An Efficient Deep Learning Approach to detect Brain Tumor Using MRI Images

Submitted By

Annur Tasnim Islam
18101021
Sakib Mashrafi Apu
21141078
Sudipta Sarker
18101631
Inzamam M. Hasan
21241084
Syeed Alam Shuvo
21141080

Group ID: T2110066

Supervised By:

Dr. Md. Ashraful Alam
Assistant Professor

Co-Supervised By:

Ahnaf Hassan Rodoshee
Lecturer

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University

October 03, 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Annur Tasnim Islam
18101021

Sakib Mashrafi Apu
21141078

Inzamam M. Hasan
21241084

Sudipta Sarker
18101631

Syeed Alam Shuvo
21141080

Approval

The thesis/project titled "An Efficient Deep Learning Approach to Detect Brain Tumor Using MRI Images" submitted by

1. Annur Tasnim Islam (18101021)
2. Sakib Mashrafi Apu (21141078)
3. Syeed Alam Shuvo (21141080)
4. Inzamam M. Hasan (21241084)
5. Sudipta Sarker (18101631)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 16, 2022.

Examining Committee:

Supervisor:
(Member)

Md. Ashraful Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Ahanaf Hassan Rodoshi
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

A brain tumor is the development of mutated cells in the human brain. Many different types of brain tumors exist nowadays. According to researchers and physicians, some brain tumors are non-cancerous while some are life-threatening. In most cases, the cancer is detected at the last stage and it is difficult to recover. This increases the mortality rate. If this could be detected in the initial stages, then a lot more lives could be saved. Nowadays, brain tumors are being detected through automated processes using artificial intelligence algorithms and brain image data. In this research, we propose an efficient approach to detect brain tumors using Magnetic Resonance Imaging (MRI) data and a deep neural network. The proposed system comprises several steps - preprocessing and classification of brain MRI images. Furthermore, we analyzed the performances of different deep neural network architectures and optimized them with an efficient one. The proposed model enables classifying brain tumors effectively with higher accuracy. To commence, we collected data and classified it using the ResNetl0l, ResNet50, InceptionV3, VGG19, and VGG19 architectures. As a consequence of our analysis, we obtained an accuracy rate of 96.72% for VGG16, 96.17% for ResNet50, and 95.55% for InceptionV3. Then, using these three top classifiers, we constructed an ensemble model and obtained an overall accuracy rate of 98.60% using EBTDM (Explainable Brain Tumor Detection Model).

Keywords: Brain Tumor, MRI, Deep Neural Network, VGG, ResNet, EfficientNet, Inception, Ensemble, EBTDM.

Acknowledgement

First and foremost, we give thanks to the Almighty Allah for allowing us to finish our thesis without any serious setbacks.
Secondly, we would like to express our gratitude to our supervisor, Dr. Md. Ashraful Alam sir and our mentor, Shakib Mahmud Dipto sir, for their invaluable assistance and advice. They came to our aid whenever we needed.
Finally, without our parents' continuous encouragement, we may not be able to achieve our goals. We are currently on the verge of graduation. Thus, thanks to their gracious support and prayers.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The brain is among the most delicate components of the human body. It maintains the whole central nervous system which is also responsible for carrying out all activities throughout the human body [1]. A brain tumor is one of the life-threatening disorders that may occur to a human being. It has been stated by The National Brain Tumor Foundation (NBTF) that the number of persons dying from brain tumors has gone as high as 300% in the previous 30 years [2]. Patients afflicted with serious brain tumors on average live for a maximum of 2 to 3 years. However, even in the early stage, it is a little difficult for a physician to recognize the disease and its severity manually. Basically, brain tumor identification starts by evaluating MRI pictures. In this study, we are attempting to build an efficient deep learning model to segment and categorize brain tumor images. Moreover, this will help us achieve greater accuracy than that of the current approaches by analyzing patients' brain MRI images. In addition to that, our technique with custom produced models also assure great accuracy.

1.2 Aims and Objective

In our research, we offer an efficient strategy for detecting brain tumors in their early stages utilizing deep neural network methods. The purpose of this paper is to develop a deep learning model that will predict the outcome of brain tumor detection using the provided MRI dataset. Our main objective is to determine the degree of brain tumor infection in humans in order to develop a reliable predictive model for tumor intensity prediction. Additionally, it is necessary to understand the various qualities in order to get more accurate results.

1.3 Research Methodology

We saw a range of machine learning models at first, including VGG16, VGG19, Inception V3, ResNet50, and ResNetl0l. We proceeded by pre-processing the MRI image data into a well-defined 224×224 format. Following that, we trained and evaluated deep learning models. Then, the machine learning models were trained and evaluated. Following assessment, we chose the best three classifiers and used them to create an ensemble. Finally, after evaluating the efficiency of the ensembles, we visualized the categorization by using our model.

1.4 Research Orientation

In Chapter 2, we demonstrated how past efforts were introduced by other researchers relatecl to our field of study. Then, in Chapter 3, we discussed in details of each approach, convolutioal layers, and activation function that we used during our research. Moreover, in chapter 4 we discussed the application of our thesis work. Additionally, we demonstrated the dispersion of our datasets and then discussed the pre-processing processes employed on the datasets. In Chapter 5, we demonstrated the findings obtained after implementing the algorithms and then discussed the analysis of the data. Finally, in Chapter 6, we discussed the conclusion and future directions for our research.

Chapter 2

Related Work

Afreen and B. Raghu studied brain tumors utilizing a deep learning network hybrid approach and transfer learning approaches[3]. For research purposes, 3640 T1-based MRI brain images were gathered from 233 patients. These MRI images showed many forms of tumors, such as glioma, meningioma, and pituitary. In addition, the authors applied the trained Inception-V3 image classification model. For validation, four types of approaches such as 10-fold cross-validation, holdout-validation, group 10-fold cross-validation, 10-fold stratified cross-validation were utilized. A large success rate was therefore found by the 10-fold cross-validation, which demonstrated 99.82% accuracy. The data set followed by this study considered MRI images to be 775 by 715 pixels. However, MRI images were used in two different ways, with pictures cropped and pictures uncropped. The authors also used DCNN and TL along with Inception V3. The CNN was constructed using tensor flow and Keras in this study. Along with that, the Inception V3 model includes a collection of pre-trained imagenet-311-layer data. In this work, the model supported by the researchers showed 99.8% accuracy in the MRI dataset classification, and the implementation speed was about 15 sec/per epoch.

In this research [4], the authors proposed a multi-level features extraction for the early diagnosis of brain tumors. Moreover, this study proposes two pre-trained deep learning models one is Inception-v3 and the other one is DensNet201, and with the assistance of these two models two unique scenarios of brain tumor recognition and classification were assessed. To begin with, the components from various inception modules were separated from the pre-trained inception model and those features were passed to the Softmax classifier for the identification of the tumor. After that,pre-prepared DensNet was utilized to extract highlights from different DensNet blocks. Then those features were passed to the Softmax classifier for the identification of brain tumors. For this proposed model 80% of the data was used for training and 20% of the data was used for testing. The proposed strategy delivered 99.34 percent and 99.51 percent accuracy respectively with inception-v3 and DensNet201 and accomplished best in the identification of brain tumors.

The authors of the study paper [5] described how they used a convolutional neural network (CNN) approach in conjunction with data augmentation and image processing to classify MRI brain images as cancerous or non-harmful. They compared the accuracy of their scratched CNN model to that of pre-trained VGG-16, ResNet-

50, and Inception-v3 models after successfully computing the accuracy of their 8 convolutional layers CNN model. The authors validated, developed, and tested the model using 253 MRI brain scans of genuine affected patients from Kaggle's library. 155 of the 253 MRI pictures in the sample were of cancerous tumors, whereas 98 were of non-cancerous tumors. Following that, the dataset was divided into three segments for the purpose of training, testing, and validating the suggested model. To calculate the model's accuracy, 185 photos were utilized for training, 20 images for testing, and 48 images for validation. Finally, the model demonstrated 96 percent accuracy on training data and 89 percent accuracy on validation data.

The authors offers a new Deep Neural Network (DNN) architecture called Low Layered U-Net (LU-Net) for tumor identification in this paper [6]. The Author proposes a new Deep Neural Network (DNN) architecture with fewer layers and a simpler architecture called Low Layered U-Net (LU-Net). Additionally, they compared VGG16, Le-net, and Lu-net. Additionally, they created 253 images from the collection of normal and atypical brain tumors. Additionally, this article states that LeNet's design is quite simple, with a small number of layers, and that the outcome is greatly dependent on the optimizer type and learning rate, but VGG16 is likewise quite simple, with fewer parameters and 16 layers. On the other hand, the LU-Net CNN Deep Neural Model is straightforward, extremely quick, and capable of more precisely directing the tumor; this model is a fusion of Le-Net and U-Net with several modifications. The advantages of Le-Net and U-Net, as well as the LU-Net, a new CNN architecture with low complexity and a relatively small number of layers. Finally, demonstrated in the study that the Lu-Net Model completely substitutes and executes other CNN models with an overall accuracy of 98%.

In this paper [7] the Authors stated that the MRI brain tumor image segmentation attempts to divide the region of the tumor into a healthy brain to give a distinct tumor border. This research divides ROI and non-ROI into a completely coevolutionary network with unique architecture, namely UNet-VGG16.Moreover, this research attempts to partition the MRI brain tumor in order to better see a 1.5 Tesla machine MRI picture. Also, they want to detect the tumor area clearly so they use the Fully Convolutional Network (FCN) since it has great performance for semantic segmentation and it used U-Net architecture. But the U-Net has some problems it takes more time for its implementation and needs a powerful pc for this so they hybridized U-Net and VGG16. It is also stated that the U-Net architecture is built using different scenarios to generate alternative models that will be evaluated for accuracy with the proposed model.VGG16 is similar to U-Net. In this paper, their objective is to minimize the computation process and to speed up the model's training time and UNet-VGG16 has the least loss but the most precision. Lastly,CCR value of 95.69 percent was used to determine the segmentation results from testing data.

In his study "Investigation The Effect Of Using Gray Level And RGB Channels On Brain Tumor Image," Ahmed B Salem Salamh [8] studied the effect of utilizing RGB channels on brain tumor imaging. Additionally, he investigated the effect of gray levels on brain tumor imaging. Similarly, he asserted that this could aid in speeding up the detection process. The gray level and any of the RGB (Red, Green, Blue) color channels were included with the primary purpose of reducing the operation's total time complexity. Similarly, this is because extracting information from photographs of complex systems, such as the human brain, can take an extended period of time. To put it another way, proper therapy cannot be ensured unless an appropriate decision is reached. Cerebrum tumors, on the other hand, have the following structure, which is somewhat complex to follow. As a result, a plan based exclusively on MRI will fall short of capturing images of the tumor and all of its subregions. Not only may this method simplify the procedure, but it also minimizes the amount of information required. As a result, less data must be analyzed and fewer calculations must be performed to obtain a conclusion, lowering the overall time required.

Onur Sevli's research [9] study exhibits the performance comparison of various pre-trained DL models for the classification of MRI images of the brain. The author concentrated on the performance of certain models such as Vgg-16, ResNet50, and Inception V3. In this paper, the author has provided a categorization methodology for MRI brain pictures, which has been entirely automated by transfer learning techniques in deep convolutional models. The author has nevertheless worked with a tiny data set consisting of 253 brain MRI pictures. The author has therefore followed pre-processing and data increase criteria for improved efficiency. In addition, the transfer learning method was utilized with a little amount of data to lessen the processing burden and obtain successful results. In this research, the author used confusion matrices for evaluating more performances. A confusion matrix provides a true and false table classification. In this paper, the performance models used by the author were adequate. Vgg-16 demonstrated 94.42% accuracy, ResNet50 showed 82.49% and Inception V3 showed 63.13% lower success. The precision of Vgg-16 and ResNet50 was 100%. In terms of recall, Vgg-16 and ResNet50 were the same. However, the Vgg-16 model in all respects was superior.

In the research paper [10], the authors mainly worked on Giloma, meningioma, and pituitary, these three types of tumors. The author's proposed a model that offers improvement in feature extraction by using deep learning and machine learning to detect brain tumors. Deep learning is utilized for feature extraction and incorporates various models such as Inception-v3 and Xception. This improvement might be adequate to help a massive role in clinical applications for brain tumor detection. Firstly, they extracted the elements from MRI images by utilizing Inception-v3, and these features were sent to deep classifiers like softmax. In the subsequent approach, the Xception model is applied for the feature separation and Classification of brain tumors from the MRI images. Besides, the separated elements are given to the AI classifiers like SVM, RF, KNN. An MRI dataset was utilized for the training and testing of the proposed model, which contains 3064 MRI images from 233 distinct patients. The test result of Inception-v3 achieved 94.34 percent accuracy; on the other hand, the Xception model showed a test accuracy of 93.79 percent.

Belaid, Ouiza Nait, and Malik Loudini developed a brain tumor identification model based on deep learning techniques and on the mixtures of pre-trained VGG-16 CNNs to brain tumors in their research paper[11]. They offered a pre-arranged VGG16 model with trainable blocks. They also separated the GLCM framework for the source images in order to shape GLCM feature images as input images. Once again, the authors developed a programmed technique that relied on a combination of Convolutional Neural Networks (CNN) and many data sources. As a result, they came up with the idea of avoiding overfitting by utilizing a small dataset MRI. Additionally, they connected the outputs of both the GLCM feature image input CNN and the original image input for CNN in order to arrange the images using the advantages of CNN and GLCM highlights for increased accuracy. The results of their exploratory analysis on their dataset indicate that combining GLCM energy images with an original image as a contribution to two CNNs is effective for contrast testing. In the final study, the authors demonstrated that the original image with an energy image as input has a higher recognition rate of highlights than other input mixes, and that their model has an average accuracy of 96.5%.

In this paper [12] the authors used four deep learning models for utilized the classification of brain cancers in this comparative research. They used AlexNet, VGG16, GoogleNet, and RestNet50 in this research and collect different types of results by using these models. These models are used to categorize brain tumors as normal or abnormal. They wanted to bring out the best results from those models. They compared those models for finding out which models give the best accuracy rate. Moreover, they also compared the time span that means which models take the shortage time for processing. Also, there are two folders in their dataset yes and no with 3000 Brain MRI pictures. By comparing those models, they found that the ResNet model has the greatest performance with a 95.8% accuracy. on the other hand, the AlexNet model is just 82.7% accurate. Therefore, they found in the second parameter that the AlextNet model is faster than the Resnet model. It is also stated that the longer time takes the best accuracy results can be found. Finally, they also used a GPU to speed up model performance and the increase in performance is between 63 and 144 times.

Palash Ghosal, Lokesh Nandanwar, Swati Kanchan, Ashok Bhadra, Jayasree C. and Debashis Nandi proposed a deep CNN-based SE-ResNet-101 architecture for automatically classifying MR images of brain tumors into three classes: meningiomas, pituitary, and gliomas tumors[13]. Simultaneously, their trials demonstrate that the proposed technique surpasses the other two competing brain tumor classification systems in terms of overall accuracy, sensitivity, and specificity. The image slice samples are loaded into a Squeeze and Excitation ResNet model based on Convolutional Neural Networks, which is used to automatically classify brain tumors from MRI data (CNN). Despite this, the use of zero centering and intensity normalization as a preprocessing step for smooth changes in intensity across tissues was also investigated when combined with data augmentation. Additionally, their experiments demonstrate that the suggested CNN achieves an overall accuracy rate of 89.93 percent without the use of augmented data. With the addition of data augmentation, the inclusive accuracy of 93.83 percent was increased to 98.67 percent, 91.81 percent, and 91.03 percent for Glioma, Meningioma, and Pituitary tumor, respectively[14]. Finally, they concluded that the proposed technique could be beneficial to physicians when it comes to brain tumor classification.

The research paper by Talo, M., Yildirim, O., Baloglu, U. B., Aydin, G., and Acharya, U.R. concentrated on several forms of CNN (Convolutional Neural Networks) brain tumor identification utilizing MRI brain images [15]. In this work, their priority was to diagnose brain tumors automatically by analyzing MRI pictures rather than manually. In addition, five pre-trained Deep Learning models have been used to automatically classify images of brain MRIs such as ResNet-18, Vgg-16, AlexNet, ResNet-34, and ResNet-50. The researchers used the data set of a widely recognized educational institution in this article. They used all images except some were utilized, and all brain images are 256 to 256 pixels in size and are axially weighted with T2. In addition, five classes for classification have been followed. The classes are regular and include four principal types of disease: brain tumor, cerebrovascular, degenerative, and inflammatory disorders. The authors also used transfer learning strategies because it decreases the requirement for a large number of data. In addition, confusion matrix approaches were used in this study to validate the data set using pre-train models. However, the researchers claimed that the best classification accuracy occurred with ResNet50 after analyzing and implementing different approaches, 95.26 %, where other models revealed the lowest accuracy. Therefore, after studying this research, we can say that deep learning is one of the recent diagnostic techniques for diagnosis purposes.

Chapter 3

Research Methodology

3.1 Working Process

First and foremost, we have gathered and pre-processed our data. After preprocessing we have splitted our datasets into a 7:2:1 ratio, Where 70% of our datasets have been used to train our mentioned deep learning models, 20% of our datasets have been for training purposes and rest of the datasets, 10% have been used for validating our implemented models.

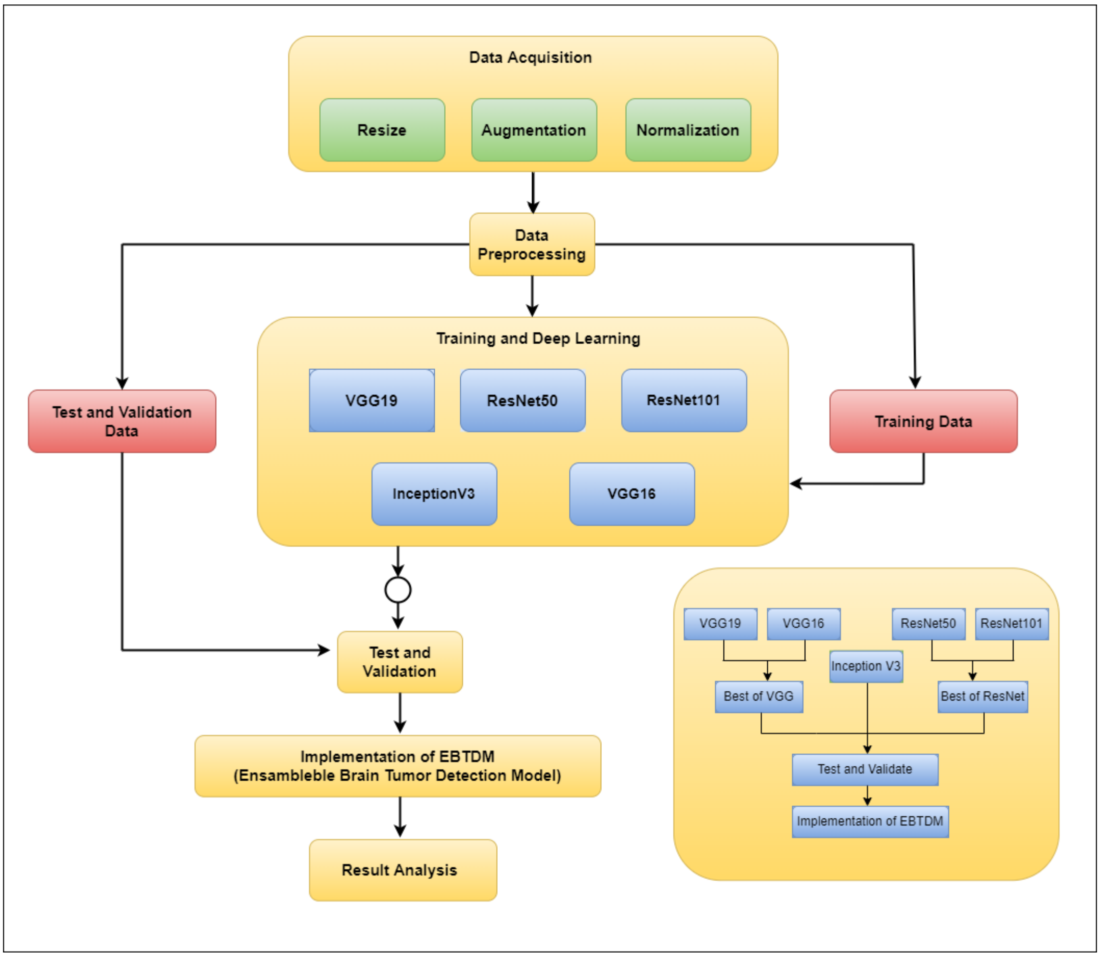

Figure 3.1: Illustration of our Working Process

In our research, multiple deep learning models have been used and implemented such as ResNet101, ResNet50, Inception V3, VGG19 and VGG16. After that, comparison of the validation accuracies have been done. From the comparison, we found out the best three performing architectures and ensembled them for a more efficient outcome. Figure 3.1 illustrates our research workflow.

3.2 Used Architectures

In our research, we have availed five architectures, which are VGG-19, VGG-16, ResNet-50, ResNet-101, and Inception V3. The details of these architectures are given below:

3.2.1 VGG-19

VGG-19 is a 19-layer convolutional neural network. One may import a pre-trained version of the network that has been trained on over a million pictures from the ImageNet database. It utilized just 3x3 filters with stride and pad of 1, as well as 2×2 max-pooling layers with a stride of 2. In order to decrease the number of parameters in such deep networks, it applies minimal 3×3 filters in all convolutional layers, which is best employed with its 7.3 percent error rate. Moreover, a total of 138M parameters were included in the VGG-19 model, which placed it in the second position in classification and first in localization. This model was developed by using a portion of ImageNet.

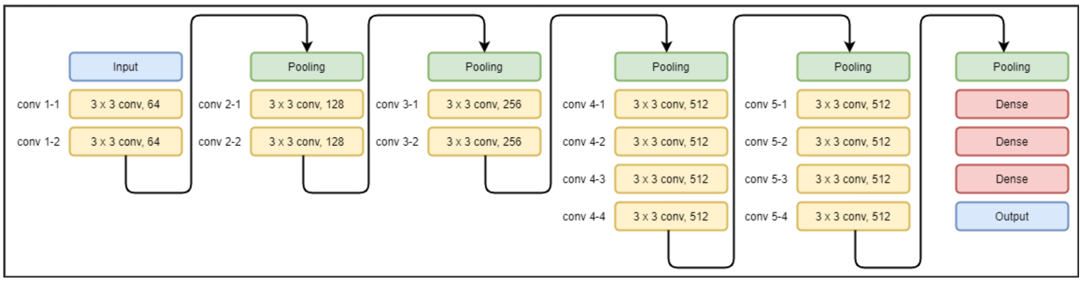

Figure 3.2: Internal Architecture of VGG19

3.2.2 VGG-16

VGG16 is a Convolutional Neural Network (CNN) architecture that is frequently utilized in ImageNet, a large visual database project that is used to produce visual object recognition software. It is widely recognized as one of the most advanced vision model architectures yet devised. Instead of having a large number of hyper-parameters, VGG16 concentrated on having 3x3 filter convolution layers with a stride 1 and always used the same padding and maxpool layer of 2x2 filter stride 2. In the end, it has two fully linked layers, followed by a softmax for output. The number 16 in VGG16 refers to the fact that it has 16 layers with different weights. With around 138 million parameters, this network is fairly large. It is now the most widely used method for extracting features from photographs taken in the field. The model achieves 92.7 percent top-5 test accuracy in ImageNet, a dataset of over 14 million images belonging to 1000 classes.

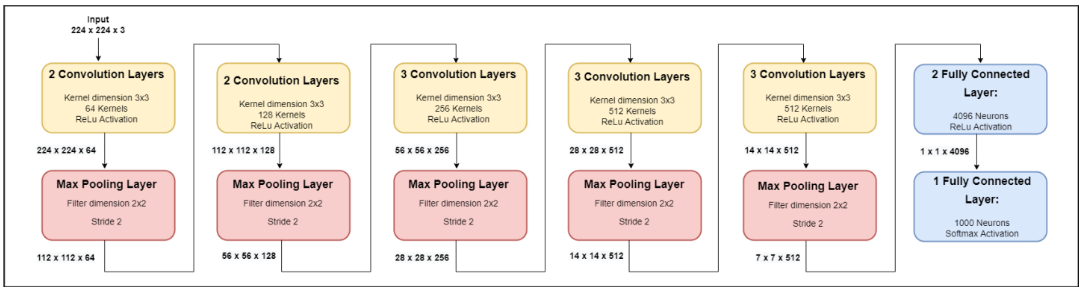

Figure 3.3: Internal Architecture of VGG16

3.2.3 ResNet50

ResNet-50 [16] enables the efficient training of extremely deep neural networks by avoiding the vanishing gradient problem.

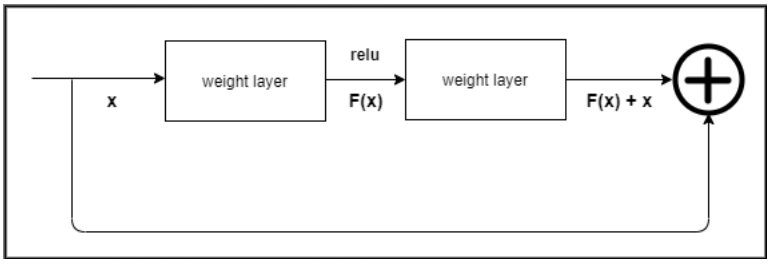

Figure 3.4: Residual Learning

The skip connection concept has been first introduced in the model shown in the above figure of Residual Learning).

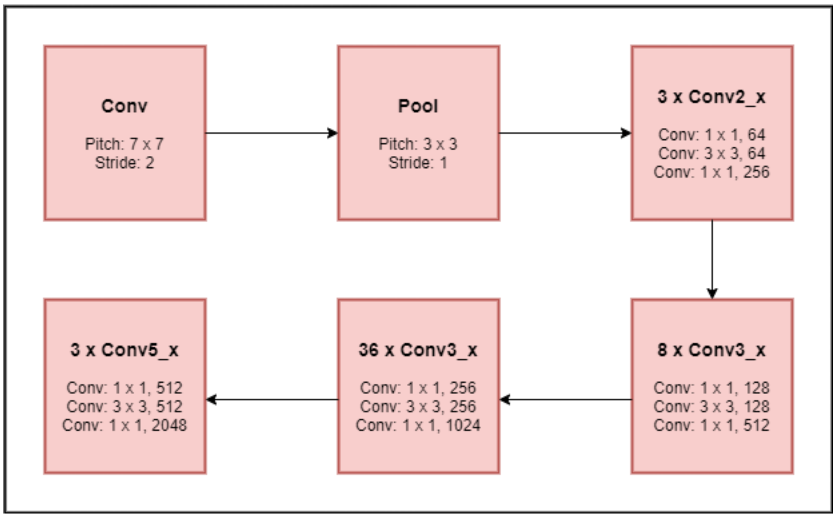

Figure 3.5: Internal Architecture of ResNet50

With the immediate layer, transmission of the output layer may occur to a distant layer in this paradigm. This implies that all layers operate at the same power level. This model is composed of five phases, each of which contains a different mix of convolutional layers. The architecture is demonstrated below.

3.2.4 ResNet101

ResNet101 is a CNN model that carries 101 layers. Here the networks that are built on one million photos obtained from the image net database and may then be imported readily into ResNet101. Then this pretrained network can sort out the images into 1000 item categories.[16]

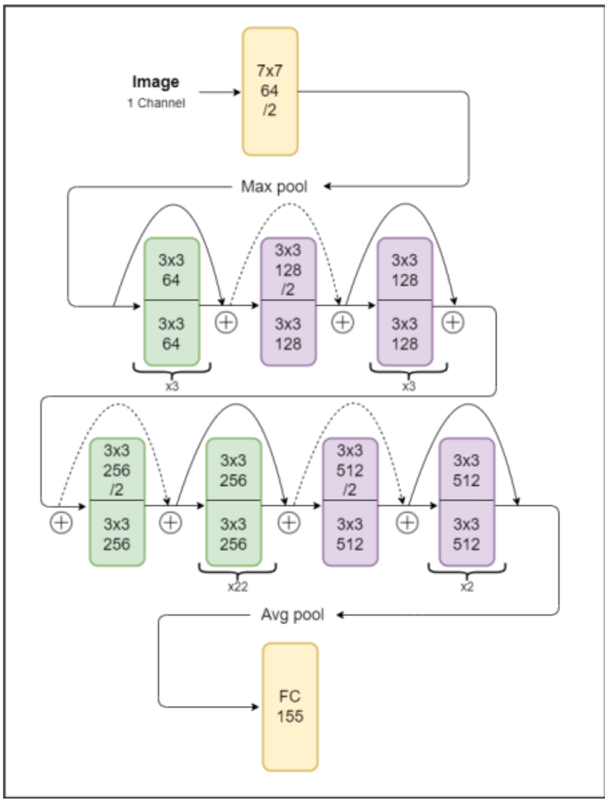

Figure 3.6: Internal Architecture of ResNet101

3.2.5 Inception V3

Inception-V3 mostly centers around consuming less computational force by altering the past Inception structures. This thought was proposed in the paper [17], published in 2015. Inception-V3 is an improved version of the well-known GoogLeNet network, which has shown mesmerizing results in different biological applications making use of transfer learning. Similar to GoogLeNet, an inception model was proposed by Inception-V3. This model concatenates many different sized convolutional filters into a new filter. As a result of this architecture, the number of parameters that must be taught is decreased, as is the computational complexity. It uses Label Smoothing, factorized 7 x 7 convolutions, and has an auxiliary classifier, all of which are advancements over the other members of the Inception family.

3.3 Convolutional Layer

The basic component of a convolutional neural network is the convolutional layer. This network is particularly well-suited for working with two-dimensional data.

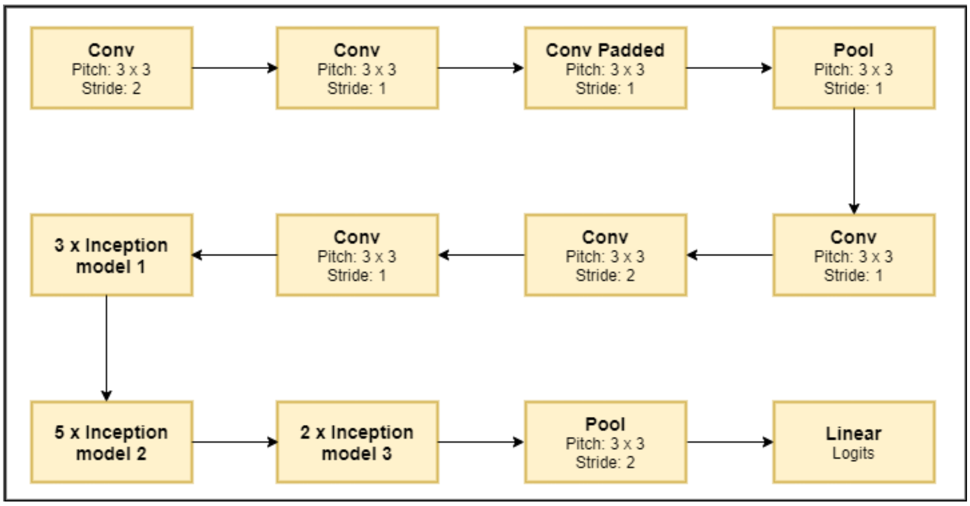

Figure 3.7: Internal Architecture of Inception V3

It is made of convolutional filters that convert two-dimensional images to three-dimensional ones and works remarkably good. Model learning occurs rapidly. CNNs are multilayer perceptrons inspired by biology. They have a predisposition to recognize visual patterns in raw image pixels. These deep networks investigate discrete sections of the input image. [18] [19]

3.3.1 Activation Function

The essential units of a neural network's structure are its gross inputs, which are processed and transformed using an activation function into a final output called unit activation. A network layer's output values can range from negative infinity to positive infinity. [20]. The output values of a network layer can range from negative infinity to positive infinity.

Rectified Linear Unit(ReLU)

In neural networks, ReLU is a widely used non-linear activation function[21]. ReLU is more efficient than other Functions because not all neurons are activated simultaneously. Additionally, during the back-propagation stage of neural network training, the weights and biases are not modified because the gradient value is 0 in some circumstances.

$$f(x) = max(0, x)$$

Softmax

In order to obtain a Softmax function, we assembled multiple sigmoid curves. It provides outputs which varies from 0 to 1. Thus, it can be used as probabilities of the data points of a certain class.[21][22] The Softmax function have been used to solve issues with multiple classes. For a particular class, the probability returns a function.

$$f(x) = \frac{e^{x_i}}{\sum_{n=1}^{k} e^{x_i}}$$

The value of k is directly proportional to the probability of the instance.
x_i = output from the ith neuron.
i $\in R+$.

3.3.2 Max Pooling Layer

It is a filter that, using the available samples, picks the brightest pixels in an image. The overall objective is to reduce the multiplicity of an input configuration (image, hidden-layer, output matrix, etc.) while allowing for assumptions about the characteristics present inside the rejected subregions. Additionally, it reduces computational work by reducing the number of elements to learn and ensures that the underlying depiction is linear. It's quite useful for darkening the image's background and brightening the pixels. It works by applying a maximum filter to generally non-overlapping subregions of the initial representation.

3.4 Optimizer: Adam

It is an optimizer that takes into account the fitness function which is targeted for machine learning applications. These applications usually comes in big datasets and hard to deal with parameters.[23] Here, we combine the advantages of two different methods (AdaGrad's dealing ability with sparse gradients, and RMSProp's dealing ability with targets that are non-stationary) which became recently popular. Since this process demands less resources, getting the final results become simpler.

3.5 Transfer Learning

Transfer learning occurs when a pre-trained machine learning model is utilized to address a similar issue. People frequently use transfer learning to improve their performance in a related activity, which decreases the amount of training data required. The transfer learning method is not only simple, but it is also incredibly effective. In the area of natural language processing, transfer learning techniques have been employed for speech recognition, document classification, and prediction. Deep learning models uncover a variety of representations, some of which may be applied to a variety of tasks, making them appropriate for transfer learning. The learning is boosted in the training sample by using information from the target task in the source task. On the other hand, transfer learning is an effective way to reduce training time to half. This strategy might be linked to the development of deep learning models for picture classification.[24][7]

3.6 Ensemble Modeling

Ensemble modeling [25][26] is a multiple layered technique for generating predictions from a collection of numerous diverse models. We have kept both the inputs and the outputs in the same format in order to modify the averaging layer. This has been done in order to minimize the errors and get rid of deceptive predictions. One thing that must be taken care of is that the individual models must have a strong architecture themselves in order to be ensambled.

Averaging Layer

We have used a combination of VGG16 and ResNet50 in our suggested "EBTDM" model. As well as the Inception V3 architectures. Here, the layer will collect the probability of the output. From a variety of commonly used structures with identical dimensions. Additionally, for both impacted brain tumors, an average of distinct probabilities will be computed. MRI scans of the normal brain and MRI images of the abnormal brain.

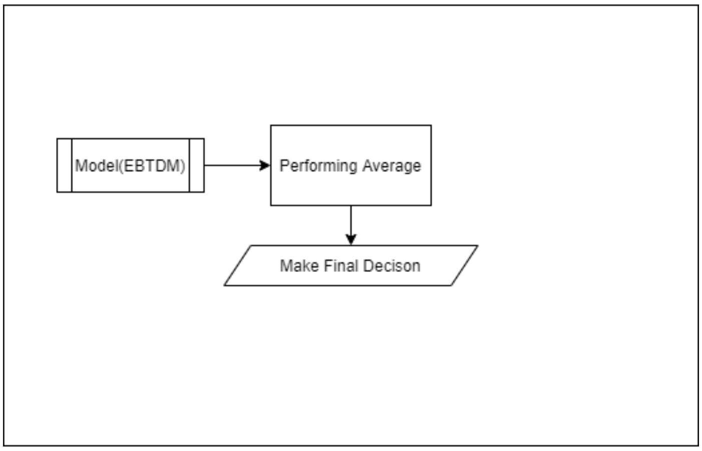

Figure 3.8: Basic structure of an ensemble model

Output Layer

Our suggested "EBTDM" model predicts whether the patient is BRAIN tumor afflicted or unaffected based on the MRI input images to the model of that patient based on the averaging layer's output probability.

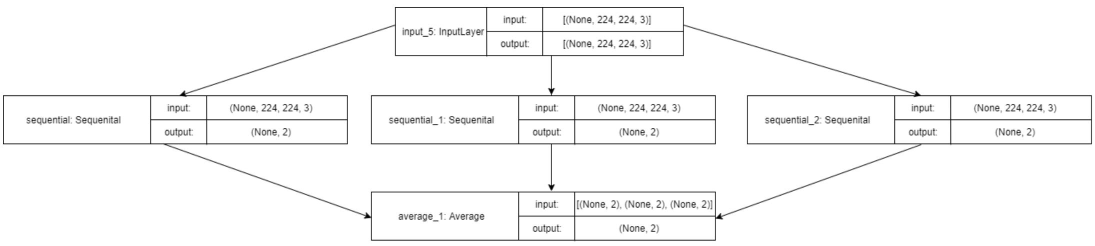

Figure 3.9: Output layers of the ensemble model hierarchy for EBTDM

3.7 Confusion Matrix

As long as the dataset includes an equal number of samples from each class, classification accuracy may be used to evaluate performance. A more comprehensive set of performance measures should be used when dealing with an uneven dataset. Confusion matrices are used in this situation. A confusion matrix is a table that lists correct and incorrect classifications [9]. In a confusion matrix, the following four outcomes are represented by values: TP stands for true positive, while FP stands for false positive. TN stands for true negative, whereas FP stands for false positive (FN). A number of metrics are generated based on the confusion matrix's values to represent the classifier's performance.

Classification accuracy is measured by the number of properly categorized samples to the total amount of data (Eq. 3.1).

$$accuracy = \frac{number\, of\, correctly\, classified\, data\, samples}{total\, amount\, of\, data} \tag{3.1}$$

Recall shows that a classification system is capable of identifying real positive results. Eq. 3.2 calculates the recall (responsiveness) based on the findings.

$$precision = \frac{TP}{TP + FN} \tag{3.2}$$

Precision is a measure of a classification's ability to detect only true positives. Eq. 3.3 shows how to compute precision.

$$recall = \frac{TP}{TP + FP} \tag{3.3}$$

It is the harmonic average of these two measures that are utilized to calculate the F1 score. Eq. 3.4 computes the F1 score.

$$F1\, score = \frac{2 * precision * recall}{precision + recall} \tag{3.4}$$

Chapter 4

Implementation

4.1 Dataset

We worked with two datasets: Br35H and Brain Tumor V3. Here, we've divided the images into test, train, and validation ratios that are optimal. The percentages are as follows: 20% for test, 70% for train, and 10% for validation. There are 6762 images in total. Among them 3579 images are normal where 3183 images are affected.

4.1.1 Source

Brain Tumor MRI image data collection

Source 1 : Jakesh, B.(2020, July). Brain Tumor, Version 3. Retrieved July 26, 2020, from https://www.kaggle.com/jakeshbohaju/brain-tumor/version/3.

Source 2: Hamada, B.(2021, November).Br35H :: Brain Tumor Detection 2020, Version 12. Retrieved November 14, 2020, from https://www.kaggle.com/ahmedhamada0 /brain-tumor-detection

4.1.2 Data Sample

In the given figure, the first three images represent tumor-affected human brain MRI images. The later three images represent normal human brain MRI images. This can be inferred that human brain images with the small white faded regions are the tumors and vice-versa.

4.1.3 Data Classification

As shown in Figure 4.1, we were able to come up with train, test, and validation data with 7:2:1 split ratios respectively.

Training Set

The step in which examples tagged with machine learning algorithm results or output labels are fed into the system.

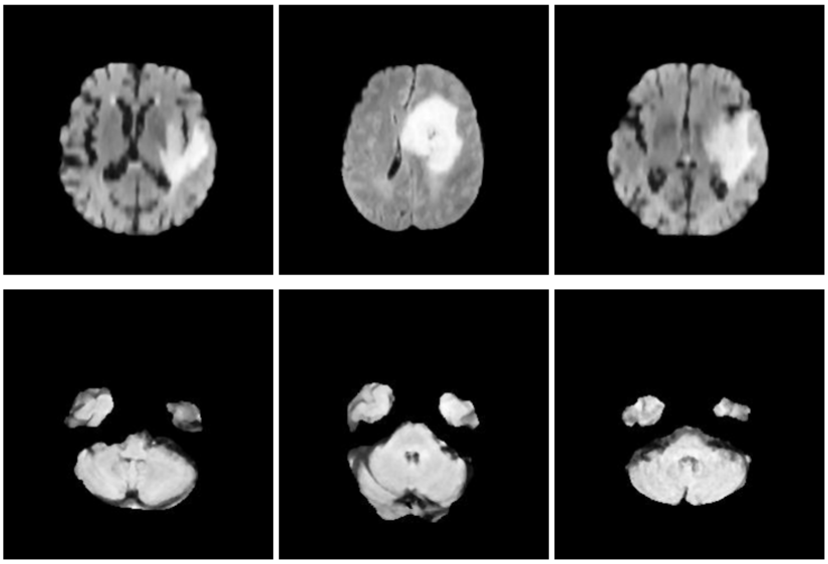

Figure 4.1: Sample Data of the Dataset

	Training Set	Validation Set	Testing Set	Total
Normal	2505	358	680	3579
Affected	2228	318	600	3183
Total	4733	676	1353	6689

Table 4.1: Classification of our Dataset

Testing Set

The algorithm may learn certain features of the training set as it converges to improve performance by using a succession of real-world samples. Good outcomes will increase confidence in the algorithm in the actual world for an unknown test collection.

Validation Set

While the model hyperparameters were being adjusted, the data sample is used to provide an impartial assessment of model fitting for the training set. For the validation dataset, when the competence is considered in the specified model, the assessment becomes more prejudiced.

4.1.4 Data Labels

Over here, we can classify our dataset into two different labels. They are human brain images affected by a brain tumor and human brain images unaffected by a brain tumor. As a result, our dataset can be represented using a binary classifier. In our dataset, the "Class" attribute defines two types of data in binary representation. In class attribute, 0 represents the number of unaffected brains and 1 represents the number of affected brains.

4.2 Data Visualisation

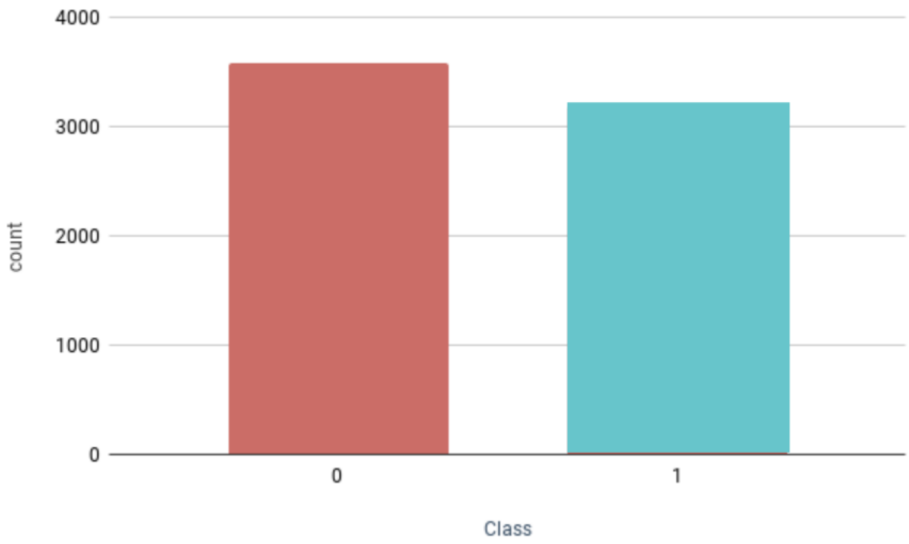

Figure 4.2: Illustration of balance in between the labels using bar-chart

From figure 4.2 we get illustration of balance in between the labels. There are two types of representation in the bar-chart. Here, 0 has higher priority than 1. That means, the number of unaffected images in our dataset are more than the number of affected images.

4.3 Data Pre-processing

4.3.1 Image Resizing

In our work, we will be using the pre-trained VGG19, VGG16, Resnet101, ResNet50 and InceptionV3 models. Hence, each MRI image of the human brain needs to be resized to a fixed size of 224 x 224 during the network training process. We used TensorFlow, Scikit Graphic, and Caffe frameworks for this purpose. [25]

4.3.2 Normalization and Scaling Images

Normalization is the process used for reducing data redundancy and for removing information of less important images. Here, the PCA(Principal Component Analysis) technique has been used for normalization. Using PCA a large data variable is converted into a small data variable, retaining most of the information [27].
Here, Eigen flat fields are generated and merged to normalise the Brain MRI images projection. Then the systematic errors of projection intensity normalization are reduced by dynamic flat fields [28] [29]. This task has been completed using the Keras ImageDataGenerator class.
By transforming re-scaled input to a ratio that could be multiplied by each pixel, normalization methods limit data to a scale of 0-1 scale. Our dataset consists of affected and unaffected brain MRI images in the .jpg format.

4.3.3 Data augmentation

In our system, we used data augmentation techniques to improve our network output by the sudden transformation of the image orientation. As the augmentation operator's translation, flip and rotation of 90, 180, 270 degrees have been used for the initial images both vertically and horizontally. [25].

Chapter 5

Result and Analysis

5.1 Result Analysis

The confusion matrix, as well as performance information such as Validation accuracy, recall, precision, and F1 score, were produced for each model as part of our study's results and analysis. These were used as performance metrics.

5.2 Individual Architecture

The performance of this research is explained using training curve and validation curve. Along with it, Confusion Matrix is used to analyze the performance of each model with some performance measures like validation accuracy,recall, precision and F1 score.

5.2.1 Performance analysis with the learning curves

VGG19

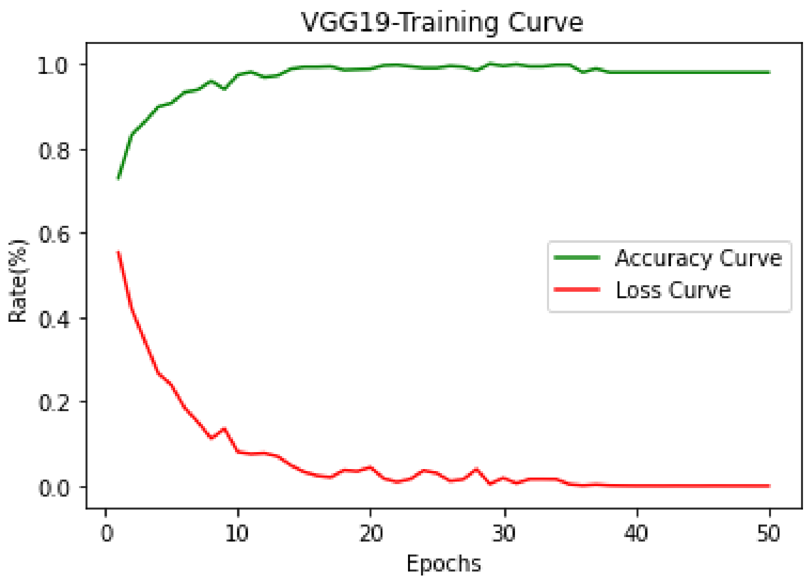

Figure 5.1: Training curve(s) of VGG19 architecture

In the illustrated VGG19 training curve above we can see that from epoch 0 to epoch 10 there is a sharp increase in the rate of change for the accuracy curve. For the loss curve the rate of change sharply decreased from around 0.6 to 0.0 from epoch 0 to epoch 18. Afterwards, the change in rate did not vary much.

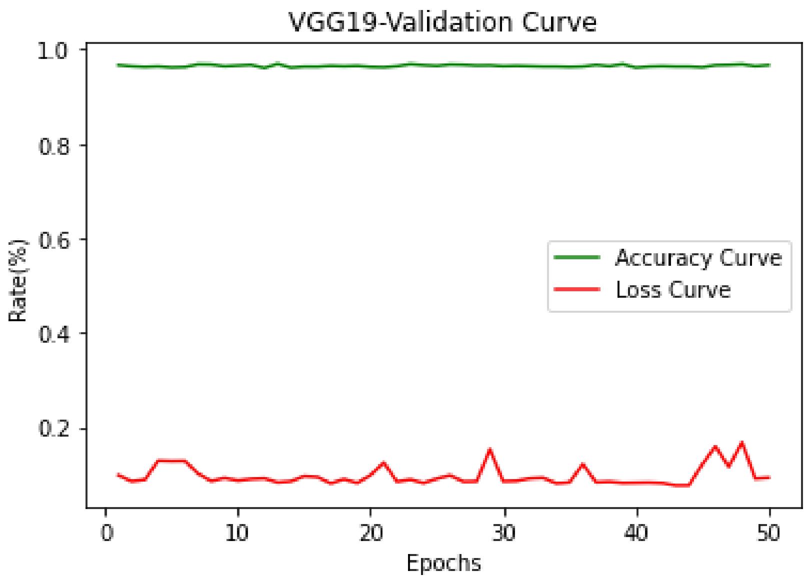

Figure 5.2: Validation curve(s) of VGG19 architecture

From the attached VGG19 validation curve we can see that the rate of change for the accuracy curve did not change much and was around 1.0. On the other hand, fluctuations of the rate of change was quite often but under or near 0.2.

VGG16

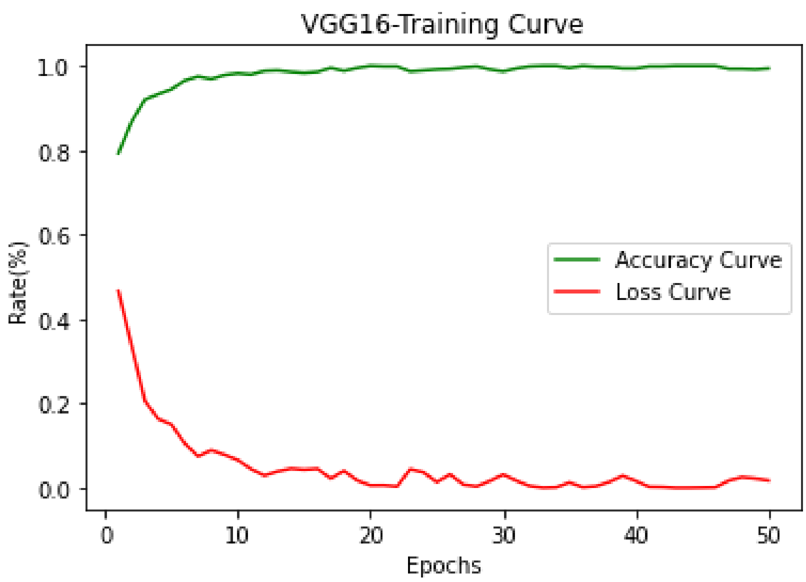

Figure 5.3: Training curve(s) of VGG16 architecture

In the VGG16 training curve there was a quick rise in the rate from epoch 0 to 8. The rate rose from 0.8 to 1.1 and then the increasing rate remained constant. On the other hand, for the loss curve the decreased from 0.5 to nearly 0 from epoch 0 to 11. After that the loss decreased at an approximate steady rate until epoch 50.

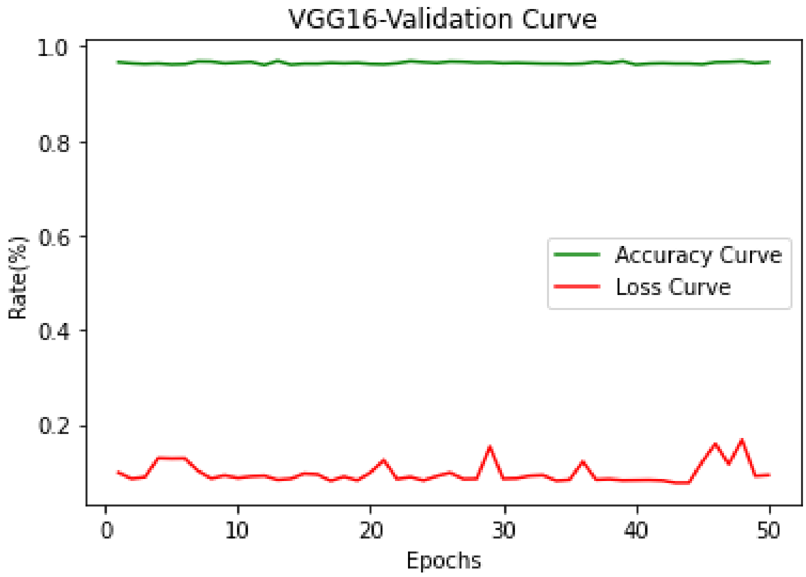

Figure 5.4: Validation curve(s) of VGG16 architecture

From the above VGG16 validation curve we can see that the accuracy curve kept increasing at a steady rate of around 1.0 throught. However, for the loss curve the rate of change slightly fluctuated from epoch 0 to epoch 50. The rate of change always remained 0.2.

ResNet50

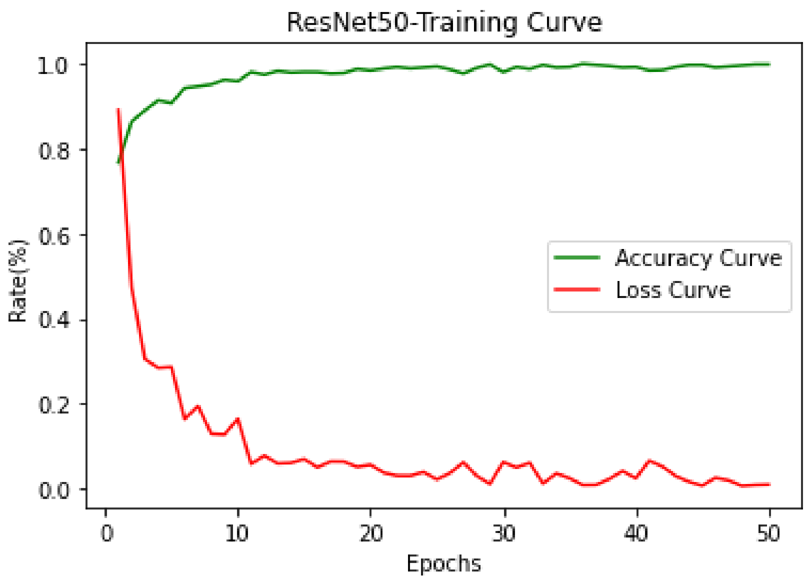

Figure 5.5: Training curve(s) of ResNet50 architecture

The training curve for ResNet50 shows that the accuracy curve is increasing at an increasing rate. From epoch 0 to 10 the increase was relatively noticeable when compared to its nature afterwards. Unlike the accuracy curve the loss curve showed a steep fall in the rate of change until 0.3. The rate gradually reached close to 0.0 until epoch 10. After that there was slight fall in the rate of loss curve.

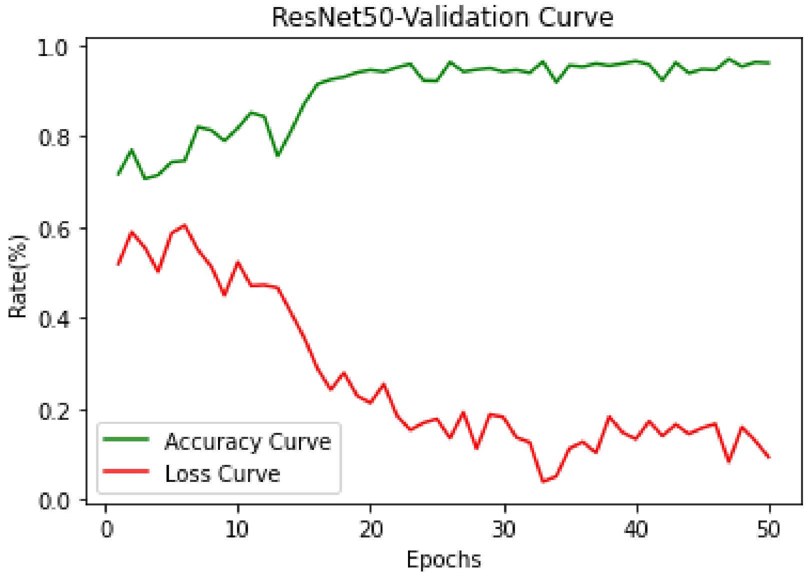

Figure 5.6: Validation curve(s) of ResNet50 architecture

Accuracy curve for the ResNet50 validation curve continued to rise from around 0.7 to close to 1.0 until epoch 22. Then the rate of change was not that observable when compared to the scenario beforehand. Similarly, for the loss curve there were noticeable changes from epoch 0 to 32. Afterwards there were small spikes in the curve but not that much noticeable compared to the previous scenario.

ResNet101

During the ResNet101 model training period, the training curves were generated which contained both accuracy and loss curves. From figure 5.7, it is visible that the accuracy and the loss curve were overlapping at first. However, their performance increased afterwaeds. For the accuracy curve, the rate kept rising till around epoch 18 and then continued to stay around 1.0. On the other hand, the rate of change of the loss curve decreased drastically until epoch 9. After that, it remained below 0.2 throughout.

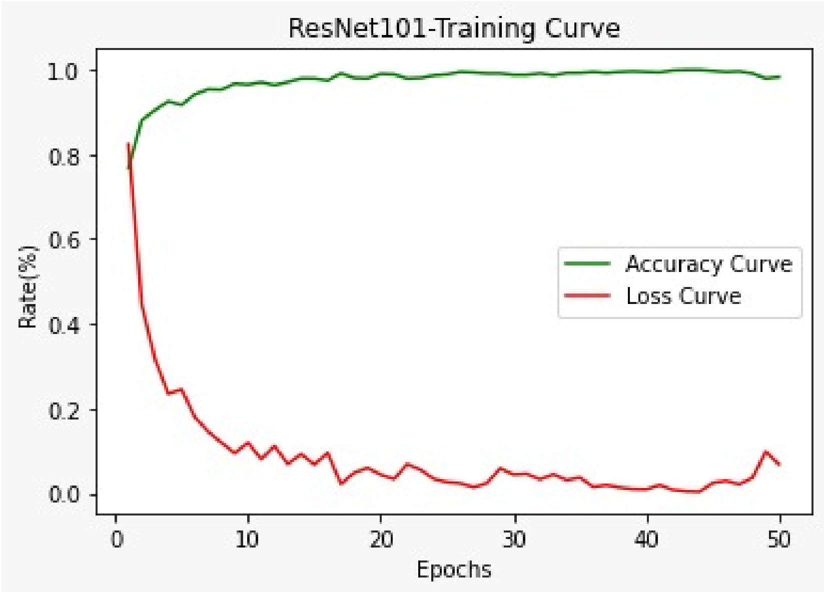

Figure 5.7: Training curve(s) of ResNet101 architecture

Here, it has been found that the loss and accuracy results were not satisfactory, from epoch 0-5, as the accuracy and loss curves were over lapping. However, after epoch 5, the rate of change of the accuracy curve started to increase gradually until it reached 1.0 in around epoch 20. Afterwards, the rate of change became steady. For the loss curve, the rate of exchange decreased till epoch 22 and then did nbot show much changes.

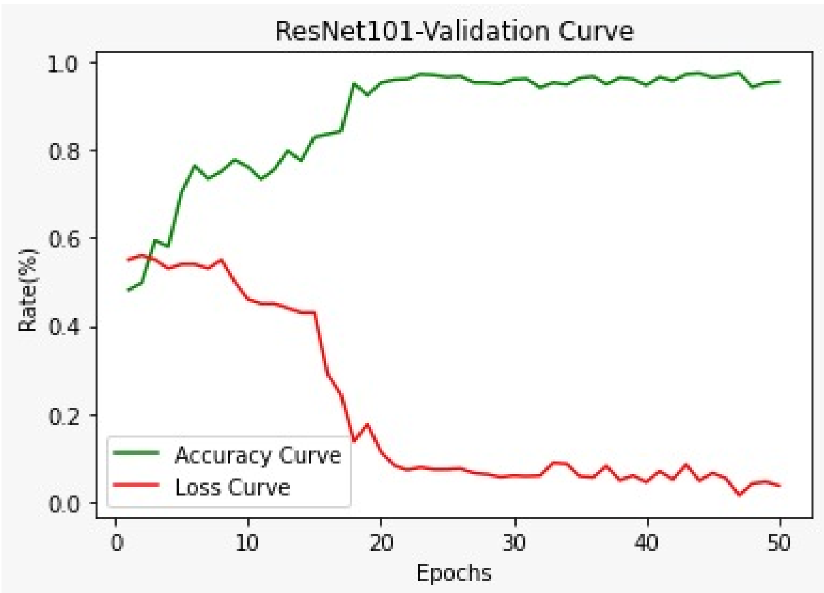

Figure 5.8: Validation curve(s) of ResNet101 architecture

InceptionV3

Throughout the InceptionV3 model's training phase, training curves for both accuracy and loss were produced. As shown in figure 5.9, accuracy increased progressively from 0.7 to 1.0 and stayed constant until epoch 20.

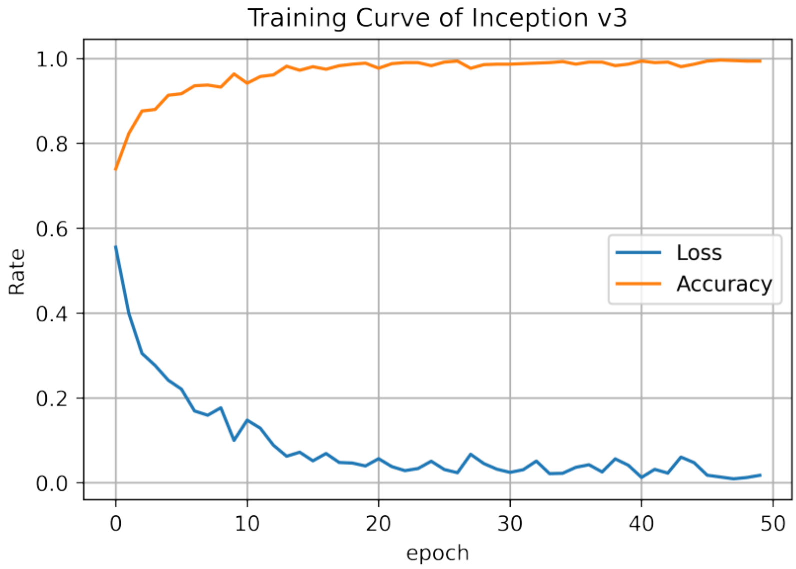

Figure 5.9: Training curve(s) of InceptionV3 architecture

Similarly, the loss curve dropped progressively from 0.6 to near 0.0 during epoch 30 and stayed stable.

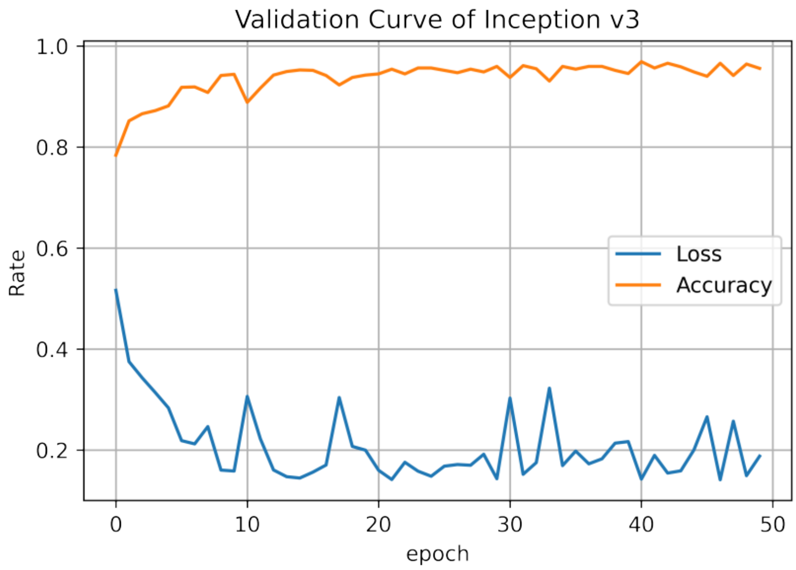

Figure 5.10: Validation curve(s) of InceptionV3 architecture

From figure 5.10 it is found that during validation the accuracy was rising from 0.8 and reached 1.0 from epoch 18 and then continued in this manner. In addition to that the loss showed frequency changes until epoch 10 after which the changes became frequent.

Ensemble Model

The training curve of the ensemble model shows that the accuracy graph did not show much changes from epoch 0 to epoch 50. Again for the loss curve, it remained flat throughout except for epoch 20.

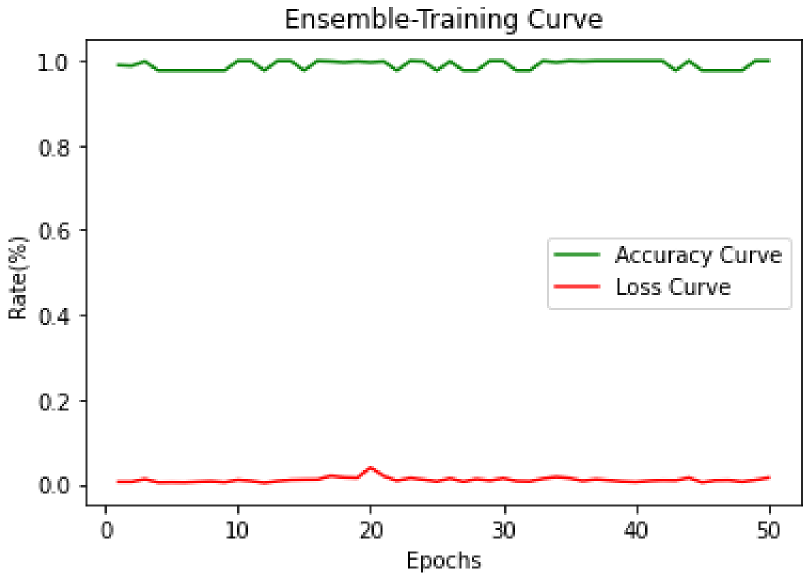

Figure 5.11: Training curve(s) of Ensemble Model architecture

The training curve of the ensemble model shows that the accuracy graph did not show much changes from epoch 0 to epoch 50. Again for the loss curve, it remained flat throughout except for epoch 20.

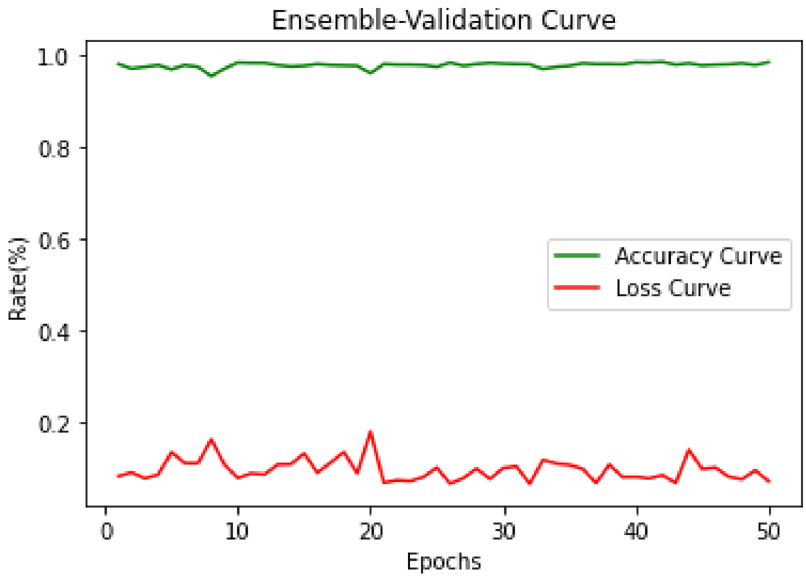

Figure 5.12: Validation curve(s) of Ensemble Model architecture

Now for the validation curve of the ensemble model, the accuracy rate remained around 1.0 from epoch 0 to epoch 50. On the other hand, for the loss curve, there were sudden rise and fall but the rate of rise did not go over 0.2 for any of the epochs shown above.

5.2.2 Confusion Matrix

The confusion matrix is an array that contains both the algorithm's accurate and incorrect predictions, as well as the actual state of the world. In our study, to describe the performance of VGG19, VGG16, ResNet101, ResNet50, and InceptionV3; we have made use of confusion matrix criteria.

VGG16

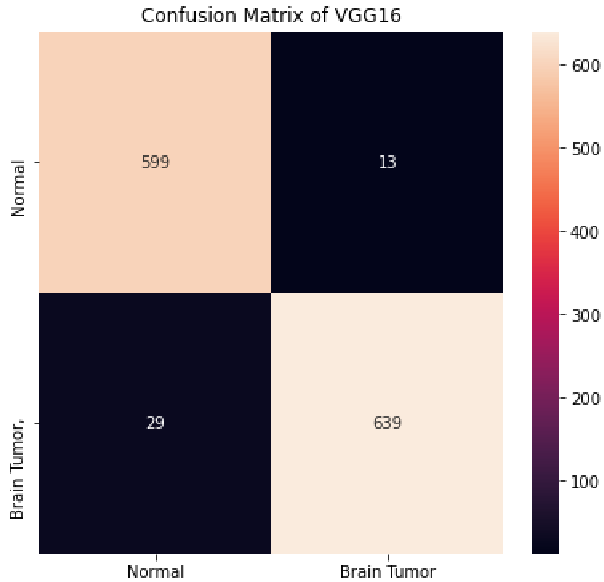

Figure 5.13: Confusion Matrix of VGG16

According to the VGG16 confusion matrix above, 639 images were classified by brain tumors. However, the algorithm incorrectly categorized 29 of the affected images as normal in the process. However, the system correctly categorized 599 images as normal, but the system incorrectly classified 13 images which is an error.

VGG19

The confusion matrix of the VGG19 illustrated in the given Figure shows that 155 images were classified by brain tumors. However, the algorithm incorrectly categorized 11 of the affected images as normal in the process. However, the system correctly categorized 200 images as normal, but the system incorrectly classified 2 images which is an error.

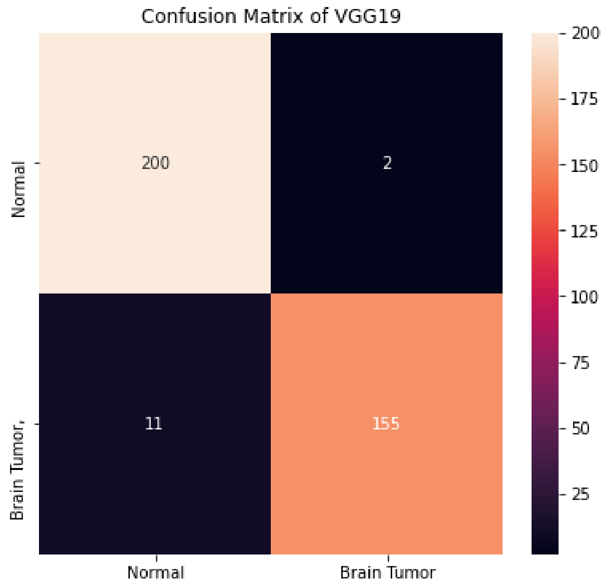

Figure 5.14: Confusion Matrix of VGG19

ResNet50

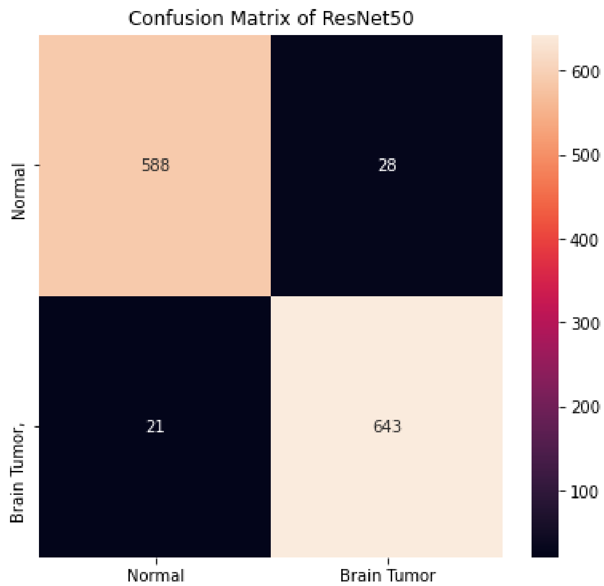

Figure 5.15: Confusion Matrix of ResNet50

According to the ResNet50 confusion matrix above, 643 images were classified by brain tumors. However, the algorithm incorrectly categorized 21 of the affected images as normal in the process. However, the system correctly categorized 588 images as normal, but the system incorrectly classified 28 images which is an error.

ResNet101

The confusion matrix of the ResNet101 illustrated in the given Figure shows that 514 images were classified by brain tumors. Hence, the algorithm incorrectly categorized 48 of the affected images as normal in the process. On the other hand, the system correctly categorized 709 images as normal, but the system incorrectly classified 9 images which is an error.

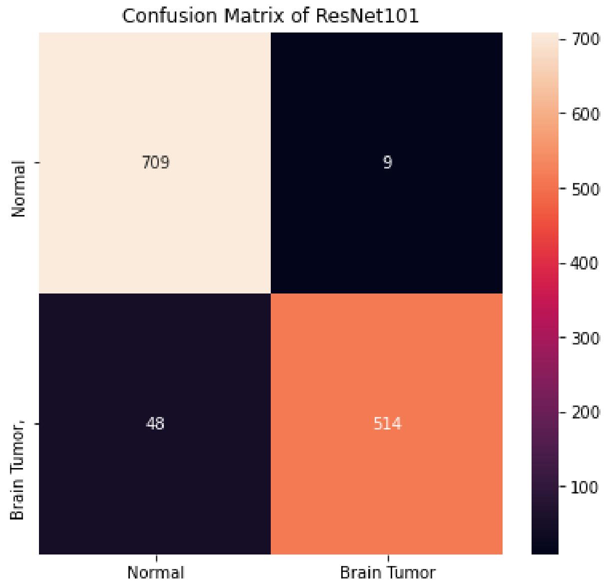

Figure 5.16: Confusion Matrix of ResNet101

InceptionV3

According to the Inception V3 confusion matrix above, 617 images were classified by brain tumors.

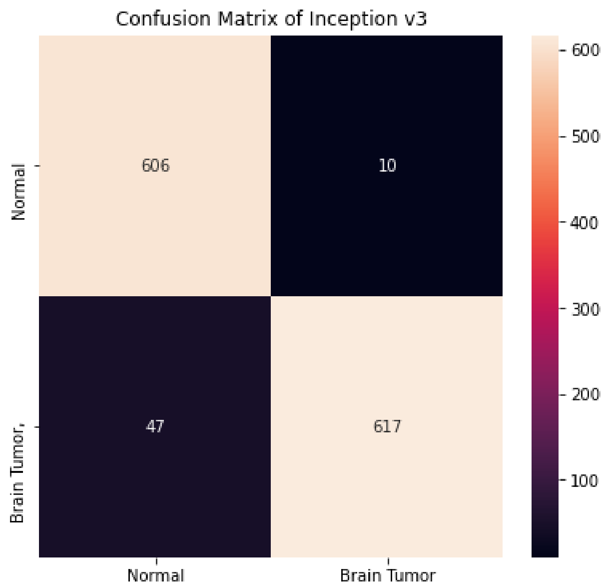

Figure 5.17: Confusion Matrix of InceptionV3

However, the algorithm incorrectly categorized 47 of the affected images as normal in the process. However, the system correctly categorized 606 images as normal, but the system incorrectly classified 10 images which is an error.

5.2.3 Result Comparison

On our splitted dataset, we applied VGG19, VGG16, ResNet50, ResNet101, and InceptionV3 architectures in the final stage of our research. We have acquired a 96.72% accuracy for VGG16, 96.47% for VGG19, 96.17% for ResNet50, 95.55% for ResNet101, and 95.55% for InceptionV3. After comparing ResNet50 with ResNet101 we observed that ResNet50 comes up with higher accuracy. Likewise, the accuracy of VGG19 and VGG16 was the same. Finally, we have decided to work with VGG16, ResNet50 and InceptionV3 to get better performance.

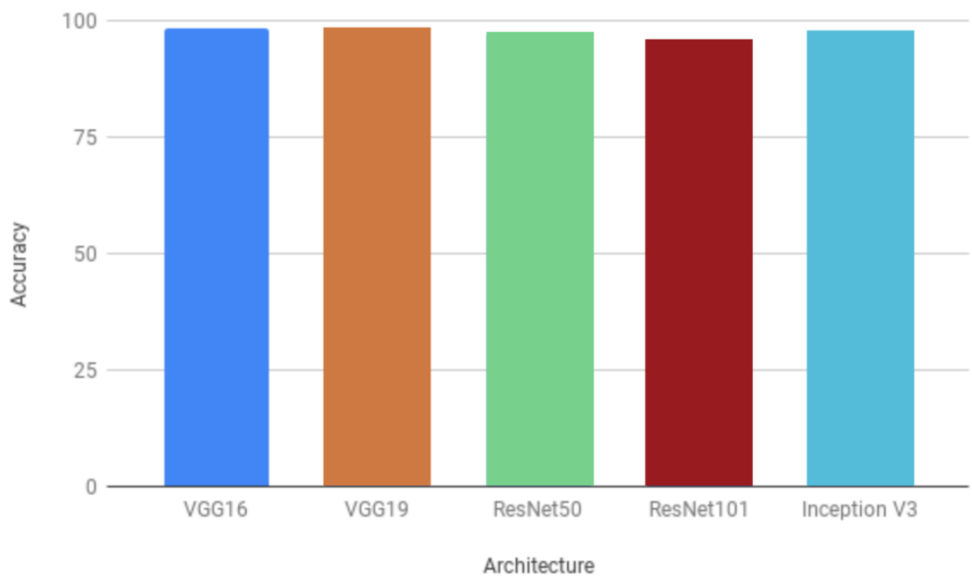

Figure 5.18: Result comparison between used architectures

After comparing ResNet50 with ResNet101 we observed that ResNet50 comes up with higher accuracy. Likewise, the accuracy of VGG19 and VGG16 was the same. Finally, we have decided to work with VGG16, ResNet50 and InceptionV3 to get better performance.

Architecture	Accuracy	Precision	Recall	F1 score
VGG16	96.72%	96.69%	96.77%	96.72%
VGG19	96.47%	96.76%	96.19%	96.47%
ResNet50	96.17%	96.19%	96.15%	96.17%
ResNet101	95.55%	95.97%	95.10%	95.55%
InceptionV3	95.55%	95.60%	95.65%	95.55%

Table 5.1: Comparison table of our implemented models

From the table , we can determine that VGG16, ResNet50 and InceptionV3 are the best performing architectures among the implemented models so far.

Ensemble Model

According to the Ensemble Model confusion matrix above, 646 images were classified by brain tumors.Whether, the algorithm erroneously categorized 15 of the affected

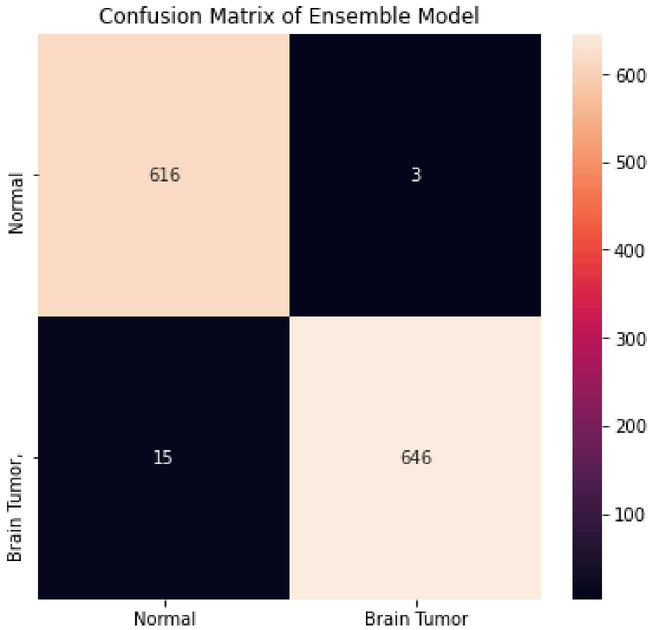

Figure 5.19: Confusion Matrix of Ensemble Model

images as normal in the process. Moreover, the system correctly categorized 616 images as normal, but the system incorrectly classified 3 images which is an error.

Chapter 6

Conclusion and Future Work

The number of people diagnosed with brain tumors, as well as the number of people who die from them, is growing day after day. Across the country, millions of people are seeking treatment at hospitals. Some patients in critical condition are not receiving adequate treatment in a reasonable timeframe due to the amount of time it takes to obtain results, which is endangering their lives. As a result, our model will be able to identify brain tumors by analyzing the brain MRI data and categorizing the tumor's intensity based on the imaging data, which will help to improve the situation considerably. We used the VGG16, VGG19, inceptionV3, Res et50, and ResNetlOl architectures to train the dataset, and then compared the results to implement the proposed architecture EBTDM (Ensemble of VGG16, ResNet50, and InceptionV3). In the future, we will attempt to collect additional data in order to improve our model. Finally, we will work on reducing time complexity of our model and developing a process or interface that will enable people out of the knowledge of medical science to detect brain tumor effectively through MRI images.

Bibliography

[1] A. Filatov, P. Sharma, F. Hindi, and P. S. Espinosa, "Neurological complications of coronavirus disease (covid-19): Encephalopathy," *Cureus*, vol. 12, no. 3, 2020.

[2] A. Jacobi, M. Chung, A. Bernheim, and C. Eber, "Portable chest x-ray in coronavirus disease-19 (covid-19): A pictorial review," *Clinical imaging*, vol. 64, pp. 35–42, 2020.

[3] A. A. Habiba *et al.*, "A novel hybrid approach of deep learning network along with transfer learning for brain tumor classification," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 1363–1373, 2021.

[4] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, "A deep learning model based on concatenation approach for the diagnosis of brain tumor," *IEEE Access*, vol. 8, pp. 55 135–55 144, 2020.

[5] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, "Brain tumor classification in mri image using convolutional neural network," *Math. Biosci. Eng*, vol. 17, p. 6203, 2020.

[6] X. Feng, N. J. Tustison, S. H. Patel, and C. H. Meyer, "Brain tumor segmentation using an ensemble of 3d u-nets and overall survival prediction using radiomic features," *Frontiers in computational neuroscience*, vol. 14, p. 25, 2020.

[7] A. A. Pravitasari, N. Iriawan, M. Almuhayar, T. Azmi, K. Fithriasari, S. W. Purnami, W. Ferriastuti, *et al.*, "Unet-vgg16 with transfer learning for mri-based brain tumor segmentation," *Telkomnika*, vol. 18, no. 3, pp. 1310–1318, 2020.

[8] A. B. S. Salamh *et al.*, "Investigation the effect of using gray level and rgb channels on brain tumor image," *Computer Science & Information Technology (CS & IT)*, pp. 141–148, 2017.

[9] O. Sevli, "Performance comparison of different pre-trained deep learning models in classifying brain mri images," *Acta Infologica*, vol. 5, no. 1, pp. 141–154, 2021.

[10] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, and M. O. Alassafi, "Brain tumor classification based on fine-tuned models and the ensemble method," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 67, no. 3, pp. 3967–3982, 2021.

[11] O. N. Belaid and M. Loudini, "Classification of brain tumor by combination of pre-trained vgg16 cnn," *Journal of Information Technology Management*, vol. 12, no. 2, pp. 13–25, 2020.

[12] A. A. Abbood, Q. M. Shallal, and M. A. Fadhel, "Automated brain tumor classification using various deep learning models: A comparative study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 1, pp. 252–259, 2021.

[13] M. Masood, T. Nazir, M. Nawaz, A. Mehmood, J. Rashid, H.-Y. Kwon, T. Mahmood, and A. Hussain, "A novel deep learning method for recognition and classification of brain tumors from mri images," *Diagnostics*, vol. 11, no. 5, p. 744, 2021.

[14] P. Ghosal, L. Nandanwar, S. Kanchan, A. Bhadra, J. Chakraborty, and D. Nandi, "Brain tumor classification using resnet-101 based squeeze and excitation deep neural network," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, IEEE, 2019, pp. 1–6.

[15] M. Talo, O. Yildirim, U. B. Baloglu, G. Aydin, and U. R. Acharya, "Convolutional neural networks for multi-class brain disease detection using mri images," *Computerized Medical Imaging and Graphics*, vol. 78, p. 101 673, 2019.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[18] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: A review," *Journal of medical systems*, vol. 42, no. 11, pp. 1–13, 2018.

[19] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," *Classification in BioApps*, pp. 323–350, 2018.

[20] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[21] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.

[22] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, "A high-speed and low-complexity architecture for softmax function in deep learning," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, IEEE, 2018, pp. 223–226.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[25] T. G. Dietterich *et al.*, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, no. 1, pp. 110–125, 2002.

[26] R. Polikar, "Ensemble learning," in *Ensemble machine learning*, Springer, 2012, pp. 1–34.

[27] Z. Jaadi, "A step-by-step explanation of principal component analysis (pca)," *Retrieved June*, vol. 7, p. 2021, 2021.

[28] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[29] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.