Detection of Brain Tumor using Several Convolutional Neural Network Architectures

by

Abrar Salehin
16301079
Md. Sizer Ahmad
16301044
Moinul Islam
16301180

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 2020

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:

Abrar Salehin

Abrar Salehin
16301079

Sizer Ahmad

Md. Sizer Ahmad
16301044

Moinul Islam

Moinul Islam
16301180

Approval

The thesis titled "Detection of Brain Tumor using Several Convolutional Neural Network Architectures" submitted by

1. Abrar Salehin (16301079)
2. Md. Sizer Ahmad (16301044)
3. Moinul Islam (16301180)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 5, 2020.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez
Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University, ou=CSE, email=zavid.parvez@bracu.ac.bd, c=US
Date: 2020.10.17 21:37:04 +06'00'

Mohammad Zavid Parvez, PhD

Assistant Professor
Department of Computer Science and Engineering
Brac University

Co- Supervisor:
(Member)

Tanzim 17.10

Md Tanzim Reza

Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar, PhD

Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

The word “brain tumor” defines the unusual expansion of the cells in the brain. Among other tumors, brain tumors are possibly one of the most alarming and life-threatening. So, detection of brain tumor in early stage is much needed because many individuals died as they were unaware of getting a tumor in the brain. For this purpose, different machine learning algorithms and image processing techniques are used for the early detection of brain tumor. The aim of this study is to detect brain tumor by observing different areas of brain and tumorous grow of brain tissues with the help of functional magnetic resonance imaging (fMRI) data. Our main goal is to determine whether the tumor is present in patient’s brain or not. After data collection, we have pre-processed the data where different steps like image extraction, data segmentation were performed. We have used CNN architectures for the classification of brain tumor. For this purpose, different pre-trained CNN model VGG16, VGG19, Inception V3, ResNet50, DenseNet121 and Xception have implemented. Among those models we have identified 3 models (Inception V3, DenseNet121, VGG19) which gave higher accuracy compared to other models and selected them for further work. Rather than taking one model as most accurate we have used ensemble method in our study which produced better predictive solution in terms of brain tumor detection.

Keywords: Brain Tumor, Classification, fMRI, Machine Learning, CNN architecture, Deep Neural Network, Ensemble Learning

Dedication

We want to thank Almighty Allah and dedicate our work to all the patients struggling against Tumor Disease to motivate us to work in this field. We also want to dedicate our work to our parents and teachers who taught us the meaning of our life and sacrificed their whole lives to make us a good human being.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Mohammad Zavid Parvez sir and co-supervisor Md Tanzim Reza sir for their kind support and advice in our work. They helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ADAM$ Adaptive Moment Estimation

ANN Artificial Neural Networks

$BOLD$ Blood Oxygen Level Dependent

CNN Convolutional Neural Network

CNS Central Nervous System

CSF CerebroSpinal Fluid

CT Computed Tomography

DL Deep learning

DTI Diffusion Tensor Imaging

FCM Fuzzy C-Means

$fMRI$ functional Magnetic Resonance Imaging

HGG High Grade Glioma

ICA Independent Component Analysis

KNN K-Nearest Neighbors

LGG Low Grade Glioma

ML Machine Learning

MRI Magnetic Resonance Imaging

NLP Natural Language Processing

NN Neural Networks

PCA Principal Component Analysis

PET Position Emission Tomography

$PSVM$ Proximal Support Vector Machine

$ReLU$ Rectified Linear Unit

RF Random Forest

RL Reinforcement Learning

RNN Recurrent Neural Network

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Motivation

In the human body, brain is the most critical organ that allows us to think and feel. Brain also controls our actions and reactions, vital functions like breathing, movement of the arms and legs as well as it controls different functions of organ within the body that is crucial for the survival of human life. Brain tumor is the abnormal growth of tissues in our brain. It can cause both mental and physical problem for ourselves. Due to brain tumor problems like headaches, vision problems, seizures, memory loss, changes in personality, anxiety or depression etc. can be happened. The symptoms of the brain tumor vary depending on the location, type and stage of the tumor. As brain works as a central computer that controls different functions of our body so any problem arise in the brain may affect our whole body. So, early detection of brain tumor is a great concern for us. The main objective of our research was the detection of brain tumor at an earlier stage so that necessary steps can be taken immediately to reduce the further growth of the tumor as well as minimize the life risk due to brain tumor which can lead to brain cancer. So, early detection of brain tumors is the key part and first phase of tumor treatment planning. There are two types of brain tumors called benign and malignant tumors. Among these two types of brain tumors, a benign tumor is a primary type of brain tumor that is not cancerous, while a malignant tumor stores cancerous cells. So, the identification of the tumor in the primary stage is much necessary. Early detection of many lethal brain tumors like Meningioma, Glioblastoma, Astrocytoma increased the survival of the patient as they can be treated better at an early stage. Many researchers have found that with the help of modern technology and different machine learning algorithms, classification of brain tumors can be a path for the detection of brain tumors. We have chosen fMRI (functional magnetic resonance imaging) data for our research. Moreover, we have used CNN (convolutional neural network) for data preprocessing and classification of the image and at the end we have applied ensemble method to get an optimal result . The aim of our research is the early detection of brain tumor with better accuracy in a easier process.

1.2 Major Contribution

For brain tumor detection, most of the time, researchers use fMRI images as it can show the cellular structure of the brain. After the extraction of fMRI images, it will be in the DICOM(.dcm) format. We need to convert it into the NIfTi(.nii) format. It is a 3D model of the brain, so we can easily detect the brain tumor of the patient by looking into the NIfTi images. Then we can use Deep Learning, Machine Learning algorithms for finding the brain tumor tissue. In our paper, we have followed the Deep Neural Network approach. We have used six different types to CNN model architectures VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception. The main reason for selecting this process is these models are pre-trained with millions of images. So, we did not have to build CNN model architecture on our own. However, we have created a dense layer and fully connected layer and trained our models with our training dataset. Furthermore, we have compared among these models and selected 3 models in terms of accuracy. After that we have performed ensemble method using those 3 models and it provided us better accuracy for detecting the brain tumor.

1.3 Thesis Orientation

The following sections of this research paper have been assembled in the following way. Chapter 2 contains the Literature Review that is the past study of the related works and different mechanisms close to our proposed methodology. The next section in Chapter 3 describes the background study that is related to our research such as fMRI, Brain Tumor, Ensemble Learning, CNN and different Machine Learning models. Chapter 4 includes the proposed model and planning of our work for the detection of brain tumors. After this, the experimental results and analysis of our study have shown in chapter 5. At the end of this paper, the conclusion and future plan of our work have been included in chapter 6.

Chapter 2

Literature Review

Many researchers have applied multiple methods and techniques for the detection of brain tumor. Several classification and segmentation techniques as well as machine learning algorithms are available in the existing literature.

M. Monica et al. [31] proposed a model for detecting the grade of a tumor. In their method, they first removed the noise from the image by using a PCNN. Then for segmentation they used Fuzzy C means and extracted the features. After this, they used naïve bayes classifier for classified the features. In their method, they achieved 91% accuracy.

Another approach for the detection of brain tumor was done by Animesh Hazra et al. [41] based on segmentation. In this method during pre-processing, firstly image were converted from RGB to gray as well as a median filter were applied for the removal of noise. They also considered the image enhancement because the contrast of the image might be low. Then sobel, canny or prewitt edge detectors were applied for the detection of edge. After this part, they executed single, multiple or various thresholding and segmented the result. Later, the k-means algorithm was implemented for clustering the binarized image.

Vidya Dhanves et al. [39] in their approach firstly they segmented the region of brain and they classified it after that. In their approach, they initially converted the image into a contrast image. Then, they preprocessed the image with the help of the morphological operators erosion and then they dilation it to perform skull stripping and for noise removal median filter was used. After this, k-means clustering was performed and the preprocessed image was segmented. Then an object labeling algorithm classified the segmented region. An accuracy of 75.51% and a True-Positive rate of 96.67% was reached by using this method.

For brain tumor segmentation Navpreet Kaur et al. [44] used self-adaptive K-means Clustering. A couple of steps were performed for the pre-processing of the data in their approach. Initially, the noise was removed by applying a median filter and then the brain surface extraction algorithm used for skull stripping. This is important because tumor like shape may create trouble during detection of the tumor as well as high-intensity value may also present in the eyes or bones. Later on, removal of Salt and Pepper noise was performed. Furthermore, the image was segmented by using self-adaptive K-means clustering and lastly to extract boundaries a Sobel edge detector is used.

Naouel Boughattas et al. [48] used multi-kernel learning for feature selection and classification in their proposed model for brain tumor segmentation. The importance of features is weighted by using the multi kernel learning those are classified by using Support Vector Machine. An expert labelled the features of boundary voxels among tumors, edema and also the healthy tissues and during training step they created a database for that. By using this dataset MKL-SVM were trained and classification of voxels also executed. Later on, post-processing is performed with the help of morphological operators because small errors usually seen in the resulting segmentation and MKL-SVM algorithm is applied one more time on a reduced area around the segment.

Another method was proposed by T. Chithambaram et al. [37] to classify brain tumors and for this purpose, they used genetic algorithms and artificial neural networks. Initially, the content-based active contour model is used to mark the tumor regions. After that, intensity and texture features are extracted by using these regions and then genetic algorithm is implemented for feature selection which optimizes the feature selection. Furthermore, an ANN or a SVM classified the features. In this method, the ANN had an accuracy of 94% and the SVM of 91.7%.

K. B. Vanishanvee et al. [25] worked on a proximal support vector machine (PSVM). Compared to standard vector machine it is considered more faster and computationally effective in terms of classifying features of the segmented image of the gray level cooccurrence matrix. They used principal component analysis (PCA) to select the features. The overall accuracy for this approach was 92% which is better compared to the accuracy of using normal support vector machine which was 82%.

Another approach was proposed by Saumya Chauhan et al. [36] where they used an algorithm for the detection of brain tumor as well as they classified the tumor whether it is benign or malignant. At the beginning of their work they used a median filter for noise removal from the image. After that they converted the gray image to RGB and $l \times a \times b$ color space respectively. Then they performed colour based k-means clustering and segmented the resulting image. Furthermore, they extracted features from the region that contains most indexes. For feature extraction they used the gray level co-occurrence matrix, the histogram of gradients and the Canny edge detecter. After that they classified these features using the IBkLG algorithm (Instance Based K-Nearest using Log and Gaussian weight kernels classifier). In this process it classified the images with an accuracy of 86.6%.

Sonali et al. [23] first classifies the tumors into three types in their proposed algorithm which are Normal, Benign and Malignant. Additionally, they divided malignant tumors into two sub types called glioma and meningioma. At the beginning, they used principal component analysis (PCA) for features extraction. After that they applied a probabilistic neural network to classify it. They used 70 samples to train their algorithm and tested it on 35. They achieved 97.14% and 100% accuracy in their approach.

Salim Ouchtati et al. [53] suggested another novel technique for the detection of brain tumor. In their approach along with the detection of brain tumor they also tried to identify the tumor location. First they reshaped the images to 128×128 pixels and then they normalized it during pre-processing. They divided the image into 64 window of the size 16x16 in the feature extraction step and they extracted one to three of the central moments of the order for the histogram of every window as features. After that neural network was performed for the classification of these features. The method reached an accuracy of 88.33%.

Nilesh et al. [34] performed the detection of brain tumor in the following way. First, they applied adaptive contrast enhancement based on modified sigmoid function to preprocess the MR images. Then they used skull stripping process so that from the MR image they can remove the non-brain tissues. After this, they segmented the preprocessed image by using a threshold again. Furthermore, the resulting binary image was processed again so that white pixels can be removed which are free from infected region and it was done with the help of erosion. The outcome can be considered as a mask on the original image. At the end, they computed the gray level co-occurrence matrix and used support vector machine (SVM) to classify the extracted features. An accuracy of 96.51% was reached from the classification by SVM.

Aye min et al. [45] suggested a method for the segmentation of brain tumor and for this purpose the fMRI image was first pre-processed using a median filter and apart from this a wiener filter was also used. Then they merged the results which were filtered. After this adaptive K-means clustering applied for image segmentation. Rather than using normal k-means they used adaptive K-means because if the value of K is fixed then it is considered as a strong bias. At last, for segmentation of brain tumor region they used morphological operators. In this method their accuracy was 98.30%.

Another algorithm was propsed by Nitish et al. [19] for brain tumor classification. In their approach they divided the tumors into four separate tumor classes. For feature extraction the gray level co-occurrence matrix was used. Then, they classified the resulting features with the help of a two layer feed forward neural network. After this, the Levenberg Marquart nonlinear optimization algorithm was used to optimize the features. They reached an accuracy of 97.5% in their proposed algorithm.

F. P. Polly et al. [54] in their suggested algorithm along with brain tumor detection they also categorized the tumor as low-grade glioma and high-grade glioma. Low grade gliomas are less destructive compared to high grade gliomas where as high grade gliomas can be life-threatening. They used Otsu binarization to process binary images and depending on the variance of the image a threshold was applied on the image by this process. Then, with the help of k-means clustering they segmented the binary image. Furthermore discrete wavelet transform was used to extract the features. Additionally, principal component analysis (PCA) was used so that most significant features can be selected from the resulting features after reducing from all the extracted features. At the end, SVM was used to classify the extracted features. Authors claimed that in this approach for brain tumor detection and classification they reached 99% accuracy.

J. Seetha et al [61] proposed a CNN classification algorithm for the detection of brain tumor. In CNN, image can be easily scalable and it can work with 3D input and output volume so five-layer convolutional neural network was used in their approach. First, the image was divided into two type. One type was those image that contained tumor and another was non-tumor image and they organized the images in different directory. In their approach they performed preprocessing, feature extraction and classification in the training stage. They resized the image and trained the fully connected layer by using python. In their approach they achieved the high performance with least time. To improve the testing accuracy they applied gradient descent algorithm for calculating loss function. In terms of category with brain tumor, the authors acquired the highest probability score value. On the other hand, lowest probability score value was acquired on non-tumor brain category. Researchers compared the performance with other existing technique like SVM. In this process, the authors observed classification output result and feature extractions value and depend on this they calculated the accuracy. After comparing with other existing method researcher found that CNN is the effcient automatic brain tumor detection techniques and they reached 97.5% testing accuracy with low complexity in their approach.

Hanwat et al. [67] also proposed a CNN based brain tumor classification method in their research and six step consisting methods was proposed by the authors. The stpes were as follows: insertion of image, image preprocessing, image segmentation, feature extraction, classification and performance evaluation. In their study, they applied CNN, Random forest (RF) and K Nearest Neighbors (KNN) classification techniques for classify the brain tumor MRI images. They took 94 MRI sample images for their study. Inception V3 architecture of CNN was implemented in their study that is consists of four layers (convolutions, pooling, max pooling, and fully connected layers). Moreover, Random forest classifier was used to make a group of many decision trees and one result was produced by each decision tree. They compared their result with other prominent machine learning classifier like K-Nearest Neighbors where K-Nearest Neighbors acquired 74% of accuracy and CNN acquired 80% of accuracy. From their study, they analyzed that CNN works better than others algorithm and it can be perhaps considered as one of the best classifier. Furthermore, the study demonstrated that the normal exactness of the brain tumor characterization with the assistance of Convolutional Neural Network classifier is 98% with 0.097 cross-entropy and is 71% is the approval precision. Researchers successfully compared between CNN, RF, and KNN and. Through their study researcher found that for performing various phases of brain tumor arrangement CNN is one of the efficient approach.

After studying several research, we have applied convolutional neural network for the detection of brain tumor. In our study we have performed six pre-trained CNN model VGG16, VGG19, Inception V3, ResNet50, DenseNet121 and Xception so that we can compare the result between them in terms of accuracy. After observing those models we selected 3 models which were Inception V3, DenseNet121 and VGG19 because their accuracy was higher compared to other models. In the past research, many studies have classified brain tumor with the help of only one classifier or model. However, in our study rather considering one model as most accurate we have implemented ensemble method among those 3 selected models which helped us to detect the brain tumor with higher accuracy compared to one individual model.

Chapter 3

Background Study

This chapter covered several topics, different methods and techniques that are related to our research area. Many articles, books, research papers were observed to gather information about brain, brain tumor, fMRI, various machine learning models, convolutional neural network and other subjects that are similar to our study. Brain tumors are break down into two sub-categories which is labelled as a primary and secondary type of brain tumor. The primary type of brain tumors emerge from brain cells whereas secondary type brain tumors metastasize into the brain from different organs [52]. In the past two decades, multiple research have been performed with modalities like EEG, fMRI or MEG etc. to classify brain tumor stages of the human brain. However, the most common technique that is using at present to detect and classify for the brain tumor is MRI or fMRI [9]. By using fMRI techniques abnormalities within the brain can be detected that may not be possible with other imaging techniques. For the detection or classification of a tumor the intensity, shape and texture of a region are used as well as those are also used to detect brain tumor in fMRI or MRI images along with the help of machine learning algorithms [10]. To solve machine learning problems, CNN can be a right choice and it works excellent in different aspects [33]. Convolutional Neural Network has a significant contribution to medical technology for the classification of brain tumor from MRI images [76]. MRI data is suitable for extraction of the image of brain tumor patients and for this purpose CNN can be useful with greater precision.

3.1 Brain

All living organisms have brain whether it is human or animal and brain is the central organ for all. The brain is accountable for all the information that process and coordinate in our body. It also functions as a control system that helps us for the understanding of speech, the movement of lips and it controls the other organs of human body [63]. Among other vertebrates, the human brain is the largest brain compared to the weight of the body that is approximately 3.3 lbs. The brain volume for the male on average 1274 cubic centimeters whereas the female brain volume is 1131 cubic centimeters. About 86 billion nerve cells (neurons) present in the brain [12]. Moreover, billions of nerve fibers (axons and dendrites) also exist in the brain. The central unit of the brain and nervous system are neurons [5]. Gathering sensory information, maintaining blood pressure and respiratory system as well as hormone releases are the most primary features of the brain [65]. Brain is consist of three main components which are the brain Stem, cerebrum and cerebellum. In the brain stem, there is reticular formation which is made up with grey and white matter. It links the spinal cord with the cerebrum and cerebellum. As reported by Mayfield Clinic, cerebrum is the core part of the human brain and it is separated into two hemispheres. Mainly both the right and left hemisphere of the brain lies in the cerebrum. The brain stem situated below, and the cerebellum sits at the back of it [22]. The hemispheres are covered with a collection of fibers called the corpus callosum, that transfer information from one end to another [14]. The outermost layer of the cerebrum is cerebral cortex which is composed of four lobes and they are the frontal, parietal, temporal and occipital lobes [1]. The cerebral cortex is remarked as the heart of complex thinking and is notably enlarged in human brains. Vision perception, problem solving, judgement related tasks, motor function, sound and expression, spatial orientation and movement etc different task are performed by these four lobes. The different parts of brain is shown in the Figure 3.1

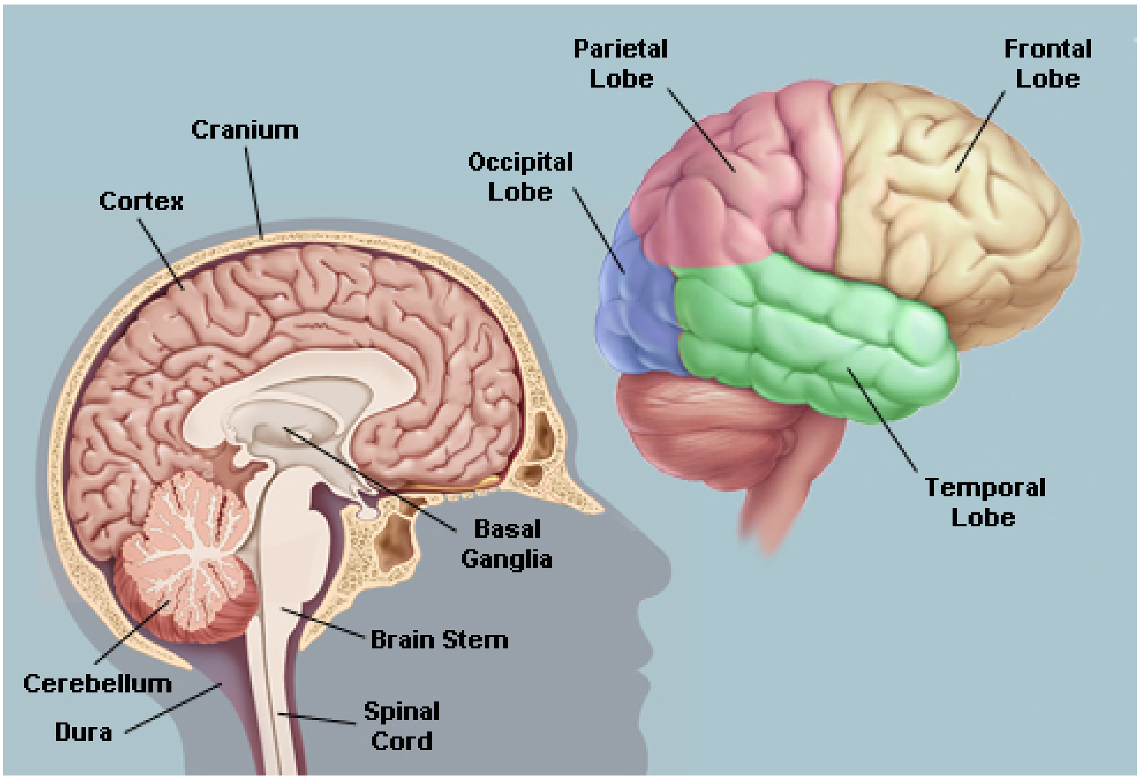

Figure 3.1: Basic structure of a Human Brain

3.2 Brain Tumor

Brain tumor is the result of the unexpected growth of the cells in the brain. Meningiomas is one type of prevalent brain tumor in adults which make up 37.1% of all primary brain tumors and approximately 81% malignant brain tumors make-up from Gliomas (such as glioblastoma, ependymomas, astrocytomas, and oligodendrogliomas) [82]. Glial cells are the main source where maximum of the primary brain tumors emerge and they are classified by low-grade glioma and high-grade glioma [60]. “Edema,” “non-enhancing (solid) core,” “necrotic (or fluid-filled) core,” and “active core” are four properties for intra-tumoral regions based on which the tumors are classified [60]. Tumor’s genetic features and type of glial cell involved in the tumor are observed and it is used to classify the Gliomas which can help to predict the future behavior of the tumor as well as the treatments that will work for it. Based on the location and rate of growth of the tumor it affects our brain function and it can be life-threatening as well. Brain tumors can be classified into two types which are benign tumors and malignant tumors [11]. Among this two types benign tumors are generally non-cancerous. Usually, it does not transmit other regions of the body or the adjacent tissues but if they interfere with portions of the brain that is responsible for vital bodily functions then it can be deadly. With the help of surgery or radiation therapy benign tumors can be usually treated as it’s transmission or growing rate is slow compared to malignant tumors. On the other hand, Malignant brain tumors are not like benign tumors and it is cancerous. It’s growing rate is much higher compared to benign tumors. So, malignant tumors are more dangerous for human life. The average survival rate is only 35% in terms of malignant brain tumor patients. Glioblastoma multiforme is the most common form of primary malignant brain tumors and only 5.6% is the five-year relative survival rate in this case [82]. The survival rate may vary based on different factors like tumor types, size, location of the tumor, age and gender of the patients. Most importantly the size of the tumor is a great matter of concern because the survival rate decreases when the tumor size increases [13]. In the Figure 3.2, some random patient’s 2D brain tumor images are given.

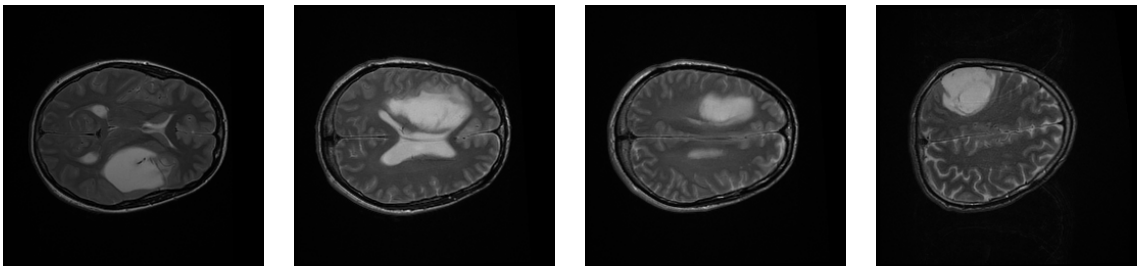

Figure 3.2: Sample 2D Images of Some Patients with Brain Tumor

3.3 Functional Magnetic Resonance Imaging

fMRI (Functional Magnetic Resonance Imaging) is a classification of imaging approach aimed to demonstrate regional and time-varying diversity in brain metabolism [3], [4]. fMRI has the ability to figure out brain function safely, noninvasively and fruitfully. It can be used as an activation maps by observing the blood flow changes that which parts of the brain are being stimulated or involved in a selective mental process [2]. fMRI system is easy to use and it works with a very high spatial resolution. The major benefit of using fMRI is that it does not use radiation like X-rays, positron emission tomography (PET) and computed tomography (CT) scans. Virtually fMRI has no risks if it can be done properly with precaution. fMRI is a unique type of magnetic scan. MRI scanner houses a very powerful electromagnet through the cylindrical tube. The magnetic field within the scanner influences the magnetic nuclei of atoms. Generally, atomic nuclei are arbitrarily organized but due to the effect of a magnetic field the nuclei become adjusted with the direction of the field. The degree of alignment is proportional with the magnetic field. The magnetic signal from hydrogen nuclei in water (H_2O) is detected in fMRI. In capillary red blood cells oxygen is supplied to neurons by haemoglobin. If neuronal activity increases then the demand for oxygen also increases. The rise of local response in blood circulation also has a connection with the regions of increased neural activity. Haemoglobin becomes diamagnetic when oxygenated. On the other hand it is paramagnetic when deoxygenated. The change in magnetic properties brings little fluctuation in MR signal of blood based on the degree of oxygenation. Since blood oxygenation changes depending on the levels of neural activity these variations can be used to distinguish brain activity. Convolutional Neural Network(CNN) is showing great potentials and it is most commonly used to figure out visual imagery. In fMRI data processing CNN is generally used as relevant features can be extracted automatically [42]. Identification and diagnose of Alzheimer's disease (AD) has become easier as CNN can be used to classify the fMRI data effectively [29].

3.4 Machine Learning

Machine Learning is defined as an application of simulated intelligence where a machine or program becomes acquainted from some learned data or knowledge and make prophecy or outcomes based on the experiment data or experience but not specifically programmed for executing fixed or appointed task [74]. Emphasis is basically given on crating data driven decisions that may attain information and utilize that to be asked for themselves. Learning from the error and developing the accuracy level much better gradually while experimented to newer data is the performing pattern of these algorithms. The process of this learning makes with espial or informative data like instances, direct knowledge or recompose in order to searching for information aptitude and creating greater options within the future. Along with that Moreover, data are classified into training set and test set by which algorithms are experienced by training data and assumed data are compared to test data to define the accuracy. Algorithms of machine learning by comparing these trained and tested data is monitored repeatedly till reaching to a vicarious accuracy level. Except human interference or assistance allowing laptop to be mechanical and to change their actions accordingly is the initial objective. The external setting may be comprised of external info, delineated in some type, that may be a supply of external info. Method of translating external info to information is the method of training. In the Figure 3.3, working flow diagram of a basic machine learning model is shown.

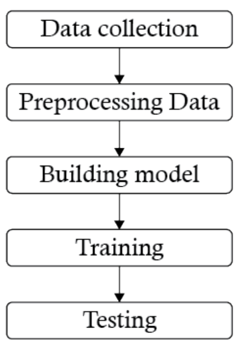

Figure 3.3: Basic Machine Learning Model

Machine learning algorithms can be classified in the four following classes:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Machine Learning

3.4.1 Supervised Machine Learning

This process concerns with teaching through examples. Application of supervised machine learning algorithms can be given through what has been attained in the former training data to new experimented data utilizing labeled examples to prophecy the expected outcome [57]. Exposure of system is significant in quantities of classified data in time of making supervised machine learning. Having a teacher supervising the entire operation is the conception from where the "supervised" is derived. The system or program does not achieve the capability of executing specific targets or outcomes to any new data given to it until enough number of training is applied on the data. Training data will be a composition of inputs collaborated with the correct outputs when a supervised learning algorithm is equipped. Scanning of the data is performed by the algorithms for patterns that equivalents to the desired outputs during training. Followingly, the completion of training, a supervised learning algorithm would take on new unfamiliar inputs and defines how to classify new inputs based on prior training results. Comparison of the system outputs with the accurate and expected output can be performed and errors can be identified by this way. This might help to modify and improve in getting more accurate outcomes and as well as reduction of errors. In the Figure 3.4, working flow diagram of supervised Machine Learning is shown.

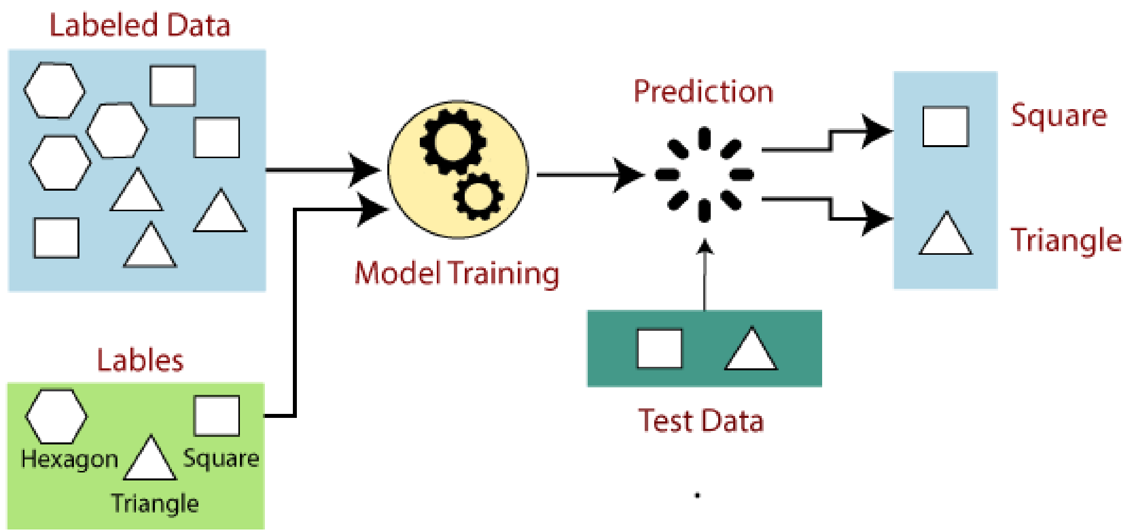

Figure 3.4: Supervised Machine Learning

3.4.2 Unsupervised Machine Learning

In case of usage of data or information that are to be trained is not labeled or classified unsupervised machine learning is embedded. For this approach of machine learning a mathematical model is to develop from a set of data that contains only inputs. It is applied against a data not containing a historic mark. Searching out a function that can show a clandestine structure helps to apply the system [20]. Instead of getting the accurate, specific outputs unsupervised learning emphasizes on exploring the data and draw hypothesis from the datasets to describe clandestine structures from unlabeled data. Association Rules are an example of such algorithms. In the Figure 3.5, working flow diagram of unsupervised Machine Learning is shown.

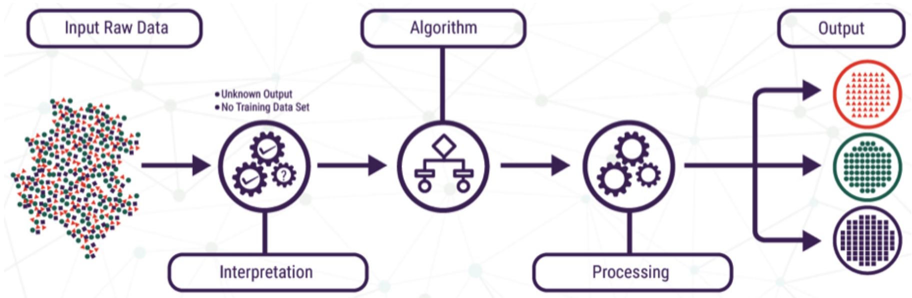

Figure 3.5: Unsupervised Machine Learning

3.4.3 Semi-Supervised Machine Learning

Semi-supervised machine learning algorithm is a combined class that utilizes both the unlabeled and labeled data for training purpose and this combination makes a decision to define this class as an intermediate version of supervised and unsupervised machine learning [17]. Generally, a vast portion of unlabeled or unclassified data and a little portion of labeled data is used to train the dataset. This method is useful and effective while attaining a vision of developing high learning accuracy. An exception is observed in case of semi supervised machine learning when obtained labeled data demands for efficient and pertinent structures for training it or gaining from it, semi supervised learning method is used. Additional resources or structures are not a requirement for gaining unlabeled data to learn. In the Figure 3.6, working flow diagram of Semi-Supervised Machine Learning is shown.

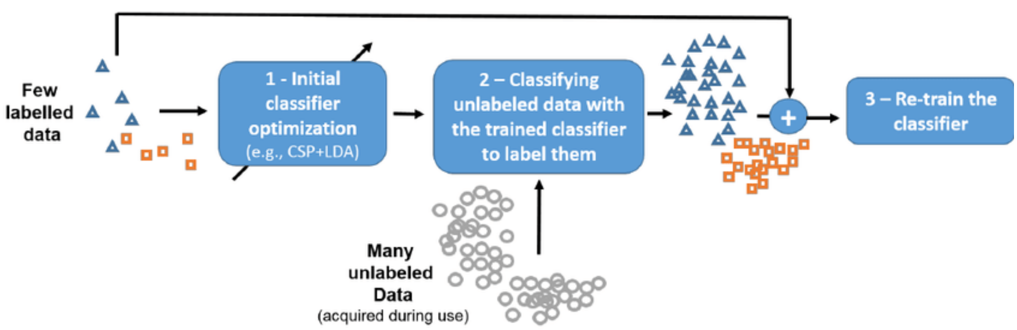

Figure 3.6: Semi-Supervised Machine Learning

3.4.4 Reinforcement Machine Learning

Reinforcement machine learning is the area of machine learning that involves it interests regarding software agent action in a process or system to optimize accrued rewards. Interactions of these algorithms with its environment take place by generating actions and this is how to find out errors or rewards. Achieving knowledge to obtain a goal in an unpredictable and a potentially intricate environment is possible by the agent through this learning. This type of knowledge provides recompose in a positive or negative form in a progressive environment. Necessity of a simple reward feedback is much important for knowing which action is most appropriate and which is not. This is defined as the reinforcement signal. Automatic learning of the model manner within a particular context is allowed for the machine and system in this method which helps it to execute maximum performance [7]. In the Figure 3.7, block diagram of Reinforcement Machine Learning is shown.

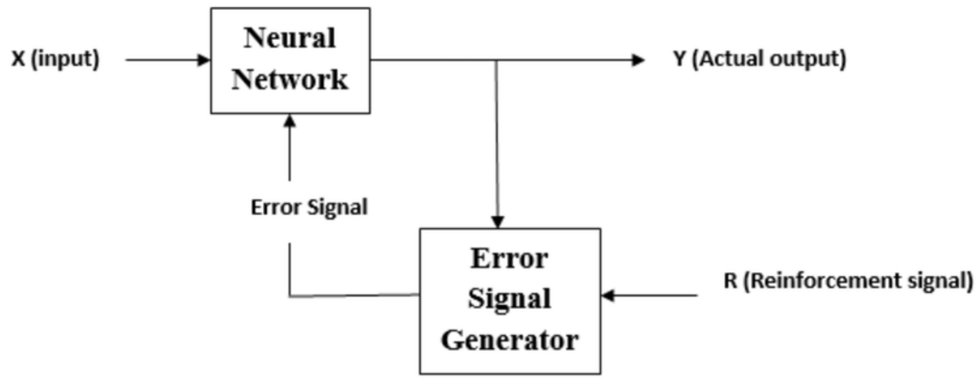

Figure 3.7: Reinforcement Machine Learning

3.5 Deep Learning

Deep learning has become an important and essential part of machine learning which was generated from ANN. Hierarchical infrastructures are used in deep learning method for the cohesiveness of their layers. Simple linear or non-linear equations are used to conjunct the output among layers in these models. Usage of low-level peculiarity to transform into higher level abstract peculiarity is a specialization of these models. In deep learning models, tracking to the extensive abstract representations of the input data is maintained and capability to know about these factors routinely is created [21].

In the Figure 3.8, Deep learning works better beyond ordination than machine learning when it concerns a case of unraveling complex problems, like speech recognition, image classification, NLP. The training time is long is case of deep learning as it uses a large number of parameters. Suggestion of classical deep learning method is using ANN, CNN, RNN and their alternative versions.

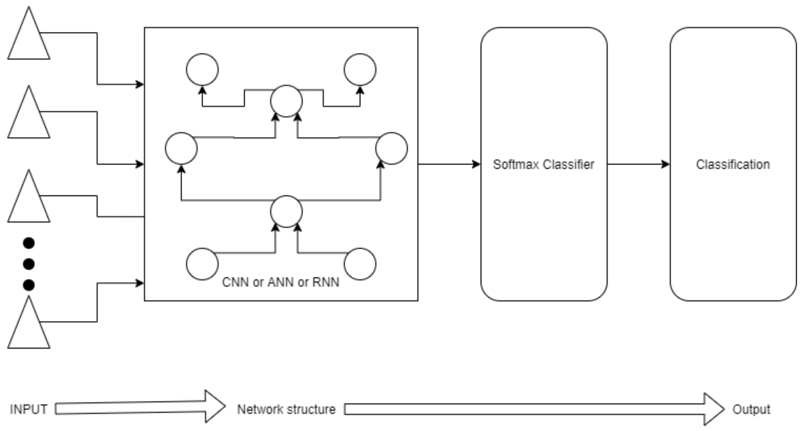

Figure 3.8: Deep Learning Model

3.6 Neural Networks

A series of algorithms that exerts for recognizing underlying relationships in a set of data through a process that imitates the way the human brain works. These poly-layers networks are useful for the division, prediction and so on. These networks have abilities to adapt to changing inputs; so, the networks generate the most effective results except necessitates redesign the output criteria. ANN has an input layer, which supplies all the input data and they are passed on to the activation function after assigning individual weights to each of the inputs [6]. Implementation of the activation function requires ways to create the output which would later pass on to the output layer. Possibility of having multiple hidden layers between the input and the output layer is also available. Classical NN models are supposed for feed forward algorithm which passes the data only in a single direction to the pathway of output layer. In case of an error signal generation at the output layer back propagation is necessary. After that, for reducing the error extent the error gets back to the former layer. Weight update delta rule is used for the purpose.

3.7 Convolutional Neural Network

A CNN i.e. Convolutional Neural Network is known as a particular form of artificial neural network in which usage of perceptron's, an algorithm for machine learning unit is practiced exploring data for supervised machine learning. This class of neural network has a wide range of utilization in the world of medical science and technologies. It's been over the years the many researchers have tried their best to introduce a standard model that is capable of detecting the tumor in a way having excellent efficiency and effectivity. Unexceptionally, a neural network is also consisting of an input layer, an output layer, and multiple hidden layers like other artificial neural networks. Some of these layers are convolution, using a process of mathematics to move findings on to successive layers. This includes couple of layers in which layer of convolution, layer of nonlinearity, layer of pooling and layer of complete relation are present. Though the layers of convolution and fully connected layers have parameters no parameter is accredited for pooling and non-linearity layers. An extraordinary level of performance is observed in case of machine learning problems through the CNN [58]. Some of the common implementation of CNN are image classification, object identification, object tracking, pose estimation, text identification and recognition, visual saliency detection, behavior recognition, scene labelling, voice and natural language processing. In the Figure 3.9, basic CNN architecture is shown.

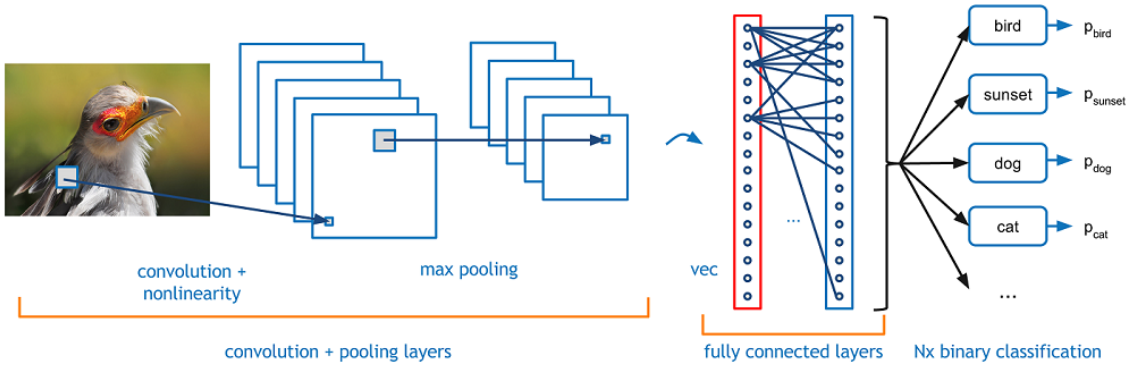

Figure 3.9: Basic CNN Architecture

3.7.1 VGG16

VGG is said to be the design of a Convolution Neural Network. Noticeable and significant contribution of VGG is to imply the classification or localization exactitude is excessed by higher growing CNN depth through the employment of little receptive field in the layers. Larger receptive fields were used in neural network before VGG (7 * seven and eleven * 11) compared three to three * 3 in VGG but they weren't as deep as VGG. There has existence of several unit versions of VGG, the deepest is with nineteen layers of weight. The key difference between VGG16 and VGG19 is that there are unit sixteen layers in VGG16 and unit nineteen layer in VGG19. There are convolutional layer and three area unit absolutely connected layer in VGG16 16 layers area [55]. In the Figure 3.10, VGG16 architecture is shown.

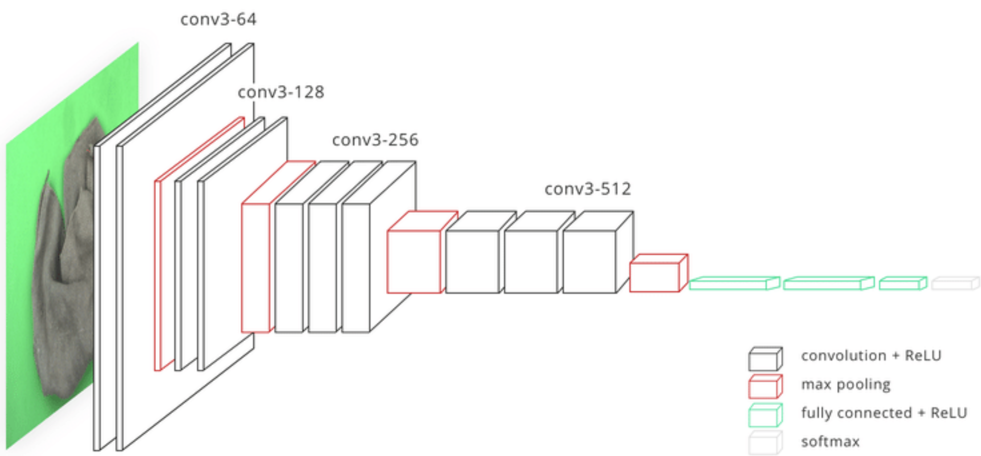

Figure 3.10: VGG16 Architecture

3.7.2 VGG19

VGG19 is a 19 unit deep convolutional neural network. Among many unit versions of VGG, the deepest one is with nineteen layers of weight. A pretrained version of the network trained on more than a million images from the ImageNet database can be loaded. The pretrained network has the ability to classify images into 1000 object categories, like keyboard, mouse, pencil and many animals. For this reason, the network is blessed with enriched feature representations for a vast range of images 224-by-224 is the input size for this class of convolutional neural network. In the Figure 3.11, VGG19 architecture is shown.

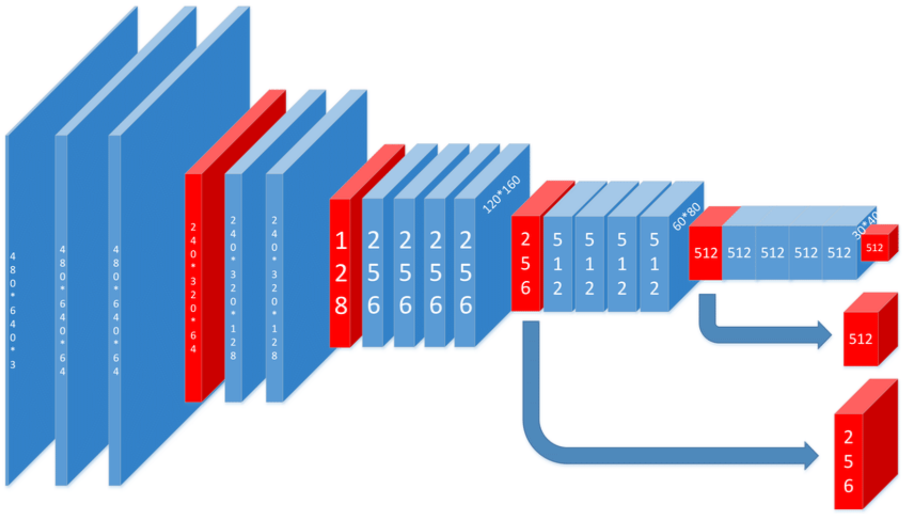

Figure 3.11: VGG19 Architecture

3.7.3 Inception V3

Inception V3 might be considered as a convolution neural network that is trained from the ImageNet information on over 1000000 images. This forty-eight layers deep neural network has a ability to categorize images in a thousand classes of things. The pre-trained Inception V3 model achieves state-of-the-art accuracy for recognizing general objects with 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher" [73]. This Inception V3 neural network may consist of two parts:

- Feature extraction part with a Convolutional Neural Network (CNN).
- Classification part with fully connected and SoftMax layers.

General features are extracted from input images in the first segment and categorization of them based on those features in the second part by the model. In case of transfer learning, while constructing a new model to classify original dataset, reusing of the feature extraction part and re-training the classification part with the dataset takes place. Training of the model with less computational resources and training time is possible as there is no need to train the feature extraction part (which is the most complex part of the model). In the Figure 3.12, Inception V3 Model is shown.

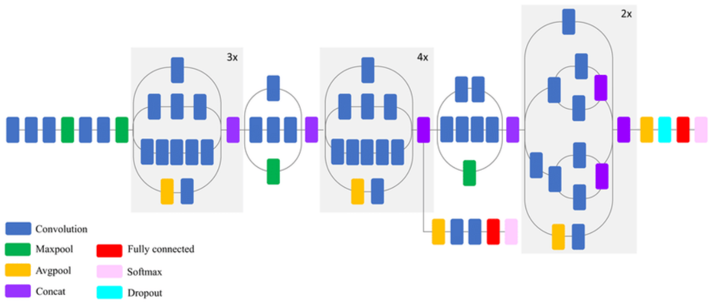

Figure 3.12: Inception V3 Model

3.7.4 ResNet

ResNet also known as Residential Energy Services Network, a residual neural network is a type of artificial neural network which is built from pyramidal cells in the cerebral cortex. Before ResNet, there had been several ways to deal the vanishing issue, for instance, adds an auxiliary loss in a middle layer as extra supervision, but none seemed to really tackle the problem once and for all. For minimizing error rate more layers were utilized in deep neural network in many architectures. Though it is suitable for few numbers of layers with the increased layers number there is a common problem in deep learning called Vanishing/Exploding gradient. This makes the gradient to become 0 or too large. As a result, with the increased layers numbers, the training and test error rate also increases. ResNet, proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network is considered as a better alternative to meet this challenge. For solving the problem of the vanishing/exploding gradient, this architecture utilized the concept called Residual Network that uses a technique called "skip connections." Task of the skip connection is to skip training from a few layers and connect directly to the output. Addition of this skip connection facilitates if any layer hurt the performance of architecture then it will be skipped by regularization [32]. So, this results fruitfully in training very deep neural network by eradicating the problems caused by vanishing/exploding gradient to a greater extent. Usage of a 34-layer plain network architecture inspired by VGG-19 in is performed in this networking system which then the shortcut connection is added. The architecture is then converted into residual network by these shortcut connections. In the Figure 3.13, ResNet architecture is shown.

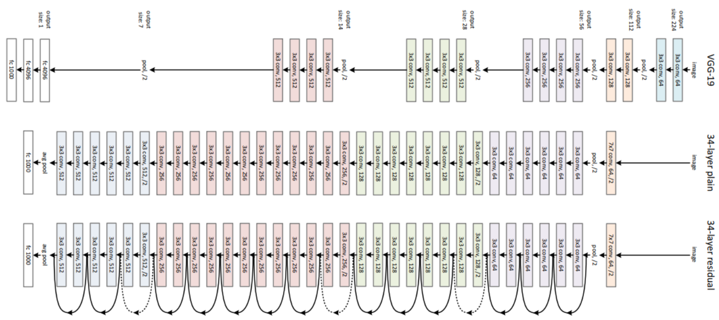

Figure 3.13: ResNet Architecture

3.7.5 DenseNet

Substantial depth of DenseNet has been observed more accurate, and effective to train if they possess shorter connections between layers near to the input and those close to the output in the recent time. DenseNet can be defined as a new CNN architecture that reached State-Of-The-Art (SOTA) leads to classification datasets (CIFAR, SVHN) through the usage of less parameters. Basically, this portion of DenseNet also represents some of the advantageous face of the DenseNet. Such as alleviating the vanishing-gradient problem, strength feature propagation, encouraging feature reuse and simultaneously reducing the number of parameters. Generally, convolution network work in such a way- having an initial image such as a shape, then applying set of convolution/pooling filters on it, squeezing width and height dimensions and increasing features dimension. So, the output from the L_i layer is input to the L_{i+1} layer. On the contrary, DenseNet paper proposes concatenating outputs from the previous layers instead of using the summation. This DenseNet architecture has an amazing connectivity pattern like connection of each layer is done with all others in a very dense block [32]. All layers have the accessibility feature maps from their preceding layers which facilitates heavy feature reuse. As a direct consequence, the model is more compact and less prone to overfitting. Along with that, each individual layer accepts direct supervision from the loss function through the shortcut way, which offers implicit deep supervision. Due to the stated characteristics DenseNet has become a natural fit for per-pixel prediction problems. In the Figure 3.14, DenseNet architecture is shown.

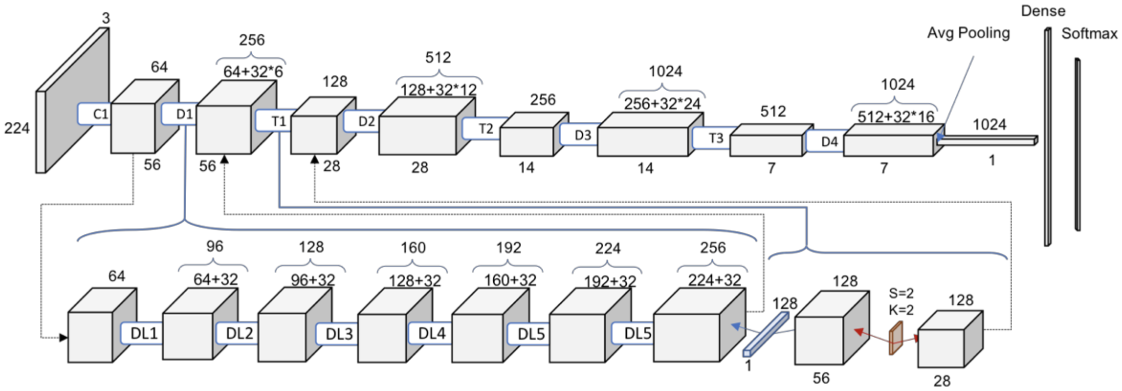

Figure 3.14: DenseNet Architecture

3.7.6 Xception

Xception is a 71 layers deep convolutional neural network which has the image input size of 299-by-299. It is the extreme version of inception. Even it is better than inception v3 with a modified depth-wise separable convolution. The original depth-wise separable convolution is the depth-wise convolution (channel-wise $n \times n$ spatial convolution) followed by a pointwise convolution (1×1 convolution to change the dimension). There is no need of performing convolution across all channel like what happens in conventional convolution. On the contrary, in case of modified depth-wise convolution in Xception point-wise convolution is followed by a depth-wise convolution. The inception module in Inception V3 motivates this modulation that

1×1 convolution is done first before any $n \times n$ spatial convolutions. It causes a slight difference from the original one. ($n = 3$ here since 3×3 spatial convolutions are used in Inception V3.) [38] And these two slight differences can be cited as:

- The order of operation: Channel-wise spatial convolution is done firstly and then 1×1 convolution is done in case of original depth-wise convolution but for modified one 1×1 convolution comes first in order and then channel-wise spatial convolution takes place.
- Presence or absence of non-linearity: There is presence of non-linearity after first operation in original Inception but in case of Xception the modified depth-wise separable convolution, any intermediate ReLU non-linearity is absent.

3.8 Optimizers in Neural Networks

Optimizers in neural network are the class of algorithms or methods that has the ability to change the attributes in a neural network. To minimize the losses what should be the way to change the weights or learning rates can be defined by using optimizer. It does the performance of optimization. Optimization algorithms are responsible for minimizing the losses and to facilitate the most accurate results possible. Various optimizers are researched within the last few couples of years each having its advantages and disadvantages among which some of the prominent are:

- Gradient Descent
- Adagrad
- RMSprop
- Adam
- Poor Conditioning
- Saddle Points
- Local Optima

We have used Adam and RMSprop to optimize our main algorithm.

3.8.1 Adam

Adam (Adaptive Moment Estimation) has a working manner which uses the momentum of first and second order. To avoid the rolling too fast is one of the major intuition or aim behind using Adam as having ability jumping over the minimum, there is expectation of decreasing the velocity slightly for getting a more careful research. Moreover, Adam also keeps an exponentially decaying average of past gradients M(t) as it desires to store an exponentially decaying average of past squared gradients like AdaDelta. As stated before, Adam has also some advantage and disadvantage as well.

Advantage:

- Too fast and rapidly converged
- Ability to rectify vanishing learning rate and to possess high variance

In spite of having it shows a little drawback whenever the case of computational cost is concerned. We set the learning rate $lr = 0.00001$ for this optimizer.

3.8.2 RMSprop

RMSprop shortly known as rprop is utilized only by using sign of the gradient. For different weights the magnitude of the gradient also shows different manner and can be changed during learning and for this reason a single global learning rate is difficult to choose for this. We set the learning rate $lr = 0.0001$ for this optimizer. Rprop is the combination of the idea of only using the sign of gradient with the idea of adapting the step size individually for each weight. This includes increasing the step size for a weight multiplicatively if the last two gradient signs do agree, otherwise decreasing the step size. This also includes limiting the step sizes in between 50 and a millionth (Mike Shuster's advice) [18].

3.9 Ensemble Learning

Ensemble learning is a way that combines several models in order to improve machine learning results and it generates one optimal predictive model. It is an effective way in terms of decision making and compared to a single model or classifier it gives better predictive performance. Basically, this method works on a voting system (weighted) and rather than considering one model as a best or most accurate, it counts the opinion of several models and produce one final result that is more effective. There are different types of ensemble techniques such as Bayesian averaging, Boosting, Bagging, Stacking etc [8], [16]. In the Figure 3.15, Ensemble Learning Model is shown.

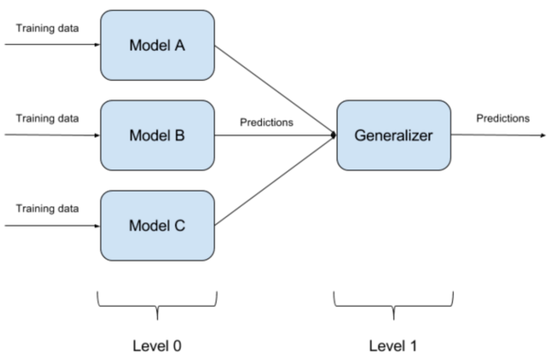

Figure 3.15: Ensemble Learning Model

Chapter 4

Proposed Model

In this chapter, we are going to discuss the workflow of our proposed model which is shown in the Figure 4.1.

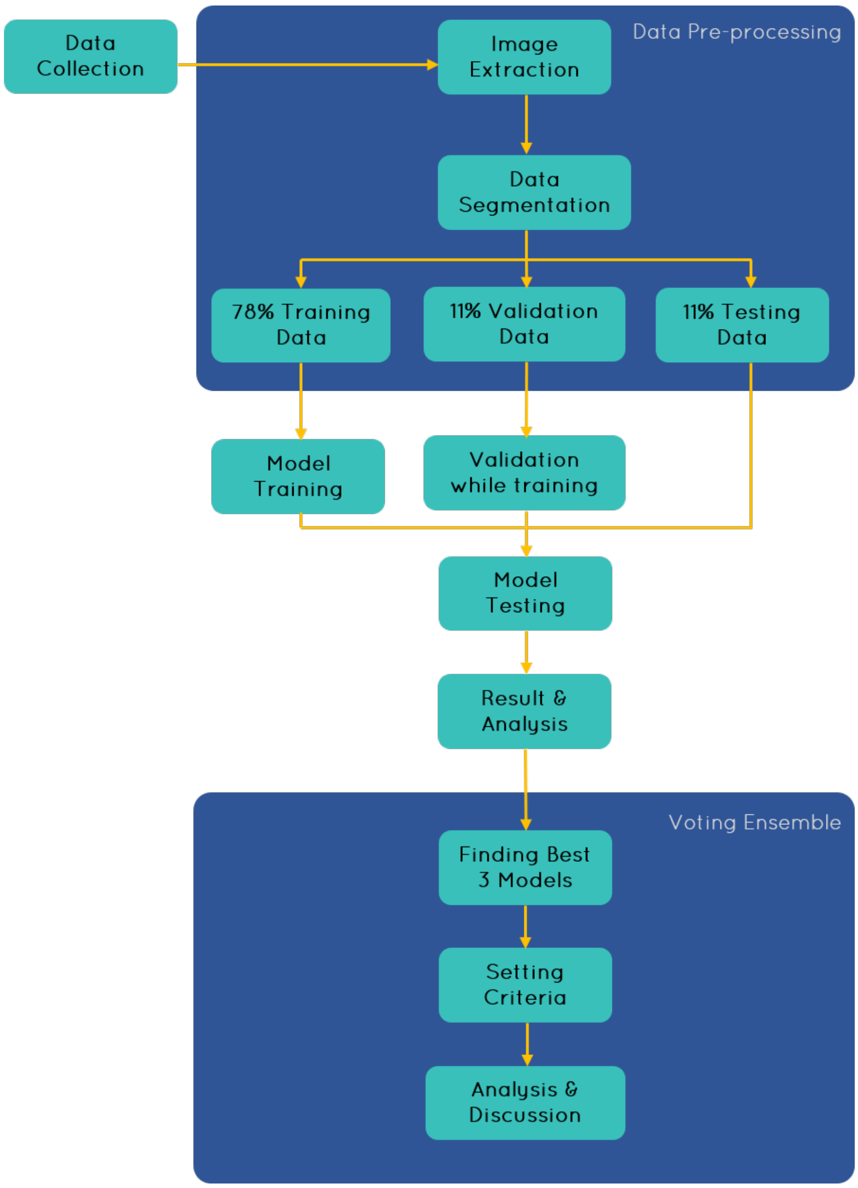

Figure 4.1: Workflow of Proposed Model

Our primary purpose is to detect tumor tissue at an early stage. In this chapter, we will try to discuss how we have built our model to detect the tumor. After the collection of data, we did pre-processing where included image extraction, data segmentation according to train, validation, and test. We divided our data into three segments named train, test, and validation. In each segment, we created a 'yes' class for the patient with a brain tumor and 'no' class for the patient having no tumor. Then We took pre-trained CNN model VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception. We have built a Convolutional Neural Network as we did not take the fully connected layer of the transfer model. After that, we start training our model as well as checked the accuracy by validation dataset. Then, we did model testing and got different accuracy for different CNN models. We analyzed by plotting the accuracy and loss function [79]. After that, we did voting ensemble method. First of all, based on the test data, we calculated the accuracy manually of all CNN models and chose the best 3 models according to the test accuracy. Then, we created the ensembling accuracy. For that, among the best 3 models, if at-least two of them giving 'yes' vote, we determined that, there is a tumor in the patient's brain and while we got two or more than two 'no' votes then we decided that there is no tumor in that brain image. Like this, we made the test accuracy of voting ensemble too and came up with a research that, which one is performing better for detecting brain tumor.

4.1 Data Collection

For any research, we need to collect data from an authentic source, and this process is called data collection. After the collection of data, we need to train our model and check the validity of our model with testing data.

We have collected our data from the UK Data Service [86]. There are data of 22 brain tumor patients collected from the Brain Research Imaging Centre, University of Edinburgh, UK. There was the information on clinical data of patients with unique MRI ID, date of treatment, and necessary notes. We collected the metadata, which contains Pathology, Location of Tumor, estimates of grey/white/CSF and tumor volumes if direct electrical stimulation was performed with its outcome, and the different types of imaging data acquired (T1, T2, fMRI, DTI). We also collected the 'Clinical Data'. There was a list of patients, demographic, handedness, and scores to various clinical tests. All the data are useful for functional magnetic resonance (fMRI) for surgical brain tumor planning.

The unique id of the patient was helpful for us to find them on an individual basis. Those data were organized in tar files with the subject ID. Under the unique id, there were subsections of word repetition, finger foot lips, silent verb generation, COR 3D IR PREP1, axial T2, tissue classes, etc. The file format of all the data was standard NIfTI format (.nii).

From our data, we have enough information about the male-female and age ratio (Table 4.1). There were 9 female patients and 13 male patients. The range of male patients is 30 to 48, and the content of female patients is 27 to 68. We also got information about the Pathology of each patient. It helped us to know better about their conditions. Some of the data shown in Table 4.1

Table 4.1: Few Patient's Information

Patient ID	Sex	Age	Pathology
19277	M	31	Astrocytoma type II
18886	M	41	Astrocytoma type II
19275	F	68	Meningioma
19357	F	49	Glioblastoma Multiform
18638	F	35	Astrocytoma type II
19085	F	28	Meningioma
19015	M	30	Glioblastoma Multiform
18716	M	48	Glioblastoma Multiform
19723	F	27	Astrocytoma type II
18582	M	41	Astrocytoma type II
19398	F	65	Met lung CA

The size of the input image is 224 × 224. There has shown some 2D images after converting from 3D NIfTI image in the Figure 4.2

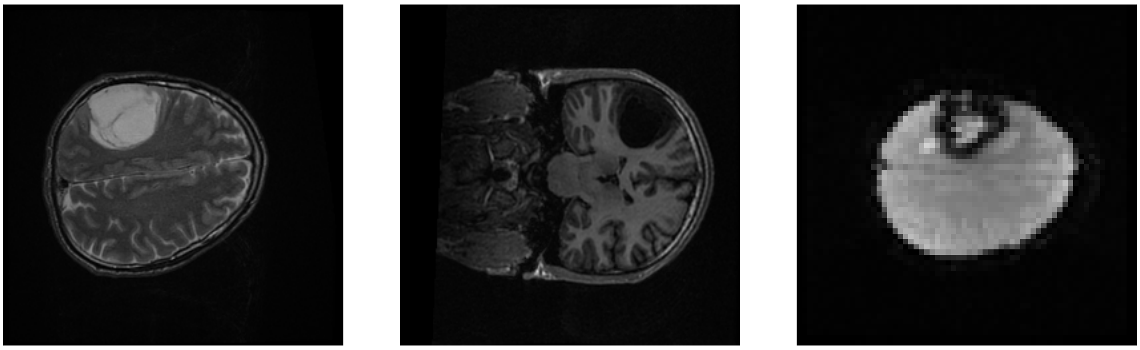

Figure 4.2: 2D Images of Brain Tumor

4.2 Image Conversion

In our dataset, every file was in the NIfTI (nii) format. Which was 3D and have voxel sizes which do not have any range like pixels. So, we needed to convert it into 2D [49]. We took some steps to process it. First of all, we imported necessary libraries such as Nibabel, CV2, NumPy, and Glob in our Python program. Then we declared the path where the NIfTI image is located and passes the path to the nibabel. Then we used data normalization by scaling the image. We used a 'For loop' as one 3D NIfTI image used to convert into more than 30 2D images. We declared a path and save all the 2D images there in PNG format by using the CV2 library. We have given the Python program In the Figure 4.3.

```
In [1]: import nibabel as nib
        import cv2
        import numpy as np
        import glob

In [47]: img_path = r"E:\CSE400\Dataset\Dataset-20200505T215011Z-001\Dataset\f9860d61-19c4-4698-95a57-b580a57e8e8c\19398\tissue_c
         img = nib.load(img_path)
         img_array = img.get_fdata()
         scaled_img_array = (img_array/np.max(img_array))*255
         for i in range(scaled_img_array.shape[2]):
             path = "E:/Data ar Data/a19/"+str(i)+".png"
             img = cv2.imwrite(path, scaled_img_array[:,:,i])
```

Figure 4.3: Data Preprocessing by Converting the 3D Image to 2D Image

4.3 Data Pre-processing

Data pre-processing is a technique in any Machine Learning process where the raw data transformed into an understandable or desired form. Now there are some factors that we used for processing our data. These factors are given below:

4.3.1 Data Normalization

Normalization is a process that is frequently applied as a part of data preparation for Machine Learning [50]. It is used to transform an image into a set of pixels which is more familiar to work. Without corrupting differences in the ranges of values or losing information, it changes the values of numeric columns in the data to a standard scale. It will maintain consistency, encourage accurate insertion, deletion, and alteration [15]. In an RGB image, the range of the pixel is 0 to 255, but the 3D images that we have collected have voxels which do not have any particular range. So, normalization will rescale it so that they end up ranging between 0 and 1. It will break our 3D images into different slices. Every slice will be saved into the desired path by using the CV2 library from our program [30].

4.3.2 Data Augmentation

Data augmentation is one of the most widely used tools in deep learning to increase dataset size and make neural networks perform better [72]. It artificially increases the number of training examples. Sometimes a big dataset is not enough for preventing overfitting [78]. With augmentation, we can rotate, flip, skew, crop, shear, and zoom our image. It will not take extra memory or will not harm our primary data. We can take an image and create a lot of other images from it though there is no need to save all the images. The advantage is while training the data, our model can learn the image in many ways. Therefore, it helps in reducing overfitting or improves generalization [66], [69].

In our model, we took an image as input. Then, if we use data augmentation, it will generate more images, and our primary input data remain the same. We can differentiate classes according to the position of the tumor in the brain. For example, when we give an image as input, then data augmentation will flip the image. So, if there is a tumor on the left side of the brain after flipping, there will be an image that contains a tumor on the right side. After that, data augmentation is simulated through the images, and our model will learn from it. In the testing time, if we give an image as input that has a tumor on the right side, it will easily recognize it. The advantage is we just trained our model with a left-sided brain tumor image, but our model can detect the right-sided brain tumor also. Infect, it will not harm or change our primary data. [40]

4.3.3 Test-Train Data Segmentation

We took 4468 images as training data, 639 images as validation data, and 655 images as testing data. We had data from 22 patients. After pre-processing, we got a total of 5762 images. Then we segmented the data into Train, Validation, and Test. We spilled 78% of data into training and 11% of data into validation, and 11% data into testing [28], [59].

Model Training

We had a total of 5762, and among these, we trained our model with 4468 images, which are 77% of real data so that our Convolution Neural Network model could learn in a better way. There is one binary classification that has two subfolders, 'yes' and 'no.' 'Yes' folder contains images with a tumor, and the 'No' folder contains images with no tumor. The segmented images with 3 x 3 filter helped our model to learn key features of the tumor. The input shape is 224,224,3. That means the height and weight of the images are 224. A sample picture of training model is shown in the Figure 4.4.

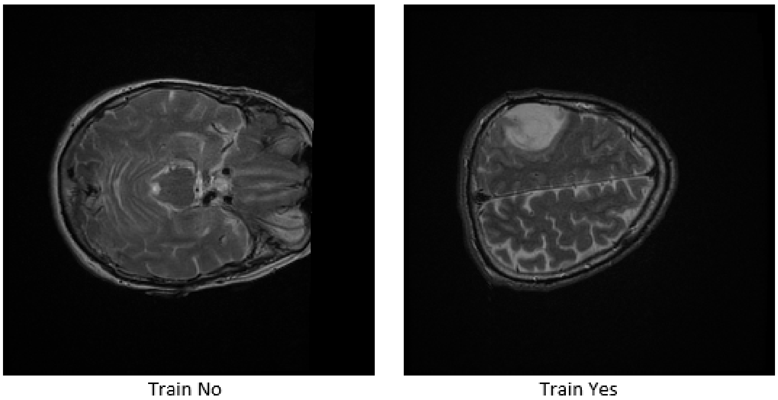

Figure 4.4: The Training Model

Model Testing

For testing our data, we split 11% of data into our train set, which is in the amount of 655 images. After training our Convolution Neural Network model, we applied these images to check the accuracy. After giving these unseen data as input, our model would detect the tumor using the previous learning experience. Based on the accuracy, we could say how good or bad our model is. We used many CNN architectures in this phase. A sample picture of testing model is shown in the Figure 4.5.

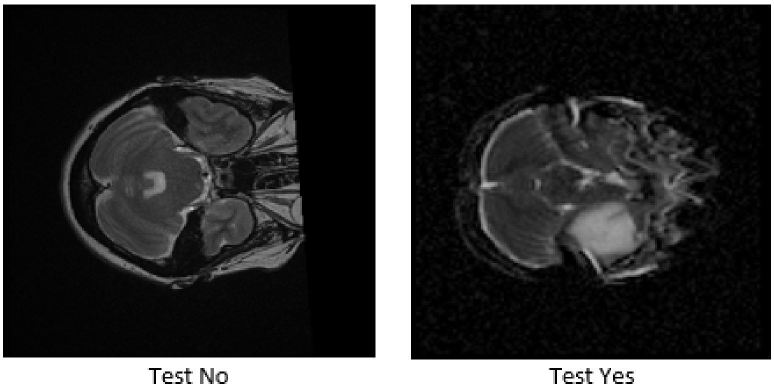

Figure 4.5: The Testing Model

Model Validation

We took 11% of our total data as validation. While training our model, we need to estimate how our model is performing. For that, he used to check the accuracy after every epoch by validation dataset. We took 639 data for the validation set. If we do not use validation, then there will be a high chance that the result we get after testing will be biased to training data. For reducing unbiased results, we need to use the validation data set. A sample picture of model validation is shown in the Figure 4.6.

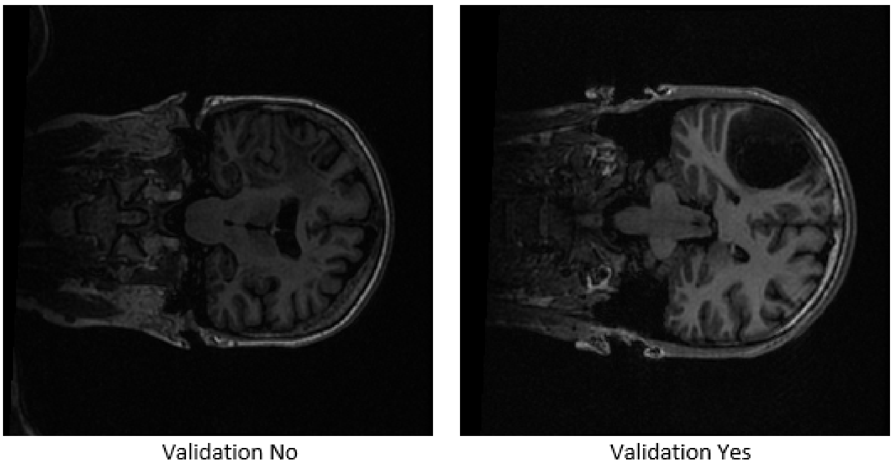

Figure 4.6: The Validation Model

4.4 CNN Architecture

After processing the data, we built our Deep Convolutional Neural Network [56]. We have used transfer learning VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception as CNN models [84]. These models are pre-trained on more than a million images from the ImageNet database [70]. We have imported TensorFlow and Keras library. Then we have used our different CNN models separately [81]. We used Adam and RMSprop as Optimizers. Then we compared with one another by measuring test accuracy. We have taken input shape 224 by 224. Though we have a transfer model, we did not use the fully connected layers. We have flattened the CNN model then added a Dropout Layer. After that, we have added two Dense layers. One was created with 1024 neurons, activation function ReLU and another one was made with two neurons activation functions Softmax. All layers are shown in the Figure 4.7

```
[ ] def print_layer_trainable():
        for layer in new_model.layers:
            print("{0}:\t{1}".format(layer.trainable, layer.name))


    print_layer_trainable()

    True:    functional_1
    True:    flatten
    True:    dropout
    True:    dense
    True:    dense_1
```

Figure 4.7: Model Layers

We have used Categorical Cross-Entropy as a loss function. For model training, we have taken 100 epochs, batch size 16, and per step, the epoch is total loaded data divided by the batch size. Model training is shown in the Figure 4.8

```
optimizer = Adam(lr=1e-5)
loss = 'categorical_crossentropy'
metrics = ['accuracy']

new_model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

epochs = 100
steps_per_epoch = generator_train.n / batch_size
steps_val = generator_val.n / batch_size
steps_test = generator_test.n / batch_size


history = new_model.fit_generator(generator=generator_train,
                                  epochs=epochs,
                                  steps_per_epoch=steps_per_epoch,
                                  validation_data=generator_test,
                                  validation_steps=steps_test)

280/279 [==============================] - 76s 273ms/step - loss: 0.0021 - accuracy: 0.9996 - val_loss: 0.2185 - val_accuracy: 0.9751
Epoch 72/100
280/279 [==============================] - 76s 272ms/step - loss: 0.0067 - accuracy: 0.9973 - val_loss: 0.0776 - val_accuracy: 0.9891
Epoch 73/100
```

Figure 4.8: Model Training

4.4.1 Categorical Cross-entropy

Categorical cross-entropy calculates the performance of more than two classification tasks. It compares the predicted probability distribution with the target probability distribution [51]. When there are multiple possible outputs, the model must choose which one is correct. It is also called Softmax Loss. Before going to the target block, it must use the activation function Softmax. It ensures the output is a probability distribution [68]. Training model of categorical cross-entropy is shown in the Figure 4.9

It measures the summation of the separate loss for each class label per observation. The formula is:

$$CCE = \sum_{c=1}^{M} y_{i,c} \log(p_{i,c})$$

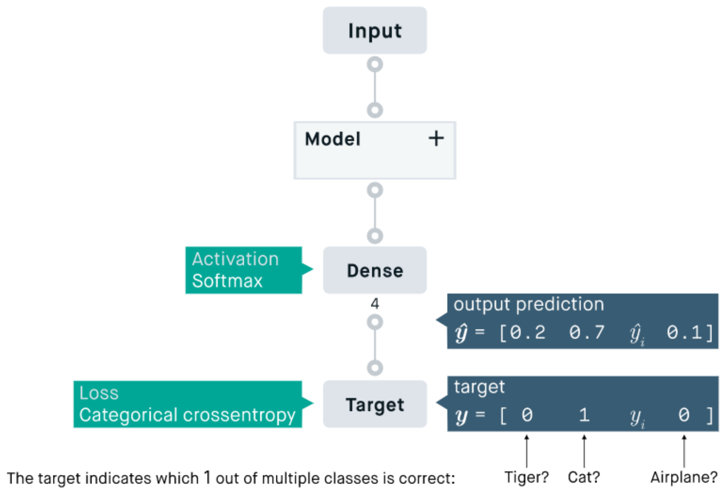

Figure 4.9: Model Training of Categorical Cross-entropy

Where,
M = the number of class
y_i = the binary indicator (0 or 1) denoting the class for the sample

4.4.2 Softmax Function

The softmax function is a function that turns any real values into probabilities [46]. The input can be any number, but the result will always be in the range of 0 to 1. When any small or negative number given, the softmax turns it into a small probability. When a big number has given as input, then the softmax turns it into a big probability. But it won't go below 0 or above 1 [85]. The formula:

$$s(\vec{x}_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Where,
$\vec{x}$ = input vector of the softmax function
x_i = input values of softmax. These values can be any positive or negative value
e^{x_i} = the exponential function will be applied to all input values. If the input is negative, it will return minimal positive value, and if the input is positive then it will produce huge value
$\sum_{j=1}^{n} e^{x_j}$ = it is the normalization term. It ensures that the output values of the function will sum to 1 and the range will be 0 to 1
n = the number of classes in the multi-class classification

4.4.3 ReLU Function

Rectified linear activation function or ReLU is a function which gives output the same input if it is positive. Otherwise, it will provide 0 [71]. It often achieves good performance in Neural Networks, and it is straightforward to train the model. That's why it is, by default activation function of many Neural Networks and commonly used in the output of CNN neurons [80]. But sometimes it decreases the ability of the model to train the data correctly as whenever it gets a negative number, the result becomes 0. In the Figure 4.10, we can not map the negative values appropriately [47]. Mathematically the formula is $y = max(0, x)$

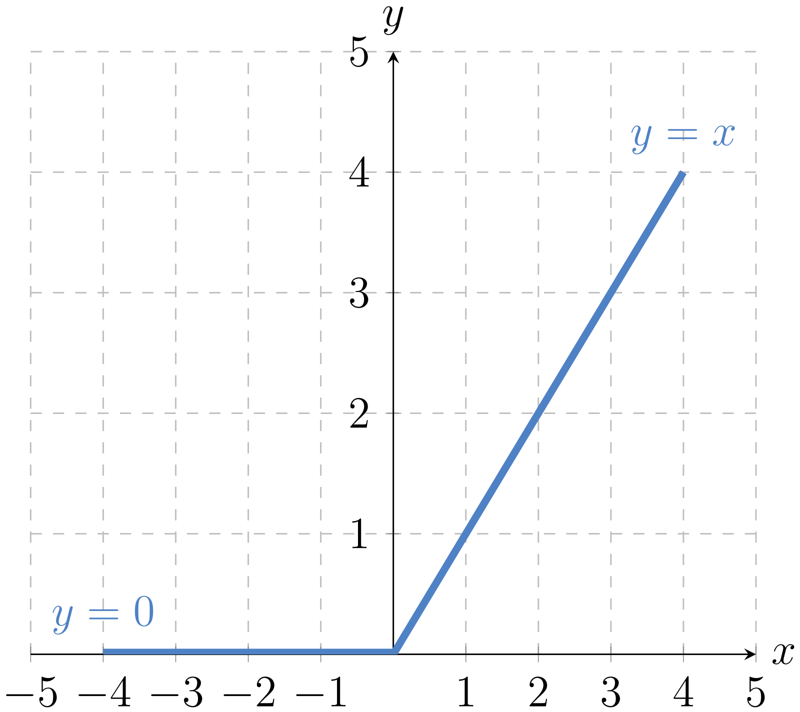

Figure 4.10: ReLU Function

There are many advantages of ReLU functions:

- There is no complicated mathematics, so it takes less time while training our model.
- Compare to other activation function like Sigmoid or tanh it does not have vanishing gradient (a difficulty that is found while training ANN).
- Even the x in the graph gets larger; it does not saturate.
- Though it gives output 0 while the input values are negative, it sometimes desirable as it does not allow some given unit not to active at all.

4.4.4 Dropout Layer

The dropout layer is dropping the unit in a neural network for reducing overfitting. The idea is it removes several neurons from our neural network randomly [43]. The dropout layer can be used in Dense fully connected layers, convolutional layers, and Recurrent layers such as the LSTM network layer. We need to scale the chosen dropout rate because the weight will be larger than before of the network. The rescaling of the weight can be performed at training time or leaving the output unchanged during test time. We have used the Keras library, so we have chosen the first one, and each weight will be updated after finishing each batch epoch [77]. Dropout Layer is shown in the Figure 4.11

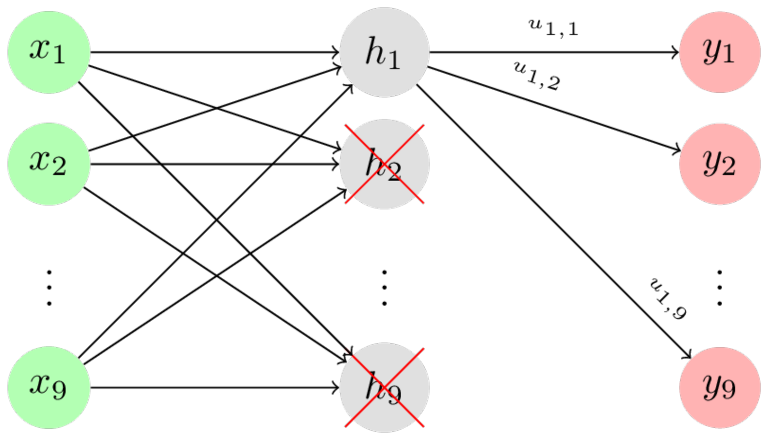

Figure 4.11: Dropout Layer

There are two advantages of dropout:

1. Often in our neural model, neighbouring neurons achieve similar weight which occurs overfitting. So, by dropping some neuron can prevent this.

2. Sometimes a neuron can over-weigh the input from a neuron in the previous layer, and can over specialize as a result.

4.5 Comparison between CNN Models

After executing six CNN models we have observed the training accuracy, loss function and testing accuracy between those models. We performed 10 epochs for those CNN models and analyzed the differences between them from the beginning to the end. In some aspect like training accuracy some model performed better where as in other cases like loss function or testing accuracy they performed average or their performance was not that much pleasant. For some model, though their starting was not good but they ended up with a surprising result. Similarly, we have noticed the opposite result and some model gave very poor result in our study. For example, in terms of train accuracy though the result was high for Xception model but loss function and test accuracy result were unsatisfactory. On the other hand, the loss function was high for VGG19 model as well as the train accuracy and test accuracy result were also satisfactory for this model. However, among those six models test accuracy result of Inception V3, VGG19 and DenseNet121 was higher compared to other models.

4.6 Voting Ensemble

Most of the cases consult with various person or asking multiple votes while taking a decision is better than depending on one particular vote or system. In this study, we have implemented similar voting system for our result which can be called ensemble method. For this purpose, we need to find the models, setting the criteria that means based on which condition our model will provide us what kind of result and at last analysis of that result is also necessary because we need to check whether the result is matching with our expected outcome or not.

4.6.1 Finding Best Three CNN Models

After comparing and analyzing between different CNN models we tried to calculate manually to find the test accuracy between those models so that, we can select best 3 models for our ensemble technique. First of all, we have taken all of the test images into a numpy array. Then, we did resize which includes normalization so that, our images stored in the range of 0 and 1. We have used python's cv2 library for that. We have taken the same input shape (224,224) for all images. We used the same model that has previously trained with our training dataset. Thus, we got '1' value for having brain tumor and '0' value for having 'no tumor' in the brain images. We made separate 'yes' and 'no' table for all CNN models and calculated the accuracy manually. Then, we selected three best performing CNN models for our next research.

4.6.2 Setting the Criteria for Ensembling

We have collected binary value '1' and '0' from our test dataset as a sign of 'Yes' or 'No' in terms of brain tumor detection. '1' means tumor is present and '0' means tumor is not present. As we used three models so for, that we counted the majority vote that means, when we got more than two 1's we considered that, tumor is present. Similarly, we considered the result '0' that means, negative when more than two 0's were present among those three CNN models. That's how we tried to find an optimal solution manually.

4.6.3 Test Accuracy Analysis

In this section, we compared the accuracy between Voting Ensemble and those three CNN models. We compared the individual accuracy separately for each model with the result that we got after ensembling. Thus, we showed that, which one was performing better for detecting the brain tumor.

Chapter 5

Results and Analysis

In this chapter, we will show the final result of our implementation and how good our model is performing for detecting the brain tumor tissue of patients. We will look at our training accuracy, which is the accuracy that we got after applying our CNN architecture model to classify our training data set. Then, testing accuracy, which is the performance of our model on the unknown data set, and loss function, which determines how bad our model is performing. Our primary target is to get the higher training and test accuracy and get a lower loss function [27]. We have taken six CNN model architectures, which are VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception [26]. Then we displayed all of the accuracies, loss functions one by one, and compared with one another. Moreover, we have taken binary value of test dataset and calculated test accuracy manually. Then, identified three best CNN models which generate higher accuracy among those 6 models. At the end, we have applied ensemble method [83] and compared with the other CNN model's accuracy individually.

5.1 CNN Model Training Accuracy

It is also called categorical accuracy. It means how well our model is classifying our training data set [75]. There are two parts to our training dataset. First, one contains the brain image of the patients having no tumor and the second one contains brain images of patients with tumors. With the help of training accuracy, our model will learn accurately how a tumor looks, the position, size, etc. After learning, our model can identify brain tumor while an unknown data will be given. So, it is essential to train our model correctly. We have checked this with training accuracy for all our CNN models. We have taken ten epochs every time.

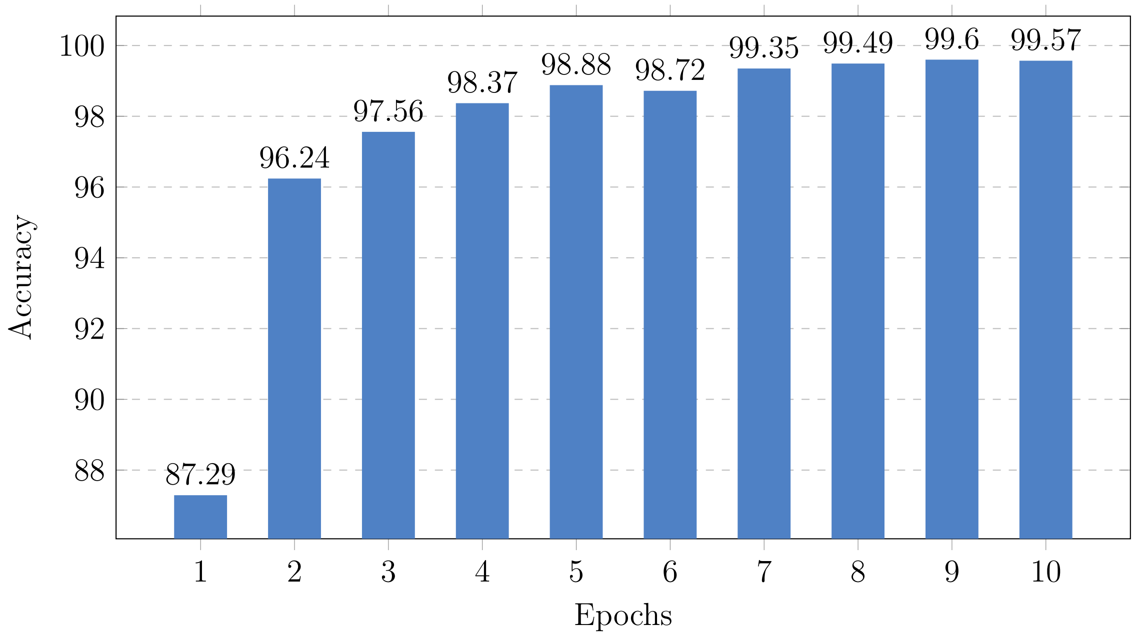

Figure 5.1: VGG16 Training Accuracy

In the Figure 5.1, we can see the training accuracy of our VGG16 model. In the beginning, the training accuracy is 87.29%. In the 9th epoch, we got an accuracy of 99.6% which is the best training accuracy and after ten epochs, our training accuracy is 99.57%. That means our model can classify 99.57% images from our training dataset.

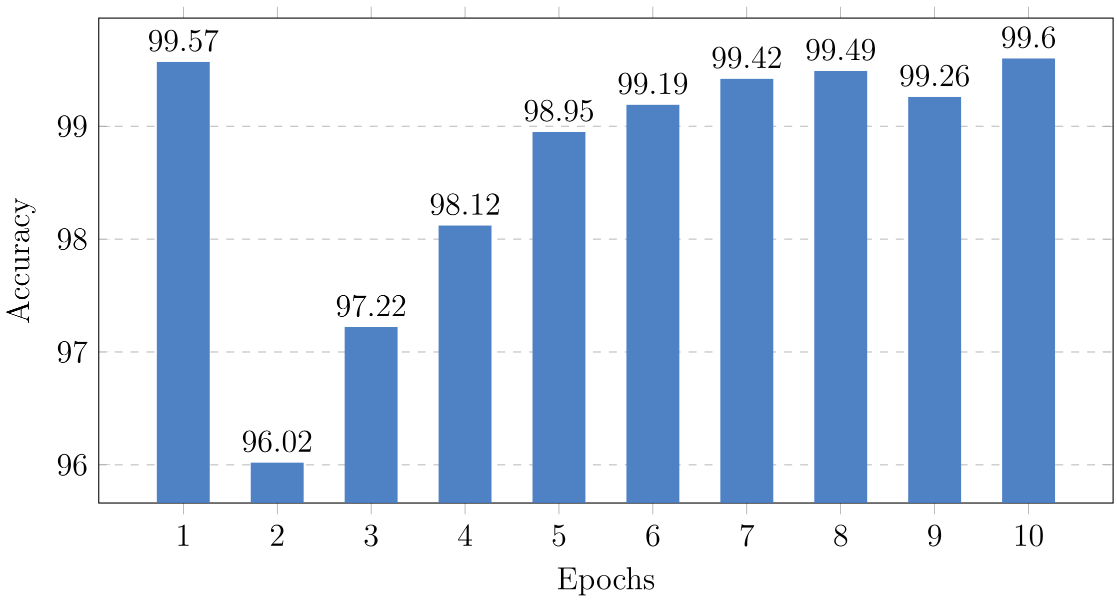

Figure 5.2: VGG19 Training Accuracy

In the Figure 5.2, we have shown the training accuracy of our VGG19 model for classifying the brain tumor in our training dataset. It has started with 99.57% accuracy. After 1st epoch though the value decreased but gradually it increaded again and in the 6th epoch the value showed 99.19% accuracy. We ended up with 99.6% accuracy after ten epochs which was the best accuracy.

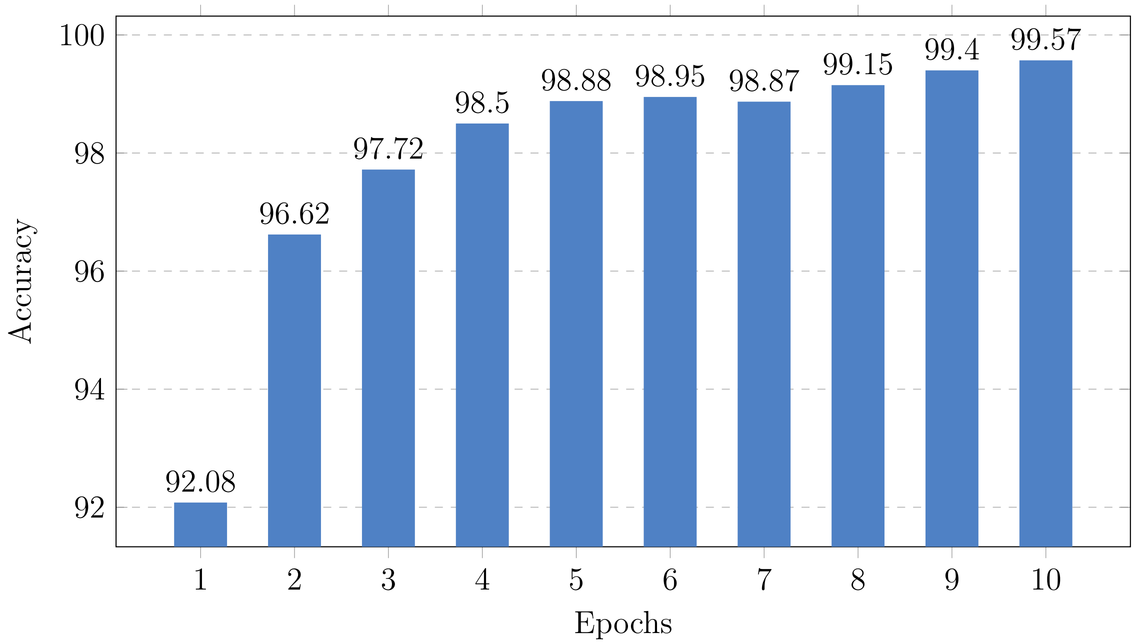

Figure 5.3: Inception V3 Training Accuracy

Figure 5.3 shows the training accuracy of Inception V3. Similarly, we have run our program for ten epochs. In the first epoch, we obtained an accuracy of 92.08%. After that, it has raised from 96.62% to 99.4%. In our last epoch, we got 99.57% accuracy.

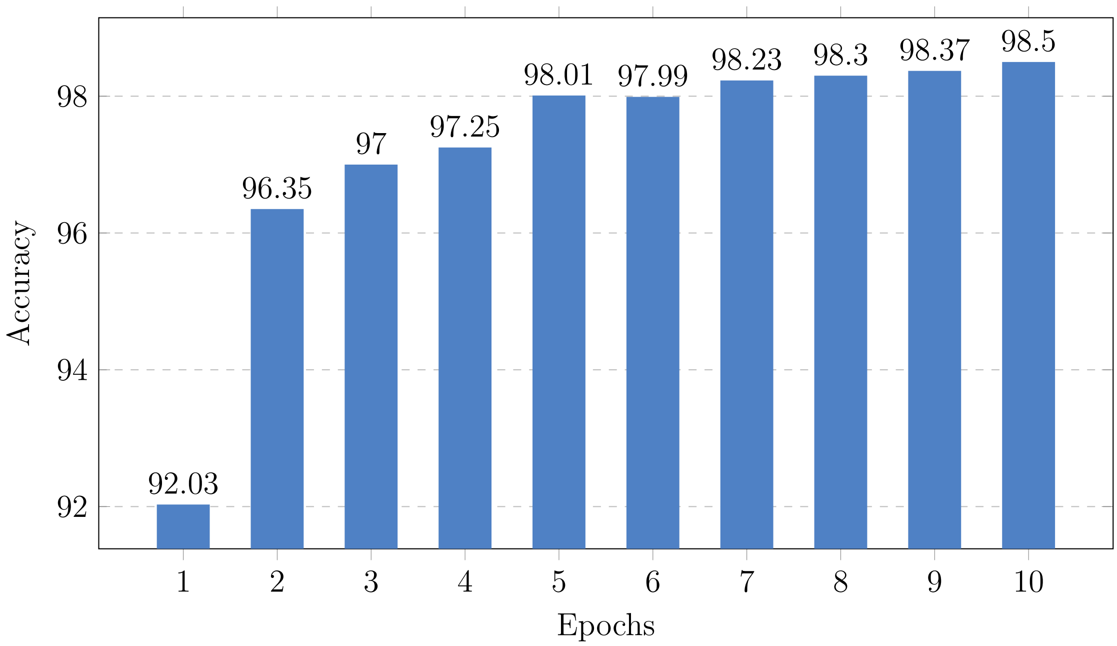

Figure 5.4: ResNet50 Training Accuracy

Figure 5.4 represents the training accuracy of ResNet50. Here, we obtained 92.03% training accuracy in our first epoch. After that, we got 96.35% accuracy. Similarly, we have gone through all other processes like our previous models and, at last, achieved 98.5% training accuracy.

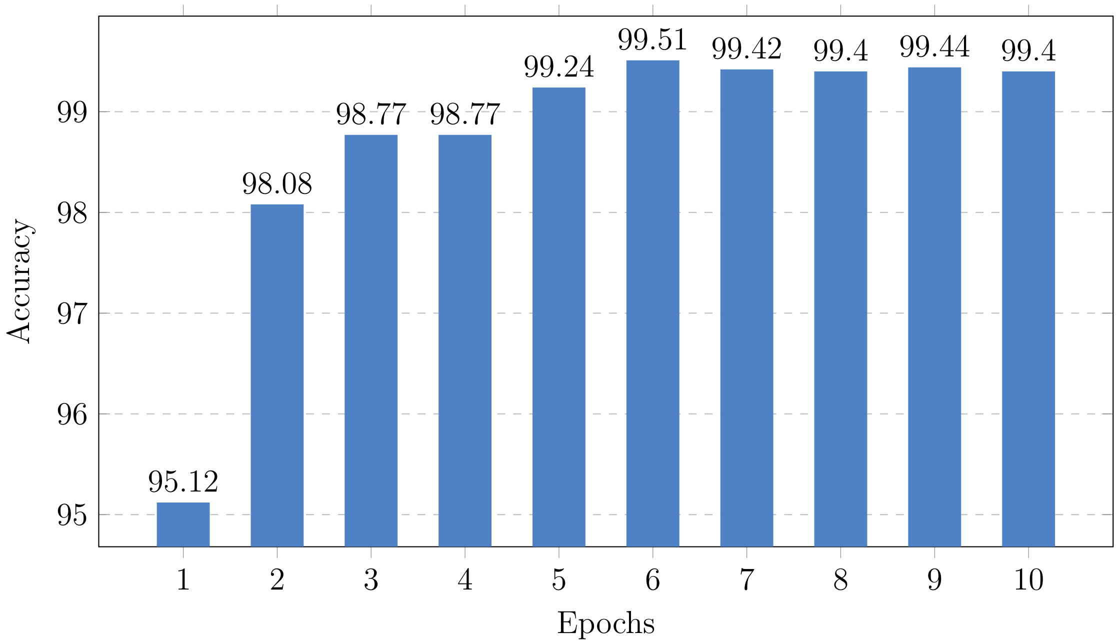

Figure 5.5: DenseNet121 Training Accuracy

Figure 5.5 has been created to show the training accuracy of DenseNet121. In the first epoch, we have got 95.12% training accuracy. After that the value increased gradually and we ended up with 99.4% in our last epoch.

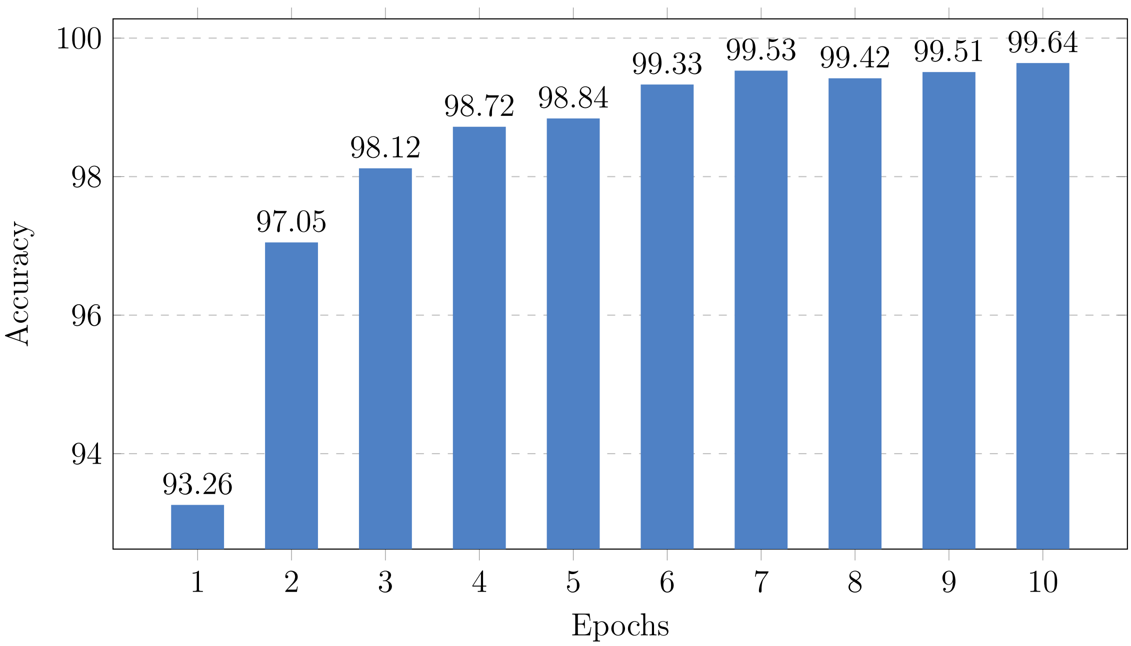

Figure 5.6: Xception Training Accuracy

Figure 5.6 presents the training accuracy of our last CNN model, which is Xception. In the first epoch, we achieved 93.26% training accuracy. However, in the 2nd epoch we got an accuracy of 97.05 which gradually increased till the end. At the end, we got 99.64% accuracy in the 10th epoch which was the highest accuracy among other CNN models.

Table 5.1: Train Accuracies of CNN Models

X-Values	VGG16	VGG19	Inception V3	ResNet50	DenseNet121	Xception
Epoch 1	87.29	99.57	92.08	92.03	95.12	93.26
Epoch 2	96.24	96.02	96.62	96.35	98.08	97.05
Epoch 3	97.56	97.22	97.72	97	98.77	98.12
Epoch 4	98.37	98.12	98.5	97.25	98.77	98.72
Epoch 5	98.88	98.95	98.88	98.01	99.24	98.84
Epoch 6	98.72	99.19	98.95	97.99	99.51	99.33
Epoch 7	99.35	99.42	98.86	98.23	99.42	99.53
Epoch 8	99.49	99.49	99.15	98.3	99.4	99.42
Epoch 9	99.6	99.26	99.4	98.37	99.44	99.51
Epoch 10	99.57	99.6	99.57	98.5	99.4	99.64

In the Table 5.1, we can see the training accuracy of all the CNN architectures on each epoch separately. With the help of this, we can compare which one is the best for classifying the training dataset. From that table, the highest accuracy was obtained by the Xception model in the 10th epoch, which is 99.64%.

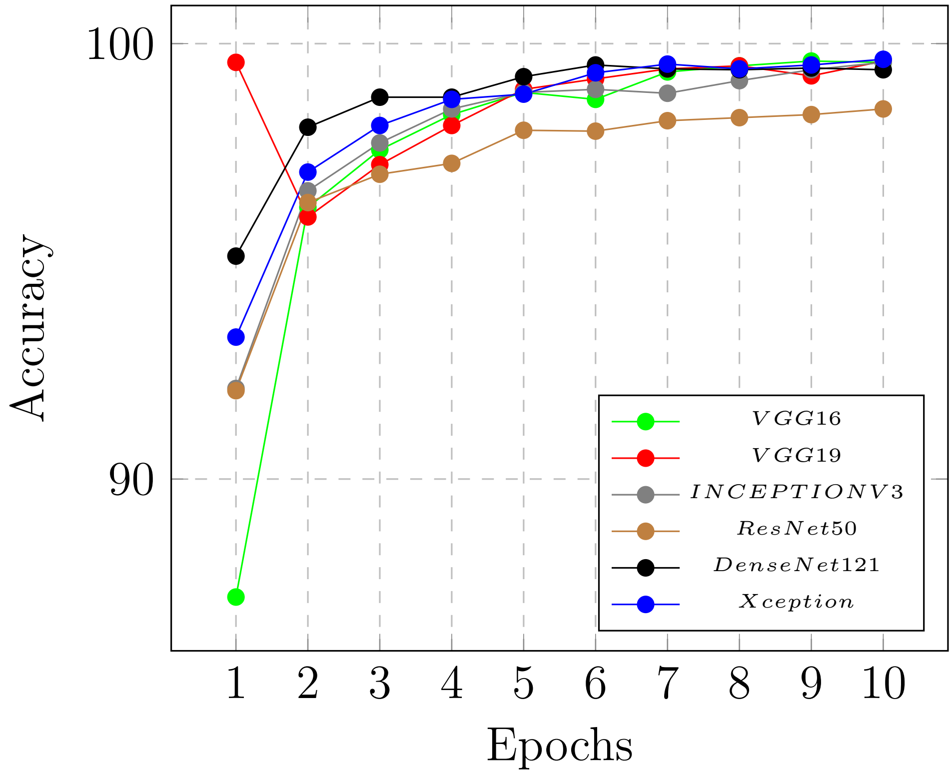

Figure 5.7: Comparison of Train Accuracies of CNN Model Architectures

The graph in Figure 5.7 shows the comparison of training accuracy between all 6 CNN architectures that we have used. It shows either our training accuracy is increasing or decreasing. It looks like all the lines are on the same axis as the training accuracies are very close to each other. But if we notice the green line that is at the top of everyone. It tells us that this is the highest training accuracy model. We also can differentiate the training accuracy in each epoch.

5.2 Loss Functions of CNN Model

The loss function is one of the most critical components of Deep Neural Network. It indicates how far our model is from the actual result. With the value of loss function, we can determine how bad our CNN model predicts the presence of a brain tumor from the dataset [62]. If our models work fine, then the value of loss will be zero; otherwise, the loss will be more significant [35]. If our model learns well from the training dataset, then the loss function will be reduced gradually. Here, our main target is to get as much lower loss function as we can [64]. We have checked the loss function of all six CNN architecture models for ten epochs.

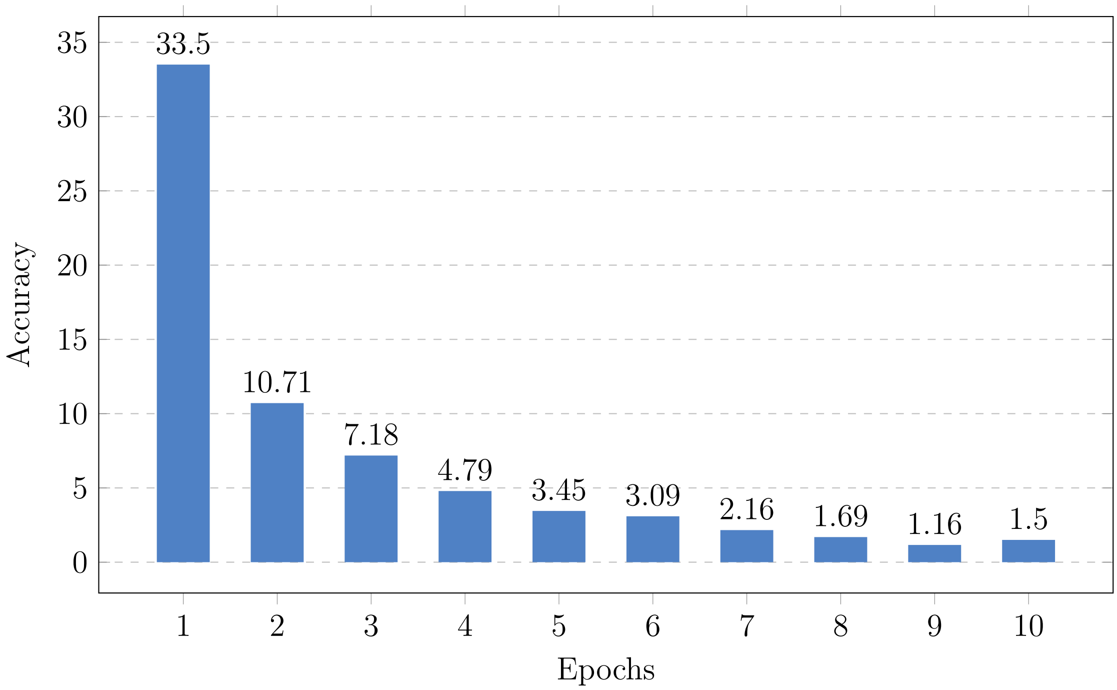

Figure 5.8: Loss Functions of VGG16

Figure 5.8, indicates the loss functions of VGG16 model architecture. In the very first epoch, the loss function was 33.5%. It decreased gradually in the following epochs. The lowest one we have obtained in the 9th epoch, which was 1.16%. That means when we train our model for more epochs the loss function used to decreased more. That is why we have taken 10 epochs everytime for getting lower loss functions.

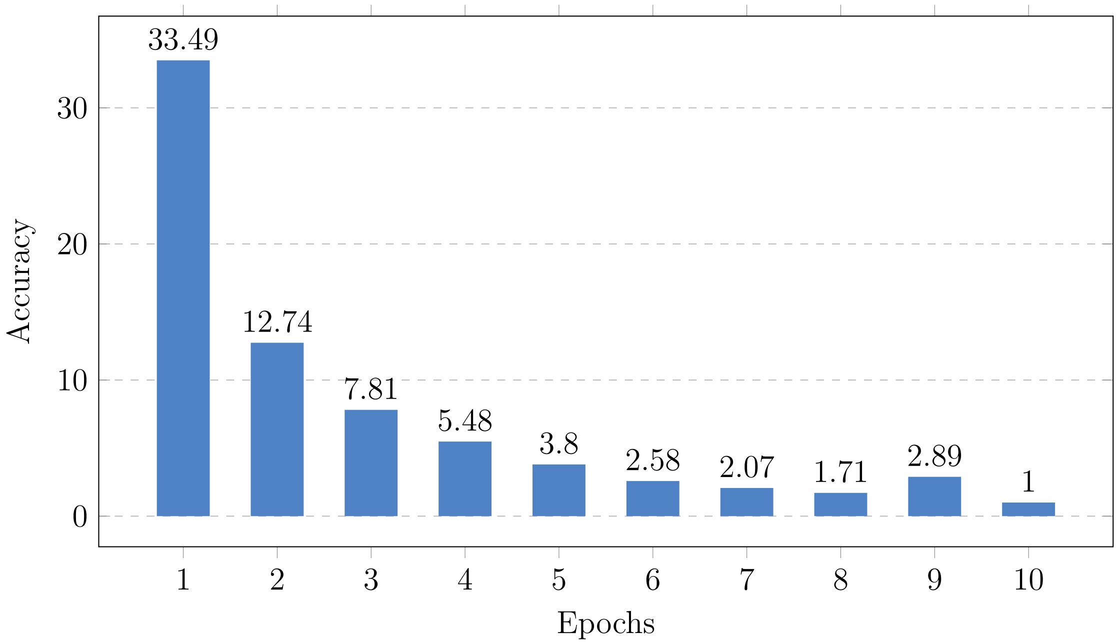

Figure 5.9: Loss Functions of VGG19

Figure 5.9 there has shown the loss functions of VGG19. We can notice that the loss functions are decreasing in one after another epoch. As our training is learning, that's why the loss functions are giving us a lower value than the previous epoch. We started with 33.49% loss function but at the end it ended up with a loss function of only 1% that means we can say our VGG19 model did very well for learning the training dataset.

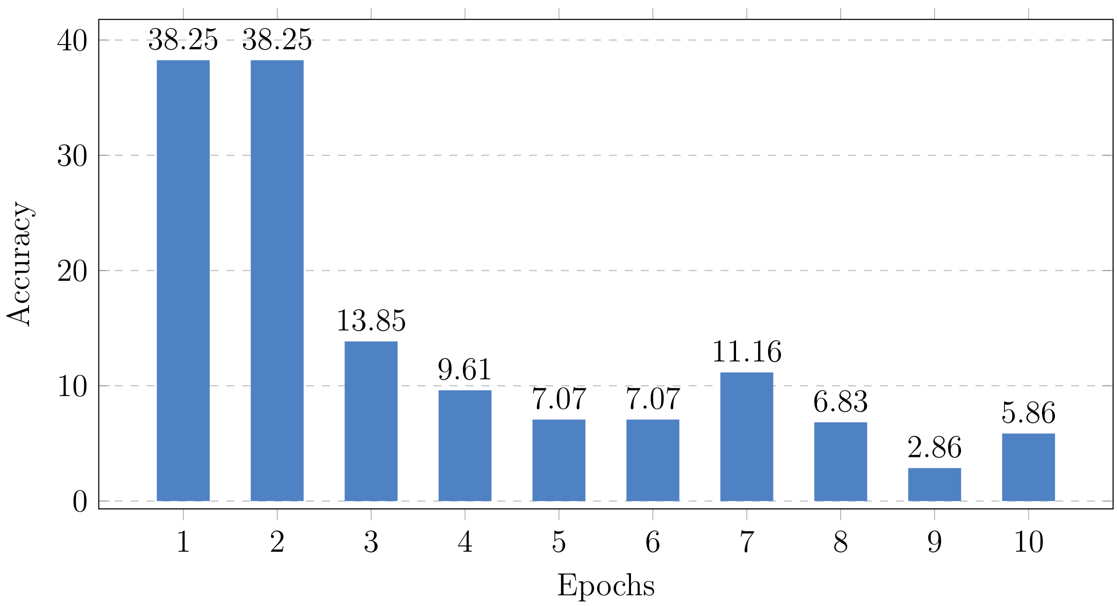

Figure 5.10: Loss Functions of Inception V3

The loss function of the Inception V3 model architecture has indicated in the Figure 5.10. 38.25% was our first loss function. After passing through different epochs, our model managed to obtain a 5.86% loss function in epoch 10.

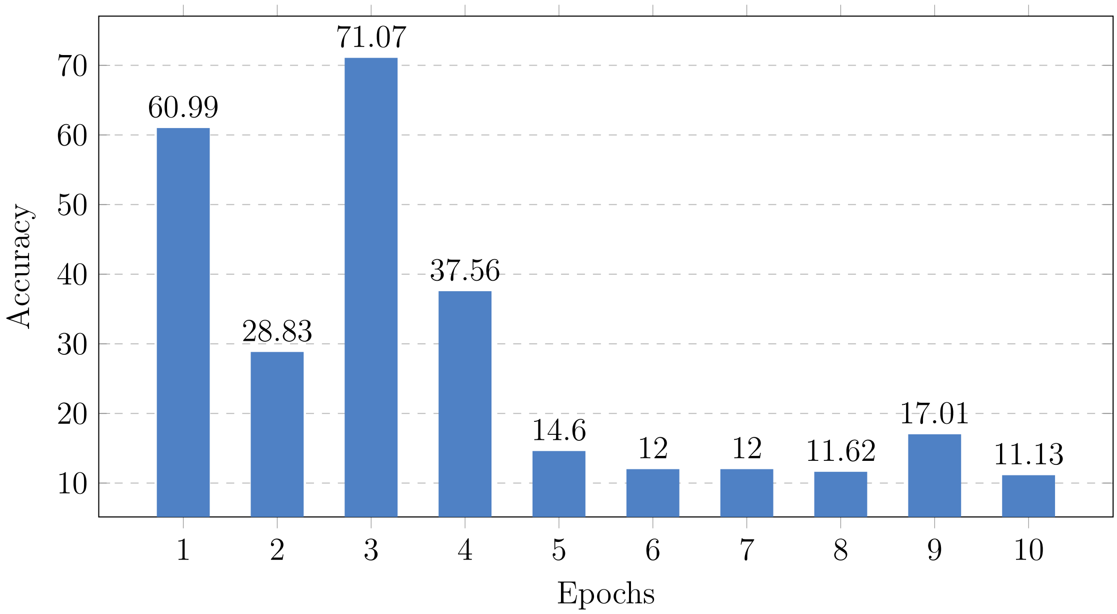

Figure 5.11: Loss Functions of ResNet50

Figure 5.11 represents the loss functions of ResNet50 model architecture. After training dataset, ResNet50 gave the worst loss function compare to other CNN model architectures that we have used. It has started with a 60.99% loss function. In the 3rd epoch, the value increased and the loss function was 74% in this case which was the worst scenario among other CNN models we have used. The lowest loss function we got here is 11.13% in our 10th epoch which was also not pleasant for us.

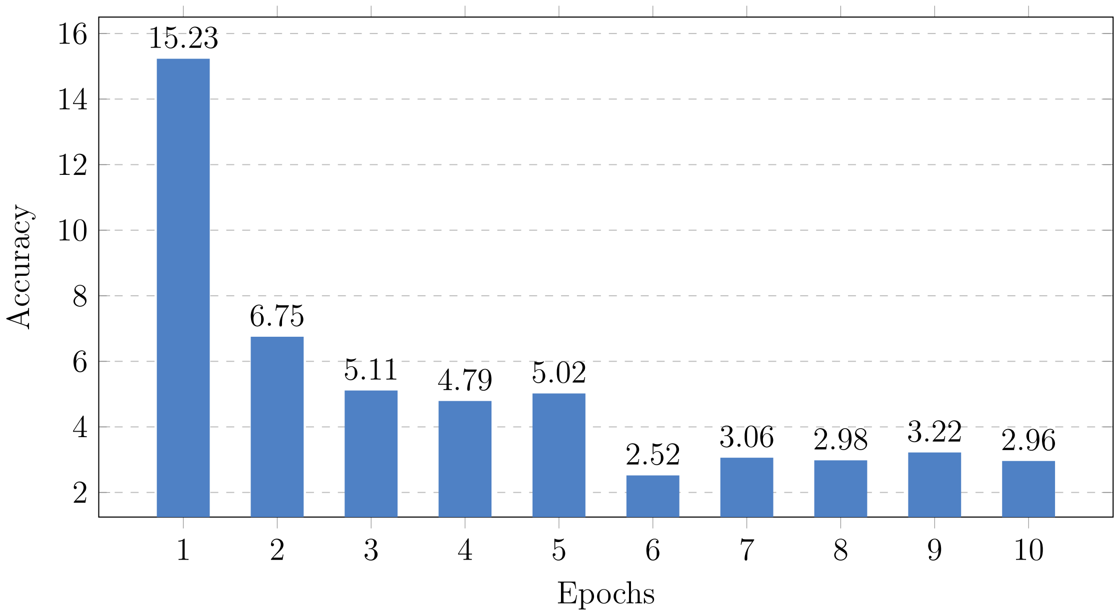

Figure 5.12: Loss Functions of DenseNet121

The loss function of DenseNet121 CNN architecture has shown in the Figure 5.12. In the first epoch, 15.23% was our loss function, and the last one is 2.96%. We obtained our best loss function in the 6th epoch, which was 2.52%.

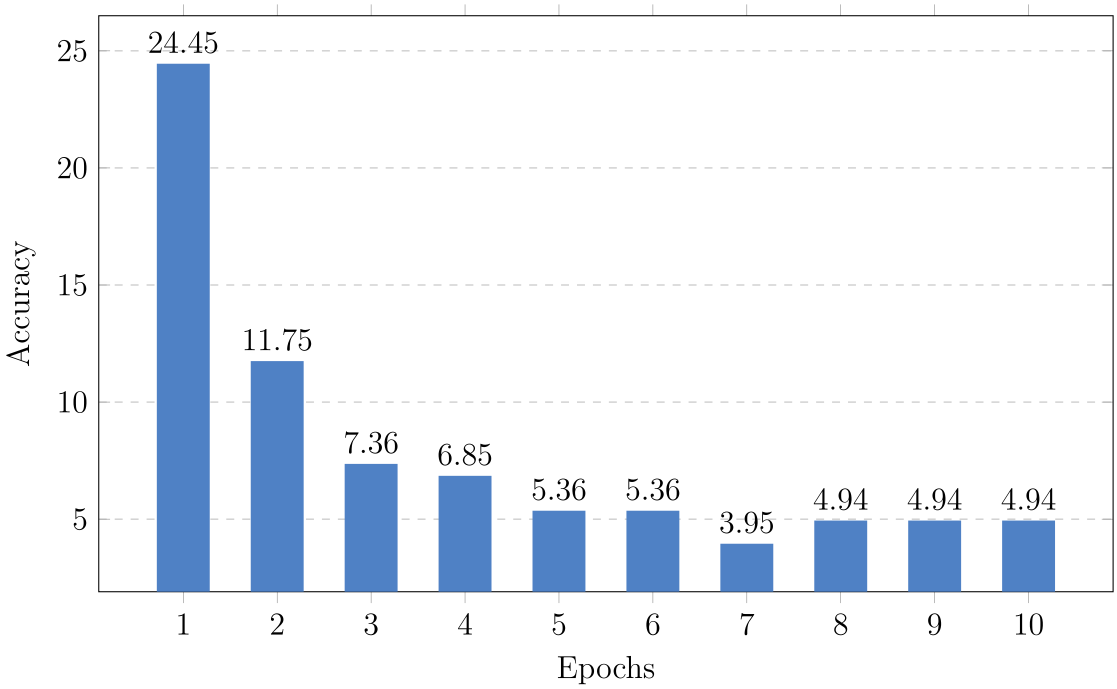

Figure 5.13: Loss Functions of Xception

In the Figure 5.13, we can see the loss functions of our Xception model. It has started with 24.45% loss function then managed to reduced up to 4.94% in the 10th epoch. For this model the best loss function was 3.95% that we got in our 7th epoch.

Table 5.2: Loss Functions of CNN Models

X-Values	**VGG16**	**VGG19**	**Inception V3**	**ResNet50**	**DenseNet121**	**Xception**
Epoch 1	33.5	33.49	38.25	60.99	15.23	24.45
Epoch 2	10.71	12.74	38.25	28.83	6.75	11.75
Epoch 3	7.18	7.81	13.85	71.07	5.11	7.36
Epoch 4	4.79	5.48	9.61	37.56	4.79	6.85
Epoch 5	3.45	3.8	7.07	14.6	5.02	5.36
Epoch 6	3.09	2.58	7.07	12	2.52	5.36
Epoch 7	2.16	2.07	11.16	12	3.06	3.95
Epoch 8	1.69	1.71	6.83	11.62	2.98	4.94
Epoch 9	1.16	2.89	2.86	17.01	3.22	4.94
Epoch 10	1.5	1	5.86	11.13	2.96	4.94

Table 5.2 showing the loss functions of every CNN model of every epoch. We can compare which model is performing well on which epoch by going through this table. The lowest loss functions we got is 1% in the 10th epoch of VGG19. So, for learning the training dataset, the VGG19 model performs better than other CNN model architectures.

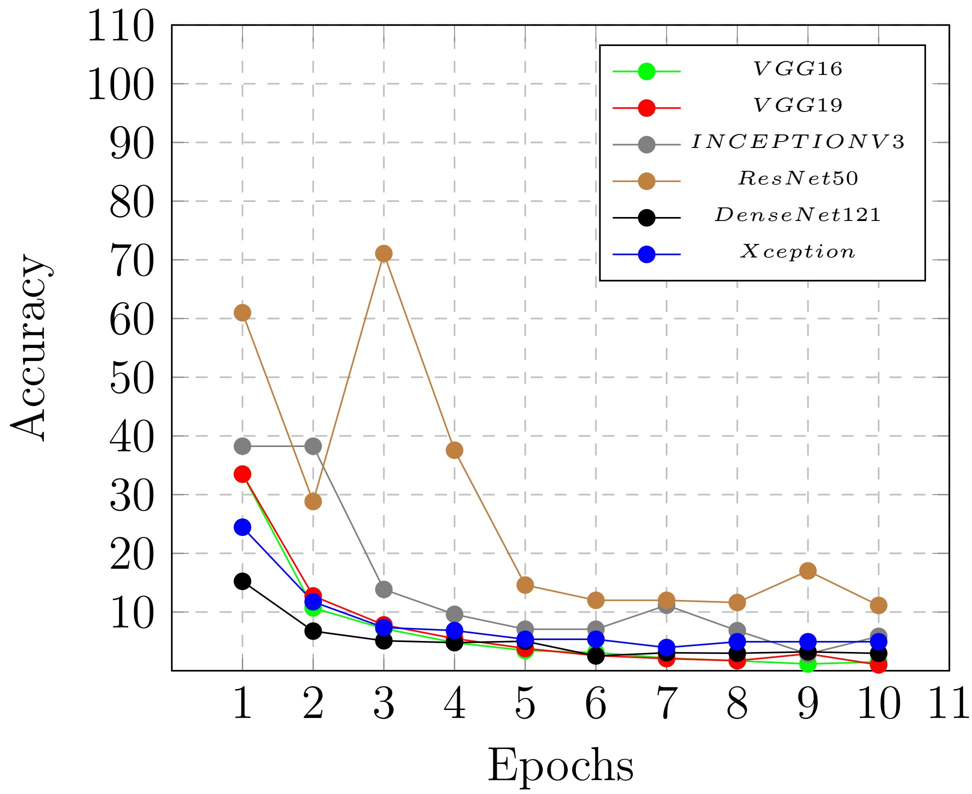

Figure 5.14: Comparison of Loss Functions of Our Used CNN Models

Figure 5.14 represents the comparison of the loss functions of our CNN models. Apart from ResNet50 and DenseNet121 every CNN model started from almost the same place. However, in the 3rd epoch, there was an unexpected increase in the loss function of ResNet50. This model performed worst during the whole training time. Rather than this model, other models did a deficient percentage in the loss functions.

5.3 CNN Model Test Accuracies

The test accuracy is to evaluate the performance of the model. After training and validation, we need to determine which one of our CNN model architectures are performing best for detecting the brain tumor tissue of the patients. We gave new data for determining the test accuracy of the CNN models. The models compared the new data with the previous training data and predicted detecting tumors. We have run our CNN models for ten epochs here [24].

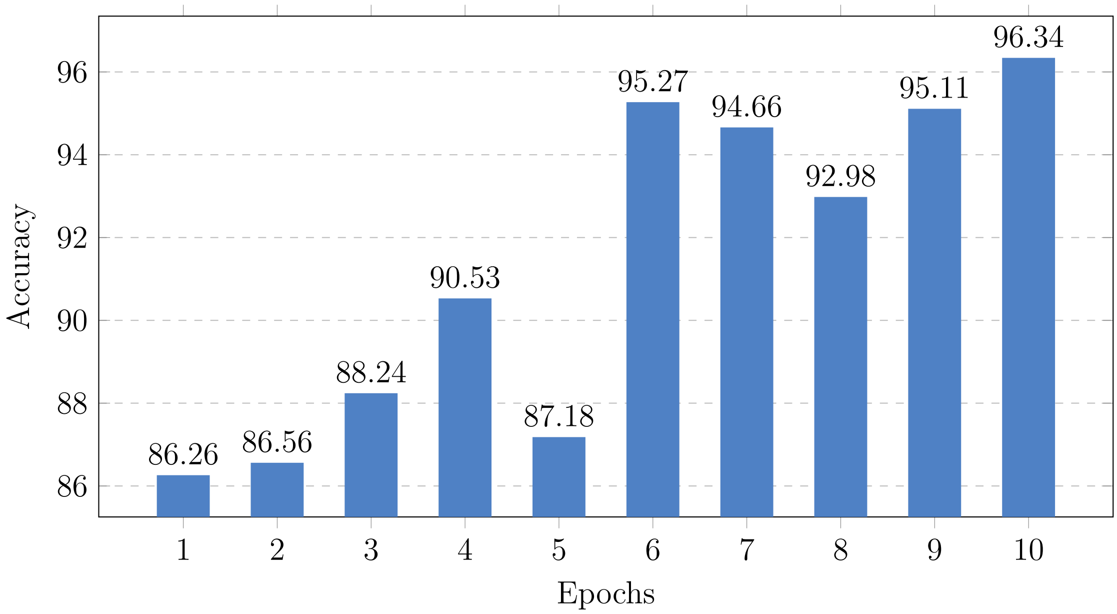

Figure 5.15: VGG16 Test Accuracy

In the Figure 5.15, we have shown the test accuracy of our CNN architecture model VGG16. We executed ten epochs here. We set batch size 16 and per step of the epoch was total image/batch size. That means we have taken 280 images per epoch. In the first epoch, the accuracy was 86.26%, and that has gradually upgraded one after another. The best accuracy we have found in the 10th epoch. It was 96.34%.

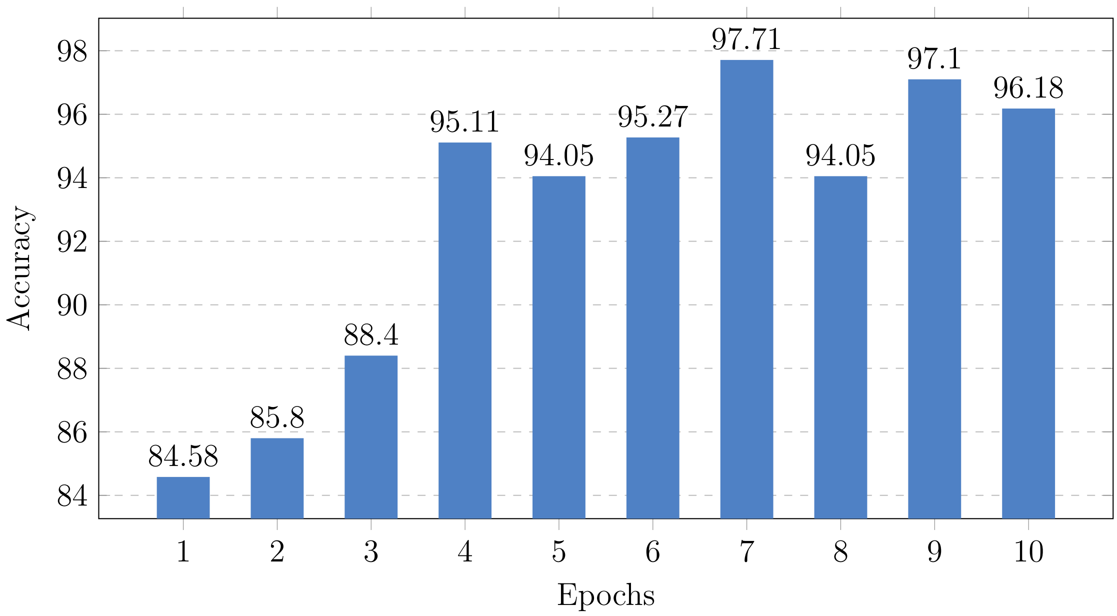

Figure 5.16: VGG19 Test Accuracy

In the Figure 5.16, we have given the test accuracy of detecting brain tumors by our CNN model architecture VGG19. Like our previous model, we have executed ten epochs. The accuracy was started at 84.58%, which was not satisfactory for us. But after passing some epoch, we have seen that the accuracy has increased. The best accuracy we obtained in epoch number 7, which is 97.71%. It means that this model performs good enough for detecting brain tumors of the patients.

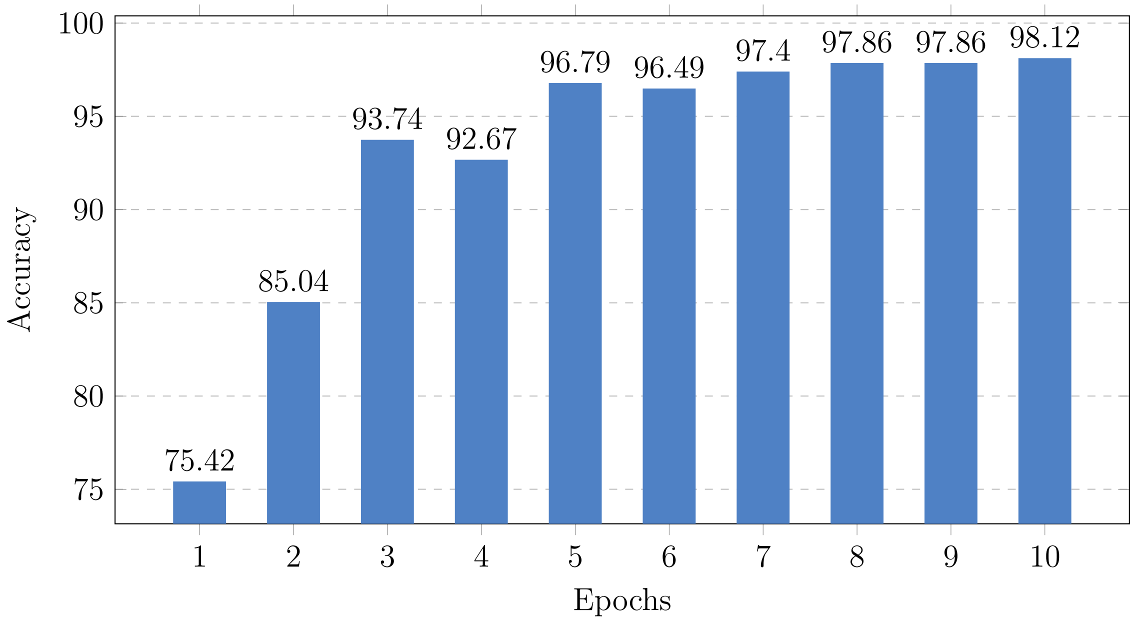

Figure 5.17: Inception V3 Test Accuracy

In the Figure 5.17, we have shown the test accuracy of our CNN model Inception V3. We started with 75.42% accuracy. Though our starting was not good but after finishing ten epochs, we obtained 98.12% accuracy in 10th epoch. This model gives the best result compare to any other CNN model we have used in our study.

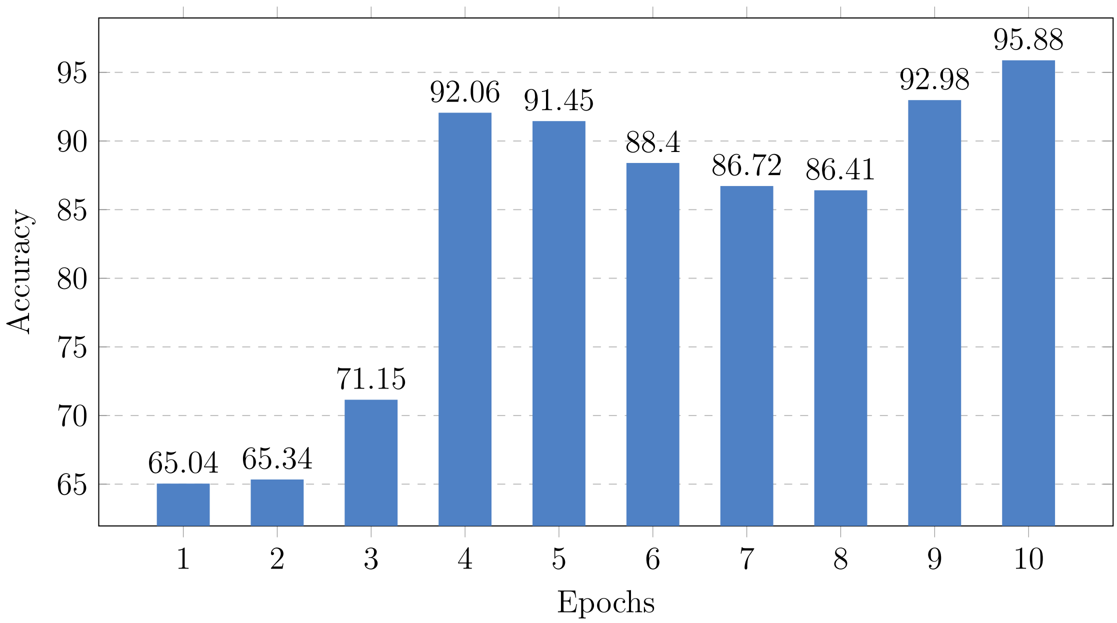

Figure 5.18: ResNet50 Test Accuracy

In the Figure 5.18, we can see the test accuracy of ResNet50. In this CNN model architecture, we obtained just 65.04% accuracy at the very beginning. But, after going through several more epochs, we saw an increase in the testing accuracy. After finishing ten epochs, we obtained 95.88% accuracy which was the highest accuracy for this model.

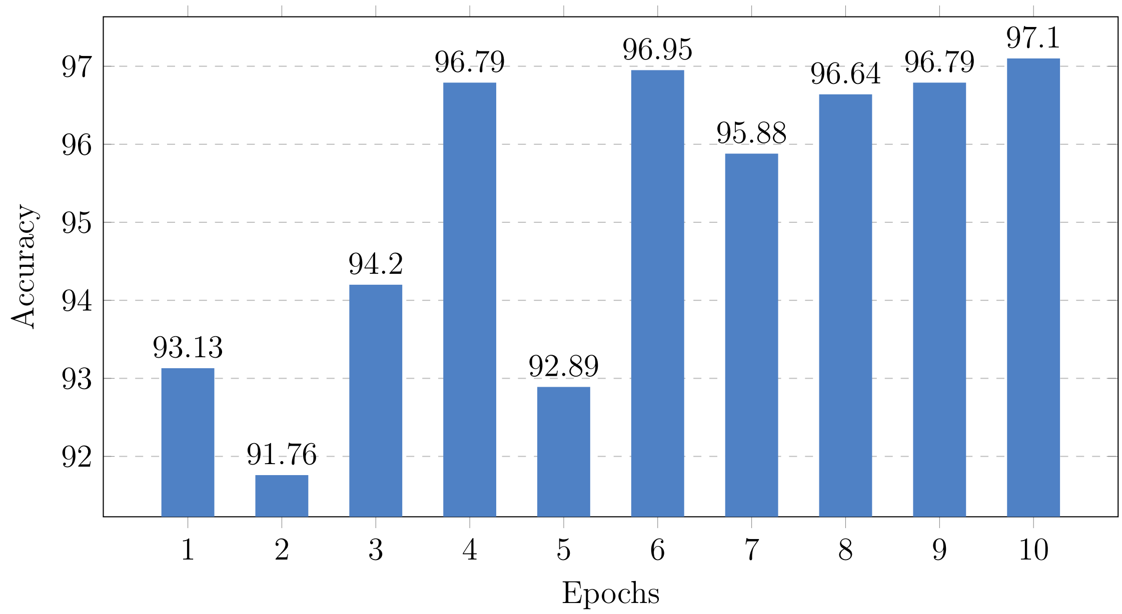

Figure 5.19: DenseNet121 Test Accuracy

In the Figure 5.19, we have shown the test accuracy of our CNN architecture model, DensNet121, for detecting the brain tumor tissue. Here, compared to other models accuracy is higher from the beginning. The first epoch has given us 93.13% accuracy. In this model the accuracy was always more than 91%. At the end, we got the highest accuracy of 97.1% in epoch number 10.

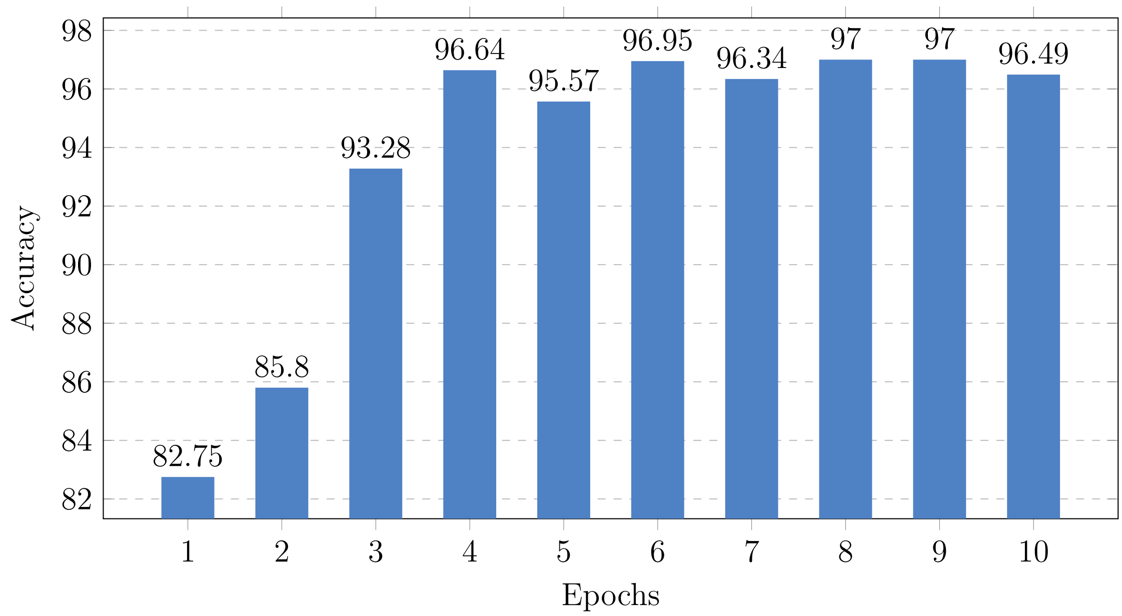

Figure 5.20: Xception Test Accuracy

In the Figure 5.20, we can observe the test accuracy of our last CNN model Xception. The accuracy was started at 82.75% and ended up with 96.49%. We have observed all the accuracies of our model and compared all by making a line chart. There we reach the test accuracies of our one model with another.

Table 5.3: Test Accuracies of CNN Models

X-Values	VGG16	VGG19	Inception V3	ResNet50	DenseNet121	Xception
Epoch 1	86.26	84.58	75.42	65.04	93.13	82.75
Epoch 2	86.56	85.8	85.04	65.34	91.76	85.8
Epoch 3	88.24	88.4	93.74	71.15	94.2	93.28
Epoch 4	90.53	95.11	92.67	92.06	96.79	96.64
Epoch 5	87.18	94.05	96.79	91.45	82.98	95.57
Epoch 6	95.27	95.27	96.49	88.4	96.95	96.95
Epoch 7	94.66	97.71	97.4	86.72	95.88	96.34
Epoch 8	92.98	94.04	97.86	86.41	96.64	97
Epoch 9	95.11	97.1	97.86	92.98	96.79	97
Epoch 10	96.34	96.18	98.12	95.88	97.1	96.49

Table 5.3 represents the test accuracies of our CNN model for every epoch. We can see the comparison of our CNN model with one another in each epoch here. In the 10th epoch of Inception V3, we obtained 98.12% test accuracy, which is the best one. The worst result we have found in the ResNet50 model.

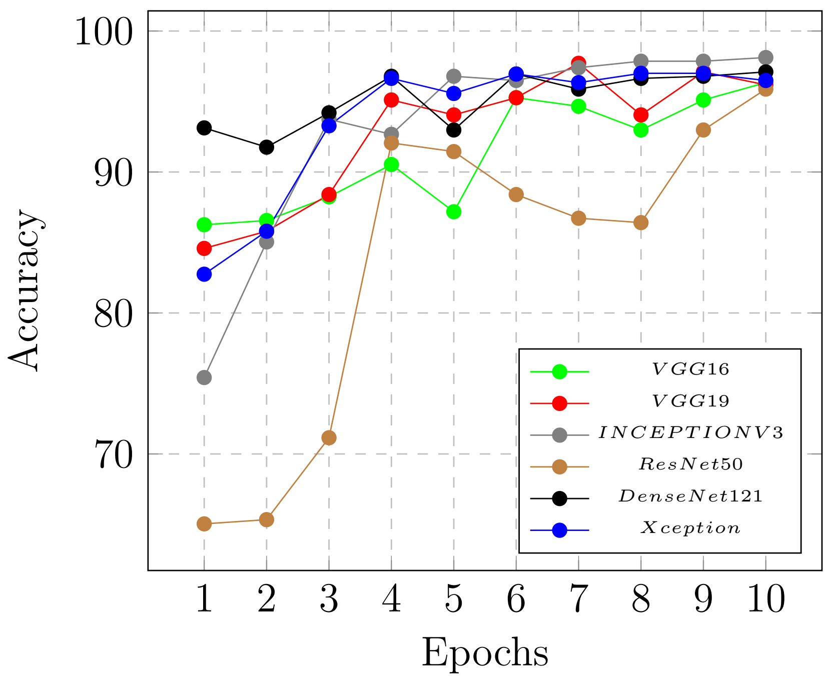

Figure 5.21: Comparison of Test Accuracies of Our CNN Model Architectures

In the Figure 5.21, we can see the comparison between our 6 CNN architectures for test accuracies. We can determine which one is performing best among VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception on each epoch. In the begining, the lowest accuracy was in ResNet50 which was about almost 65% but at the end, it has managed to obtain about 96 % accuracy. To conclude, we have found the gray line which is Inception V3 model, provides the highest accuracy in 10th epoch, which is about 98%.

5.4 Ensemble Method

Here, we made prediction on our test dataset. Then, we got accuracy of our six CNN model architectures. There were 655 images, 229 images have brain tumor and 426 images have no brain tumor. We have taken all these images and generated accuracy. Among these six models, we have taken the best three CNN models and did voting ensemble. When the best three or two CNN models giving 'yes' votes, we have determined that there presents a tumor in that image otherwise, there is no tumor. And, when three or more than two CNN models were giving 'no' votes then, we have determined that, there is 'no' tumor in this brain tumor patient.

Table 5.4: Finding best 3 CNN Models for doing Voting Ensembling

VGG16 (YES)	**VGG19 (YES)**	**Inception V3 (YES)**	**ResNet50 (YES)**	**DensNet121 (YES)**	**Xception (YES)**
0	0	0	0	0	0
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	1	0
1	1	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	0	1	1
1	1	1	1	1	1
... ...					
... ...					
Table of 'No Brain' Tumor					
(NO)	**(NO)**	**(NO)**	**(NO)**	**(NO)**	**(NO)**
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	0	0	0
1	0	0	0	0	0
0	0	0	1	0	0
... ...					
... ...					
Accuracy					
96.49	96.79	98.02	95.57	97.09	96.64

Table 5.4 shows the prediction for finding best three CNN models based on our test dataset. We had taken all 655 images from our test dataset and we did prediction manually. So, there has showed just 20 predicted data in this section. We did separate table for having tumor or not having any tumor in the images. In the upper table, there is the predicted value of having brain tumor tissue. '1' represents that, there has tumor in that image but sometimes we have found '0' in that table which represents there has no tumor. Similarly, the next table is for 'No' class of brain tumor tissue. Here, also we have got some '1' values which is telling us that there has tumor in that image. So, we made an accuracy test for all six CNN models and compared with each other. Thus, we have selected 3 best CNN models for doing our voting ensemble which are Inception V3 with 98.02% accuracy, DenseNet121 with 97.02% accuracy and VGG19 with 96.79% accuracy.

Table 5.5: Voting Ensemble Data

VGG19 (YES)	**Inception V3 (YES)**	**DensNet121 (YES)**	**Voting Ensemble (YES)**
0	0	0	0
1	1	1	1
1	1	1	1
1	1	1	1
1	1	0	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	0	1	1
... ...			
... ...			
Table of 'No Brain' Tumor			
(NO)	**(NO)**	**(NO)**	**(NO)**
0	0	0	0
1	0	0	0
0	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
... ...			
... ...			
Accuracy			
96.79	98.02	97.09	98.17

In the 5.5 table, we have showed the result of our voting ensemble. Here, we have taken the best three CNN models from our previous comparison based on the test accuracy. Now, in the upper 'YES' part, when we got at-least two '1' from our models, on that time, we have given '1' in the Voting Ensemble section otherwise we gave '0'. In the lower table, which is for 'NO' part, when we got at-least two '0', we have written '0' in the Voting Ensemble section otherwise we wrote '1'. We predicted like this for all images on our test dateset manually one by one then, calculated our test accuracy for Voting Ensemble. This time, we got 98.17% accuracy which was the best result we have found so far. Thus, we can conclude that, when we make a decision best on the best CNN models then we can detect brain tumor more accurately.

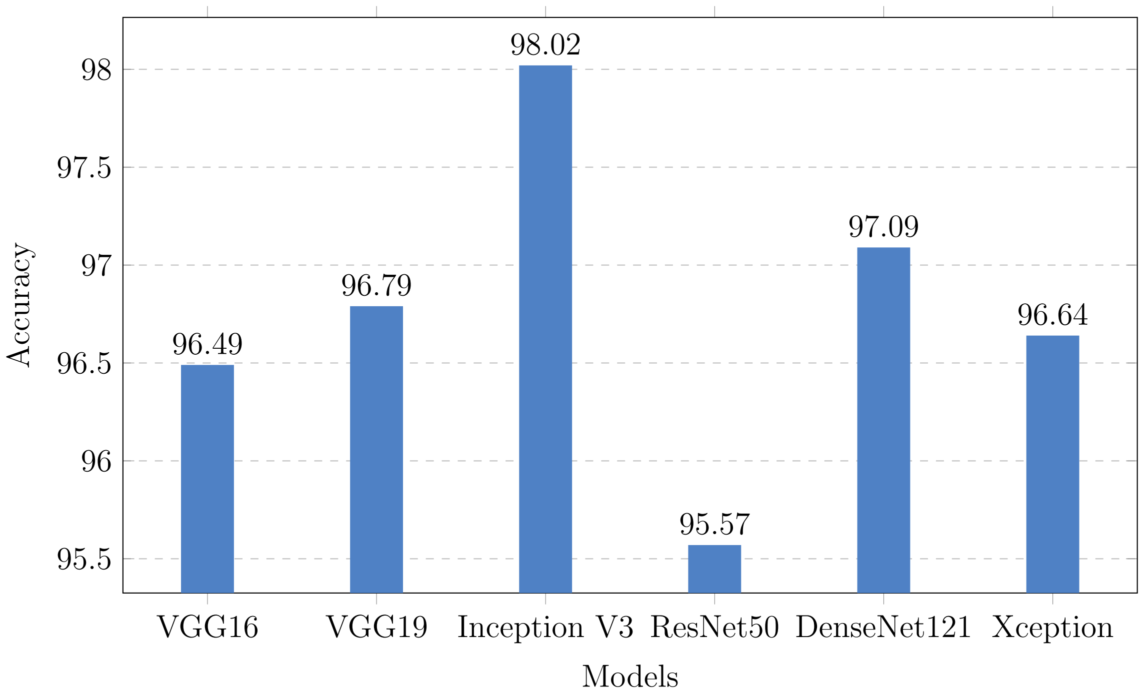

Figure 5.22: Predicted Test Accuracy for finding best CNN Models

In the Figure 5.22, we have shown the predicted test accuracy of all six CNN models. This accuracy is not similar to our previously analysed result where we have compared the accuracy and loss functions for 10 epochs. Here, after getting the predicted values from test dataset which are '0' and '1' we calculated accuracy manually. Then, we tried to find our the best three models. From this figure, we can easily determine that the best three models are Inception V3 which has accuracy of 98.02 %, DenseNet121 which has accuracy of 97.09 % and VGG19 which has accuraccy of 96.79 % .

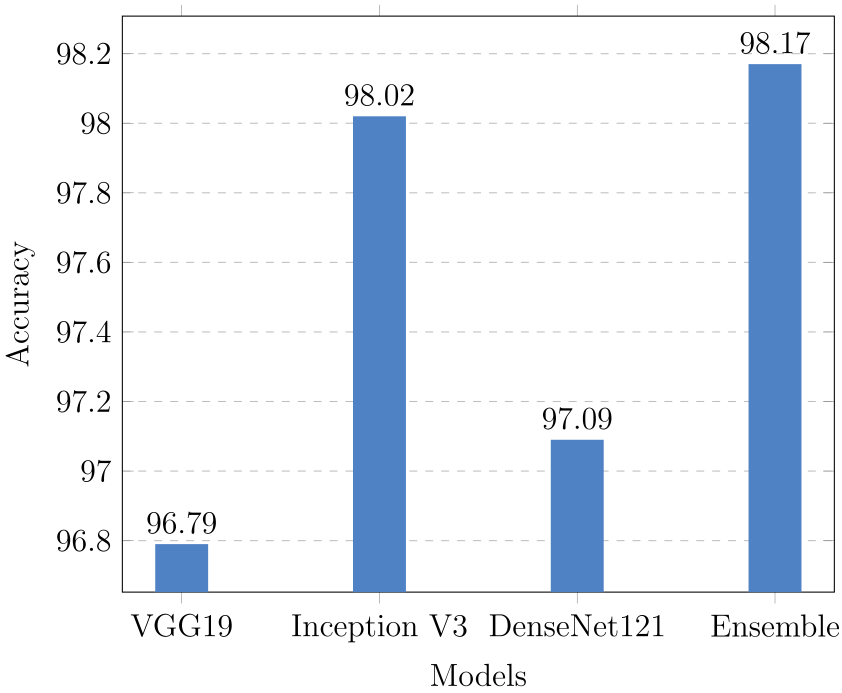

Figure 5.23: Test Accuracy Comparison between best 3 CNN Models and Voting Ensemble

In the 5.23, along with the voting ensemble test accuracy it represents the individual three best CNN models test accuracy as well. Here, the accuracy of Inception V3 is 98.02% which is higher than DenseNet121 and VGG19. However, the accuracy of Ensemble is 98.17% which is even higher than the Inception V3 model. From our research, after comparing individual model with the ensemble one we can conclude by stated that, ensemble method is more efficient than using single classifier as well as ensemble method produce better predictive result with higher accuracy in terms of brain tumor detection.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Our primary purpose is, to build a system that will detect brain tumors at an early stage so that, people can take proper treatment from the very beginning. In this research, we have worked with fMRI brain imaging data and Convolutional Neural Network architecture. We did binary classification, which detects either there has any brain tumor or not. As all the data were in 3D NIfTI images so, after collecting the data, we did pre-processing, which includes image conversion, data segmentation, data augmentation, and data normalization. Then, we separated our data into a training dataset, validation dataset, and testing dataset. After that, we did data training, validation, testing and for that purpose, we have used the TensorFlow library and Keras library. We have performed six different types of CNN architectures VGG16, VGG19, Inception V3, ResNet50, DenseNet121, and Xception. These models are already pre-trained with millions of ImageNet images. So, we do not need to create a CNN model on our own. But we did not take the Dense layer and fully connected layer of the CNN models, so we made our Dense layer with 1024 neurons and fully connected layer with two classes. These CNN architectures gave us different types of results and accuracy when we tested them. Based on the test accuracy, we did comparison between these six models and we have found 3 best models: Inception V3, DenseNet121 and VGG19. After that, we did voting ensemble manually. For that, we did binary value of Test dataset. '0' means 'No' tumor and '1' means there has tumor. After analyzing the results of those models, we have taken Inception V3, DenseNet121 and VGG19 as they managed to produce higher accuracy among all our CNN models. Then, we performed ensemble method with those 3 models. When our best 3 models giving more than or equal to 2 'Yes' votes on that time, we told that there presents a tumor in the brain. Our study showed that, the accuracy we achieved after applying ensemble technique, it was better than the individual accuracy provided by the CNN models.

6.2 Future Work

In our study we have applied ensemble method and in future we have plan to work with more data as well as more models so that the ensemble learning can generate more accuracy in terms of brain tumor detection. Apart from those models that we have already implemented in our study, we will try to use different CNN models to check either it can give us better accuracy and solutions or not. Moreover, we would like to add some features that will show us the result of brain tumor along with the location and size of the tumor. Additionally, we have plan to detect the tumor type as well either it is malignant or benign tumor. Furthermore, we would like to distinguish between high-grade glioma and low-grade glioma. To improve accuracy, we are planning to use some algorithms along with CNN model architecture. After that, we want to extend our work to detect different types of tumor tissue in the whole human body.

References

[1] D. Valentino, J. Mazziotta, and H. Huang, “Visualization of human brain structure-function relationships,” in *Images of the Twenty-First Century. Proceedings of the Annual International Engineering in Medicine and Biology Society,*, IEEE, 1989, pp. 1737–1738.

[2] S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank, “Brain magnetic resonance imaging with contrast dependent on blood oxygenation,” *proceedings of the National Academy of Sciences*, vol. 87, no. 24, pp. 9868–9872, 1990.

[3] P. A. Bandettini, E. C. Wong, R. S. Hinks, R. S. Tikofsky, and J. S. Hyde, “Time course epi of human brain function during task activation,” *Magnetic resonance in medicine*, vol. 25, no. 2, pp. 390–397, 1992.

[4] K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weisskoff, B. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, and R. Turner, “Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation.,” *Proceedings of the National Academy of Sciences*, vol. 89, no. 12, pp. 5675–5679, 1992.

[5] J. Martin, A. Pentland, and R. Kikinis, “Shape analysis of brain structures using physical and experimental modes,” in *CVPR*, 1994, pp. 752–755.

[6] W. S. Sarle, “Neural networks and statistical models,” 1994.

[7] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[8] T. Dietterich, “Ensemble learning. the handbook of brain theory and neural networks,” *Arbib MA*, 2002.

[9] N. K. Logothetis, “The underpinnings of the bold functional magnetic resonance imaging signal,” *Journal of Neuroscience*, vol. 23, no. 10, pp. 3963–3971, 2003.

[10] Y. Zang, T. Jiang, Y. Lu, Y. He, and L. Tian, “Regional homogeneity approach to fmri data analysis,” *Neuroimage*, vol. 22, no. 1, pp. 394–400, 2004.

[11] L. Tian, T. Jiang, Y. Wang, Y. Zang, Y. He, M. Liang, M. Sui, Q. Cao, S. Hu, M. Peng, *et al.*, “Altered resting-state functional connectivity patterns of anterior cingulate cortex in adolescents with attention deficit hyperactivity disorder,” *Neuroscience letters*, vol. 400, no. 1-2, pp. 39–43, 2006.

[12] G. Beattie, “What we know about how the human brain works,” in *How public service advertising works*, World Advertising Research Center, 2008.

[13] C. M. Chen, B. L. Hou, and A. I. Holodny, “Effect of age and tumor grade on bold functional mr imaging in preoperative assessment of patients with glioma,” *Radiology*, vol. 248, no. 3, pp. 971–978, 2008.

[14] C. R. Cloninger, "Evolution of human brain functions: The functional structure of human consciousness," *Australian & New Zealand Journal of Psychiatry*, vol. 43, no. 11, pp. 994–1006, 2009.

[15] K. M. Iftekharuddin, J. Zheng, M. A. Islam, and R. J. Ogg, "Fractal-based brain tumor detection in multimodal mri," *Applied Mathematics and Computation*, vol. 207, no. 1, pp. 23–41, 2009.

[16] Z.-H. Zhou, *Ensemble learning. encyclopedia of biometrics*, 2009.

[17] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[18] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, 2012.

[19] N. Zulpe and V. Pawar, "Glcm textural features for brain tumor classification," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 3, p. 354, 2012.

[20] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv preprint arXiv:1307.0411*, 2013.

[21] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[22] S. Budday, P. Steinmann III, and E. Kuhl, "Physical biology of human brain development," *Frontiers in cellular neuroscience*, vol. 9, p. 257, 2015.

[23] S. B. Gaikwad and M. S. Joshi, "Brain tumor classification using principal component analysis and probabilistic neural network," *International Journal of Computer Applications*, vol. 120, no. 3, 2015.

[24] Y. Pan, W. Huang, Z. Lin, W. Zhu, J. Zhou, J. Wong, and Z. Ding, "Brain tumor grading based on neural networks and convolutional neural networks," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 699–702.

[25] K. Vaishnavee and K. Amshakala, "An automated mri brain image segmentation and tumor detection using som-clustering and proximal support vector machine classifier," in *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, IEEE, 2015, pp. 1–6.

[26] L. Zhao and K. Jia, "Deep feature learning with discrimination mechanism for brain tumor segmentation and diagnosis," in *2015 international conference on intelligent information hiding and multimedia signal processing (IIH-MSP)*, IEEE, 2015, pp. 306–309.

[27] A. Casamitjana, S. Puch, A. Aduriz, and V. Vilaplana, "3d convolutional neural networks for brain tumor segmentation: A comparison of multi-resolution architectures," in *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer, 2016, pp. 150–161.

[28] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in mri images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.

[29] S. Sarraf and G. Tofighi, "Deep learning-based pipeline to recognize alzheimer's disease using fmri data," in *2016 Future Technologies Conference (FTC)*, IEEE, 2016, pp. 816–820.

[30] G. Singh and M. Ansari, "Efficient detection of brain tumor from mris using k-means segmentation and normalized histogram," in *2016 1st India International Conference on Information Processing (IICIP)*, IEEE, 2016, pp. 1–6.

[31] M. M. Subashini, S. K. Sahoo, V. Sunil, and S. Easwaran, "A non-invasive methodology for the grade identification of astrocytoma using image processing and artificial intelligence techniques," *Expert systems with Applications*, vol. 43, pp. 186–196, 2016.

[32] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[33] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, 2017, pp. 1–6.

[34] N. B. Bahadure, A. K. Ray, and H. P. Thethi, "Image analysis for mri based brain tumor detection and feature extraction using biologically inspired bwt and svm," *International journal of biomedical imaging*, vol. 2017, 2017.

[35] N. M. Balasooriya and R. D. Nawarathna, "A sophisticated convolutional neural network model for brain tumor classification," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, IEEE, 2017, pp. 1–5.

[36] S. Chauhan, A. More, R. Uikey, P. Malviya, and A. Moghe, "Brain tumor detection and classification in mri images using image and data mining," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, IEEE, 2017, pp. 223–231.

[37] T. Chithambaram and K. Perumal, "Brain tumor segmentation using genetic algorithm and ann techniques," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, 2017, pp. 970–982.

[38] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[39] V. Dhanve and M. Kumar, "Detection of brain tumor using k-means segmentation based on object labeling algorithm," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, 2017, pp. 944–951.

[40] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, pp. 18–31, 2017.

[41] A. Hazra, A. Dey, S. K. Gupta, and M. A. Ansari, "Brain tumor detection based on segmentation using matlab," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, IEEE, 2017, pp. 425–430.

[42] T. Horikawa and Y. Kamitani, "Generic decoding of seen and imagined objects using hierarchical visual features," *Nature communications*, vol. 8, no. 1, pp. 1–15, 2017.

[43] S. Hussain, S. M. Anwar, and M. Majid, "Brain tumor segmentation using cascaded deep convolutional neural network," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2017, pp. 1998–2001.

[44] N. Kaur and M. Sharma, "Brain tumor detection using self-adaptive k-means clustering," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, IEEE, 2017, pp. 1861–1865.

[45] A. Min and Z. M. Kyu, "Mri images enhancement and tumor segmentation for brain," in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, IEEE, 2017, pp. 270–275.

[46] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes, "Big data analysis for brain tumor detection: Deep convolutional neural networks," *Future Generation Computer Systems*, vol. 87, pp. 290–297, 2018.

[47] S. Banerjee, S. Mitra, F. Masulli, and S. Rovetta, "Brain tumor detection and classification from multi-sequence mri: Study using convnets," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 170–179.

[48] N. Boughattas, M. Berar, K. Hamrouni, and S. Ruan, "Feature selection and classification using multiple kernel learning for brain tumor segmentation," in *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, IEEE, 2018, pp. 1–5.

[49] L. Dong, C. Luo, X. Liu, S. Jiang, F. Li, H. Feng, J. Li, D. Gong, and D. Yao, "Neuroscience information toolbox: An open source toolbox for eeg–fmri multimodal fusion analysis," *Frontiers in Neuroinformatics*, vol. 12, p. 56, 2018.

[50] S. Hussain, S. M. Anwar, and M. Majid, "Segmentation of glioma tumors in brain using deep convolutional neural network," *Neurocomputing*, vol. 282, pp. 248–261, 2018.

[51] R. Mehta and T. Arbel, "3d u-net for brain tumour segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 254–266.

[52] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 311–320.

[53] S. Ouchtati, J. Sequeira, B. Aissa, R. Djemili, and M. Lashab, "Brain tumors classification from mr images using a neural network and the central moments," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, IEEE, 2018, pp. 455–460.

[54] F. Polly, S. Shil, M. Hossain, A. Ayman, and Y. Jang, "Detection and classification of hgg and lgg brain tumor using machine learning," in *2018 International Conference on Information Networking (ICOIN)*, IEEE, 2018, pp. 813–817.

[55] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2018, pp. 169–175.

[56] M. I. Razzak, M. Imran, and G. Xu, "Efficient brain tumor segmentation with multiscale two-pathway-group conventional neural networks," *IEEE journal of biomedical and health informatics*, vol. 23, no. 5, pp. 1911–1919, 2018.

[57] D. R. Schrider and A. D. Kern, "Supervised machine learning for population genetics: A new paradigm," *Trends in Genetics*, vol. 34, no. 4, pp. 301–312, 2018.

[58] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecological Informatics*, vol. 48, pp. 257–268, 2018.

[59] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 61–72.

[60] Z. Zhao, G. Yang, Y. Lin, H. Pang, and M. Wang, "Automated glioma detection and segmentation using graphical models," *PloS one*, vol. 13, no. 8, e0200745, 2018.

[61] 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*, Springer, 2019, pp. 183–189.

[62] P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Capsule networks for brain tumor classification based on mri images and coarse tumor boundaries," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 1368–1372.

[63] R. Carter, *The human brain book: An illustrated guide to its structure, function, and disorders*. Penguin, 2019.

[64] S. Chen, C. Ding, and M. Liu, "Dual-force convolutional neural networks for accurate brain tumor segmentation," *Pattern Recognition*, vol. 88, pp. 90–100, 2019.

[65] W. Du, S. Li, and Z. Wang, "Research on the human brain, the external brain and the public external brain," in *Journal of Physics: Conference Series*, vol. 1168, 2019.

[66] C. Han, K. Murao, T. Noguchi, Y. Kawata, F. Uchiyama, L. Rundo, H. Nakayama, and S. Satoh, "Learning more with less: Conditional pggan-based data augmentation for brain metastases detection using highly-rough annotation on mr images," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 119–127.

[67] S. Hanwat and J. Chandra, "Convolutional neural network for brain tumor analysis using mri images," *International Journal of Engineering and Technology (IJET)*, vol. 11, pp. 67–77, 2019.

[68] T. C. Hollon, B. Pandian, S. S. S. Khalsa, R. D'Amico, M. B. Sisti, J. N. Bruce, K. M. Muraszko, B. G. Thompson, C. Freudiger, S. L. Hervey-Jumper, *et al.*, "Near real-time intraoperative brain tumor diagnosis using stimulated raman histology and deep neural networks," *Neurosurgery*, vol. 66, no. Supplement_1, nyz310_634, 2019.

[69] J. Nalepa, M. Marcinkiewicz, and M. Kawulok, "Data augmentation for brain-tumor segmentation: A review," *Frontiers in Computational Neuroscience*, vol. 13, 2019.

[70] A. Rehman, S. Naz, and M. I. Razzak, "Writer identification using machine learning approaches: A comprehensive review," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 10 889–10 931, 2019.

[71] S. Sajid, S. Hussain, and A. Sarwar, "Brain tumor detection and segmentation in mr images using deep learning," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9249–9261, 2019.

[72] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep cnn with extensive data augmentation," *Journal of computational science*, vol. 30, pp. 174–182, 2019.

[73] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146 533–146 541, 2019.

[74] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[75] A. M. Alqudah, H. Alquraan, I. A. Qasmieh, A. Alqudah, and W. Al-Sharu, "Brain tumor classification using deep learning technique–a comparison between cropped, uncropped, and segmented lesion images with different sizes," *arXiv preprint arXiv:2001.08844*, 2020.

[76] M. M. Badža and M. Č. Barjaktarović, "Classification of brain tumors from mri images using a convolutional neural network," *Applied Sciences*, vol. 10, no. 6, p. 1999, 2020.

[77] A. Çinar and M. Yıldırım, "Detection of tumors on brain mri images using the hybrid convolutional neural network architecture," *Medical Hypotheses*, p. 109 684, 2020.

[78] C. Han, L. Rundo, R. Araki, Y. Furukawa, G. Mauri, H. Nakayama, and H. Hayashi, "Infinite brain mr images: Pggan-based data augmentation for tumor detection," in *Neural approaches to dynamics of signal exchanges*, Springer, 2020, pp. 291–303.

[79] S. A. A. Ismael, A. Mohammed, and H. Hefny, "An enhanced deep learning approach for brain cancer mri images classification using residual networks," *Artificial Intelligence in Medicine*, vol. 102, p. 101 779, 2020.

[80] T. Kalaiselvi, T. Padmapriya, P. Sriramakrishnan, and V. Priyadharshini, "Development of automatic glioma brain tumor detection system using deep convolutional neural networks," *International Journal of Imaging Systems and Technology*, 2020.

[81] A. Naseer, M. Rani, S. Naz, M. I. Razzak, M. Imran, and G. Xu, "Refining parkinson's neurological disorder identification through deep transfer learning," *Neural Computing and Applications*, vol. 32, no. 3, pp. 839–854, 2020.

[82] *Quick brain tumor facts*, Apr. 2020. [Online]. Available: https://braintumor.org/brain-tumor-information/brain-tumor-facts/.

[83] M. I. Razzak, M. Imran, and G. Xu, "Big data analytics for preventive medicine," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4417–4451, 2020.

[84] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A deep learning-based framework for automatic brain tumors classification using transfer learning," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 757–775, 2020.

[85] M. Toğaçar, B. Ergen, and Z. Cömert, "Brainmrnet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model," *Medical Hypotheses*, vol. 134, p. 109 531, 2020.

[86] C. Pernet, K. Gorgolewski, and W. Ian, *A neuroimaging dataset of brain tumour patients*. [Online]. Available: http://reshare.ukdataservice.ac.uk/851861/.