VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018, KARNATAKA.

A PROJECT REPORT

On

"IDENTIFICATION AND CLASSIFICATION OF BRAIN TUMOR"

Submitted in Partial Fulfillment for the Award of the Degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

1SG18CS047   **Manisha L**

Under the Guidance of
Prof. Swetha B N
Assistant Professor,
Dept of CSE

Department of Computer Science and Engineering
(Accredited by NBA)
SAPTHAGIRI COLLEGE OF ENGINEERING
(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
14/5, Chikkasandra, Hesarghatta Main Road
Bengaluru – 560057.
2021-2022

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
Bengaluru-560057

Department of Computer Science and Engineering
(Accredited by NBA)

SAPTHAGIRI
College of Engineering

Certificate

Certified that the Project Work entitled **"IDENTIFICATION AND CLASSIFICATION OF BRAIN TUMOR"** carried out by MANISHA L (1SG18CS047), bonafide student of Sapthagiri College of Engineering, in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The project report has been approved as it satisfies the academic requirements in respect of **Project Work Phase II (18CSP83)** prescribed for the said degree.

21/09/2022

Signature of the Guide
Prof. Swetha B N
Assistant Professor

Signature of the HOD
Dr. Kamalakshi Naganna
Professor & Head

Signature of the Principal
Dr. H Ramakrishna
Principal

EXTERNAL EXAMINATION:

Name of the Examiners

1. Dr. Praveen Kumar M

2. Dr. Lavanya N L

Signature with Date

29/9/22

29/9/22

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

I consider it is a privilege to express my gratitude and respect to all those who guided me in completion of project.

I would like to express our profound thanks to **Sri. G Dayanand**, Chairman, Sapthagiri College of Engineering, Bangalore, for his continuous support in providing amenities to carry out this Project work.

Special thanks to **Sri. Manoj G D**, Executive Director, Sapthagiri College of Engineering, Bangalore for his valuable suggestions.

I am, grateful to thank our Principal **Dr. H Ramakrishna**, Sapthagiri College of Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

It's a great privilege to place on record my deep sense of gratitude to our beloved HOD **Dr. Kamalakshi Naganna** of Computer Science & Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

I am also grateful to thank my project Guide Prof. **Swetha B N, Assistant Professor** of CSE department for **her** invaluable support and guidance.

I would also like to thank the teaching and non-teaching staff members who have helped us directly or indirectly during the project work.

Lastly but most importantly I would like thank my family and friends for their co-operation and motivation to complete this project successfully.

<div align="right">

MANISHA L   1SG18CS047

</div>

ABSTRACT

A brain tumor is a disease caused due to the growth of abnormal cells in the brain. There are two main categories of brain tumor, they are non-cancerous (benign) brain tumor and cancerous(malignant) brain tumor. Survival rate of a tumor prone patient is difficult to predict because brain tumor is uncommon and are different types. Treatment for brain tumor depends on various factors like: the type of tumor, how abnormal the cells are and where it is in the brain etc. With the growth of Artificial Intelligence, Deep learning models are used to diagnose the brain tumor by taking the images of magnetic resonance imaging (MRI). This project detects whether the brain tumor is present or not from MRI scans. It uses the CNN classification technique which is used to classify the type of brain tumor present. If the detected tumor is non-cancerous(benign), the further classification of the type of tumor is done. The implementation of the suggested model is applied in the Python and TensorFlow environment. Algorithms and methodologies used to solve specific research problems are included in the results and along with their strengths and limitations. This examines the quantitative characteristics of brain tumors, such as shape, texture, and signal intensity, to predict high accuracy with a low error rate.

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
#14/5, Chikkasandra, Hesaraghatta Main Road,
Bengaluru-560057

Department of Computer Science and Engineering
(Accredited by NBA)

DECLARATION

I **MANISHA L (1SG18CS047),** bonafide student of **Sapthagiri College of Engineering,** hereby declare that the project entitled "**IDENTIFICATION AND CLASSIFICATION OF BRAIN TUMOR**" submitted in partial fulfilment for the award of Bachelor of Engineering in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2021-2022 is our original work and the project has not formed the basis for the award of any other degree, fellowship or any other similar titles.

Name & Signature of the Student with date

1) Manisha L (1SG18CS047)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1   BRIEF INTRODUCTION

Fig 1.1 Different types of brain tumor

The brain contains billions of cells and with lots of complex functions happening every second. When the brain is damaged, there will be many things affected. Unusual development of tissue in the human brain that frustrates the best possible working of the brain is known as the brain tumor. Because of the fact that it will boost the outcome of study in a space where ill-advised bad situations must be at a low point, computer assistance is strongly advanced in therapeutic organizations. The cost of two-dimensional MRI readings is exorbitant, but excellent programming to assist people in clinical settings is of great interest these days.

A brain tumor, known as an intracranial tumor, is an abnormal mass of tissue in which cells grow and multiply uncontrollably, seemingly unchecked by the mechanisms that control normal cells.

The two main groups of brain tumors are termed primary and metastatic. benign brain tumors can cause many serious issues, they are not cancerous, meaning that they grow slowly and don't typically spread to other tissues.

They also usually have more clearly defined borders, making them easier to remove surgically, and they don't usually come back after removal. On the other hand, malignant brain tumors are cancerous, grow rapidly, and can spread to other parts of your brain or central nervous system, which can cause life-threatening complications.

Brain tumors are thought to arise when certain genes on the chromosomes of a cell are damaged and no longer function properly. These genes normally regulate the rate at which the cell divides (if it divides at all) and repair genes that fix defects of other genes, as well as genes that should cause the cell to self-destruct if the damage is beyond repair. In some cases, an individual may be born with partial defects in one or more of these genes. Environmental factors may then lead to further damage. In other cases, the environmental injury to the genes may be the only cause. It is not known why some people in an "environment" develop brain tumors, while others do not.

Once a cell is dividing rapidly and internal mechanisms to check its growth are damaged, the cell can eventually grow into a tumor. Another line of defense may be the body's immune system, which optimally would detect the abnormal cell and kill it. Tumors may produce substances that block the immune system from recognizing the abnormal tumor cells and eventually overpower all internal and external deterrents to its growth.

Human onlookers describe the tumor's highlights of traditional techniques. A mechanised symptomatic structure with certain anatomical highlights has been implemented to increase the accuracy of the present framework. To improve accuracy, a symptomatic system has been developed. Manual segmentation takes sensible time and precision will be less and it prompts bury and intra rater blunders. Programmed segmentation is required because of this reason.

Programmed segmentation gives the data about the encompassing tissues around the tumor. This is because of the intensity variation happening among the same groupings. Segmentation in MRI utilized in treatment checking and gaining up prominence with progress in picture guided surgical methodologies.

Laying out of tumor contours is an essential advance. This technique depends on CNN and learn highlights that are explicit to gliomas detection and segmentation.

1.1.1 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.
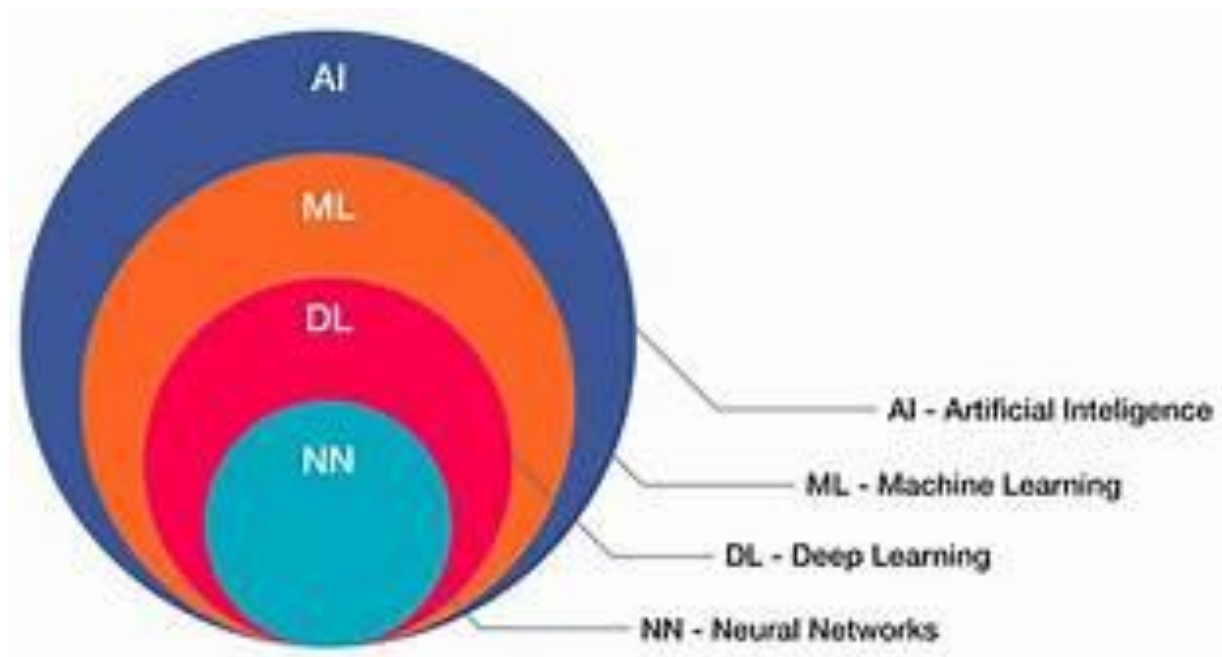

Fig 1.2 Venn diagram to represent fields of artificial intelligence

1.1.2 MACHINE LEARNING ALGORITHMS

> **SUPERVISED LEARNING**

A support-vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white. Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data and consists of a set of training examples.

In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Types of supervised learning algorithms include active learning, classification and regression.

Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, a function is generated that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data.

➢ **UN-SUPERVISED LEARNING**

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labeled. The algorithms are thus allowed to classify, label and/or group the data points contained within the data sets without having any external guidance in performing that task. In other words, unsupervised learning allows the system to identify patterns within data sets on its own.

In unsupervised learning, an AI system will group unsorted information according to similarities and differences even though there are no categories provided. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. Additionally, subjecting a system to unsupervised learning is one way of testing AI.

However, unsupervised learning can be more unpredictable than a supervised learning model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order. AI systems capable of unsupervised learning are often associated with generative learning models, although they may also use a retrieval-based approach. Chatbots, self-driving cars, facial recognition programs, expert systems and robots are among the systems that may use either supervised or unsupervised learning approaches, or both.

This algorithm is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.
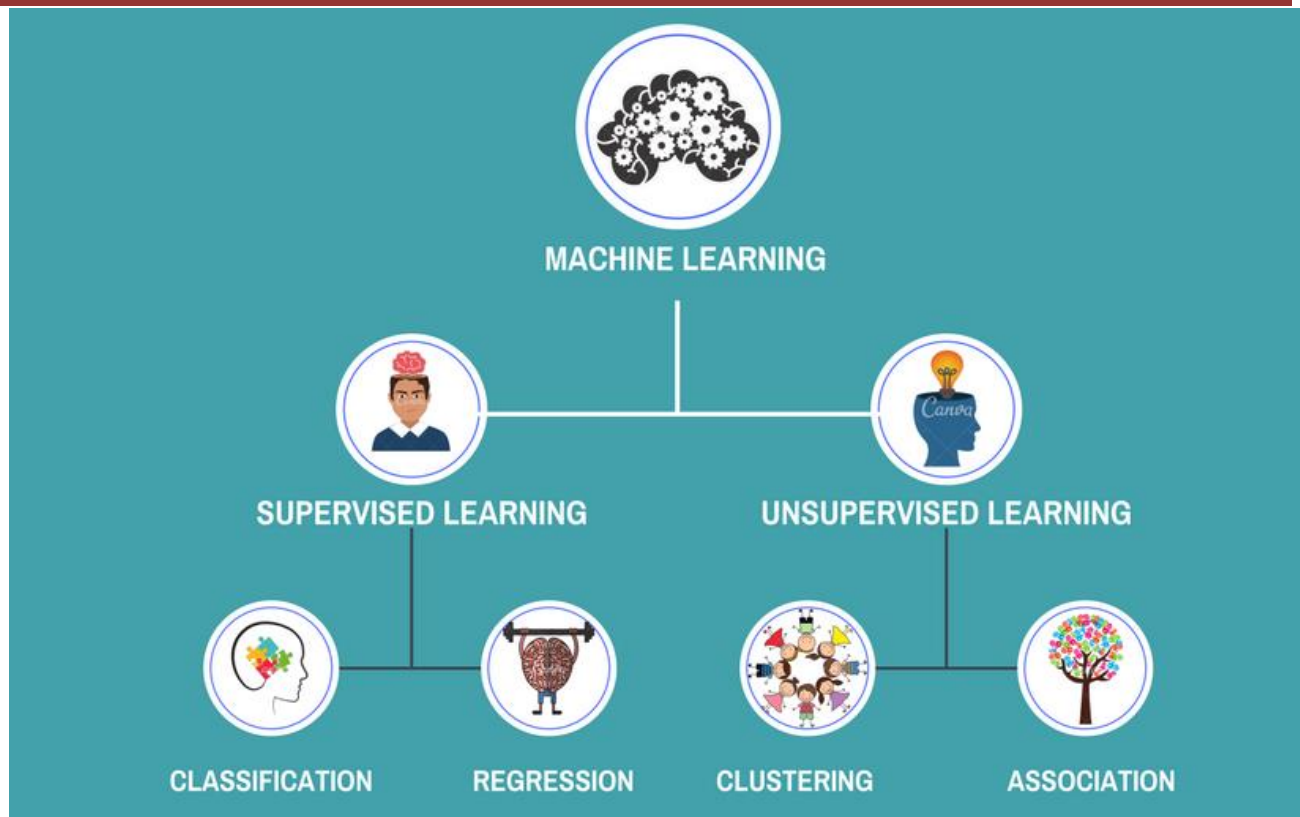
Fig 1.3 supervised and un-supervised learning

Table 1.1 Difference between supervised and un-supervised learning

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.

Supervised learning can be categorized in Classification and Regression problems.	Unsupervised Learning can be classified in Clustering and Associations problems.
Supervised learning can be used for those cases where the input as well as corresponding outputs are known.	Unsupervised learning can be used for those cases where only input data is known and no corresponding output data.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
Supervised learning is not close to true Artificial intelligence as in this, first train the model for each data, and then only it can predict the correct output.	Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

➢ DECISION TREE

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Decision tree learning uses a decision tree as a predictive model to go from observations about an item to conclusions about the item's target value.

It is one of the predictive modeling approaches used in statistics, data mining, and machine learning. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

Decision trees where target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

➢ SUPPORT VECTOR MACHINES

Support-vector machines (SVMs), also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high dimensional feature spaces.

➢ REGRESSION ANALYSIS

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is often extended by regularization (mathematics) methods to mitigate overfitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression logistic regression or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

➢ BAYESIAN NETWORK

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic

relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagram.

1.1.3   ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another.

An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs.

Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.
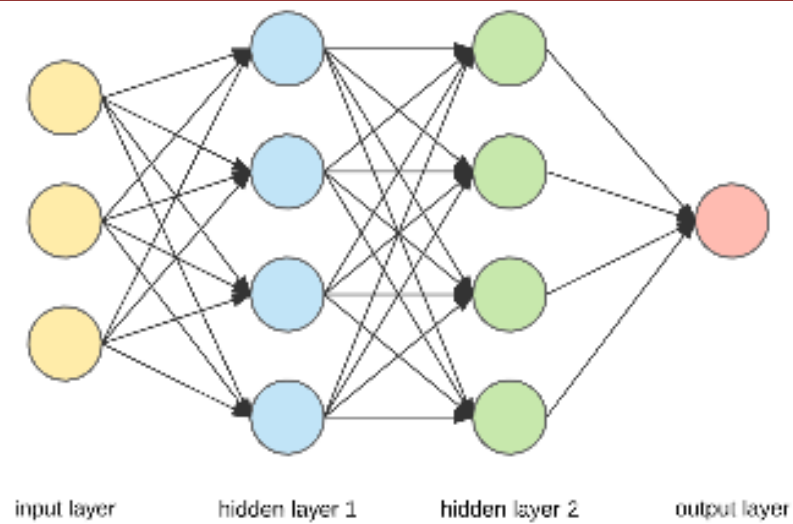
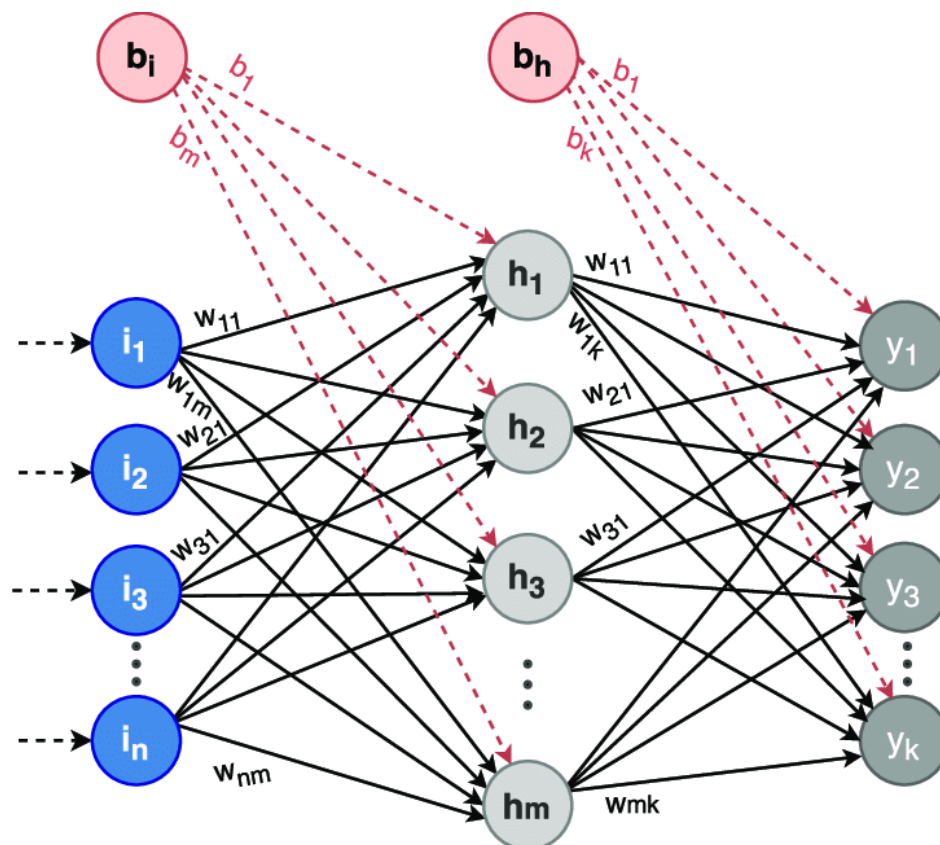**Fig 1.4 Artificial neural network**

Fig 1.5 ANN Architecture

1.1.4 BRAIN TUMOR

A brain tumor is an ailment caused due to the growth of anomalous cells in the brain. There are two main essential classes of brain tumor, they are non-cancerous (benign) brain tumor and cancerous(malignant) brain tumor. Survival rate of a tumor prone patient is difficult to predict because brain tumor is rare and are of different types. Treatment for brain tumor depends on various factors like: the type of tumor, how abnormal the cells are and where it is in the brain etc.
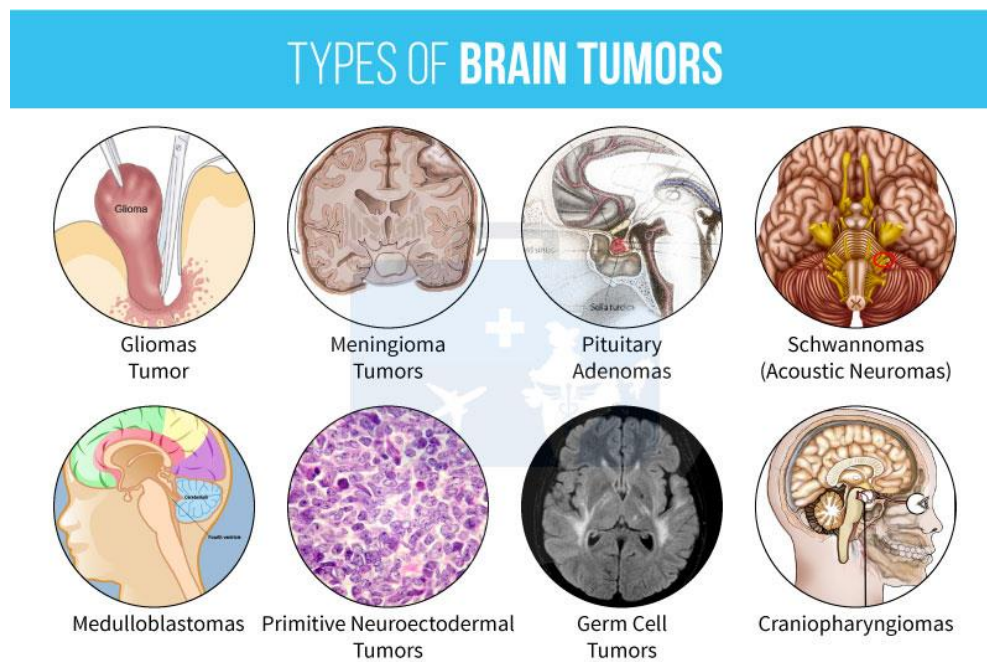
Fig 1.6 Types of brain tumors

Malignant tumors are considered cancerous and should be treated with proper measures. Types of malignant tumor are:

➢ **Carcinoma**: These tumors form from epithelial cells, which are present in the skin and the tissue that covers or lines the body's organs. Carcinomas can occur in the stomach, prostate, pancreas, lung, liver, colon, or breast. They are a common type of malignant tumor.

➢ **Sarcoma**: Sarcomas start in connective tissue, such as cartilage, bones, fat, and nerves. They originate in the cells outside the bone marrow. Most sarcomas are malignant.

➢ **Germ cell tumor**: These tumors develop in the cells that produce sperm and eggs. They usually occur in the ovaries or testicles but may also appear in the brain, abdomen, or chest.

- ➢ **Blastoma**: These tumors form from embryonic tissue or developing cells. Blastomas are much more common in children than in adults. They can lead to tumors in the brain, eye, or nervous system.

- ➢ **Meningiomas**: These are among the most common types of brain tumors and sometimes require excision or treatment if they are causing symptoms.

Benign tumors are considered non-cancerous. Detection in early stages can result in complete cure. Common types of benign brain tumors include:

- ➢ **Acoustic neuroma** (also called a vestibular schwannoma) grows on the nerves near your inner ear. It may affect your balance or hearing abilities.
- ➢ **Meningioma** is a type of brain tumor that is usually benign and tends to grow slowly.
- ➢ **Pituitary adenoma** is a tumor that occurs on the pituitary gland, located at the bottom of the skull. Pituitary tumors can sometimes affect your hormone production, which may cause you to feel sick or not like yourself.

Classifications discussed in this project are:

- ➢ Glioma tumor: Glioma is a type of tumor that occurs in the brain and spinal cord. Gliomas begin in the gluey supportive cells (glial cells) that surround nerve cells and help them function.
- ➢ Meningioma tumor: A meningioma is a tumor that arises from the meninges — the membranes that surround the brain and spinal cord. Although not technically a brain tumor, it is included in this category because it may compress or squeeze the adjacent brain, nerves and vessels. Meningioma is the most common type of tumor that forms in the head.

- ➢ Oligodendroglioma tumor: Oligodendroglioma is a brain tumor arising from oligodendrocytes, the name for cells that normally wrap around and provide support to nerve fibers in the brain. These tumors can develop anywhere in the brain, but are more commonly found in the frontal and temporal lobes.

- ➢ Glioblastoma Multiforme tumor: Glioblastoma (GBM), also referred to as a grade IV astrocytoma, is a fast-growing and aggressive brain tumor. It invades the nearby brain tissue, but generally does not spread to distant organs. GBMs can arise in the brain de novo or evolve from lower-grade astrocytoma.

> ➤ Pituitary tumor: Pituitary tumors are abnormal growths that develop in your pituitary gland. Some pituitary tumors result in too much of the hormones that regulate important functions of your body. Some pituitary tumors can cause your pituitary gland to produce lower levels of hormones.

1.2   Problem Statement

The aim of this project is to detect and classify the type of brain tumor.

1.3   Objectives

The main objective behind this project is to detect the brain tumor present or not and further it detects whether the tumor is cancerous(malignant) or non-cancerous(benign).If the tumor is non-cancerous this project further classifies it into different types such as Glioma, Meningioma and Pituitary tumor.

1.4   Organisation of the project

> ➤ Chapter 1: This chapter provides brief introduction about the project.
> ➤ Chapter 2: Provides information about the literature survey of the existing system.
> ➤ Chapter 3: Provides information about the system requirements needed to run the project.
> ➤ Chapter 4: Provides information about the system and methodology of the project.
> ➤ Chapter 5: Provides information about the implementation of each module in a precise manner.
> ➤ Chapter 6: Provides information about the experiments conducted and the results obtained.
> ➤ Chapter 7: Provides information about testing and results.
> ➤ Chapter 8: Provides information conclusion and future enhancement.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEWS

➢ **Brain tumor segmentation using deep learning by type specific sorting of images [2018]**

Zahra Sobhaninia and Safiyeh Rezaei presented a solution for brain tumor segmenting by using deep learning. In this work, they studied different angles of brain MR images and applied different networks for segmentation. The effect of using separate networks for segmentation of MR images is evaluated by comparing the results with a single network. Experimental evaluations of the networks show that Dice score of 0.73 is achieved for a single network and 0.79 in obtained for multiple networks.

➢ **Brain Tumor detection from brain MRI using Deep Learning [2018]**

P Gokila Brindha1 and M Kavinraj proposed a method comprises of a classification network which classifies the input MR images into 2 classes: on with tumor and the second without tumor. In this work, detection of brain tumor is done via classification by retraining the classifier using the technique known as transfer learning. The obtained result shows that our method outperforms the existing methods.

➢ **Tumor Detection and Classification of MRI Brain Image using Different Wavelet Transforms and Support Vector Machines [2019]**

Mircea Gurbină and Mihaela Lascu proposed a methodology that aims to differentiate between normal brain and tumor brain (benign or malign). The study of some types of brain tumors such as metastatic bronchogenic carcinoma tumors, glioblastoma and sarcoma are performed using brain magnetic resonance imaging (MRI). The detection and classification of MRI brain tumors are implemented using different wavelet transforms and support vector machines. Accurate and automated classification of MRI brain images is extremely important for medical analysis and interpretation.

➢ **Image Mining Methodology for Detection of Brain Tumor: A Review [2020]**

Shinde Apurva Swapnil; Vengurlekar Samidha Girish presented a study that shows that the filtering operations, edge detection algorithms, morphological operations and clustering are some of the important steps employed for detecting the various levels of brain tumor. This work mainly focuses on preparing the comparison review on the basis of the referenced proposed methodology, feature extraction and classification methods with its results.

➢ **Preliminary assessment for the development of CADe system for brain tumor in MRI images utilizing transfer learning in Xception model [2019]**

Daisuke Hirahara presented a study applied a novel architecture, named Xception, which enabled both high performance and reduced size and computational cost of convolutional neural networks (CNNs) using depth wise separable convolution to develop high-performance computer aided diagnosis (CADe) system for brain tumor detection from MRI. Preliminary assessment for the Xception model utilizing transfer learning demonstrated good performance with high accuracy and prediction probability. Interestingly prediction probabilities were different when different layers were relearned. The prediction probability values were highest in both normal and tumor cases when the 109 [th] layer and after (ie, the exit flow) were relearned, indicating the importance of re-learning for the Exit flow. The study results suggest that a high-performance CADe system for brain tumor in MRIs could be developed with relatively cheap small-scale learning utilizing transfer learning in the Xception model

➢ **Automated Brain Tumor Segmentation from MRI Data Based on Local Region Analysis [2019]**

Tamjid Imtiaz and Shahriar Rifat presented a work, a fully automated brain tumor segmentation method is proposed in which the local region of pixels is considered as a basic processing unit of classification instead of the traditional pixel-based classification scheme. The extracted local regions are then clustered into two groups using K means clustering algorithm for multiple feature points which effectively increases the clustering performance. The features used here are different statistical measures of intensity values and textural features obtained from the Gaussian filter bank. The segmented tumor volume is constructed by combining all the local tumor regions.
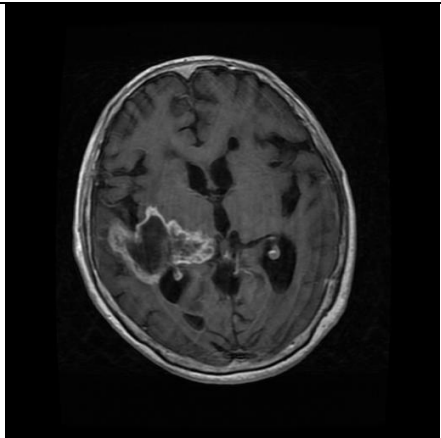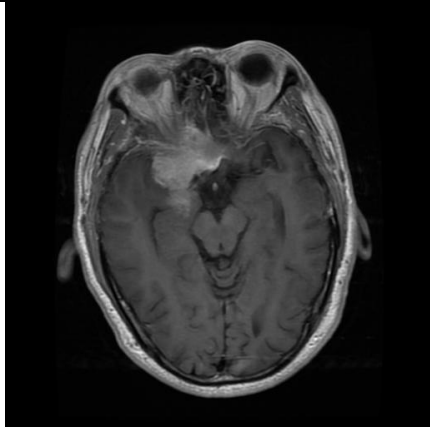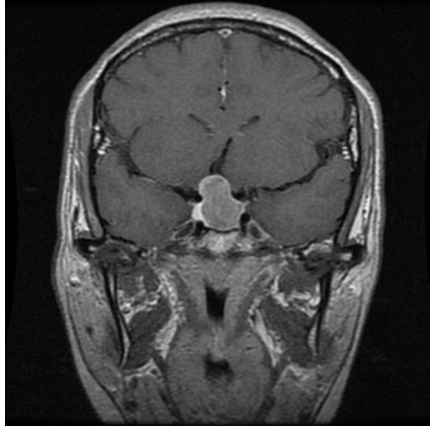
2.2   PROPOSED SYSTEM

In this project,   the augmented dataset (MRI images of brain), performed certain data pre-processing steps to convert the raw data, further investigated two deep learning models namely CNN and VGG-16 and have presented the comparative analysis in the results section. Convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers.
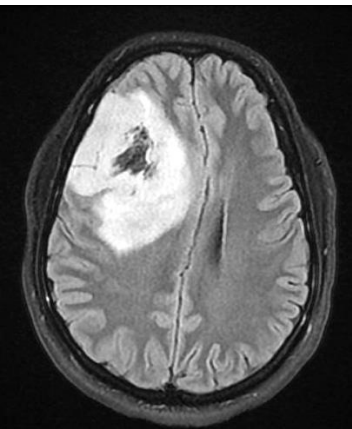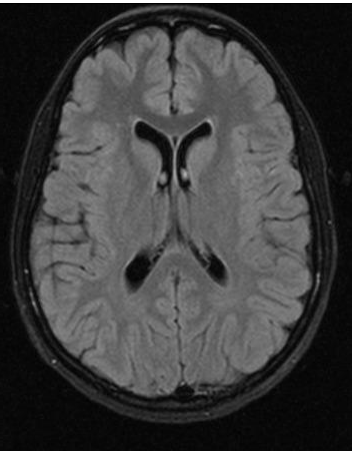
This proposed project uses CNN to train the network. The network has to learn how to recognize each of the categories in the labelled data. The datasets being used are sourced from Kaggle Dataset Platform. Further, we apply a form of gradient descent, then tabulate the predictions of the model for an image in the testing set. Finally, these model predictions are compared to the ground-truth labels from our testing set. The ground-truth labels represent what the image category actually is. From there, compute the number of predictions our classifier got correct and compute aggregate reports such as precision, recall, and f-measure, which are used to quantify the performance of our network as a whole.

Application of a data augmentation technique to test the generalizability of the TL models. Data augmentation, or increasing the amount of available data without acquiring new data by applying multiple processes to the current data, has been proven to be advantageous in image classification. Due to the limited number of images in the dataset, the data augmentation technique is applied in this study. The images in the training set were rotated at a random angle between −20 and 20 degrees, and were arbitrarily translated up to thirty pixels vertically and horizontally to create additional images. It is also worth noting that the image Data Augmenter function was utilized to dynamically create sets of augmented images during each training phase. The number of images in the training set was significantly expanded using this data augmentation method, enabling more effective use of our DL model by training with a much higher number of training images.

Table 2.1 Classification of brain tumors based MRI scans

Name of the tumor	MRI scan image	Description	Location
Glioma Tumor		Glioma is a type of tumor that occurs in the brain and spinal cord. Gliomas begin in the gluey supportive cells (glial cells) that surround nerve cells and help them function.	Glial cells (surrounding nerve cells)
Meningioma Tumor		A meningioma is a tumor that arises from the meninges — the membranes that surround your brain and spinal cord.	Meninges (membranes in the brain)
Pituitary Tumor		Pituitary tumors originate in the pituitary, a pea-sized gland in the brain that secretes and regulates hormones controlling many important body functions.	Pituitary gland and brain

Glioblastoma Multiforme Tumor		Glioblastoma multiforme (GBM) is a fast-growing glioma that develops from star-shaped glial cells that support the health of the nerve cells within the brain.	Astrocytes and Oligodendrocytes
Oligodendroglioma Tumor		Oligodendroglioma is a rare brain tumor that begin in a certain type of glial cells. Glial cells are glue-like cells that surround nerve cells and help them function.	Frontal and temporal lobes of the brain
No Tumor		No description	No description
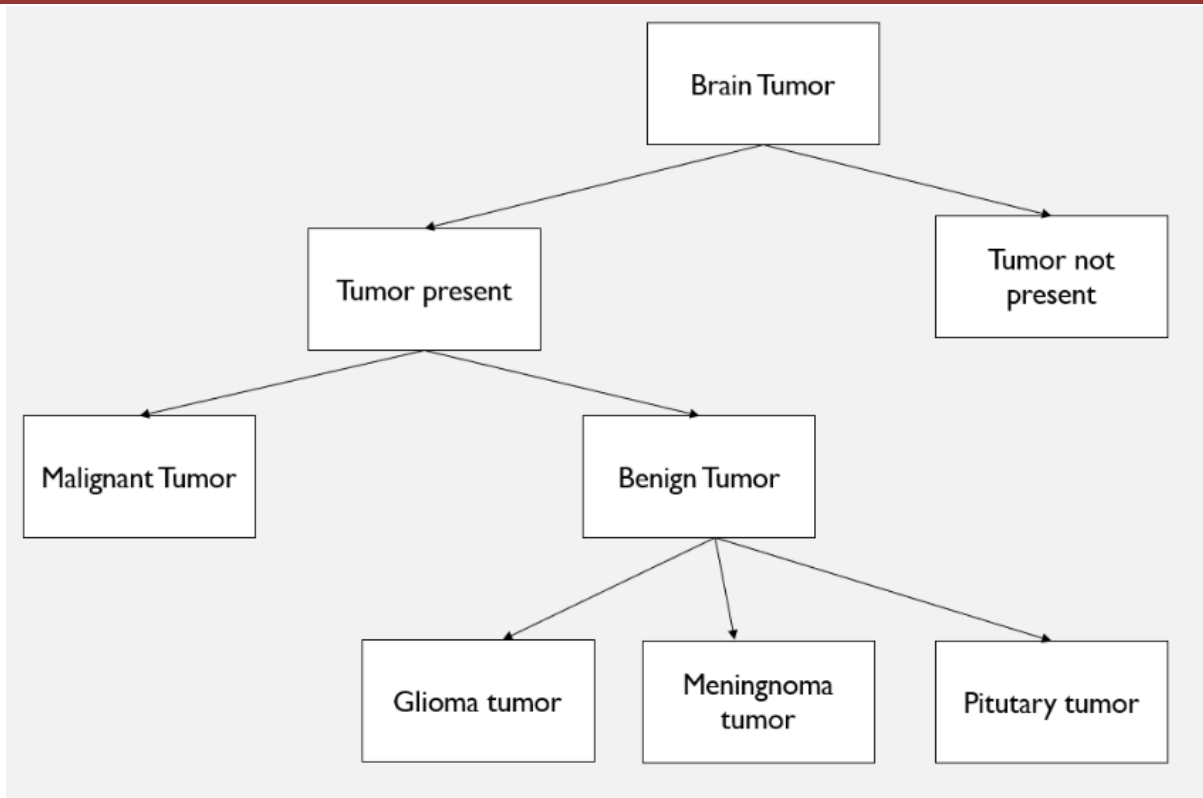
Fig 2.1 Classification done in the proposed system

➢ **ADVANTAGES:**

  ➢ The proposed project eliminates the need for BIOPSY which is a risky medical procedure required to detect the type of brain tumor.

  ➢ Early and accurate diagnosis of type of brain tumor.

  ➢ The classification of non-cancerous brain tumor such as Glioma, Meningioma, Pituitary tumors.

  ➢ This project provides a low-cost solution for the diagnosis of brain tumor.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing, and other specific functionality. In this system following are the functional requirements:

➤ The control is provided to the end user (for example patients, doctors)

➤ The MRI scan image is required to be the input in the model

➤ As soon as the person clicks on detect button, whether the tumor is present or not is displayed.

➤ After detection, if the tumor is present it is classified into benign or malignant tumor.

➤ If the detected tumor is benign, it is further classified into different types.

3.2 NON FUNTIONAL REQUIREMENTS

➤ **Dependability:**

The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not 'fail' in normal use.

➤ **Availability:**

The ability of the system to deliver services when requested. There is no error in the program while executing the program.

➤ **Reliability:**

The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

➢ **Safety:**

The ability of the system to operate without catastrophic failure. This program is user friendly and it will never effects the system.

➢ **Security:**

The ability of the system to protect itself against accidental or deliberate intrusion.

3.3   SOFTWARE/HARDWARE REQUIREMENTS

3.3.1 HARDWARE REQUIREMENTS

➢ **Operating System**
   - ➢ Microsoft: Windows (32 – or -64 bit)
   - ➢ Mac: OS 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)
   - ➢ Linux: GNOME or KDE desktop Tested on gLinux based on Debian (4.19.67-2rodete2)

➢ **RAM (Random Access Memory)**
   - ➢ GB RAM minimum but 8GB RAM recommended.

3.3.2 SOFTWARE REQUIREMENTS

➢ **Python**

Python is an interpreted high-level language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale   projects.   Python   is dynamically-typed and collected.   It   supports multiple programming   paradigms,   including structured (particularly, procedural), object-oriented and functional programming.

➢ **Python IDLE**

Every Python installation comes with an Integrated Development and Learning Environment, which you'll see shortened to IDLE or even IDE.

These are a class of applications that help you write code more efficiently. While there are many <u>IDEs</u> for you to choose from, Python IDLE is very bare-bones, which makes it the perfect tool for a beginning programmer.

➢ **OPENCV**

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV

It is used for:

      ➢ Reading an image
      ➢ Extracting the RGB values of a pixel
      ➢ Extracting the Region of Interest (ROI)
      ➢ Resizing the Image
      ➢ Rotating the Image
      ➢ Drawing a Rectangle
      ➢ Displaying text

➢ **Imutils**

Imutils is Python basic image processing functional package to do image translation, rotation, resizing, skeletonization, or blur amount detection. Imutils also check to find functions if you already have NumPy, SciPy, Matplotlib, and OpenCV installed.

➢ **Numpy**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. In Python the lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

➢ **Keras and tensorflow**

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation
Keras is:

➢ Simple : but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
➢ Flexible : Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible
➢ Powerful : Keras provides industry-strength performance and scalability

TensorFlow is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

➢ Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
➢ Computing the gradient of arbitrary differentiable expressions.
➢ Scaling computation to many devices, such as clusters of hundreds of GPUs.
➢ Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of Tensor Flow: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning.

➤ **Tkinter**

Python provides various options for developing graphical user interfaces (GUIs). The most important features are listed below.

➤ **Tkinter** − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.

➤ **wxPython** − This is an open-source Python interface for wxWidgets GUI toolkit.

➤ **PyQt** −This is also a Python interface for a popular cross-platform Qt GUI library. TutorialsPoint has a very good tutorial on PyQt

➤ **JPython** − JPython is a Python port for Java, which gives Python scripts seamless access to the Java class libraries on the local machine

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps :

➤ Import the Tkinter module.

➤ Create the GUI application main window.

➤ Add one or more of the above-mentioned widgets to the GUI application.

➤ Enter the main event loop to take action against each event triggered by the user.

➤ **Pillow**

Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new image.

CHAPTER 4
SYSTEM ARCHITECTURE

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the development of a system. Software design is a process through which the requirements are translated into a representation of software.

4.1 Architecture

A system architecture is a conceptual model using which the structure and behavior of that system can be defined. It is a formal representation of a system. Depending on the context, the system architecture can be used to refer to either a model to describe the system or a method used to build the system. Building a proper system architecture helps in the analysis of the project, especially in the early stages.
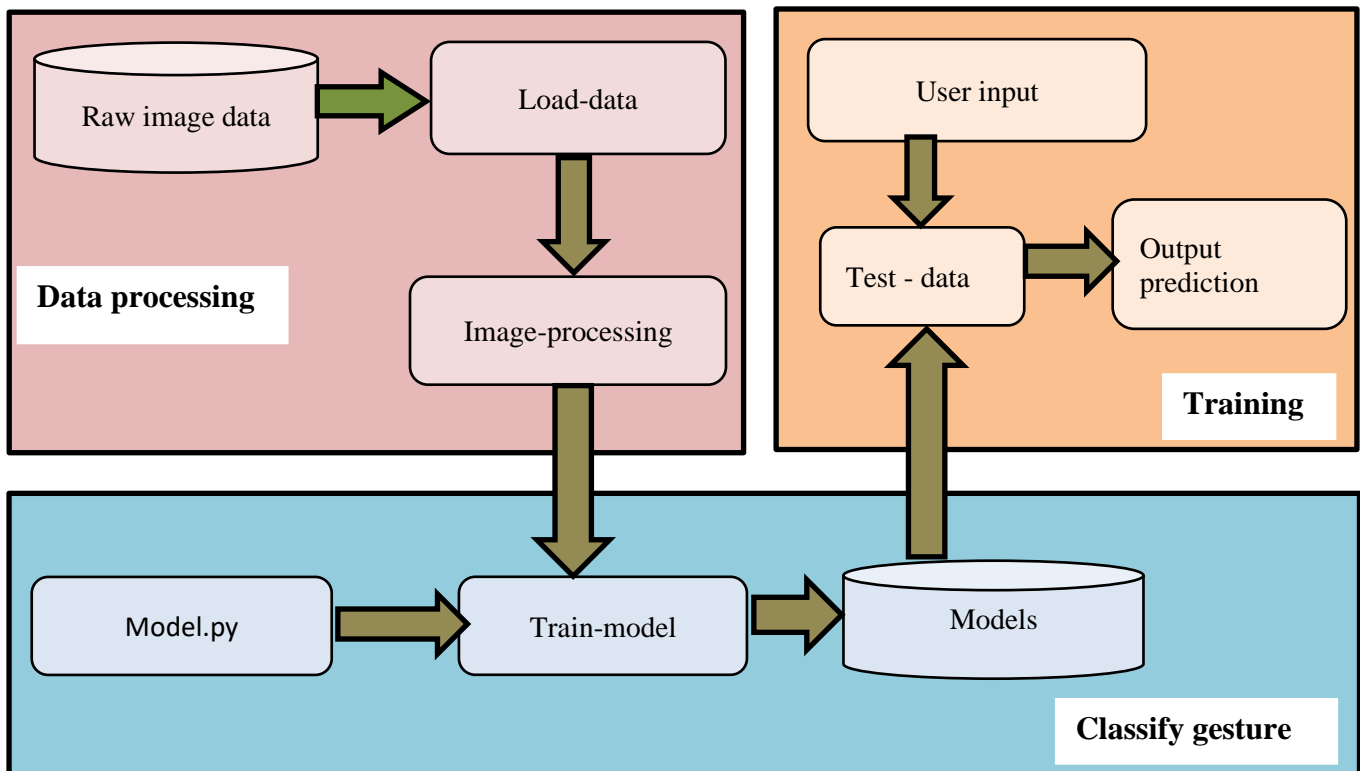

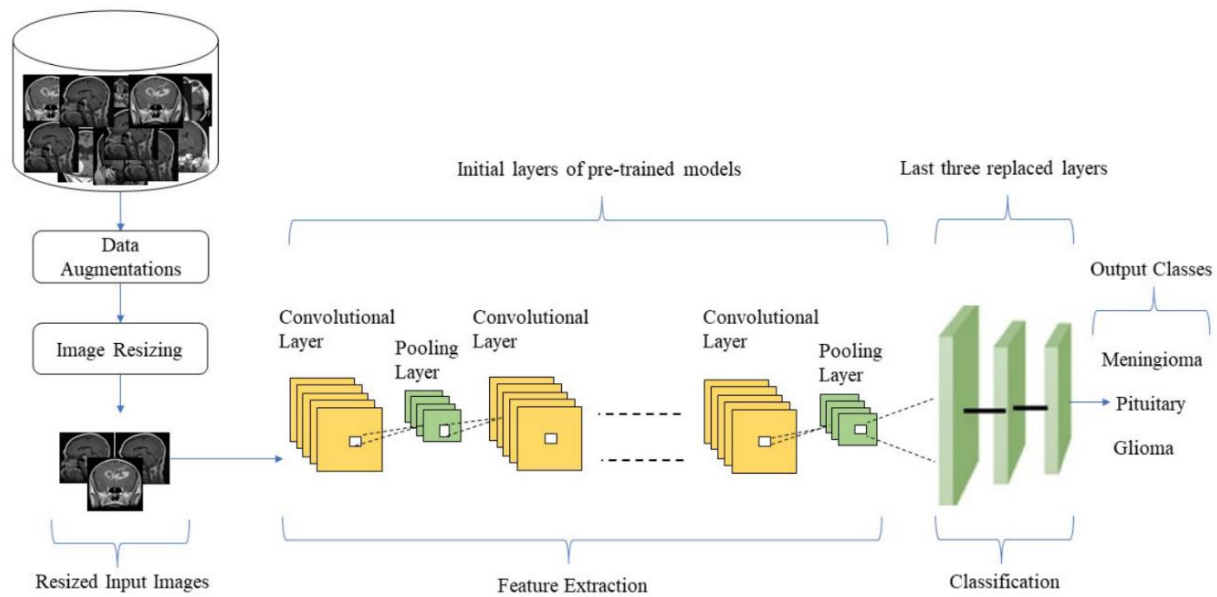**Fig 4.1   System architecture of the proposed system**

Fig 4.2 Overview of the proposed method for brain tumor classification
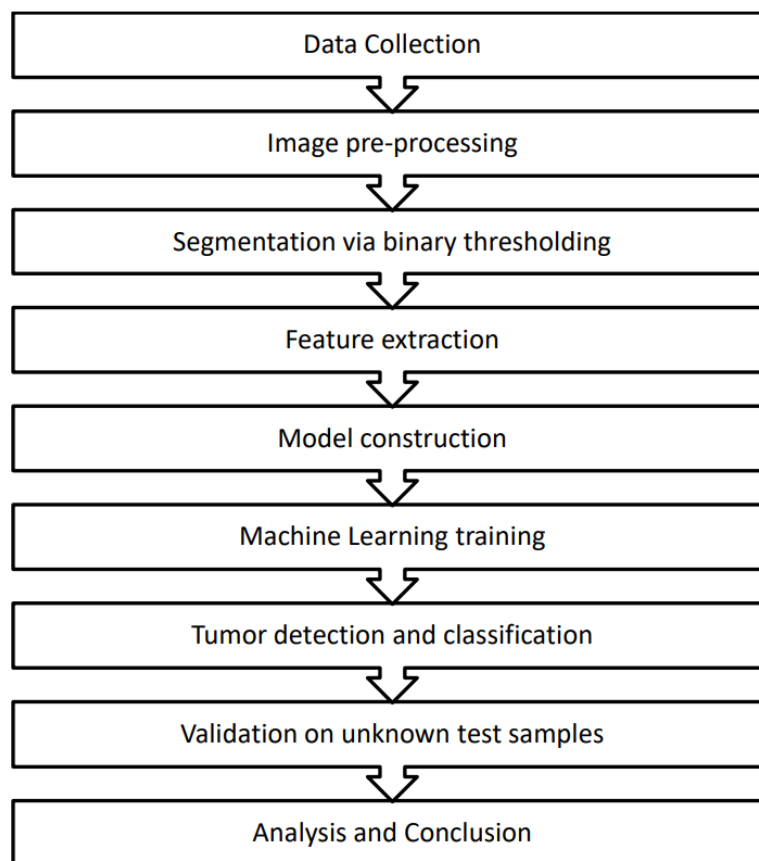

Fig 4.3 Flow Chart of the project

The proposed research methodology is depicted in Figure 1, which demonstrates an abstract view of the proposed TL-based approach for brain tumor classification using MRI images. The proposed TL-based brain tumor classification comprises the following steps. Firstly, we downloaded the freely available Kaggle MR image dataset , including glioma, meningioma and pituitary MR images, and we placed the dataset into the training directory. Secondly, we employed image Datastore to read the MR images of the dataset from the training directory.

In the third step, we applied a data augmentation technique to test the generalizability of the TL models. Data augmentation, or increasing the amount of available data without acquiring new data by applying multiple processes to the current data, has been proven to be advantageous in image classification. Due to the limited number of images in the dataset, we applied the data augmentation technique in this study. The images in the training set were rotated at a random angle between −20 and 20 degrees, and were arbitrarily translated up to thirty pixels vertically and horizontally to create additional images. It is also worth noting that the image Data Augmenter function was utilized to dynamically create sets of augmented images during each training phase. The number of images in the training set was significantly expanded using this data augmentation method, enabling more effective use of our DL model by training with a much higher number of training images. Furthermore, the augmented images were only used to train the proposed framework, not to test it; hence, only real images from the dataset were utilized to test the learned framework. In the fourth step, the input MRI images of the dataset were resized according to the pre-trained CNN model's input image requirements.

The images in the dataset were of various sizes, and different models required input images of various sizes, such as the TL mobilenetv2 classifier, accepting 224×224 input images, and the inceptionv3 classifier, requiring 229×229 input images.

Therefore, before being inserted into the DL network, the training and testing images were automatically scaled utilizing augmented image data stores of TL. Next, we employed different pre-trained deep neural networks, i.e., Inceptionresnetv2, Inceptionv3, Xception, Resnet18, Resnet50, Resnet101, Shufflenet Densenet201 and Mobilenetv2, to identify their performance in identifying and classifying different kinds of brain tumors. The proposed TL models consisted of layers from the pre-trained networks and three new layers, i.e., the last three layers modified to suit the new image categories (meningioma, pituitary and glioma). The transfer learned models had a softmax layer, classifying images into meningioma, pituitary and glioma.

For example, for "Inceptionv3" and "InceptionResNetV2", we replaced "predictions", "predictions_softmax" and "ClassificationLayer_predictions" with a "fully connected layer", a "softmax layer" and a "classification output" layer. We connected the additional layers to the network's last remaining transferred layer, i.e., "avg pool". We replaced the network's last three layers, i.e., "fc1000", "fc1000_softmax" and "ClassificationLayer_fc1000", with a "completely connected layer", a "softmax layer" and "classification output" layers for "ResNet50" and connected the additional layers to the network's last remaining transferred layer ("avg pool"). Similarly, we replaced the "fc1000", "prob" and "ClassificationLayer_predictions" layers of the network with a "fully connected layer", a "softmax layer" and a "classification output" layer for "ResNet101" and connected the new layers to the network's last remaining transferred layer ("pool5"). The detailed structure of the proposed DL framework is shown in Figure 2, extended from the concept of TL. Furthermore, we evaluated and validated each model to assess the performance of different pre-trained TL algorithms in identifying brain tumor types. For this purpose, we divided the dataset into training and testing sets to obtain accurate and reliable results; more specifically, we used 80% of the data for model training and the remaining 20% for testing. The overall process of pre-trained TL classification for brain tumor identification and classification is shown in Figure 4.2

4.2 Major Algorithm

CNN is an efficient recognition algorithm which is widely used in pattern recognition and image processing. It has many features such as simple structure, less training parameters and adaptability. It has become a hot topic in voice analysis and image recognition.

CNN algorithm need experience in architecture design, and need to debug unceasingly in the practical application, in order to obtain the most suitable for a particular application architecture of CNN. Based on gray image as the input of 96×96, in the preprocess stage, turning it into 32×32 of the size of the image. Design depth of the layer 7 convolution model: input layer, convolution layer C1, sub sampling layer S1, convolution layer C2, sampling layer S2, hidden layer H and output layer F.
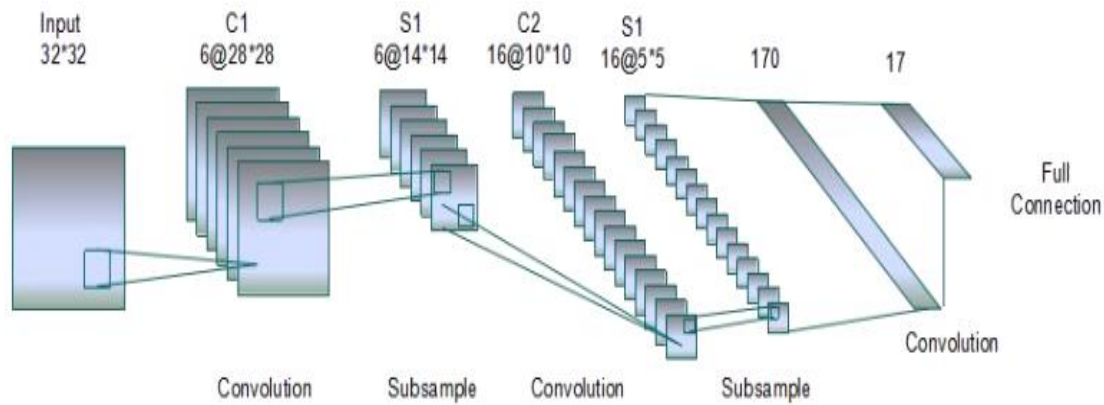
Fig 4.4 Architecture of CNN in training faces

In view of the 32×32 input after preprocessing, There is a total of 17 different pictures. C1 layer for convolution, convolution layer adopts 6 convolution kernels, each the size of the convolution kernels is 5×5, can produce six feature map, each feature map contains $(32-5+1) \times (32-5+1) = 28 \times 28 = 784$ neurons. At this point, a total of $6 \times (5 \times 5 + 1) = 156$ parameters to be trained . S1 layer for sub sampling, contains six feature map, each feature map contains $14 * 14 = 196$ neurons. the sub sampling window is 2×2 matrix, sub sampling step size is 1, so the S1layer contains $6 \times 196 \times (2 \times 2 + 1) = 5880$ connections. Every feature map in the S1 layer contains a weights and bias, so a total of 12 parameters can be trained in S1 layer . C2 is convolution layer, containing 16 feature graph, each feature graph contains $(14-5+1)(14-5+1) = 100$ neurons and adopts full connection, namely each characteristic figure used to belong to own 6 convolution kernels with six characteristics of the sample layer S1 convolution and figure.

Each feature graph contains $6 \times 5 \times 5 = 150$ weights and a bias. So, C2 layer contains a total of $16 \times (150 + 1) = 150$ parameters to be trained. S2 is sub sampling layer, containing 16 feature map, each feature map contains 5×5 neurons, S2 total containing $25 \times 16 = 400$ neurons. S2 on characteristic figure of sub sampling window for 2×2, so there is 32 trainable S2 parameters. As a whole connection layer, hidden layer H contains 170 neurons, each neuron is connected to 400 neurons on S2. So H layer contains $170 \times (400 + 1) = 48120$ parameters feature map. Output layer F for all connections, including 17 neurons. A total of $17 \times (170 + 1) = 2907$ parameters to be trained.

Designing a Convolutional Neural Network

Now that we understand the various components, we can build a convolutional neural network. We will be using Fashion-MNIST, which is a dataset of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. The dataset can be downloaded here.

Our convolutional neural network has architecture as follows:

[INPUT]

→ [CONV 1] → [BATCH NORM] → [ReLU] → [POOL 1]

→ [CONV 2] → [BATCH NORM] → [ReLU] → [POOL 2]

→ [FC LAYER] → [RESULT]

For both conv layers, we will use kernel of spatial size 5 x 5 with stride size 1 and padding of 2. For both pooling layers, we will use max pool operation with kernel size 2, stride 2, and zero padding.

Performances Metrics

To evaluate the proposed DWAE model efficiency, we compare the proposed model overall performance using accuracy, sensitivity, specificity, DSC, Precision, JSI, FPR, and FNR values.

➢ **Accuracy (ACC)**

Accuracy (ACC) is utilized to compute the degree of correct tumor classification rate, and is calculated using the following Equation

$$Accuracy = \frac{(TP+TN)}{(TP+TN)+(FP+FN)} \times 100$$

> **Sensitivity (SE)**

Sensitivity (SE) is utilized to calculate the degree of how much approach is sensitive to measure the tumor identification rate, and is calculated using the following

$$Sensitivity = \frac{(TP)}{(TP+FN)} \times 100$$

> **Specificity (SP)**

Specificity (SP) is the rate between true negative (TN) and true positive (TP), and is calculated using the following Equation

$$Specificity = \frac{(TN)}{(TN+FP)} \times 100$$

> **Dice Similarity Coefficient (DSC)**

Dice similarity coefficient (DSC) is utilized to compute the ratio between the actual tumor and non-tumor, which are compared with predicted tumor and non-tumor pixels, and is calculated using the following Equation

$$DSC = \frac{2TP}{FP+2TP+FN} \times 100$$

> **PRECISION (PRE)**

PRECISION (PRE) describes the number of digits that are used to express a value, and is calculated using the following Equation

$$Precision = \frac{(TP)}{(TP+FP)} \times 100$$

➢ **JACCARD Similarity Index (JSI)**

JACCARD similarity index (JSI) is utilized to compute the similarity between the actual tumor pixels and predicted tumor pixels and is calculated using the following Equation

$$JSI = \frac{TP}{TP+FN+FP} \times 100$$

➢ **FALSE Positive Rate (FPR)**

FALSE positive rate (FPR) is utilized to compute the ratio of wrongly identified pixels, corrected identified pixels, and is calculated using the following Equation

$$FPR = 1 - Specificity$$

➢ **FALSE Negative Rate (FNR)**

FALSE negative rate (FNR) is utilized to compute the positive proportion, but the approach-identified negative and is calculated using the following Equation

$$FNR = 1 - Sensitivity$$

where true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Applications

Below are some applications of Convolutional Neural Networks used today:

➢ Object detection: With CNN, we now have sophisticated models like R-CNN, Fast R-CNN, and Faster R-CNN that are the predominant pipeline for many object detection models deployed in autonomous vehicles, facial detection, and more.

➢ Semantic segmentation: In 2015, a group of researchers from Hong Kong developed a CNN-based Deep Parsing Network to incorporate rich information into an image segmentation model. Researchers from UC Berkeley also built fully convolutional networks that improved upon state-of-the-art semantic segmentation.

➢ Image captioning: CNNs are used with recurrent neural networks to write captions for images and videos. This can be used for many applications such as activity recognition or describing videos and images for the visually impaired. It has been heavily deployed by YouTube to make sense to the huge number of videos uploaded to the platform on a regular basis.

CHAPTER 5

IMPLEMENTATION

5.1 Module 1: Dataset Collection

The first component of building a deep learning network is to gather our initial dataset. The images themselves as well as the labels associated with each image are required. These labels should come from a finite set of categories, such as: glioma tumor, meningioma tumor, pituitary tumor and no tumor. Furthermore, the number of images for each category should be approximately uniform (i.e., the same number of examples per category) then the classifier will become naturally biased to overfitting nto these heavily-represented categories. Class mbalance is a common problem in machine learning and there exist a number of ways to overcome it. As our system is mainly focusing on detection of brain tumor, the gathered data are MRI images. The dataset used consists of several images in which tumor is present and in several images in which tumor is not present.
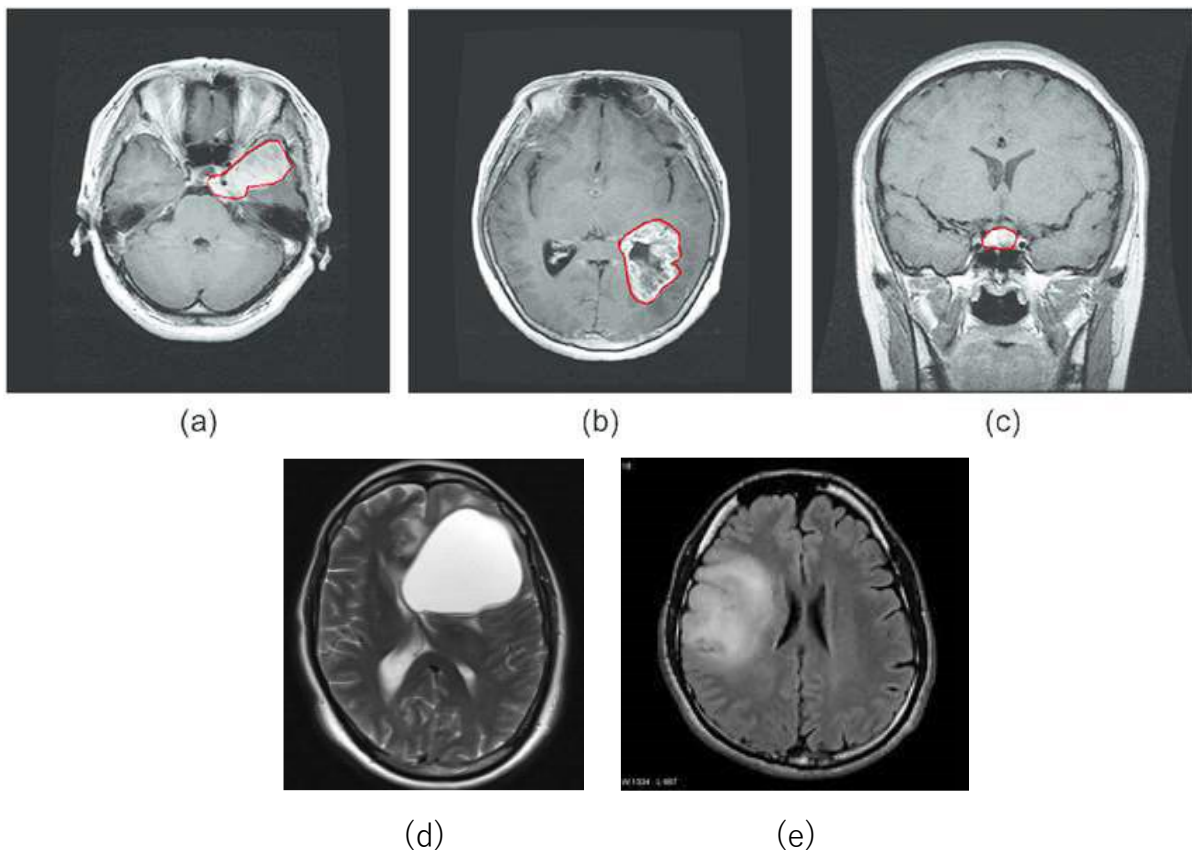

(a)   (b)   (c)

(d)   (e)

Figure 5.1 Dataset of tumors (a) Glioma (b)Meningioma (c) Pituitary (d) Glioblastoma Multiforme (e) Oligodendroglioma Tumors

5.2 Module 2: Image Pre-processing

The primary target is to improve image highlights needed for additional processing.

The main issues of image imperfections are due to the following :

- ➢ Low resolution
- ➢ Simulation
- ➢ Presence of image artifacts
- ➢ Geometric Distortion
- ➢ Low contrast
- ➢ High level of noise

These are the methods used for Image pre-processing :

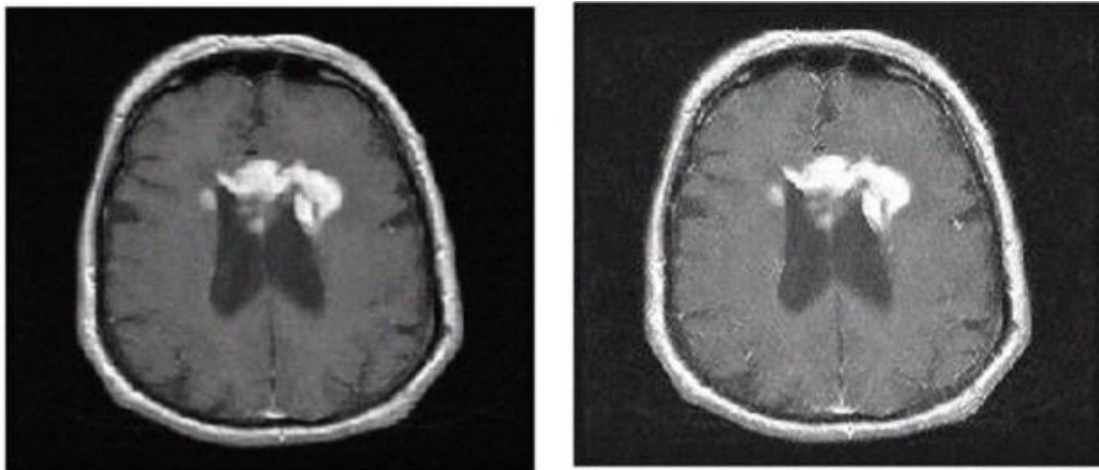### 5.2.1 Greyscale Image Conversion

Figure 5.2 Grey scale a) Original image b) Enhanced Image

The aim of grey scale contrast enhancement is to improve the perception of information which is needed for further processing. Contrast enhancement process makes the image brighter, it improves the visual details in the image. There are two methods they are direct methods and indirect methods. In the case of the direct method of grey scale contrast enhancement, a contrast measure is first initialized, which is then changed by a mapping function to generate the pixel value of the enhanced image. In indirect method, it improves the contrast by exploiting the under-utilized regions of the dynamic range.

Pixels: The Building Blocks of Images

Pixels are the raw building blocks of an image. Every image consists of a set of pixels. There is no finer granularity than the pixel. Normally, a pixel is considered the "colour" or the "intensity"

of light that appears in a given place in our image. If we think of an image as a grid, each square contains a single pixel. For example, take a look at Figure 5.2.

The image in Figure 5.2 above has a resolution of 1000_750, meaning that it is 1000 pixels wide and 750 pixels tall. We can conceptualize an image as a (multidimensional) matrix. In this case, our matrix has 1000 columns (the width) with 750 rows (the height). Overall, there are 1000_750 = 750000 total pixels in our image.

Most pixels are represented in two ways:

- ➢ Grayscale/single channel
- ➢ Colour

In a grayscale image, each pixel is a scalar value between 0 and 255, where zero corresponds to "black" and 255 being "white". Values between 0 and 255 are varying shades of gray, where values closer to 0 are darker and values closer to 255 are lighter. The grayscale gradient image in Figure 3.2 demonstrates darker pixels on the left-hand side and progressively lighter pixels on the right-hand side. Color pixels; however, are normally represented in the RGB color space.

Figure 5.3:Color image to grey scale conversion

This image is 1000 pixels wide and 750 pixels tall, for a total of 1000x750=750000 total pixels.

Figure 5.4: Image gradient demonstrating pixel values going from black (0) to white (255).

Pixels in the RGB color space are no longer a scalar value like in a grayscale/single channel image – instead, the pixels are represented by a list of three values: one value for the Red component, one for Green, and another for Blue. To define a color in the RGB color model, all we need to do is define the amount of Red, Green, and Blue contained in a single pixel.

Each Red, Green, and Blue channel can have values defined in the range [0;255] for a total of 256 "shades", where 0 indicates no representation and 255 demonstrates full representation. Given that the pixel value only needs to be in the range [0;255], we normally use 8-bit unsigned integers to represent the intensity. As we'll see once we build our first neural network, we'll often preprocess our image by performing mean subtraction or scaling, which will require us to convert the image to a floating point data type. Keep this point in mind as the data types used by libraries loading images from disk (such as OpenCV) will often need to be converted before we apply learning algorithms to the images directly. Given our three Red, Green, and Blue values, we can combine them into an RGB tuple in the form (red, green, blue). This tuple represents a given color in the RGB color space. The RGB color space is an example of an additive color space: the more of each color is added, the brighter the pixel becomes and closer to white. We can visualize the RGB color space in Figure 3.3 (left). As you can see, adding red and green leads to yellow. Adding red and blue yields pink. And adding all three red, green, and blue together, we create white.

To make this example more concrete, let's again consider the color "white" – we would fill each of the red, green, and blue buckets up completely, like this: (255, 255, 255). Then, to create the color black, we would empty each of the buckets out (0, 0, 0), as black is the absence of color.

To create a pure red color, we would fill up the red bucket (and only the red bucket) completely: (255, 0, 0).
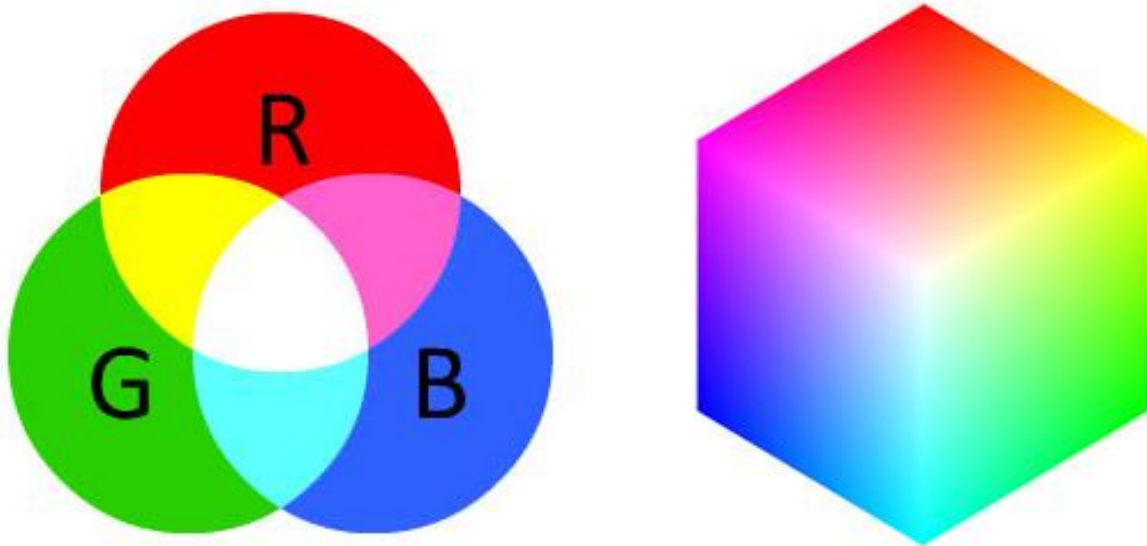
Figure 5.5:RGB color model

Left: The RGB color space is additive. The more red, green and blue you mix together, the closer you get to white. Right: The RGB cube.

The RGB color space is also commonly visualized as a cube (Figure 3.3, right) Since an RGB color is defined as a 3-valued tuple, which each value in the range [0;255] we can thus think of the cube containing 256_256_256 = 16;777;216 possible colors, depending on how much Red,

Green, and Blue are placed into each bucket.

As an example, let's consider how "much" red, green and blue we would need to create a single color (Figure 3.4, top). Here we set R=252, G=198, B=188 to create a color tone similar to the skin of a Caucasian (perhaps useful when building an application to detect the amount of skin/flesh in an image). As we can see, the Red component is heavily represented with the bucket almost filled. Green and Blue are represented almost equally. Combining these colors in an additive manner, we obtain a color tone similar to Caucasian skin.
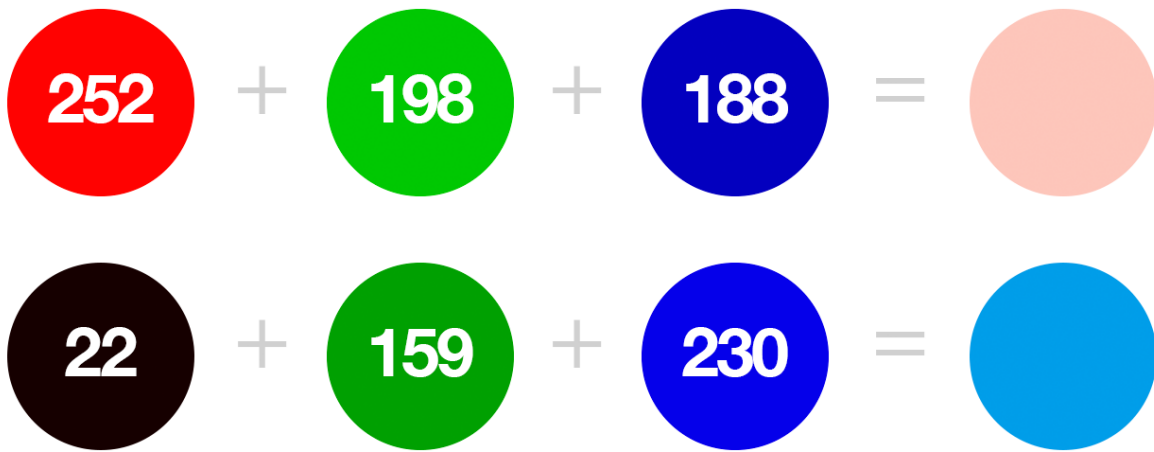
Figure 5.6:Examples of adding color components together

Top: An example of adding various Red, Green, and Blue color components together to create a "caucasian flesh tone", perhaps useful in a skin detection program.

The Image Coordinate System As mentioned in Figure 3.1 earlier in this chapter, an image is represented as a grid of pixels. To make this point more clear, imagine our grid as a piece of graph paper. Using this graph paper, the origin point (0;0) corresponds to the upper-left corner of the image. As we move down and to the right, both the x and y values increase. Figure 3.6 provides a visual representation of this "graph paper" representation. Here we have the letter "I" on a piece of our graph paper. We see that this is an 8_8 grid with a total of 64 pixels.

It's important to note that we are counting from zero rather than one. The Python language is zero indexed, meaning that we always start counting from zero.
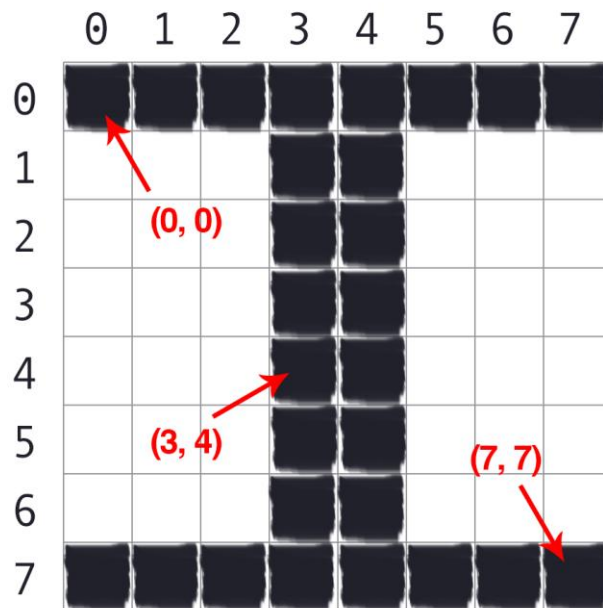

Figure 5.7: Pixel accessed by the co-ordinates.

The letter "I" placed on a piece of graph paper. Pixels are accessed by their (x;y):Coordinates, where we go x columns to the right and y rows down, keeping in mind that Python is zero-indexed. As an example of zero-indexing, consider the pixel 4 columns to the right and 5 rows down is indexed by the point (3;4), again keeping in mind that we are counting from zero rather than one. Images as NumPy Arrays

Figure 5.8: Loading image from disk and displaying on screen

➢ **OPENCV**.

Image processing libraries such as OpenCV and scikit-image represent RGB images as multidimensional NumPy arrays with shape (height, width, depth). Readers who are using image processing libraries for the first time are often confused by this representation – why does the height come before the width when we normally think of an image in terms of width first then height? The answer is due to matrix notation.

When defining the dimensions of matrix, we always write it as rows x columns. The number Of rows in an image is its height whereas the number of columns is the image's width. The depth Will still remain the depth.

Therefore, while it may be slightly confusing to see the .shape of a NumPy array represented as (height, width, depth), this representation actually makes intuitive sense when considering how a matrix is constructed and annotated

> **RGB and BGR Ordering**

It's important to note that OpenCV stores RGB channels in reverse order. While we normally think in terms of Red, Green, and Blue, OpenCV actually stores the pixel values in Blue, Green, Red order.Early developers of the OpenCV library chose the BGR colour format because the BGR ordering was popular among camera manufacturers and other software developers at the time. Simply put – this BGR ordering was made for historical reasons and a choice that we now have to live with. It's a small caveat, but an important one to keep in mind when working with OpenCV.

5.2.2 Scaling and aspect ratios

Scaling or resizing, is the task of increasing or decreasing the size of an image's width and height. When resizing an image, it's important to keep in mind the aspect ratio. Aspect ratio of image is the comparable relationship of the width to the height. For example, a 4x2 inch image's aspect ratio will be 2:1. Aspect ratio represents how large the width is in comparison to the height of the image.
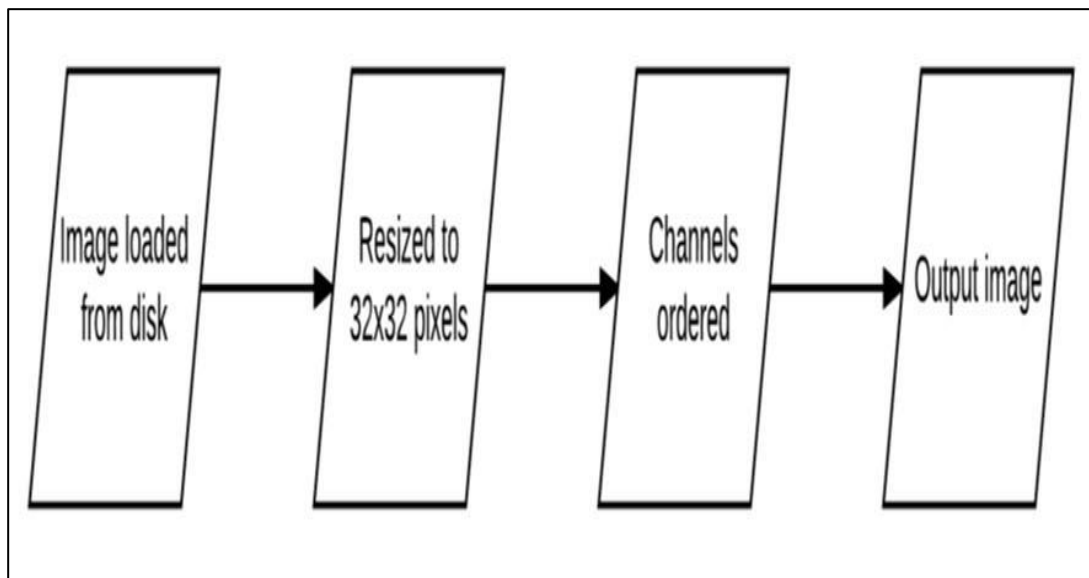

Figure 5.9 Image pre-processing pipeline that (1) loads an image from disk, (2) resizes it to 32_32 pixels, (3) orders the channel dimensions, and (4) outputs the image.

5.2.3 Histogram Equalisation

Histogram equalization normalizes the pixel intensities; thus it normalizes some of the illumination problems. The method is used in normalizing the pixel intensities of the brain MRI images. Histogram Equalization is mapping of each pixel of the input image to the relating pixels of the output image. This technique equalizes the intensity values to full range of the histogram to get an enhanced output image. It enhances the contrast and brightness of the input image by increasing the values of each pixel giving rise to dynamic range expansion.
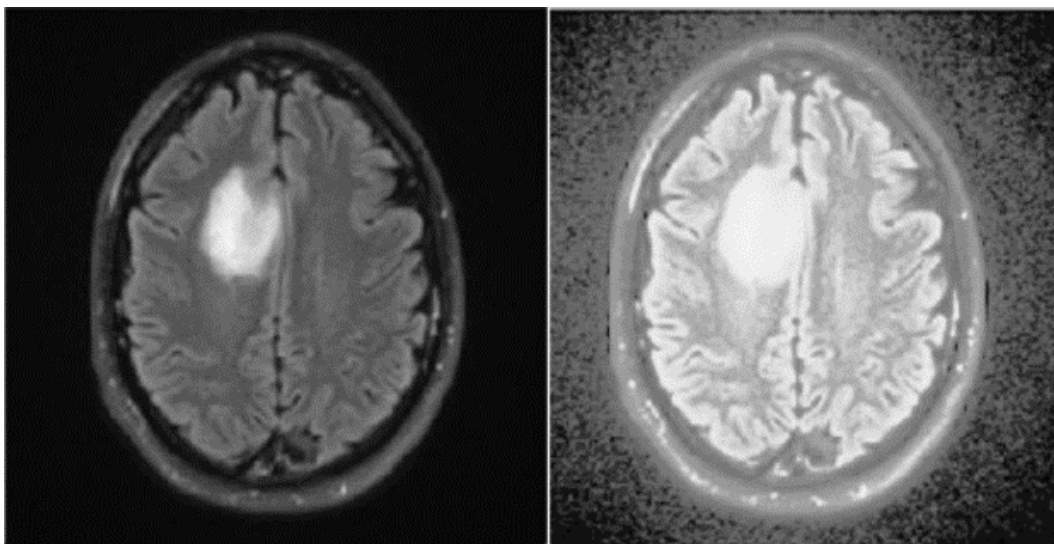
(a)Original image   (b)Histogram equalization
Figure 5.10: Histogram Equalization of brain MRI image

5.2.4 Adaptive Thresholding

Thresholding is used to part an image by setting all pixels of the image whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value. Adaptive thresholding takes a grayscale or colour image as input and, outputs a binary image that represents the segmentation.

For every pixel in the image, a threshold is been calculated. If the pixel value is lesser than the threshold value then it is set to the background value, or else it is assumed the foreground value.
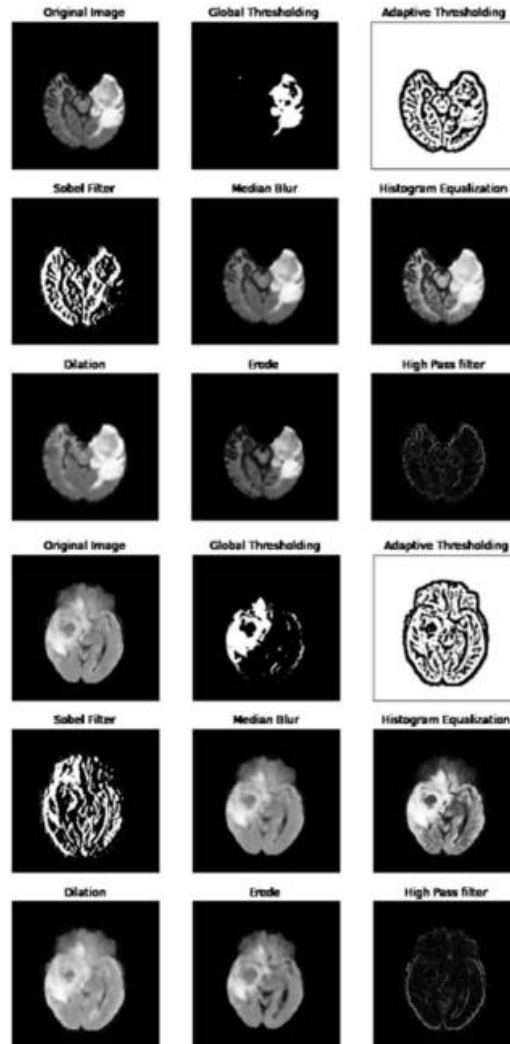
5.2.5 Erosion and Dilation

Figure 5.11 Image pre-processing employed MRI scans

Dilation and erosion are the two most important operations of morphology. Morphology is to identify the properties of the structure and shape of any element. In the MRI images may contain numerous deformities. The objectives is removing the imperfections by representing picture shape and structure. Dilation operations add pixels to the boundary region of the object, while erosion operations are used to remove the pixels from the boundary region of the objects. Thus these operation of addition and removing pixels from or to the boundary region of the objects structures the element of the selected image.

5.3 Module 3:Splitting the Datasets

The datasets are split into two parts:

- ➢ A training set
- ➢ A testing set

A training set is used by the classifier to learn what each category looks like by making predictions on the input data and then correct itself when predictions are wrong. After the classifier has been trained, it can evaluate the performing on a testing set. It's extremely important that the training set and testing set are independent of each other and do not overlap. Common split sizes for training and testing sets include 66:6%33:3%, 75%=25%, and 90%=10%, respectively.
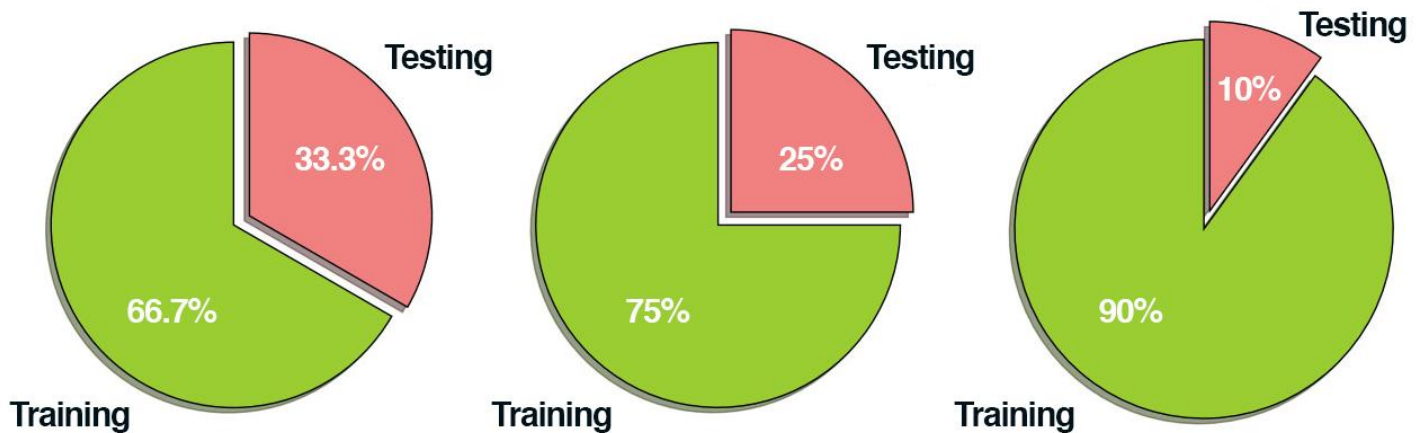

Fig 5.12 Common training and testing splits

It is important that the training set and testing set are independent of each other and are exclusive of each other.

Neural networks have a number of knobs (ex., learning rate, decay, regularization, etc.) that need to be modified to obtain accurate performance. These are the called as the types of hyperparameters.

Validation set is created as the testing set is only used in analysing the performance of the network. This set of the data is from the training dataset and is used as "fake test data" so we can amplify the hyperparameters. After determining the hyperparameter values using the validation set, we proceed to predict the values.

5.4 Module 4: Training

Training data is the data used to train an algorithm or machine learning model to predict the outcome to design the model to predict. The network should learn how to recognize each of the categories in the labelled data. A form of gradient descent is then applied.

The training dataset is used to train the model using CNN algorithm for every epoch or iteration, the model is being tested on the validation (fake dataset).

The validation dataset is a type of test data for the model to tune the hyperparameters such as learning rate, decay and hence the validation set becomes unseen data. CNN is implemented to get better result. Past layers are interconnected with weights of the kernel. Back propagation is used to increase the quality and clarity of the image. The issue of over- fitting is fixed since all units are shared by the kernels. The neighbors of the data are taken by using the kernels. Kernel is a major source of information. Output of neural network is subjected to activation function.

➢ Convolutional Layers

Extraction of the features from the image serves as a main objective in convolution layers, the part of picture is link to the following convolution layer.

➢ Padding

Padding is integrating a zero layer outside the input volume so the data on surrounding won't be missed and we can get a similar aspect of output as input volume. Zero padding is used .

➢ Activation Function

Non- linear activation function ReLU (Rectifier Activation function) is used to provide accurate results and combines the results, feeding the output into the next layer in the network. During training, a CNN automatically learns the values for these filters.

In the context of image classification, our CNN may learn to:

➢ Detect edges from raw pixel data in the first layer.
➢ Use these edges to detect shapes (i.e., "blobs") in the second layer.
➢ Use these shapes to detect higher-level features such as facial structures, parts of a car, etc.in the highest layers of the network.

➢ Pooling layer

It is used for combining spatially nearby features. Max-pooling is generally used to join features. It decreases the dimension of input image and controls over-fitting.

Validation set

A supervised AI is trained on a corpus of training data. Training, tuning, model selection and testing are performed with three different datasets: the training set, the validation set and the testing set. Validation sets are used to select and tune the final AI model.

Training sets make up the majority of the total data, averaging 60 percent. In testing, the models are fit to parameters in a process that is known as adjusting weights.

The validation set makes up about 20 percent of the bulk of data used. The validation set contrasts with training and test sets in that it is an intermediate phase used for choosing the best model and optimizing it. Validation is sometimes considered a part of the training phase. It is in this phase that parameter tuning occurs for optimizing the selected model. Overfitting is checked and avoided in the validation set to eliminate errors that can be caused for future predictions and observations if an analysis corresponds too precisely to a specific dataset.

Testing sets make up 20 percent of the bulk of the data. These sets are ideal data and results with which to verify correct operation of an AI. The test set is ensured to be the input data grouped together with verified correct outputs, generally by human verification. This ideal set is used to test results and assess the performance of the final model.

It is generally considered unwise to attempt further adjustment past the testing phase. Attempting to add further optimization outside the validation phase will likely to increase overfitting.
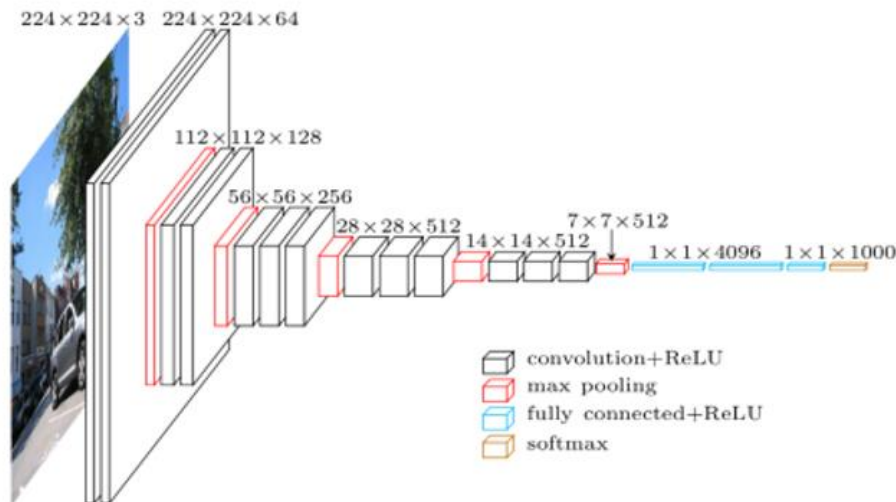
Figure 5.13 Visualization of VGG architecture

It consists of 11 number of inception modules and each module comprises of convolutional layer, activation layer, pooling layer, and batch normalization layer. These modules are concatenated to get multiscale maximal features from input images. The global average pooling, and fully connected layers are added with the last Inception module layer along with different Inception modules features for brain tumor classification.

The dropout layer after global average pooling layer has been used for regularization, and this layer minimized the over fitting problem for training the proposed model. The features are extracted from fully connected layer and after the concatenation from different Inception blocks are passed to the classifier. The features taken from Inception block C, D and E modules are concatenated and passed to the traditional classifier such as SoftMax.

The features from Inception block A and B did not produce good performance and simply these features are not extracted for classification is shown in Figure 5.8. The deep layers Inception module such as C, D and E Inception blocks produced better performance and these Inception module-based features have been used for classification and assessment of multiclass brain tumor types.
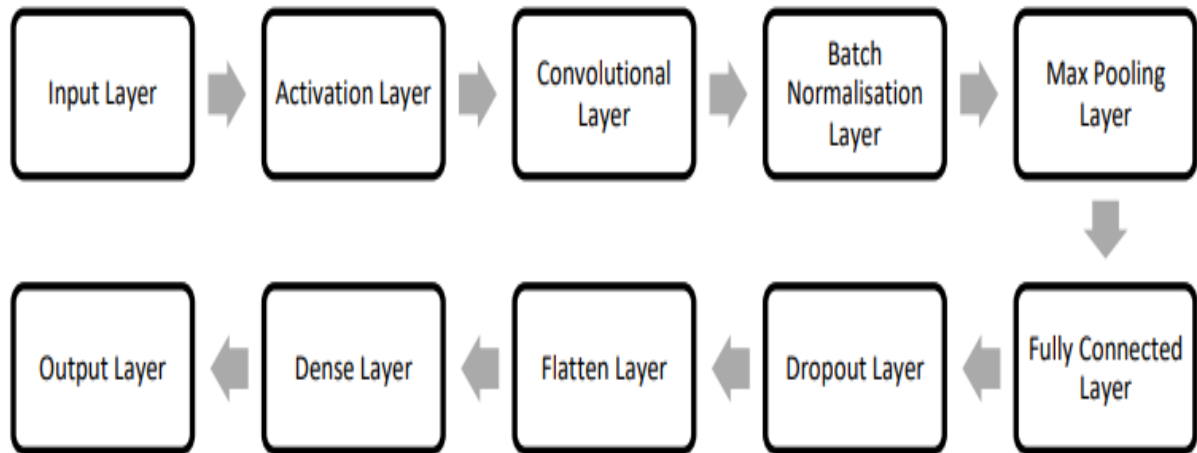
Convolution Neural Network

Fig:5.14 Diagram of a model trained from scratch using CNN architecture

Classifier models can be basically divided into two categories respectively which are generative models based on hand- crafted features and discriminative models based on traditional learning such as support vector machine (SVM), Random Forest (RF) and Convolutional Neural Network (CNN). One difficulty with methods based on hand-crafted features is that they often require the computation of a large number of features in order to be accurate when used with many traditional machine learning techniques. This can make them slow to compute and expensive memory-wise. More efficient techniques employ lower numbers of features, using dimensionality reduction like PCA (Principle Component Analysis) or feature selection methods, but the reduction in the number of features is often at the cost of reduced accuracy. Brain tumour segmentation employ discriminative models because unlike generative modelling approaches, these approaches exploit little prior knowledge on the brain 's anatomy and instead rely mostly on the extraction of [a large number of] low level image features, directly modelling the relationship between these features and the label of a given voxel.

In our project, we have used the Convolutional Neural Network architecture for Brain tumor Detection and Classification.

Convolutional neural network processes closely knitted data used for image classification, image processing, face detection etc. It is a specialised 3D structure with specialised NN analysing RGB layers of an image. Unlike others, it analyses one image at a time ,identifies and extracts important features and uses them to classify the image .Convolutional Neural Networks (ConvNets) automatically learns mid-level and high-level representations or abstractions from the input training data. The main building block used to construct a CNN architecture is the convolutional layer. It also consists of several other layers, some of which are described as bellow:

➢ Input Layer-It takes in the raw pixel value of input image
➢ Convolutional Layer- It is the first layer to extract features from an input image.Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel to generate a feature map Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters
➢ Activation Layer-It produces a single output based on the weighted sum of inputs
➢ Pooling Layer-Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling (also called subsampling or down sampling) reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:
  ➢ Max Pooling – taking the largest element in the feature map
  ➢ Average Pooling - taking the average of elements in the feature map
  ➢ Sum Pooling – taking the sum of all elements in the feature map
➢ Fully Connected Layer-The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network. the feature map matrix will be converted as column vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Forclassifying input image into various classes based on training set.
➢ Dropout Layer-It prevents nodes in a network from co-adapting to each other.

Advantages

➢ It is considered as the best ml technique for image classification due to high accuracy.

➢ Image pre-processing required is much less compared to other algorithms.

➢ It is used over feed forward neural networks as it can be trained better in case of complex images to have higher accuracies.

➢ It reduces images to a form which is easier to process without losing features which are critical for a good prediction by applying relevant filters and reusability of weights

➢ It can automatically learn to perform any task just by going through the training data i.e. there no need for prior knowledge

➢ There is no need for specialised hand-crafted image features like that in case of SVM, Random Forest etc.
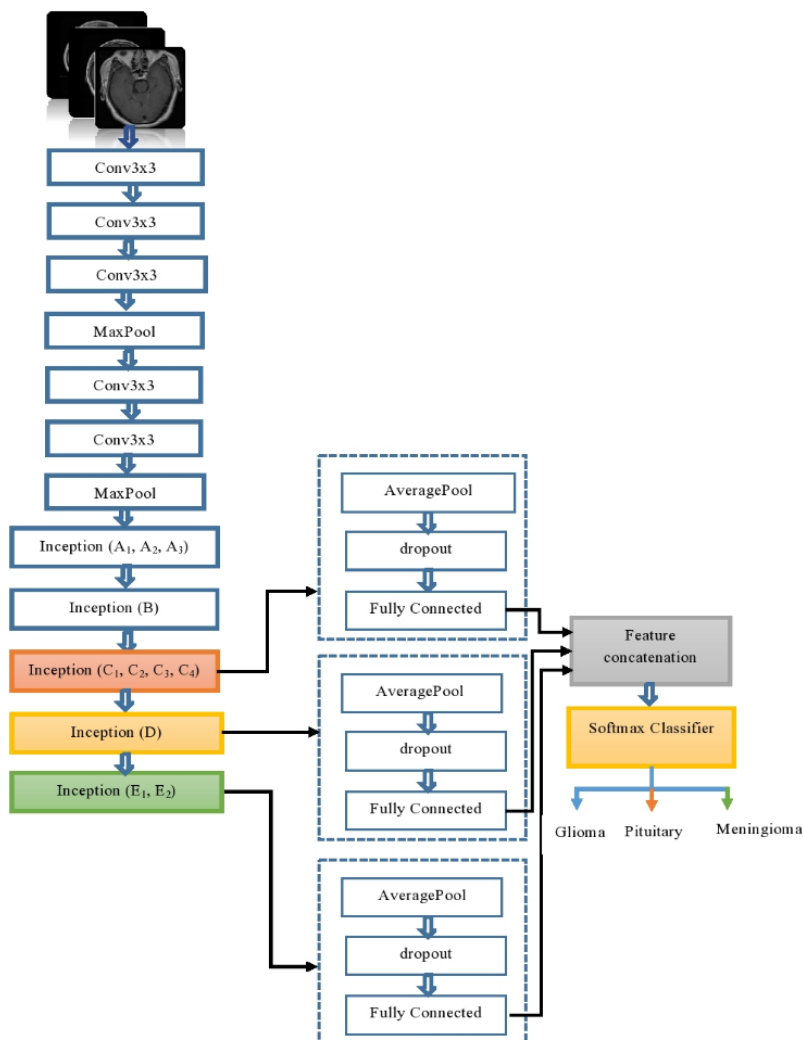

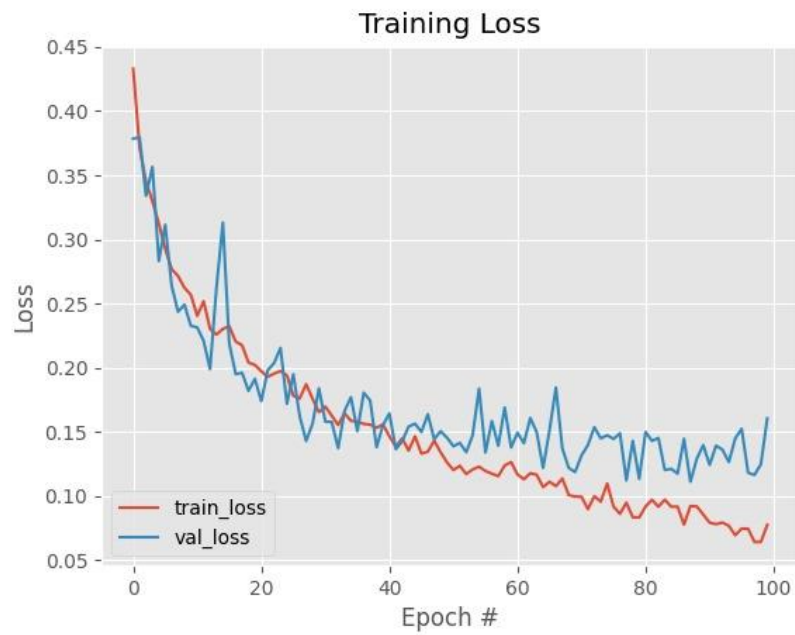**Fig 5.15   Model of pre-trained Inception-v3**

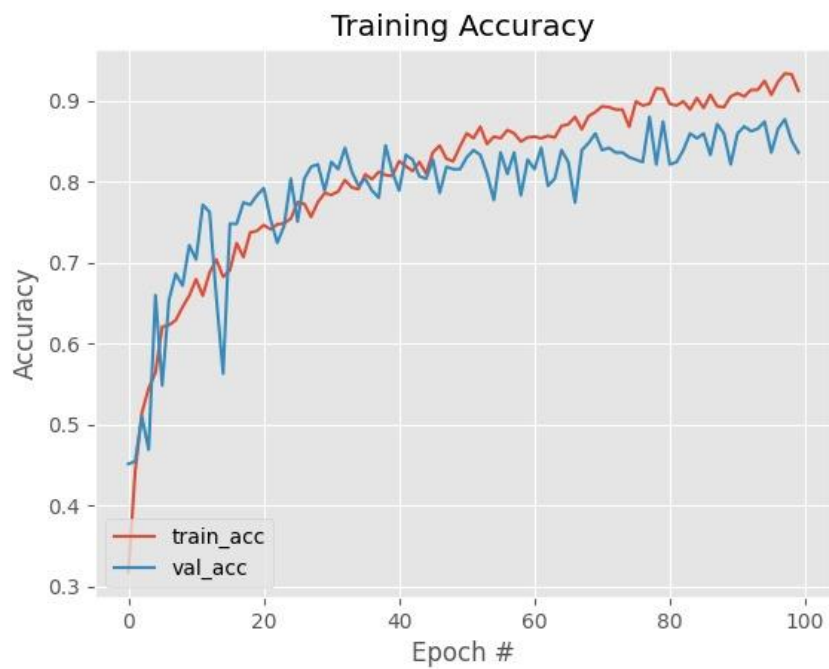**Fig 5.16 Graph determining the training loss**

Fig 5.17 Graph determining the training accuracy

5.5 Module 5: Testing

The evaluation of the trained network is done as follows. For each of the images in the testing set, it is presented to the network and asked to predict what it thinks the label of the image is. Then it is tabulated and gives the predictions of the model for an image in the testing set. Finally, these model predictions are compared to the ground-truth labels from the testing set. The ground-truth labels represent what the image category actually is. From there, it can compute the number of predictions the classifier got correct and compute aggregate reports such as precision, recall, and f-measure, which are used to quantify the performance of the network as a whole.

➢ **Pre-Train Test**

Pre-train tests allow you to catch bugs before running the model. This type of test is performed early on and doesn't need training parameters to run. Its main objective is to avoid wasted training jobs.

Some examples of tests include:

  ➢ Checking if any labels are missing in your training and validation datasets

  ➢ Check the single gradient to find the loss of data

➢ **Post-Train Test**

Post-train tests deal with job behavior. They are carried out on a trained model and check whether it performs correctly. The main purpose here is to investigate the logic behind the algorithm and detect any bugs that exist. There are three types of tests that can be used to report the behavior of the program:

  ➢ Invariant test– This testing technique allows us to check how much we can change the input data without affecting the performance of the machine learning model. We can use paired input examples (original and changed) and check for consistency in the model predictions. If the trained model performs correctly, the slight changes in the input should not affect the model predictions. For example, we can carry out a pattern recognition model on two pictures of oranges. We wouldn't expect the result to change between the two.

➤ Minimum functionality test– Minimum functionality tests are carried out similarly to unit tests in traditional software testing. The components of the program are split up, and testing is applied over those components. Minimum functionality testing allows you to assess the model performance based on specific cases found in your data. This allows you to identify critical instances where prediction errors can have serious consequences.

➤ Directional test– Unlike invariant tests, directional testing checks how perturbations in the input change the model behavior. If the trained model performs correctly, the changes in input will affect the model prediction. For example, if we had a model that estimates the price of a house, taking into account square footage, we would want to see that added space makes the house prices go up.

➤ **Test Set**

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. If the test set does contain examples from the training set, it will be difficult to assess whether the algorithm has learned to generalize from the training set or has simply memorized it.

A program that generalizes well will be able to effectively perform a task with new data. In contrast, a program that memorizes the training data by learning an overly complex model could predict the values of the response variable for the training set accurately, but will fail to predict the value of the response variable for new examples. Memorizing the training set is called over-fitting. A program that memorizes its observations may not perform its task well, as it could memorize relations and structures that are noise or coincidence. Balancing memorization and generalization, or over-fitting and under-fitting, is a problem common to many machine learning algorithms. Regularization may be applied to many models to reduce over-fitting.

In addition to the training and test data, a third set of observations, called a validation or hold-out set, is sometimes required. The validation set is used to tune variables called hyper parameters, which control how the model is learned. The program is still evaluated on the test set to provide an estimate of its performance in the real world; its performance on the validation set should not be used as an estimate of the model's real-world performance since the program has been tuned specifically to the validation data. It is common to partition a single set of supervised observations into training, validation, and test sets.

There are no requirements for the sizes of the partitions, and they may vary according to the amount of data available. It is common to allocate 50 percent or more of the data to the training set, 25 percent to the test set, and the remainder to the validation set.

However, machine learning algorithms also follow the maxim "garbage in, garbage out." Similarly, an algorithm trained on a large collection of noisy, irrelevant, or incorrectly labeled data will not perform better than an algorithm trained on a smaller set of data that is more representative of problems in the real world.

Many supervised training sets are prepared manually, or by semi-automated processes. Creating a large collection of supervised data can be costly in some domains. Fortunately, several datasets are bundled with scikit-learn, allowing developers to focus on experimenting with models instead.

During development, and particularly when training data is scarce, a practice called cross-validation can be used to train and validate an algorithm on the same data. In cross-validation, the training data is partitioned. The algorithm is trained using all but one of the partitions, and tested on the remaining partition. The partitions are then rotated several times so that the algorithm is trained and evaluated on all of the data.

Performance Measures − Bias and Variance

Many metrics can be used to measure whether or not a program is learning to perform its task more effectively. For supervised learning problems, many performance metrics measure the number of prediction errors.

There are two fundamental causes of prediction error for a model -bias and variance. Assume that you have many training sets that are all unique, but equally representative of the population. A model with a high bias will produce similar errors for an input regardless of the training set it was trained with; the model biases its own assumptions about the real relationship over the relationship demonstrated in the training data. A model with high variance, conversely, will produce different errors for an input depending on the training set that it was trained with.

A model with high bias is inflexible, but a model with high variance may be so flexible that it models the noise in the training set. That is, a model with high variance over-fits the training data, while a model with high bias under-fits the training data.

Ideally, a model will have both low bias and variance, but efforts to decrease one will frequently increase the other. This is known as the bias-variance trade-off. We may have to consider the bias-variance tradeoffs of several models introduced in this tutorial. Unsupervised learning problems do not have an error signal to measure; instead, performance metrics for unsupervised learning problems measure some attributes of the structure discovered in the data. Most performance measures can only be worked out for a specific type of task.

Machine learning systems should be evaluated using performance measures that represent the costs of making errors in the real world. While this looks trivial, the following example illustrates the use of a performance measure that is right for the task in general but not for its specific application.

Accuracy, Precision and Recall

Consider a classification task in which a machine learning system observes tumors and has to predict whether these tumors are benign or malignant. Accuracy, or the fraction of instances that were classified correctly, is an obvious measure of the program's performance. While accuracy does measure the program's performance, it does not make distinction between malignant tumors that were classified as being benign, and benign tumors that were classified as being malignant. In some applications, the costs incurred on all types of errors may be the same. In this problem, however, failing to identify malignant tumors is a more serious error than classifying benign tumors as being malignant by mistake.

We can measure each of the possible prediction outcomes to create different snapshots of the classifier's performance. When the system correctly classifies a tumor as being malignant, the prediction is called a true positive. When the system incorrectly classifies a benign tumor as being malignant, the prediction is a false positive. Similarly, a false negative is an incorrect prediction that the tumor is benign, and a true negative is a correct prediction that a tumor is benign. These four outcomes can be used to calculate several common measures of classification performance, like accuracy, precision, recall and so on.

Accuracy is calculated with the following formula:

$$ACC = (TP + TN)/(TP + TN + FP + FN)$$

Where, TP is the number of true positives

TN is the number of true negatives

FP is the number of false positives

FN is the number of false negatives.

Precision is the fraction of the tumors that were predicted to be malignant that are actually malignant. Precision is calculated with the following formula:

$$PREC = TP/(TP + FP)$$

Recall is the fraction of malignant tumors that the system identified. Recall is calculated with the following formula:

$$R = TP/(TP + FN)$$

Precision measures the fraction of tumors that were predicted to be malignant that are actually malignant. Recall measures the fraction of truly malignant tumors that were detected. The precision and recall measures could reveal that a classifier with impressive accuracy actually fails to detect most of the malignant tumors. If most tumors are benign, even a classifier that never predicts malignancy could have high accuracy. A different classifier with lower accuracy and higher recall might be better suited to the task, since it will detect more of the malignant tumors. Many other performance measures for classification can also be used.

5.6 UML diagram

5.6.1 Use case diagram

Use case consists of user and processor where user is used to provide the input to the system and processor is used to process the input data and provide output. The flow is shown in the above diagram. First user as to run the system and run the code, model and library packages are imported and loaded. After the run of code GUI is being displayed and click on select file and load the test image. After loading the image, click in prediction button to analyze the image and to give predicted output and displayed.
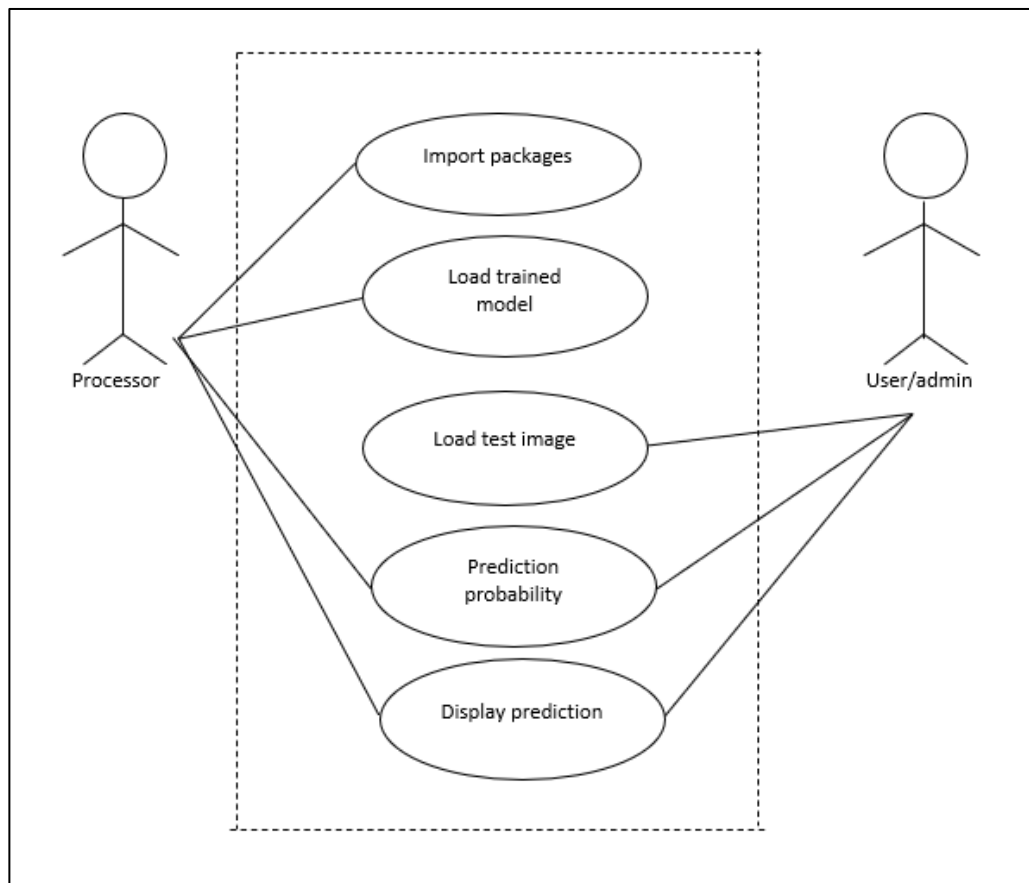
Fig 5.18   Use Case diagram

5.6.2 Sequence Diagram

Sequence diagram consists of 5 different blocks namely user, processor, memory, Model and labels. User will provide the input image through the files already saved image is being taken in consideration which is been captured and sent to the processor where pre-processing of data is done which is resizing, reshaping and other parameters and after that those are stored in the

memory unit. After pre-processing and storing of image, CNN trained model file is loaded where the featured of the image is extracted for classifying the output.

After classifying the output, label is provided such as glioma tumor, meningioma tumor, no tumor and pituitary tumor.
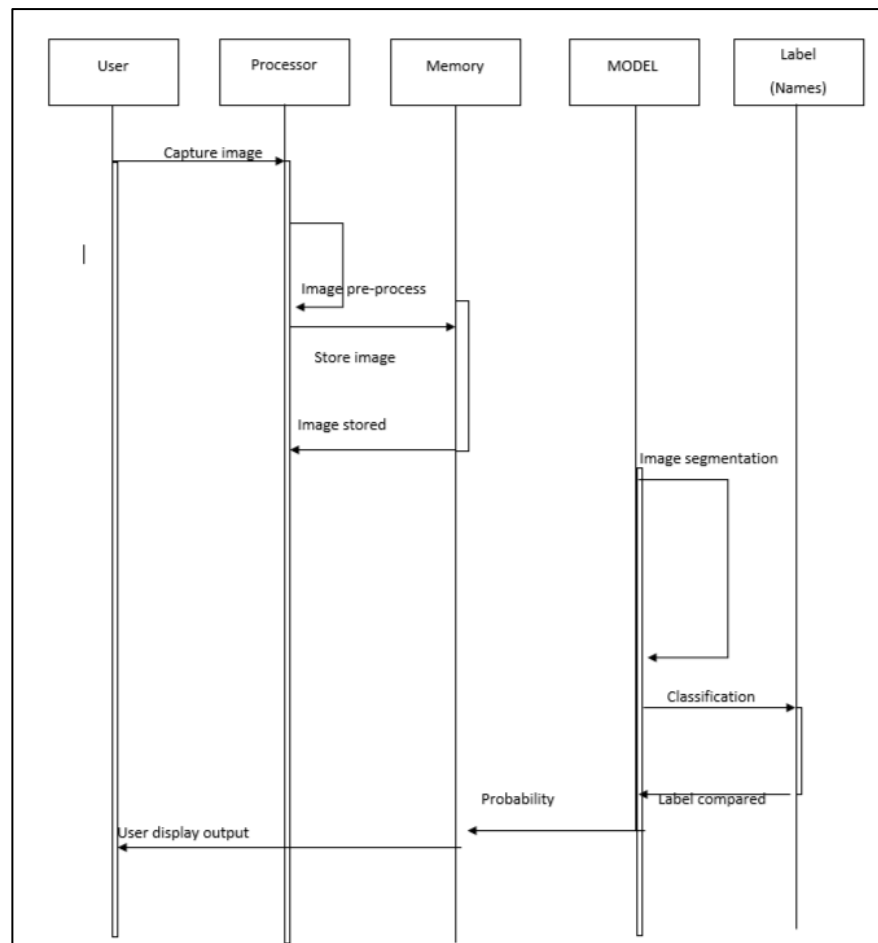

Fig 5.19   Sequence Diagram

5.6.3 Data Flow Diagram

Data Flow Diagram 0 provides u the content diagram overview of the whole system. It is designed to be an at- a-glance view, showing the system as single high-level process. Here from the file image is be loaded to the application where the loaded image is sent to classification unit to predict the result with the help of CNN model file.
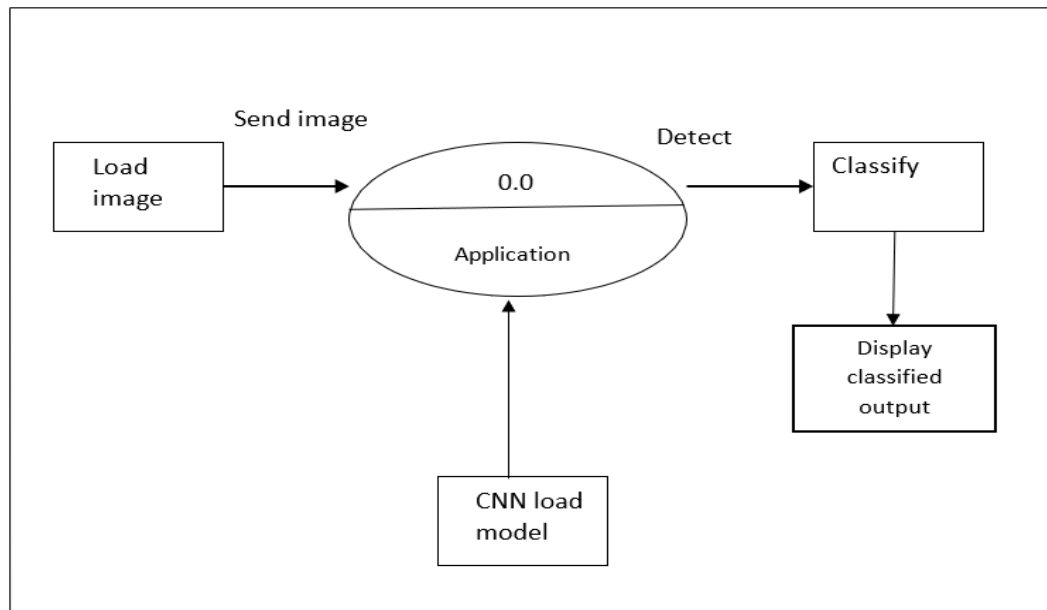
Fig 5.20 Data Flow Diagram 0

Fig 5.21 Data Flow Diagram 1 is the representation of DFD1. The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Here from the file image is be loaded to the application where the loaded image is sent to classification unit to predict the result and classes are classified given a label as glioma tumor, meningioma tumor, no tumor and pituitary tumor.
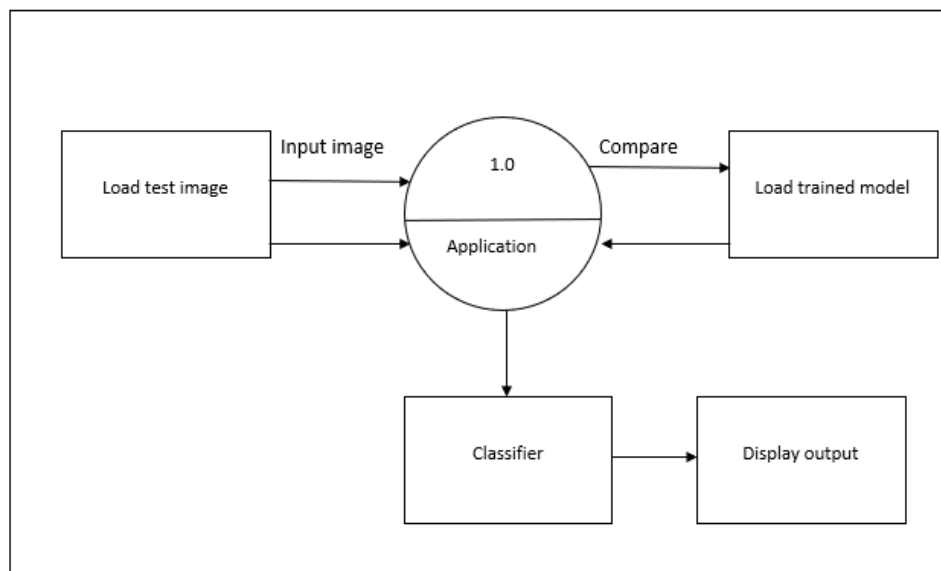
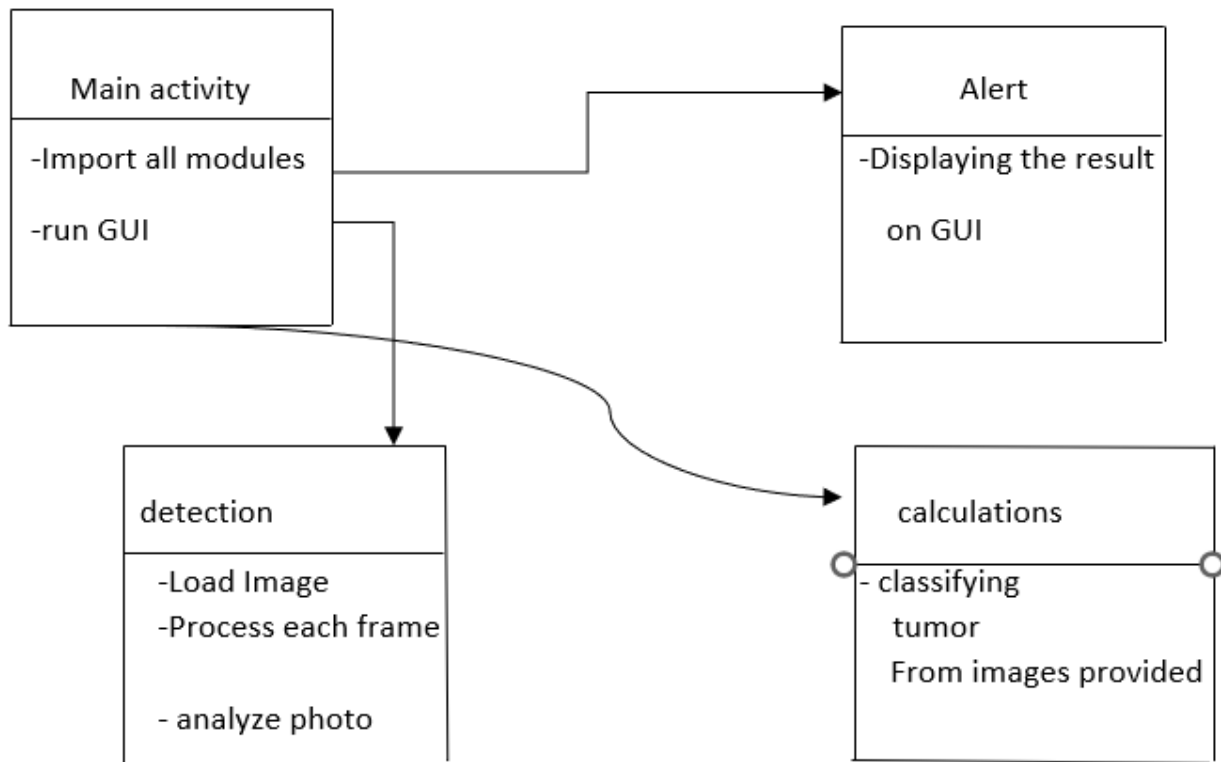**Fig 5.21 Data Flow Diagram 1**

5.6.4   Class Diagram

Fig 5.22   Class Diagram

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 TESTING

6.1.1 GUI LAYOUT Test case:

Test Case	1
Name of Test	GUI layout
Input	Resolution(width, height)
Expected output	Display window specified by user resolution
Actual output	User specified window is displayed
Result	Successful

6.1.2 Input button's Test case:

Test Case	1
Name of Test	Input button
Input	Number of buttons and there position in the window(column, row)
Expected output	Display button's in specified column, row
Actual output	Buttons are displayed in the specified position
Result	Successful

6.1.3 Input image selection Test case:

Test Case	1
Name of Test	**Input image selection**
Input	Image files present in folders
Expected output	When user click select image , it should prompt user to select image from directory
Actual output	Select directory for image's was displayed
Result	Successful

6.1.4 Display image in GUI Test case:

Test Case	1
Name of Test	**Display image in GUI**
Input	User selected input image
Expected output	Display user select image in GUI
Actual output	User Selected image displayed in GUI
Result	Successful

6.1.5 Predict/Analysis Button Test case:

Test Case	1
Name of Test	**Predict button**
Input	User selected input image
Expected output	Display predicted output in the GUI
Actual output	Predicted output was displayed
Result	Successful

6.1.6 Quit button Test case:

Test Case	1
Name of Test	**Quit button**
Input	User input / click
Expected output	To kill or destroy GUI window
Actual output	Window was killed when user click's exit button
Result	Successful

6.2 RESULTS
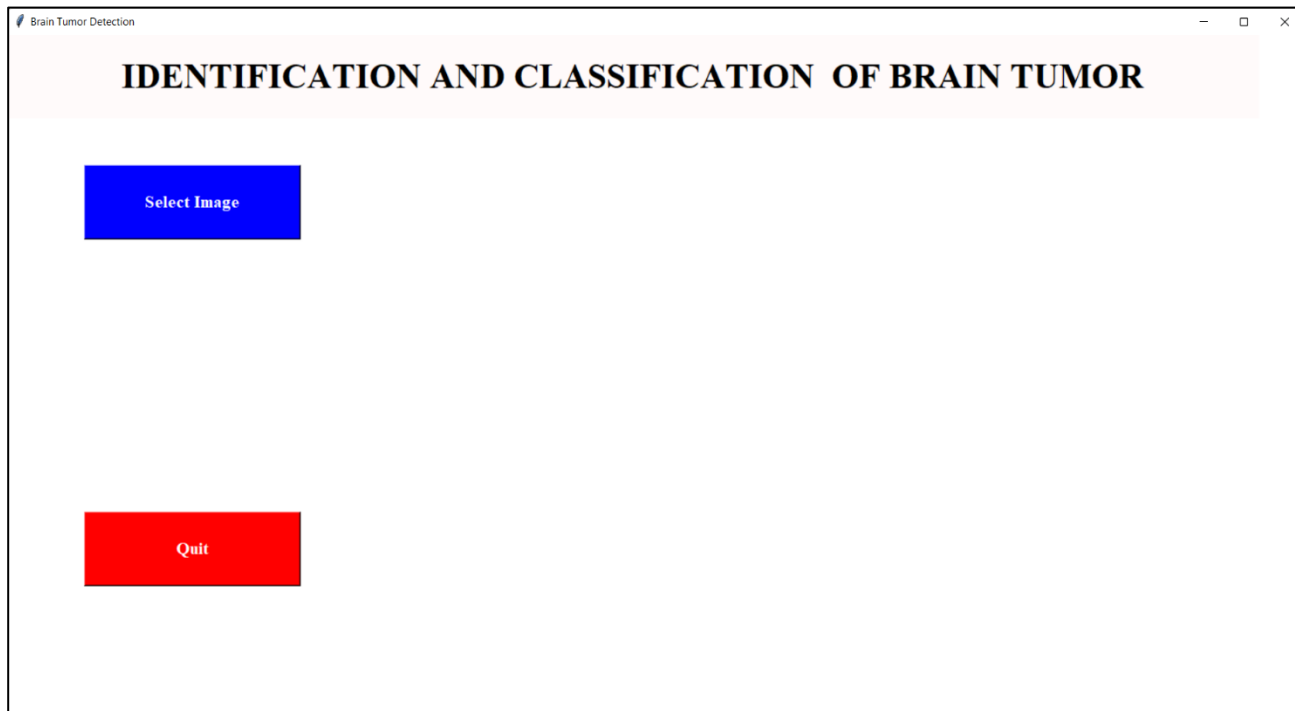
Fig 6.1 Home Page
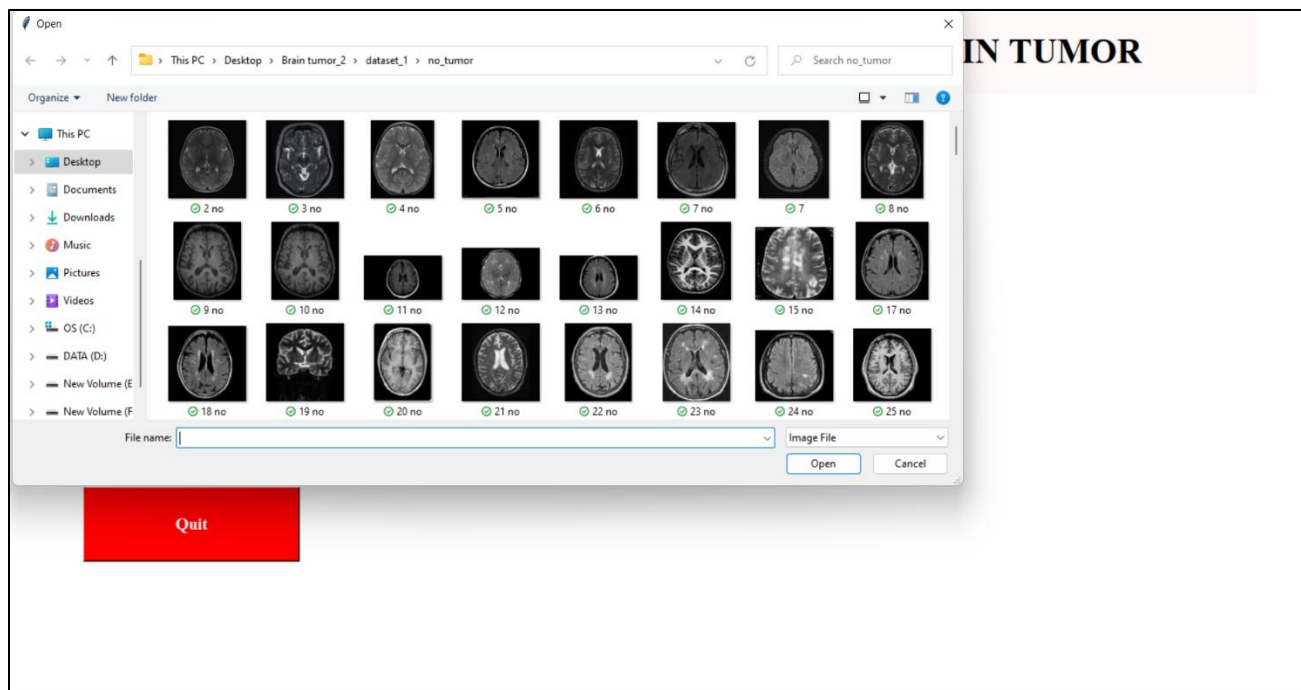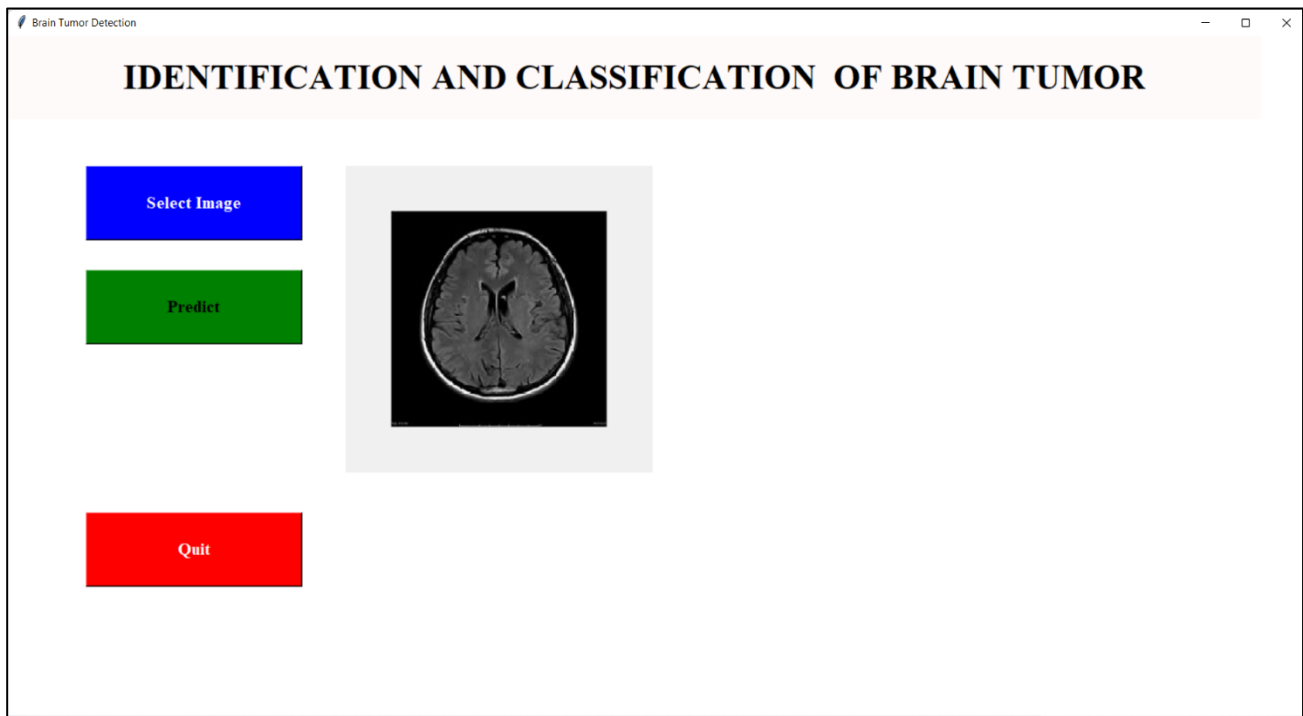

Fig 6.2 Select MRI scan
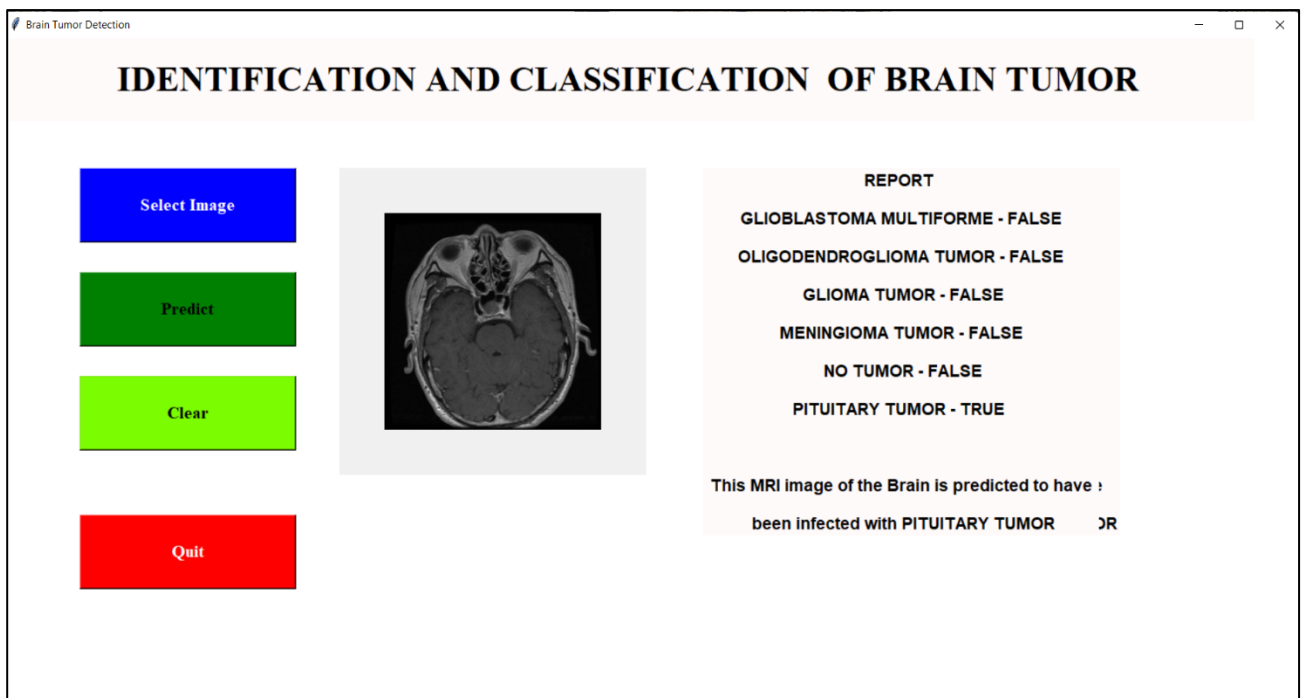
Fig 6.3 MRI scan fed to the system

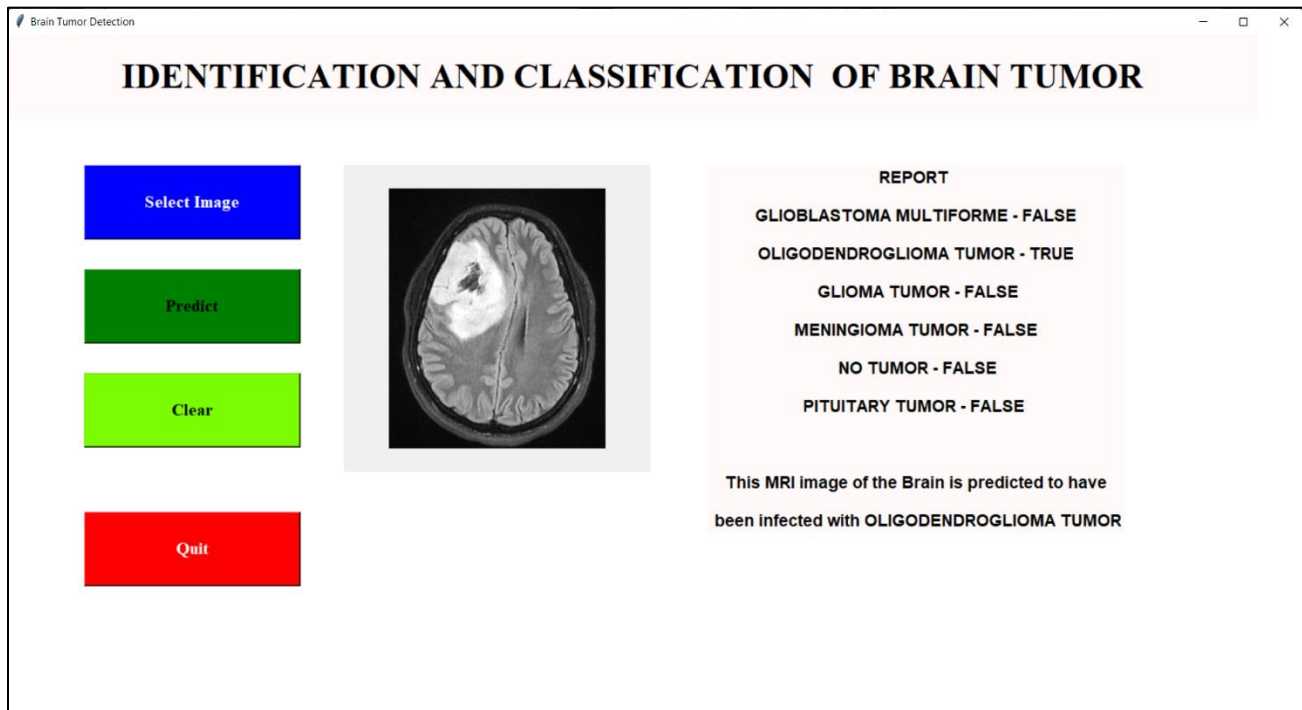**Figure 6.4. Prediction of Pituitary Brain Tumor**

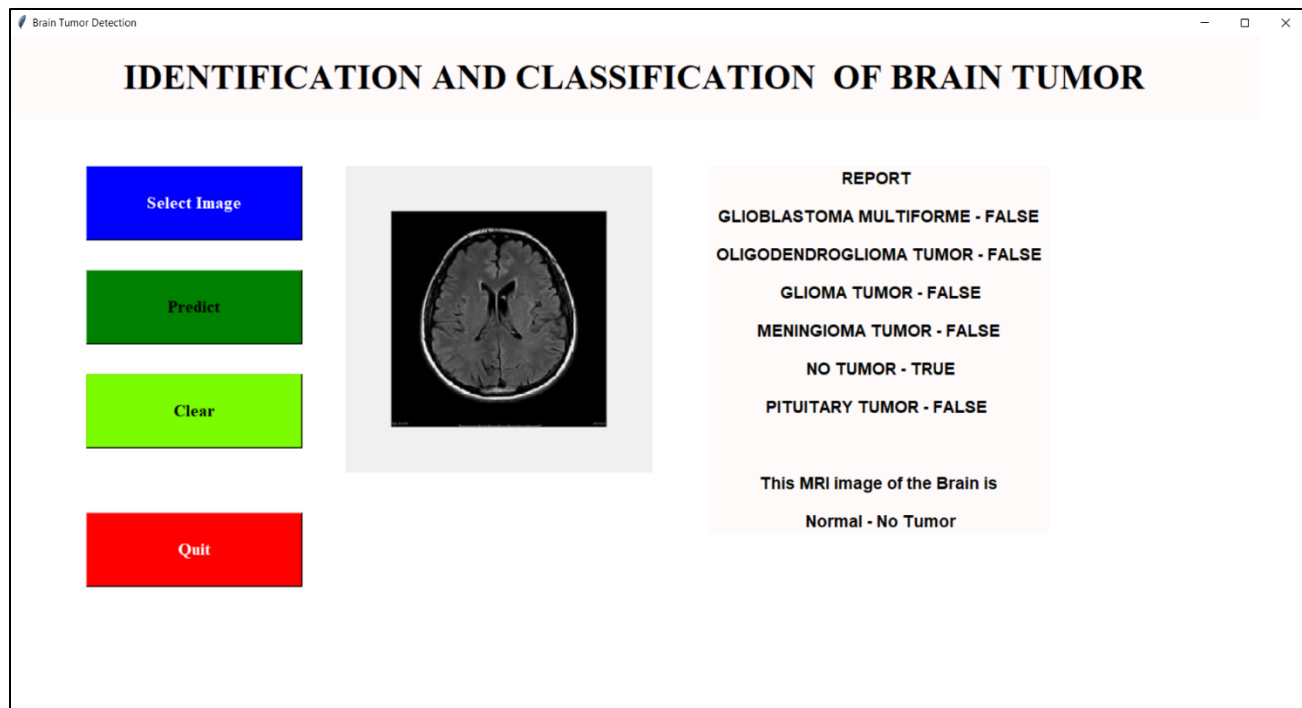**Figure 6.5. Prediction of Oligodendroglioma Tumor**

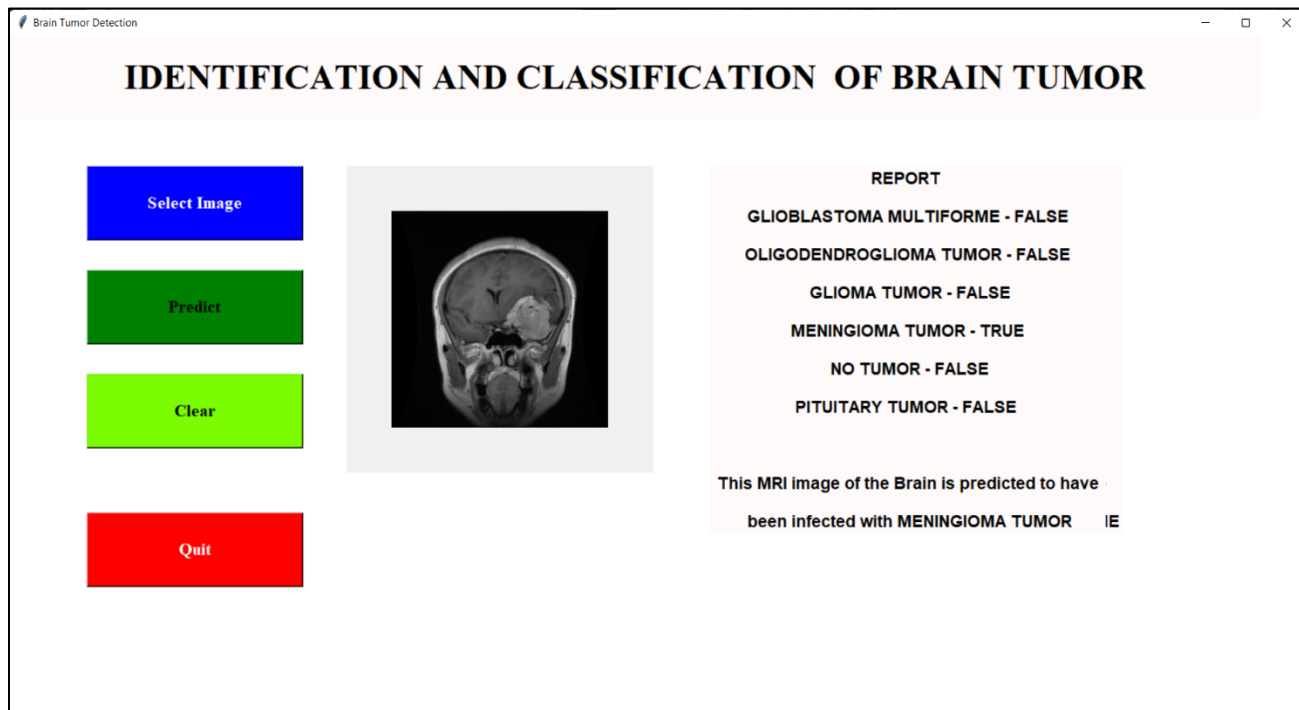**Figure 6.6. Prediction of No Tumor MRI scans**
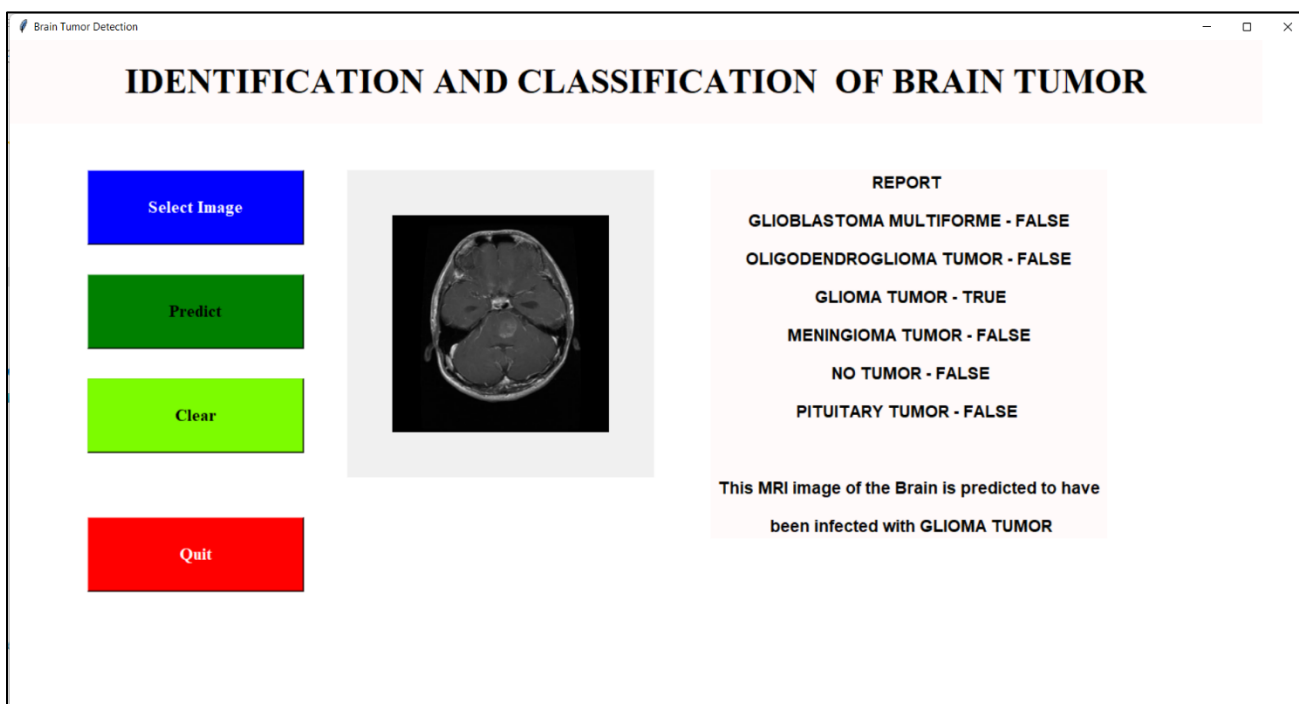
Figure 6.7. Prediction of Meningioma Tumor

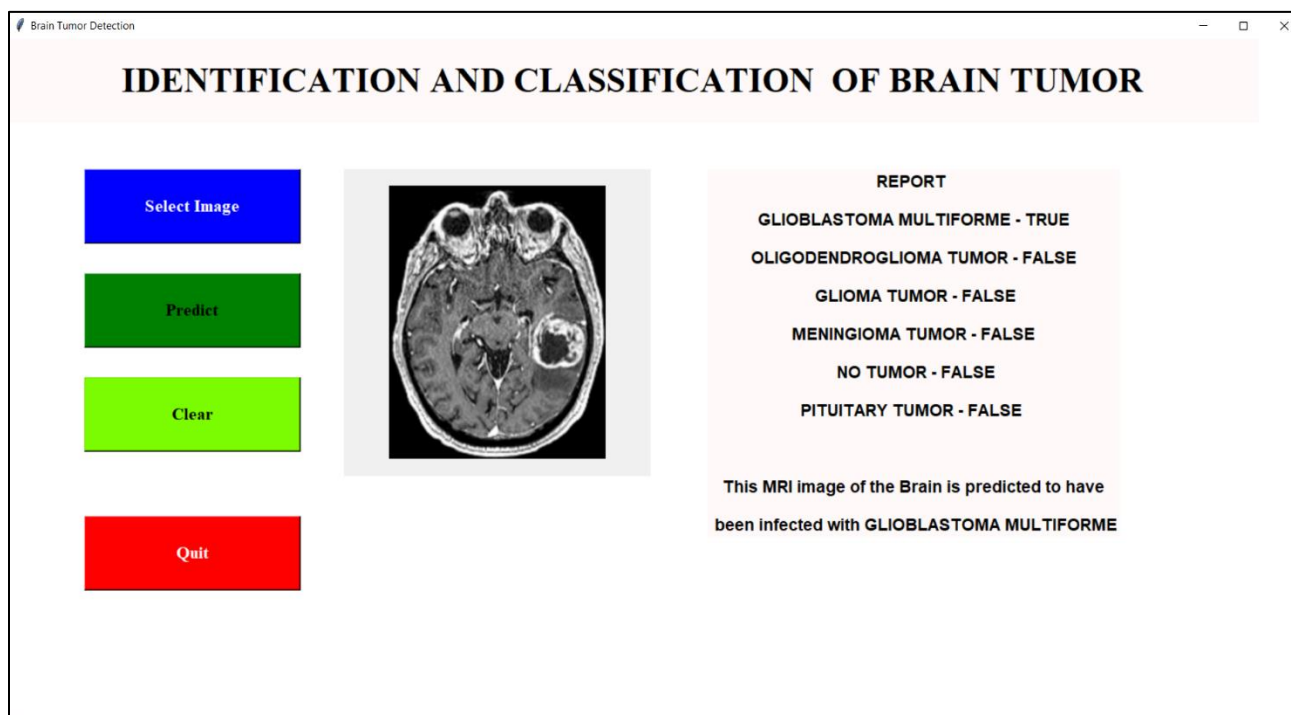**Figure 6.8. Prediction of Glioma Tumor**

Figure 6.9 Prediction of Glioblastoma Multiforme Tumor

CHAPTER 7

CONCLUSION AND FUTURE WORKS

CONCLUSION

Brain tumors are relatively rare compared, that is, 1.4% of new cases per year, in developed countries. Deaths from brain tumors have increased in the last few decades. Therefore, the scope of this project is expanded. Brain tumors, especially those that are harmful, are considered to be incurable and deadly. The need for early detection stems from the fact that brain tumors may have invisible and frightening symptoms at first. To identify the type of tumor in the brain, a dangerous medical procedure was performed and a biopsy was performed. This project proposes a way to detect and diagnose the type of brain tumor that is based on MRI (Magnetic Resonance Imaging) of a patient-fed scan of the system. It also provides an accurate diagnosis of existing brain tumors based on CNN classification algorithm. Biopsy can be greatly avoided due to the precise discharge of the system. This is considered to be the least expensive way to solve a brain tumor problem. Previous systems have used various algorithms and methods to prove the existence of a brain tumor by machine learning. This system provides a clear picture of the precise tumor present in the patient by considering various factors such as location, size etc. This method proves to be an accurate and effective method of diagnosing a brain tumor problem.

FUTURE SCOPE

Build an app-based user interface in hospitals which allows doctors to easily determine the impact of tumor and suggest treatment accordingly Since performance and complexity of ConvNets depend on the input data representation we can try to predict the location as well as stage of the tumor from Volume based 3D images.

By creating three dimensional (3D) anatomical models from individual patients, training, planning and computer guidance during surgery is improved. Using VolumeNet with LOPO (Leave-One-Patient-Out) scheme has proved to give a high training as well as validation accuracy(>95%).

In LOPO test scheme, in each iteration, one patient is used for testing and remaining patients are used for training the ConvNets, this iterates for each patient.

Although LOPO test scheme is computationally expensive, using this we can have more training data which is required for ConvNets training. LOPO testing is robust and most applicable to our application, where we get test result for each individual patient.

So, if classifier misclassifies a patient, then we can further investigate it separately. Improve testing accuracy and computation time by using classifier boosting techniques like using more number images with more data augmentation, fine-tuning hyper parameters, training for a longer time i.e. using more epochs, adding more appropriate layers etc.. Classifier boosting is done by building a model from the training data then creating a second model that attempts to correct the errors from the first model for faster prognosis. Such techniques can be used to raise the accuracy even higher and reach a level that will allow this tool to be a significant asset to any medical facility dealing with brain tumors. For more complex datasets, we can use U-Net architecture rather than CNN where the max pooling layers are just replaced by up sampling ones. Ultimately we would like to use very large and deep convolutional nets on video sequences where the temporal structure provides very helpful information that is missing or far less obvious in static images. Unsupervised transfer learning may attract more and more attention in the future.

REFERENCES

[1] Jakesh Bohaju, (2020, July). Brain Tumor, Version 3. Retrieved September 22, 2020, from https://www.kaggle.com/jakeshbohaju/brain-tumor.

[2] Navoneel Chakrabarty, (2019, April). Brain MRI Images for Brain Tumor Detection, Version 1. Retrieved September 18, 2020, from https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection.

[3] Sartaj, (2020, May). Brain Tumor Classification (MRI), Version 2. Retrieved September 23, 2020 from https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

[4] Nilesh Bhaskarrao Bahadure, Arun Kumar Ray, Har Pal 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, Article ID 9749108, 12 pages, 2017. https://doi.org/10.1155/2017/9749108.

[5] Arakeri, M.P., Ram Mohana Reddy, G. An intelligent content-based image retrieval system for clinical decision support in brain tumor diagnosis. Int J Multimed Info Retr 2, 175–188 (2013). https://doi.org/10.1007/s13735-013-0037-5

6] Islam, Jyoti, and Yanqing Zhang. "A novel deep learning based multiclass classification method for Alzheimer's disease detection using brain MRI data." International conference on brain informatics. Springer, Cham, 2017.

[7] D. Sridhar and I. Murali Krishna, "Brain Tumor Classification using Discrete Cosine Transform and Probabilistic Neural Network," 2013 International Conference on Signal Processing, Image Processing & Pattern Recognition, Coimbatore, 2013, pp. 92-96, doi: 10.1109/ICSIPR.2013.6497966.

[8] Sachdeva J, Kumar V, Gupta I, Khandelwal N, Ahuja CK. Segmentation, feature extraction, and multiclass brain tumor classification. Journal of Digital Imaging. 2013 Dec;26(6):1141-1150. DOI: 10.1007/s10278-013-9600-0.

[9] Suhara, S. Fathima, and M. Safish Mary. "Fully Connected Pyramid Pooling Network (FCPPN)–A Method For Brain Tumor Segmentation." 11.3 : 1457 (2018).

[10] Deepak, S., and P. M. Ameer. "Brain tumor classification using deep CNN features via transfer learning." Computers in biology and medicine 111 (2019): 103345.

[12] Wu, Wentao, et al. "An Intelligent Diagnosis Method of Brain MRI Tumor Segmentation Using Deep Convolutional Neural Network and SVM Algorithm." Computational and Mathematical Methods in Medicine 2020 (2020).

[14] Demirhan, Ays¸e, Mustafa Tor¨ u, and ¨ ˙Inan Guler. "Segmentation of tumor ¨ and edema along with healthy tissues of brain using wavelets and neural networks." IEEE journal of biomedical and health informatics 19.4 (2014): 1451-1458.

[16] Aneja, Deepali, and Tarun Kumar Rawat. "Fuzzy clustering algorithms for effective medical image segmentation." International Journal of Intelligent Systems and Applications 5.11 (2013).

[17]Yuehao Pan, Weimin Huang et.al "Brain Tumor Grading based on Neural networks and Convolutional Neural Networks",IEEE.

[18] Darko Zikic, Yani Ioannou et.al "Segmentation of Brain tumor tisssues with convolutional neural networks", MiCCAI,2014,Brats challenge.

[19] Karen, Zisserman "Very deep convolutional networks for large scale image recognition",ICLR.[Published in 2014].

[20] Taranjit Kaur, Tapan Kuamar Gandhi, "Automated brain image classification based on VGG-16 and tranfer learning",ICIT [Published in 2019].