comprehensive overview of various image classification methods, including traditional techniques and deep learning-based approaches. They discussed the benefits and drawbacks of various approaches and how they were applied in various fields [7].

The motivation behind employing neural networks, particularly CNNs, for image segmentation and classification lies in their demonstrated effectiveness in various computer vision tasks. Neural networks offer improved accuracy, faster processing speeds, adaptability to diverse data types, and the potential for novel applications. By leveraging the structure and operation inspired by the human brain, these models can autonomously process and analyze images, reducing the reliance on manual intervention.

However, the adoption of neural networks in image processing poses challenges. Training large-scale deep neural networks demand substantial computational resources, and the scalability of such models remains a significant challenge. Additionally, ensuring generalizability across diverse datasets and real- world scenarios require addressing issues related to overfitting and model robustness. Furthermore, interpretability and explainability of neural network decisions can be challenging, especially in critical applications where human understanding is crucial. Balancing the trade-off between model complexity and computational efficiency is an ongoing challenge in the deployment of neural networks for image segmentation and classification. Despite challenges related to interpretability and generalizability, the use of artificial intelligence-based visual systems, as demonstrated in fruit classification through cameras and algorithms, showcase the potential for autonomous image analysis. Continued research and innovation in neural network techniques are essential to overcome challenges and unlock the full potential of these models in revolutionizing image processing and analysis.

We have structured the rest of the paper as follows: Section 2 reviews the related works in this field, highlighting the gaps in current knowledge and explaining how this work addresses those gaps. Section 3 describes the models used for classification. Section 4 describes the methodology used to conduct the work. Section 5 presents the result of this work, including performance analysis. Section 6 summarizes the findings of our work, concludes, and suggests avenues for future research.

2 LITERATURE REVIEW

This section reviews the related works in this field, highlights the gaps in current knowledge, and explains how this work addresses those gaps. Methodological innovation is an important contribution to method articles, and editors typically ask how the technique in question differs from previously published methods. Image segmentation and classification is a supreme task in the execution of computer vision. Object recognition, medical image analysis, autonomous driving, etc. are part of computer vision. Deep learning methods such as CNN, VGG 16, and k-means clustering have been admired in recent years for these applications. We have organized our literature review into two distinct sections, each addressing a specific aspect of our work: Convolutional Neural Networks (CNNs) and Transfer Learning.

In the section dedicated to Convolutional Neural Networks, we have extensively reviewed prior research endeavors related to CNN models. Specifically, we have delved into the historical body of work surrounding CNNs, analyzing their development and various applications.

In the context of Transfer Learning, our focus has been on VGG 16, a prominent model in this domain. Within this section, we have incorporated relevant studies and findings concerning VGG 16's usage and its adaptations in the realm of transfer learning. This approach enables us to comprehensively explore the landscape of previous research, encompassing both the broader CNN field and the specific contributions of VGG 16 in transfer learning applications.

2.1 Convolutional Neural Network (CNN)

CNNs have an architecture that helps them understand images step by step. They do this by using three layers(convolutional layers, pooling layers, and fully connected layers) that scan the image, and group information, and then make sense of it. AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet are just a few of the CNN designs that Sultana *et al.* offered an overview of, along with information on how well they performed on well-known image classification benchmarks like ImageNet [8]. A CNN model with two completely connected layers and four convolutional layers was produced by Khan *et al.* To avoid overfitting, they employed a dropout and Rectified Linear Unit (ReLU) activation function. The rate of dropout is 0.5 [9]. A well-known benchmark dataset for computer vision is the MNIST dataset. To attain high accuracy, Chattopadhyay *et al.* suggested a CNN architecture that combines convolutional, max-pooling, dropout, and fully connected layers with optimal hyperparameters [10]. Kumar *et al.* proposed convolutional layers, which make up the CNN's architecture have various filter sizes and pooling layers for down-sampling the feature maps. In addition, the authors employ batch normalization and dropout methods to reduce overfitting and increase the network's generalization capabilities [11]. As proposed by Kaushik *et al.* an approach that uses a CNN to learn the features of the image and predict the segmentation map. Gomez *et al.* presented an approach that is evaluated using a dataset of thermal images obtained from breast cancer patients and healthy subjects. According to the results, the suggested method works well in categorizing thermal pictures into normal and malignant breast tissues, and it has the potential to be employed as a non-invasive tool for early breast cancer testing [12]. Tripathi analyzed the effect of different factors, such as network architecture, data augmentation, and hyperparameters, on classification accuracy. They concluded that deeper networks with appropriate regularization techniques and data augmentation can significantly improve classification accuracy [13].

2.2 Transfer Learning

Transfer learning is a strong approach that allows pre-trained models to be utilized for new image categorization problems. It has proven cutting-edge performance on many benchmarks. Transfer learning is applied to a range of applications that also involve medical picture classification. To identify brain tumors in MRI images, Siddique *et al.* proposed a CNN model that has a high level of accuracy. The proposed model is used as a diagnostic tool in clinical settings to aid radiologists in the detection of brain tumors [14]. Agarwal proposed a deep-learning approach for classifying cooking images into different states using the VGG 19 network [15]. D.C. Febrianto *et al.* suggested a method for training a CNN using a dataset of brain magnetic resonance imaging (MRI) images that contain both normal and tumor scans [16]. Hoque provided valuable insights into the application of deep learning models for medical image analysis, specifically for brain tumor detection, and highlighted the potential benefits of using CNNs for this task. The comparative analysis of VGG 16 and VGG 19 models provides a useful benchmark for future studies in this area [17]. Abd-Ellah*et al.* compared the performance of the VGG 16 and VGG 19 networks on a dataset of 500 MRI scans, consisting of 250 normal scans and 250 scans with tumors. They also compared the performance of the VGG networks with a conventional CNN approach [18]. Pravallika and Baskar suggested a brain tumor classification method based on image processing that employs the VGG 16 CNN and classifiers using support vector machines (SVM). Analysis of the proposed system on a dataset of 210 brain MRI images, and the results show that the VGG 16 network outperforms the SVM classifier with an accuracy of 95.2% compared to 89.5% [19]. Agus *et al.* proposed a system that involves training the VGG 16 model on the MRI images to extract features followed by a classification layer using softmax regression for classifying the image into one of the two categories, glioma or non glioma [20]. Simonyan *et al.* introduced the VGGNet architecture, which achieved outstanding performance on the ImageNet dataset. They also demonstrated the efficiency of transfer learning for image classification by adjusting the previously trained VGGNet on a smaller dataset [34]. Long *et al.* presented a deep adaptation network (DAN) that can learn domainagnostic characteristics. They used various photos to highlight the usefulness of DAN.

3 MODELS USED IN IMAGE CLASSIFICATION

In this section, we have discussed the models that have been used for classification. We have used CNN and VGG 16 to classify our images.

3.1 Convolutional Neural Network

A convolutional neural network has three layers. Input layer, hidden layer, and output layer. Data is received by the input layer, then it is sent to hidden layers. The process of removing features from the data is carried out by the hidden layers, and each layer has multiple nodes or neurons that perform the calculations. Based on the features that The output layer has been gathered by the hidden layers generates the outcome or prediction. At the time of training, the weights of the nodes are adjusted. This has been done to minimize the error between the predicted output and the actual output. This process has been repeated iteratively until the network achieves the desired accuracy. CNNs, in particular, are specialized neural networks that have been designed to process images and other types of multidimensional data [30]. Convolutional Neural Networks are often employed in image categorization, object identification, and other computer vision applications. CNNs are built to automatically learn and extract features from pictures using convolution and pooling. Convolution is performed by applying a tiny filter or kernel over an input picture and computing the dot product of the filter with each patch of pixel then the output is routed via an activation function like ReLU to induce nonlinearity. The feature maps are then down-sampled and the output dimensionality is reduced via pooling. Figure 1 shows us the architecture of convolutional neural network [32]. Here are the key details of CNN:

1. **Convolutional Layer:** To extract characteristics from the input image, this layer has used several filters (kernels). The filters convolve over the image and produce a feature map. Convolutional layers can learn low-level features like edges, lines, and curves. The

formula for the 2D convolution operation in a convolutional layer can be expressed as follows:

2. **ReLU Activation Layer:** ReLU is an activation function for Rectified Linear Units. This activation function has been applied to the convolutional layer's output. This aids in introducing non-linearity into the model and permits it to pick up on more intricate aspects. The ReLU function could be defined as:

$$f(x) = \max(0, x) \tag{1}$$

3. **Pooling Layer:** This layer has been used to reduce the size of the feature maps. It has taken the maximum or average value of each patch of pixels. This has decreased the size of feature maps which helped to reduce overfitting.
4. **Fully Connected Layer:** It has taken the output of the layer and runs it through a group of neurons that are completely linked to every neuron in the layer before. To categorize the supplied image, utilize this layer.
5. **Softmax Layer:** For generating a probability distribution over the classes, it has applied a softmax function to the output of the fully connected layer. The softmax function follow as:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2}$$

6. **Loss Function:** A loss function computes the differentiation between two values. a model's anticipated output and the actual result (i.e., the ground truth) for a certain input. It is a crucial component of building a machine-learning model because it has directed the optimization process to the best possible collection of model parameters. CNNs can be trained using a variety of optimization algorithms, such as SDG (Stochastic gradient descent) or Adam. They can also be trained on large datasets, such as ImageNet, and fine-tuned on smaller datasets for specific tasks. The most advanced performance on a wide range of computer vision tasks is possible with CNNs when trained via backpropagation and gradient descent. Numerous more uses include style transfer, segmentation, object identification, and image classification.

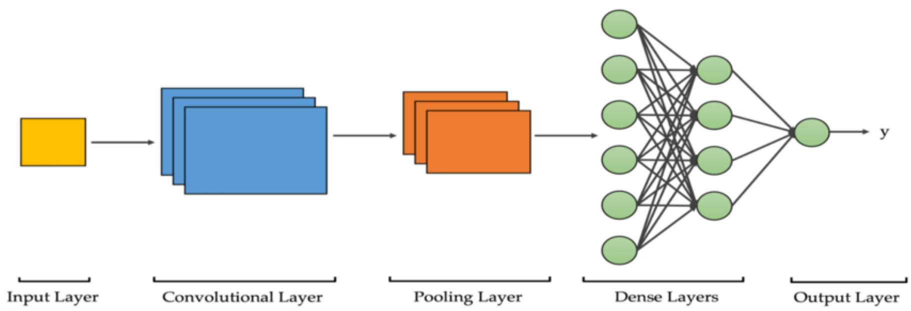


Fig. 1: CNN Architecture.

3.2 Visual Geometry Group

VGG (Visual Geometry Group) is a group of researchers from Oxford University that specializes in computer vision and deep learning. They are well-known for their contributions to the area of convolutional neural networks that excel in image recognition tasks.

In 2014, Simonyan *et al.* suggested a new architecture for CNNs that delivered outstanding outcomes on several image recognition benchmarks [34]. This architecture is known as VGGNet. This consists of a series of convolutional layers with small 3x3 filters. These layers are followed by max-pooling layers and end with several fully connected layers. The VGGNet is known for its simplicity and is still widely used today as a baseline for image recognition tasks. VGG 16 deep neural network with 16 layers. Figure 2 shows us how the VGG 16 model works layer by layer [33]. Here are the details of the VGG 16 architecture:

1. **Input Layer:** A RGB image with the dimensions 224 x 224 x 3 have been used as the input layer.
2. **Convolutional Layers:** There are 13 convolutional layers in VGG 16. Each convolutional layer has a 3 x 3 kernel and used a stride of 1 pixel. Each layer's number of filters increased as it gone deeper into the network, starting with 64 filters in the first layer and doubling after each max pooling layer.
3. **Max Pooling Layers:** There are 5 max-pooling layers in VGG 16. Each max-pooling layer has a 2 x 2 kernel and a stride of 2 pixels.
4. **Fully Connected Layers:** VGG 16 has 3 layers that have been completely linked. The amount of neurons in the first two totally connected layers (4,096 neurons each) and the third fully connected layer (1,000 neurons) correspond to the number of classes in the ImageNet dataset.
5. **Softmax Layer:** For obtaining the final probability distribution across the 1,000 classes, a softmax function have applied to the output of the last fully connected layer.

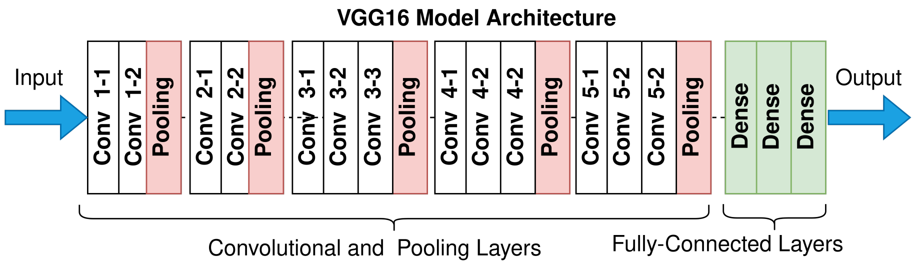


Fig. 2: VGG 16 Architecture.

4 METHODOLOGY

This section describes the methodology used to conduct the work. In our work, we have followed a procedure that contains all of the methods or tools that we have used to analyze image segmentation and classification in our work. Figure 3 shows the procedure of our work. First of all, we have collected a dataset. We have worked on three datasets. All of them have been collected from Kaggle [1].

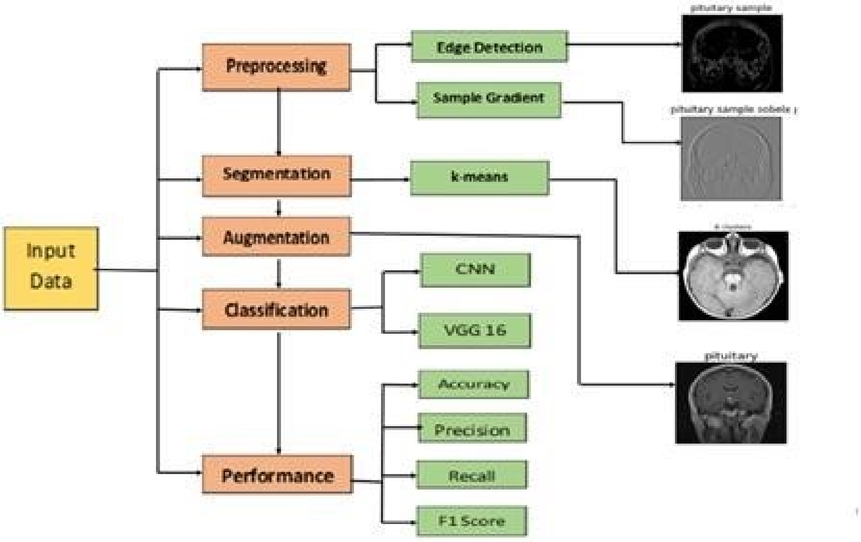


Fig. 3: Workflow of the work

4.1 Data Collection

In our work, we have chosen three datasets shown in Table 1 from Kaggle based on two classes of training and testing [1]. The first and second datasets are about brain tumor images [21] and [22]. Dataset 1 and Dataset 2 contain around 3274 images and 7023 images in total. They have 4 classes including glioma tumors, no tumors, meningioma tumors, and pituitary tumors. The third dataset is about natural images. This dataset contains a collection of 7 categories of natural images with a total of 8,789 images [1]. In our work, we select four categories such as dog, cat, fruit and flower.

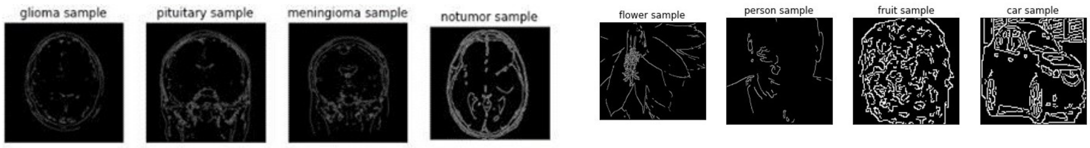
Splitting Category					
Dataset 1		Dataset 2		Dataset 3	
Train	Test	Train	Test	Train	Test
2880	394	5712	1311	2642	788

Table 1: Splitting Category of all Datasets

4.2 Dataset Preprocessing

Before analyzing the dataset, it is necessary to preprocess the dataset for better results. The goal of preprocessing is to prepare the data for analysis and to increase the model's accuracy by reducing noise and removing inconsistencies. We have used Canny Edge Detection and Sample Gradient as preprocessing steps.

Canny Edge Detection: This is a widely used algorithm for detecting edges in images. It worked by identifying areas in the image with a significant change in intensity or color and marking them as edges. The algorithm involves several steps, including smoothing the image to remove noise and calculating the gradient of the image to determine the edges, and non-maximum suppression is used to soften the edges [31]. The edges serve as the final threshold to create a binary picture, with the background represented by black pixels and the edges by white pixels. Figure 4 shows us how canny edge detection works on four types of MRI images and natural images.



(a) MRI Images (b) Natural Images

Fig. 4: Canny Edge Detection

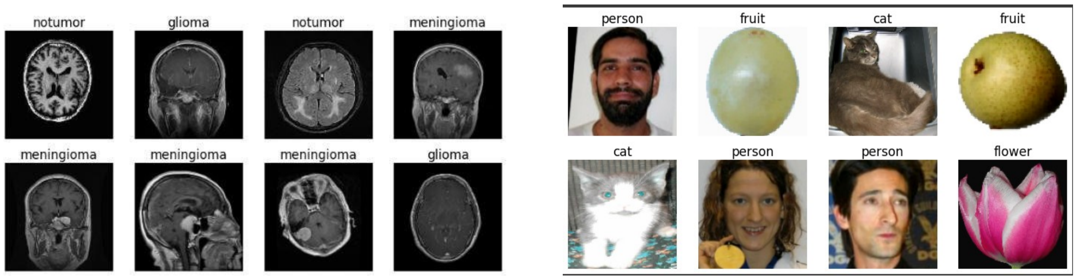
Sample Gradient: In image processing, the gradient is a measure of the rate at which the pixel intensity changes in an image. In our work, we have used the Sobel gradient for image preprocessing. The Sobel gradient is a simple and widely used method for edge detection in image processing. S. B. Kulkarni and S. G. Bhirud described the Sobel gradient as a simple and effective method for edge detection. They explain that the Sobel gradient works by calculating the gradient magnitude of an image by convolving it with two filters, one for the horizontal edges and another for the vertical edges [23]. Mustafa *et al.* explained that the Sobel gradient is a popular edge detection method due to its simplicity and effectiveness [24]. Wang proposed work to develop Laplacian operator-based edge detectors. The detectors utilized the second-order derivative of Gaussian filters to extract the edges from an image [30].

4.3 Image Segmentation

The technique of segmenting an image involves dividing it up into different sections or areas that are uniform in terms of things like color, texture, or intensity. Minaee *et al.* explain the importance of image segmentation in different uses of computer vision including object identification, recognition, and classification [25]. FCNs are a kind of deep neural network that can segment images pixel by pixel. To learn how to divide an image into several areas according to their semantic significance, Long *et al.* suggested an approach that involves training an FCN on a sizable dataset of annotated images [26]. One way to perform image segmentation is using the K-means. Data are divided into K clusters based on similarity using the unsupervised learning method K-means. Burney *et al.* presented a clear and short overview of the application of K-means clustering for segmentation. It highlights the advantages of this approach and demonstrates its effectiveness through experimental results [27].

4.4 Data Augmentation

Augmentation is used to boost the quantity and variety of training data without actually gathering any new data. By applying various transformations to existing data, such as rotating, cropping, or adding noise, it has created a new sample that is still representative of the original data for MRI images and natural images as shown in figure 5. The goal of data augmentation is to increase the diversity and volume of training data, which can improve deep learning models' generalization performance [28].



(a) MRI Images

(b) Natural Images

Fig. 5: Data Augmentation

4.5 Our CNN Model Structure

In CNN models, various layers work together in a structured manner. Convolutional layers extract image features, pooling layers decrease spatial resolution for efficiency, and fully connected layers handle the final classification or regression task, depending on the problem at hand. This is a sequential neural network model with the following layers:

1. Conv2D layer with 32 filters, kernel size of (3,3), ReLU activation function, and input shape of (IMAGE SIZE, IMAGE SIZE, 3). This layer applies 32 different filters to the input image, each of size 3x3, and applies the ReLU activation function to the output.
2. MaxPooling2D layer with pool size of (2,2). This layer reduces the size of the input image by taking the maximum value within each 2x2 window.
3. Conv2D layer with 64 filters, kernel size of (3,3), and ReLU activation function. This layer applies 64 different filters to the output of the previous layer, each of size 3x3, and applies the ReLU activation function to the output.
4. MaxPooling2D layer with pool size of (2,2). This layer reduces the size of the input image by taking the maximum value within each 2x2 window.
5. Conv2D layer with 32 filters, kernel size of (3,3), and ReLU activation function. This layer applies 32 different filters to the output of the previous layer, each of size 3x3, and applies the ReLU activation function to the output.
6. Flatten layer. This layer flattens the output of the previous layer into a 1D array.
7. Dense layer with 16 neurons and ReLU activation function. This layer applies a fully connected layer to the output of the previous layer, with 16 neurons and ReLU activation function.
8. Dense layer with 4 neurons and softmax activation function. This layer applies a fully connected layer to the output of the previous layer, with 4 neurons and softmax activation function.

The model have compiled using the 'adam' optimizer, 'sparse_ categorical_crossentropy' as the loss function, and 'accuracy' as the evaluation metric.

4.6 Our Transfer Learning Model

Transfer learning approach entails leveraging previously learned models as a foundation for new models. The VGG 16 architecture is a popular choice for transfer learning due to its excellent performance in image recognition tasks. We have built a transfer learning model based on the VGG 16 architecture using the Keras API. The pre-trained VGG 16 model is loaded with the imagenet weights, and all the layers are set to non-trainable, except for the last three layers in the VGG block, which are set to trainable. The pre-trained VGG

16 model comes first followed by a flattened layer, two dropout layers, a dense layer with 128 neurons with ReLU activation, and a final dense layer with neurons. The neurons are equal to the number of unique labels in the dataset and a softmax activation function. This new sequential model is then defined. The model has been made for tasks requiring several classes of categorization.

5 RESULT ANALYSIS

This section presents the result of this work, including performance analysis and shows how Convolutional Neural Networks (CNN) and VGG 16 architecture classify our different datasets into different categories, focusing on evaluating their performance.

This analysis used CNN and VGG 16 for classifying our datasets into different categories. The Adam optimizer have been used to train the neural network. We have used 10 epochs, a batch volume of 20, and sparse categorical cross-entropy as the loss function. The objective was to evaluate the performance of these models in accurately classifying MRI images and comparing their results. We have reported the results for 3 datasets which shows that the accuracy of VGG 16 is better than CNN for all datasets. For dataset 1, the CNN model training accuracy is 99% and test accuracy is 72%, while the VGG 16 achieved 98% in the training and 75% in the testing phase. In dataset 2, the CNN model achieved and VGG 16 both models achieved 99% in the training phase, but in the testing phase VGG 16 model result is better than the CNN model. In dataset 3, the CNN model achieved a training accuracy of 99% and a test accuracy of 82%. But VGG 16 model achieved 99% in the training phase and 95% in the testing. Table 2 shows the accuracy performance of the training and test sample for all models.

Model	Dataset 1		Dataset 2		Dataset 3	
	Train	Test	Train	Test	Train	Test
CNN	99	72	99	95	99	82
VGG16	98	75	99	97	99	95

Table 2: Accuracy performance between CNN and VGG 16

5.1 Evaluation Matrix

In machine learning, performance metrics are used to evaluate the effectiveness of a model. One such metric is the classification report, which evaluates the quality of a classification model. The report includes various performance metrics such as accuracy, precision, recall, F1-score, and support [2].

- **True Positive(TP):** The model successfully predicts the positive class and the actual outcome is positive.
- **True Negative(TN):** The model successfully predicts the negative class when the actual outcome is truly negative.
- **False Positive(FP):** The model predicts the positive class inaccurately when the real class is negative.
- **False Negative(FN):** The model predicts the negative class inaccurately when the real class is positive.

- **Accuracy:**Accuracy measures the percentage of correct predictions made by the model

$$Accuracy = TP + TN/P + N \tag{3}$$

- **Precision:** Precision measures the per centage of positive images that were correctly classified out of all the images that the model predicted as positive.

$$Precision = TP/TP + FP \tag{4}$$

- **Recall:**Recall measures the percentage of positive images that were correctly classified out of all the actual positive images.

$$Recall = TP/TP + FN \tag{5}$$

- **F1-Score:**The F1-score is the harmonic mean of precision and recall and is used to evaluate the overall performance of the model.

$$F1score = 2 * (Precision * Recall)/(Precision + Recall) \tag{6}$$

- **Support:** Support is the total number of images in a certain class. It represents the number of instances in the dataset that belong to a particular class.

By evaluating these metrics, researchers can determine the quality of their model and how well it has being performing in accurately identifying the correct labels for a given set of images. Finally, we have calculated the Precision, Recall, and F1-Score of the proposed CNN and VGG 16 model, which are commonly used performance evaluation metrics to measure the effectiveness of an image classification model in accurately identifying the correct labels for a given set of images.
In this work, experiments are conducted using three distinct datasets to train Convolutional Neural Network (CNN) and VGG16 models, varying in the sizes of training and test samples. Notably, Dataset 1, with 3000 data points, exhibited lower F1 scores compared to the other two datasets. The decision to exclude results from Dataset 1 is based on multiple considerations. Firstly, the relatively small size of the dataset may limit the models' ability to generalize effectively, potentially leading to decrease performance. Additionally, concerns about an imbalanced distribution of samples across classes in Dataset 1 could bias the models, particularly affecting performance metrics like the F1 score, especially for minority classes. In conclusion, the exclusion of results from Dataset 1 is a strategic choice aimed at ensuring the robustness and reliability of the conclusions derived from the experiments.

Convolutional Neural Network				
	Precision	Recall	F1 score	Support
no tumor	1.00	0.99	0.99	405
meningioma tumor	0.94	0.87	0.90	306
glioma tumor	0.90	0.96	0.93	300
pituitary tumor	0.97	0.98	0.98	300

(a) Dataset 2

Convolutional Neural Network				
	Precision	Recall	F1 score	Support
cat	0.81	0.67	0.73	259
dog	0.58	0.67	0.62	159
flower	0.86	0.93	0.90	183
fruit	0.97	0.99	0.98	187

(b) Dataset 3

Table 3: Evaluation Matrix using CNN

The accuracy, recall, F1 score, and support for the CNN using Dataset 2 and Dataset 3 are shown in Table 3. The table demonstrates that Dataset 3 have a lower precision, recall, and F1 scores than Dataset 2. Dataset 2 shown in Table 3a has the best precision, recall, and F1 score.

VGG 16				
	Precision	Recall	F1 score	Support
glioma tumor	0.94	0.98	0.96	300
meningioma tumor	0.97	0.94	0.96	306
no tumor	1.00	1.00	1.00	405
pituitary tumor	0.99	0.98	0.99	300

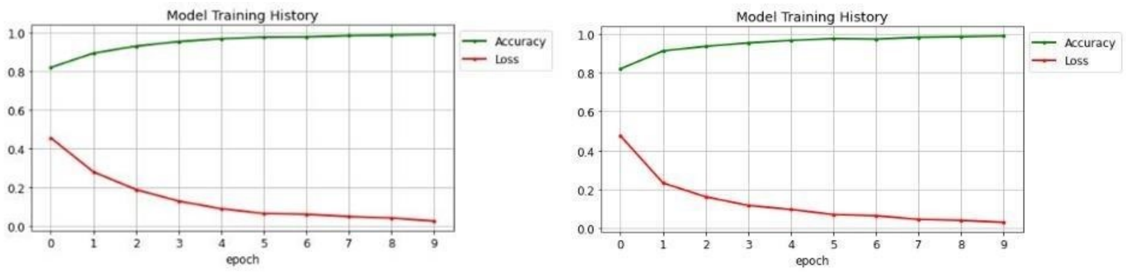
(a) Dataset 2

VGG 16				
	Precision	Recall	F1 score	Support
cat	0.92	0.98	0.95	259
dog	0.96	0.84	0.90	159
flower	0.98	0.99	0.99	183
fruit	1.00	1.00	1.00	187

(b) Dataset 3

Table 4: Evaluation Matrix using VGG 16

Table 4 displays the Precision, Recall, F1 score, and Support results for the VGG 16 using Dataset 2 and Dataset 3. The table demonstrates that Dtaset 2 has the highest F1 score, recall, or accuracy than Dataset 3. For Dataset 2 shown in Table 4a the model performs flawlessly for the assigned task, as seen by its 100% accuracy, 100% recall, and 100% F1 score for no tumor. This would suggest that there were no false positives or false negative predictions made by the model, meaning that it properly categorised every incident in the dataset.



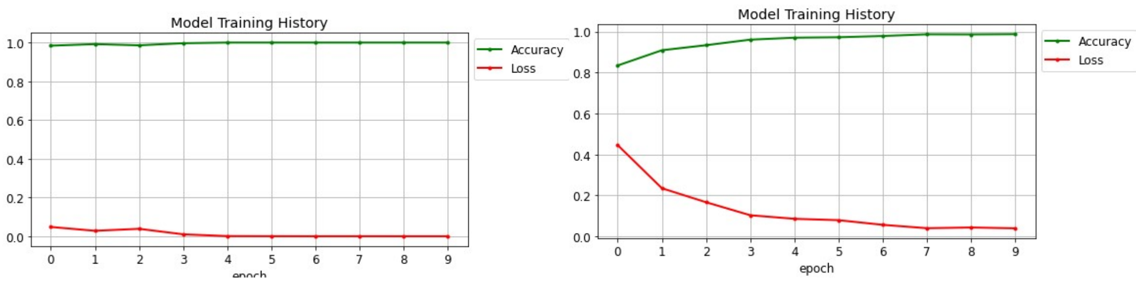
(a) CNN

(b) VGG 16

Fig. 6: Model Training History of Dataset 2

Figure 6 is a training graph of accuracy and loss for CNN and VGG 16 that shows the performance of the model throughout training for Dataset 2. The accuracy graph of CNN in figure 6a shows how the accuracy of the model changes over time during training. Initially, the accuracy may be low, but as the model is trained on more data, it gradually improves to 99.99%. The loss graph shows how the loss of the model changes over time during training. The loss should generally decrease over time, as the model learns to make better predictions.

The accuracy graph for VGG 16 in figure 6b would show the percentage of images that were correctly classified by the model. Initially, the accuracy would be low, but as the model is trained on more data, the accuracy would increase by 99.99%. The loss graph for VGG 16 shows the amount of error between the predicted and actual labels for the training data. The loss should generally decrease over time as the model learns to make better predictions.



(a) CNN (b) VGG 16

Fig. 7: Model Training History of Dataset 3

Figure 7 is a training graph of accuracy and loss for CNN and VGG 16 that shows the performance of the model throughout training for Dataset 3. The accuracy graph of CNN in figure 7a shows how the accuracy of the model changes over time during training. Initially, the accuracy may be low, but as the model is trained on more data, it gradually improves to 99%. The loss graph shows how the loss of the model changes over time during training. The loss should generally decrease over time, as the model learns to make better predictions.
The accuracy graph for VGG 16 in figure 7b would show the percentage of images that were correctly classified by the model. Initially, the accuracy would be low, but as the model is trained on more data, the accuracy would increase by 99.99%. The loss graph for VGG 16 shows the amount of error between the predicted and actual labels for the training data. The loss should generally decrease over time as the model learns to make better predictions.

6 CONCLUSION

This study presents a comprehensive approach for image segmentation and classification using pre-processing techniques, K-means algorithm for segmentation, and deep learning models (CNN and VGG16) for classification. Higher accuracy both for MRI and natural images is achieved by proposed approach. The VGG 16 model outperforms than the other models in terms of accuracy. It is shown that pre-processing techniques, segmentation, and deep learning models can be combined to achieve high accuracy in image classification tasks. The proposed method has the potential to be used in a variety of medical conditions, as MRI imaging is commonly used in medical diagnosis and treatment planning. In future research we can explore larger datasets, alternative segmentation techniques, transfer learning, ensemble methods, and alternative metrics. These techniques can be used to improve the robustness, accuracy, and applicability of the system. Further research can explore the deployment of these systems in clinical settings and real-time segmentation on low-power devices. As the field continues to evolve, there will be exciting opportunities for more advanced techniques for analyzing and understanding visual data.

References

1. Kaggle, `https://www.kaggle.com/`, [Online; accessed 22-March-2023]

2. “Javatpoint.” https://www.javatpoint.com/performance-metrics-in-machine-learning [Online; accessed 25 December-2023].
3. Roy, Prasun,Natural Images, year = 2021, `https://www.kaggle.com/prasunroy/natural-images`, [Accessed: March 23, 2023]
4. N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, “Brain tumor classification using convolutional neural network,” in World Congress on Medical Physics and Biomedical Engineering 2018: June 3-8, 2018, Prague, Czech Republic (Vol. 1), pp. 183–189, Springer, 2019
5. E. Miranda, M. Aryuni, and E. Irwansyah, “A survey of medical image classification tech- niques,” in 2016 International Conference on Information Management and Technology (ICIMTech), pp. 56-61, 2016
6. B. Padmini, C. Johnson, B. A. Kumar, and G. R. Yadav, “Brain tumor detection by using cnn and vgg-16,”.
7. 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), pp. 554-557, 2014.
8. F. Sultana, A. Sufian, and P. Dutta, “Advancements in image classification using convolu- tional neural network,” CoRR, vol. abs/1905.03288, 2019.
9. H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in mri image using convolutional neural network,” Mathematical Biosciences and Engineering, vol. 17, no. 5, pp. 6203-6216, 2020.
10. N. Chattyopadhyay, H. Maity, B. Debnath, D. Ghosh, R. Chakraborty, S. Mitra, R. Islam, and N. Saha, “Classification of mnist image dataset using improved convolutional neural network”.
11. P.Kumar and U.Dugal, “Tensorflow based image classification using advanced convo- lutional neural network,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, pp. 994-998, 2020.
12. J. Zuluaga-Gomez, Z. Al Masry, K. Benaggoune, S. Meraghni, and N. Zerhouni, “A cnn-based methodology for breast cancer diagnosis using thermal images,” Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, vol. 9, no. 2, pp. 131-145, 2021.
13. M. Tripathi, “Analysis of convolutional neural network based image classification tech- niques,” Journal of Innovative Image Processing (JIIP), vol. 3, no. 02, pp. 100-117, 2021.
14. M. A. B. Siddique, S. Sakib, M. M. R. Khan, A. K. Tanzeem, M. Chowdhury, and N. Yas- min, “Deep convolutional neural networks model-based brain tumor detection in brain mri images,” in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 909-914, IEEE, 2020.
15. R. Agarwal, “State classification of cooking images using vgg19 network”.
16. D. Febrianto, I. Soesanti, and H. Nugroho, “Convolutional neural network for brain tu- mor detection,” in IOP Conference Series: Materials Science and Engineering, vol. 771, p. 012-031, IOP Publishing, 2020.
17. S. U. Hoque, “Performance comparison between vgg16 & vgg19 deep learning method with cnn for brain tumor detection”.
18. M. K. Abd-Ellah, A. I. Awad, A. A. Khalaf, and H. F. Hamed, “Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convo- lutional neural networks,” EURASIP Journal on Image and Video Processing, vol. 2018, no. 1, pp. 1-10, 2018.
19. C. R. Pravallika and R. Baskar, “Image processing based brain tumor classification using vgg16 compared with svm to improve accuracy,” pp. 1398-1401, 2022.
20. M. E. Agus, S. Y. Bagas, M. Yuda, N. A. Hanung, and Z. Ibrahim, “Convolutional neural network featuring vgg-16 model for glioma classification,” JOIV: International Journal on Informatics Visualization, vol. 6, no. 3, pp. 660-666, 2022.
21. S. Bhuvaji, A. Kadam, P. Bhumkar, S. Dedge, and S. Kanchan, “Brain tumor classification (mri),” 2020.
22. M. Nickparvar, “Brain tumor mri dataset,” 2021.
23. S. B. Kulkarni and S. G. Bhirud, “Edge detection techniques for image segmentation: a survey of soft computing approaches,” International Journal of Emerging Technology and Advanced Engineering, vol. 4, no. 8, pp. 610-614, 2014.
24. Z. Mustafa, M. A. Khan, and M. Riaz, “An overview of image edge detection techniques,” Journal of Information Processing Systems, vol. 13, no. 1, pp. 1-20, 2017.
25. S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3523-3542, 2022.
26. . Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic seg- mentation,” in Proceedings of the IEEE conference on computer vision and pattern recog- nition, pp. 3431-3440, 2015.
27. S. A. Burney and H. Tariq, “K-means cluster analysis for image segmentation,” Interna- tional Journal of Computer Applications, vol. 96, no. 4, 2014.

28. L . Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” CoRR, vol. abs/1712.04621, 2017.
29. N. Dhanachandra, K. Manglem, and Y. J. Chanu, “Image segmentation using k-means clustering algorithm and subtractive clustering algorithm,” Procedia Computer Science, vol. 54, pp. 764–771, 2015.
30. X. Wang, “Laplacian operator-based edge detectors,” IEEE transactions on pattern analysis and machine intelligence, vol. 29, pp. 886–90, 06 2007.
31. Sekehravani, E. A., Babulak, E., and Masoodi, M. “Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics”, 9(4), 1404-1410,2020.
32. R. Kaushik and S. Kumar, “Image segmentation using convolutional neural network,” Int. J. Sci. Technol. Res, vol. 8, no. 11, pp. 667–675, 2019.
33. M. E. Agus, S. Y. Bagas, M. Yuda, N. A. Hanung, and Z. Ibrahim, “Convolutional neural network featuring vgg-16 model for glioma classification,” JOIV: International Journal on Informatics Visualization, vol. 6, no. 3, pp. 660–666, 2022.
34. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

Authors

Fatema Tuj Zohra is a recent B.Sc. graduate in Computer Science and Engineering. With a passion for advancing the field of machine learning, Fatema combines a strong foundation in computer science with a profound curiosity for exploring innovative solutions. She seeks to revolutionize real-world applications through their insightful contributions and unwavering dedication to technological progress. .

Rifa Tasfia Ratri received B.Sc. Engineering degree in Computer Science and Engineering (CSE) from Notre Dame University Bangladesh. Her current research focuses on machine learning and image processing.

Dr. Shaheena Sultana is a Professor and Chairman in the department of Computer Science and Engineering at Notre Dame University Bangladesh. She received M.Sc. Engineering degree in Computer Science and Engineering (CSE) from Bangladesh University of Engineering and Technology (BUET). She did B.Sc. Engineering degree in Electrical and Electronic Engineering (EEE) from Khulna University of Engineering and Technology (KUET) She obtained Ph.D. in CSE from BUET. Her research interests include Graph Drawing, Graph Theory, VLSI Design, Embedded System, Data Mining.

Humayara Binte Rashid is currently teaching as a Lecturer in the Department of Computer Science and Engineering of Notre Dame University Bangladesh (NDUB). She passed B.Sc. in Computer Science and Engineering (CSE) from Military Institute of Science and Technology (MIST). Her current research focuses on machine learning and Data Mining.