

ENHANCED ACCURACY IN PREDICTION OF BRAIN TUMOR GROWTH AND LONGITUDINAL BRAIN MRI SEGMENTATION

Submitted in partial fulfilment of the requirements for the award of a Bachelor of Technology degree in Biomedical Engineering.

By,

DARREN AKASH J (39240013)

SANTHIP SHANMUGAM A (39240035)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DARREN AKASH J (39240019)**, **SANTHIP SHANMUGAM A (39240035)** who carried out the project entitled **"ENHANCED ACCURACY IN PREDICTION OF BRAIN TUMOR GROWTH AND LONGITUDINAL BRAIN MRI SEGMENTATION"** under my supervision from **October 2022 to April 2023**.

Internal Guide

(Dr. J. PREMKUMAR, M.Sc., Ph.D.,)

Head of the Department

(Dr. T. SUDHAKAR, M.Sc., Ph.D.,)

HEAD OF THE DEPARTMENT
DEPARTMENT OF BIOMEDICAL ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Kodambakkam Nagar, Rajiv Gandhi Salai
Chennai - 600 079

Submitted for Viva-voce Examination held on 18-04-2023

Internal Examiner

External Examiner

DECLARATION

We, **DARREN AKASH J (39240019)** and **SANTHIP SHANMUGAM A (39240035)** hereby declare that the Project Report entitled “**ENHANCED ACCURACY IN PREDICTION OF BRAIN TUMOR GROWTH AND LONGITUDINAL BRAIN MRI SEGMENTATION**” done by us under the guidance of **Dr. J. PREMKUMAR**, Department of Biomedical Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

DATE: 18/04/2023

PLACE: Chennai



1. Darren Akash J



2. Santhip Shanmugam A

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **the Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sudhakar, Head of the Department, and Dept. of Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project **Guide Dr.J. Premkumar, Department of Biomedical Engineering** for his valuable guidance, suggestions, and constant encouragement paving the way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching staff members of the Department of Biomedical Engineering** who were helpful in many ways for the completion of the Project.

ABSTRACT

The brain tumors, are the most common and aggressive disease, leading to a very short life expectancy in their highest grade. Thus, treatment planning is a key stage to improve the quality of life of patients. Generally, various image techniques such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and ultrasound image are used to evaluate the tumor in a brain, lung, liver, breast, prostate...etc. Especially, in this work MRI images are used to diagnose tumor in the brain. However the huge amount of data generated by MRI scan thwarts manual classification of tumor vs. non-tumor in a particular time. But it having some limitation (i.e.) accurate quantitative measurements is provided for limited number of images. Hence trusted and automatic classification scheme are essential to prevent the death rate of human. The automatic brain tumor classification is very challenging task in large spatial and structural variability of surrounding region of brain tumor. In this work, automatic brain tumor detection is proposed by using Convolutional Neural Networks (CNN) classification.

Keywords - Tumor, Joint Label Fusion, Glioma, Tumor Segmentation, MRI, Tumor Prediction, Tumor Growth, Segmentation.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF ABBREVIATION	iv
	LIST OF FIGURES	v
	LIST OF TABLES	vi
1	INTRODUCTION	01
	1.1 BASIC INTRODUCTION	01
	1.2 BRAIN AND BRAIN TUMOR	02
	1.3 MODULES	04
	1.3.1 DATA COLLECTION	04
	1.3.2 IMAGE PROCESSING	04
	1.3.3 SEGMENTATION	04
	1.3.4 CLASSIFICATION	04
2	LITERATURE REVIEW	05
3	AIM AND SCOPE	10
	3.1 AIM	10
	3.2 OBJECTIVE	10
	3.3 MOTIVATION	11
	3.4 EXISTING METHOD	11
	3.4.1 DISADVANTAGES	11
	3.5 PROPOSED METHOD	12
	3.5.1 ADVANTAGES	12
4	MATERIALS AND METHOD	13
	4.1 ARTIFICIAL INTELLIGENCE	13
	4.2 MACHINE LEARNING	13
	4.2.1 FEATURES OF MACHINE LEARNING	14
	4.2.2 CLASSIFICATION OF MACHINE LEARNING	14
	4.3 DEEP LEARNING	15

	4.3.1 ARCHITECTURE OF DEEP LEARNING	16
	4.4 CONVOLUTIONAL NEURAL NETWORK (CNN)	16
	4.4.1 TYPES OF LAYERS	18
	4.5 DENSE LAYER	19
	4.6 SYSTEM REQUIREMENTS	20
	4.7 REQUIREMENT ANALYSIS	20
	4.7.1 PYTHON	20
	4.7.2 ANACONDA	21
	4.7.3 ANACONDA NAVIGATOR	22
	4.7.4 JUPYTER NOTEBOOK	22
	4.8 SYSTEM ARCHITECTURE	24
	4.9 SYSTEM MODULES	24
	4.9.1 MODULE 1: DATASET COLLECTION	25
	4.9.2 MODULE 2: PREPROCESSING	25
	4.9.3 MODULE 3: SEGMENTATION	26
	4.9.4 MODULE 4: CLASSIFICATION	27
	4.10 FEASIBILITY STUDY	27
	4.10.1 TYPES OF FEASIBILITY STUDY	28
	4.11 TESTING	29
	4.11.1 TYPES OF TESTING	29
	4.12 IMPLEMENTATION	31
	4.12.1 PYTHON	31
	4.12.2 JUPYTER NOTEBOOK	32
	4.12.3 ANACONDA	33
5	RESULTS	35
	5.1 COLLECTED DATASET	35
	5.2 SEGMENTATION	38
	5.3 TUMOR PREDICTION OUTPUT	39
6	CONCLUSION	41
	REFERENCES	
	APPENDIX	

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANTION

ACC	Accuracy
AC	Active Contour
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CT	Computed Tomography
DL	Deep Learning
DNN	Deep Neural Network
DBN	Deep Belief Network
FNR	False Negative Rate
HID	Hybrid Image Processing
IDM	Inverse Difference Movement
JFL	Joint Label Fusion
ML	Machine Learning
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
PET	Positron Emission Tomography
RELU	Rectified Linear Unit
RF	Radio Frequency
RMS	Root Mean Square
ROI	Region of Interest
SE	Shannon's Entropy
SGO	Social Group Optimization
SRG	Seed Region Growing

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE-NO
1.1	BASIC STRUCTURE OF HUMAN BRAIN	02
4.1	WORKING OF MACHINE LEARNING ALGORITHM	14
4.2	LAYERS IN CNN	18
4.3	SYSTEM MODULES	25
5.1	TRAINING DATASET	35
5.2	TESTING DATASET	36
5.3	VALIDATION DATASET	36
5.4	SAMPLE NEGATIVE BRAIN MRI SCANS	37
5.5	SAMPLE POSITIVE BRAIN MRI SCANS	37
5.6	SEGMENTATION PROCESS	38
5.7	SEGMENTATION OUTPUT	38
5.8	LONGITUDINAL TUMOR SEGMENTATION RESULTS	39
5.9	OUTPUT FOR NORMAL BRAIN MRI (NOT AFFECTED)	39
5.10	OUTPUT FOR ABNORMAL BRAIN MRI (AFFECTED)	40

LIST OF TABLES

TABLE NO	NAME OF TABLE	PAGE-NO
1.1	TYPES OF BRAIN TUMORS	03
4.6.1	SYSTEM REQUIREMENTS	21

CHAPTER 1

INTRODUCTION

1.1 BASIC INTRODUCTION

Brain tumor is one of the vital organs in the human body, which consists of billions of cells. The abnormal group of cell is formed from the uncontrolled division of cells, which is also called as tumor. Brain tumor are divided into two types such low grade (grade1 and grade2) and high grade (grade3 and grade4) tumor. Low grade brain tumor is called as benign. Similarly, the high grade tumor is also called as malignant. Benign tumor is not cancerous tumor. Hence it doesn't spread other parts of the brains. However the malignant tumor is a cancerous tumor. So it spreads rapidly with indefinite boundaries to other region of the body easily. It leads to immediate death. Brain MRI image is mainly used to detect the tumor and tumor progress modeling process. This information is mainly used for tumor detection and treatment processes. MRI image gives more information about given medical image than the CT or ultrasound image. MRI image provides detailed information about brain structure and anomaly detection in brain tissue. Actually, Scholars offered unlike automated methods for brain tumors finding and type cataloging using brain MRI images from the time when it became possible to scan and freight medical images to the computer. Conversely, Neural Networks (NN) and Support Vector Machine (SVM) are the usually used methods for their good enactment over the most recent few years¹¹. However freshly, Deep Learning (DL) models fixed a stirring trend in machine learning as the subterranean architecture can efficiently represent complex relationships without needing a large number of nodes like in the superficial architectures e.g. K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). Consequently, they grew quickly to become the state of the art in unlike health informatics areas for example medical image analysis, medical informatics and bio-informatics.

1.2 BRAIN AND BRAIN TUMOR

The human brain is the central part of the nervous system, and it is situated inside the human head, protected by the skull. Its primary function is to regulate and control all parts of the human body, enabling us to adapt and withstand various environmental conditions. As a complex organ, the human brain facilitates human actions and enables us to express our thoughts and emotions. To understand the fundamental aspects of the brain, it's necessary to explore its structure.

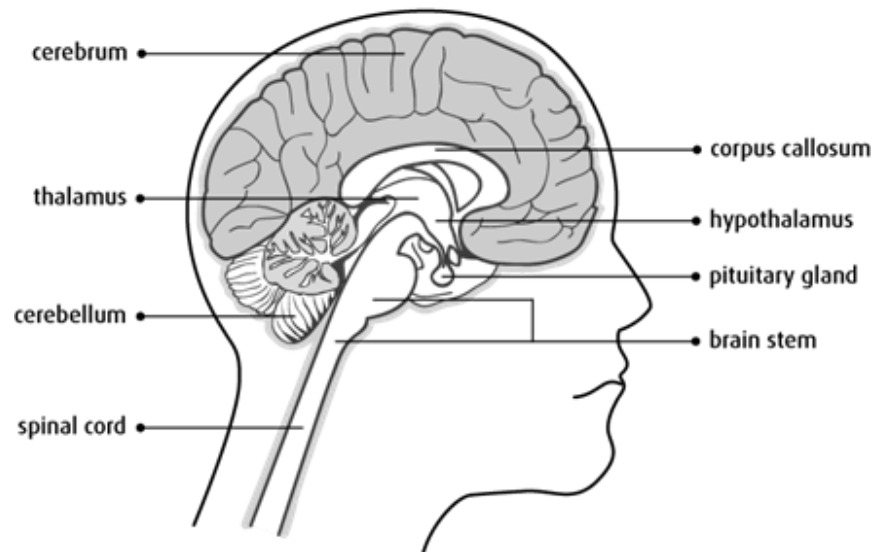


FIG 1.1 BASIC STRUCTURE OF HUMAN BRAIN

(Source credit: Canada cancer society)

A brain tumor is an abnormal growth of cells in the brain. Tumors can be either benign (noncancerous) or malignant (cancerous), and they can originate from different types of cells in the brain. Brain tumors can cause a variety of symptoms depending on their location and size, including headaches, seizures, vision problems, and cognitive impairments. The treatment for a brain tumor depends on its type, location, and other factors, and may include surgery, radiation therapy, chemotherapy, and other treatments. Regular monitoring and follow-up care is important for people who have been diagnosed with a brain tumor, as some types of tumors can recur even after treatment. The exact cause of most brain tumors is unknown, but some risk factors have been identified, including exposure to radiation, a family history of brain tumors, and certain genetic conditions. Symptoms can vary depending on the location and size

of the tumor, and may include headaches, seizures, vision problems, cognitive problems, speech difficulties, balance problems, and personality changes. Diagnosis is usually made using a combination of imaging tests and a biopsy. Treatment options depend on the type and location of the tumor, and may include surgery, radiation therapy, chemotherapy, targeted therapy, immunotherapy, and supportive care. Prognosis varies widely depending on the type of tumor and other factors, and regular monitoring and follow-up care is important for people who have been diagnosed with a brain tumor. While not all brain tumors are cancerous or life-threatening, any changes in cognitive function, behavior, or physical function that persist or worsen should be evaluated by a medical professional to determine the cause and appropriate course of action.

Table 1.1 Types of Brain Tumors

Type of Brain Tumor	Origin	Common Symptoms
Gliomas	Glial cells	Headaches, seizures, cognitive problems
Meningiomas	Meninges	Headaches, vision problems, seizures
Pituitary adenomas	Pituitary gland	Hormonal imbalances, vision problems
Medulloblastomas	Cerebellum	Headaches, balance problems, vomiting
Schwannomas	Schwann cells	Hearing loss, tinnitus, balance problems
Craniopharyngiomas	Pituitary gland	Hormonal imbalances, vision problems, headaches

1.3 MODULES

1.3.1 Data Collection

We have chosen to use a publicly-available Healthcare dataset which contains a relatively small number of inputs and cases. The data is arranged in such a way that will allow those trained in medical disciplines to easily draw parallels between familiar statistical and novel ML techniques. Additionally, the compact dataset enables short computational times on almost all modern computers.

1.3.2 Image processing

The preprocessing step involves reduction, aligning, and bias field correction. We use the CNN method that used for improve the accuracy and to reduce the computation time. CNN is one of the deep learning methods, which contains sequence of feed forward layers.

1.3.3 Segmentation

Image segmentation is the process of dividing the image into non- overlapping meaningful regions. The main objective if an image segmentation is to divide an image into many sections for the further analysis, so we can get the only necessary or a segment of information. We use various image segmentation algorithms to split and group a certain set of pixels together from the image. By doing so, we are actually assigning labels to pixels and the pixels with the same label fall under a category where they have some or the other thing common in them.

1.3.4 Classification

Image classification is to identify and portray, as a unique grey level (or colour).Image classification is perhaps the most important part of digital image analysis. We have using a CNN algorithm. Apply convolution filter in first layer. The sensitivity of filter is reduced by smoothing the convolution filter (i.e.) subsampling. The signal transfers from one layer to another layer is controlled by activation layer. Fasten the training period by using Rectified Linear Unit (RELU).

CHAPTER 2

LITERATURE SURVEY

Brain tumor is the cancerous disease where abnormal cells found in the brain. This can be cured if we detect the brain tumor at an early stage. In this proposed system the tumor area is marked and defined what kind of tumor present in the brain tumor MRI image. AlexNet model is used for the classification of different types of tumors as a base model along with Region Proposal Network (RPN) by Faster R-CNN algorithm. Here, the concept of transfer learning is used during training. The proposed system helps to predict the correct type of tumor with better accuracy (R. Ezhilarasi and P. Varalakshmi, 2018).

The brain tumors, are the most common and aggressive disease, leading to a very short life expectancy in their highest grade. Thus, treatment planning is a key stage to improve the quality of life of patients. Generally, various image techniques such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and ultrasound image are used to evaluate the tumor in a brain, lung, liver, breast, prostate...etc. Especially, in this work MRI images are used to diagnose tumor in the brain. However the huge amount of data generated by MRI scan thwarts manual classification of tumor vs. non-tumor in a particular time. But it having some limitation (i.e.) accurate quantitative measurements is provided for limited number of images. Hence trusted and automatic classification scheme are essential to prevent the death rate of human. The automatic brain tumor classification is very challenging task in large spatial and structural variability of surrounding region of brain tumor. In this work, automatic brain tumor detection is proposed by using Convolutional Neural Networks (CNN) classification. The deeper architecture design is performed by using small kernels. The weight of the neuron is given as small. Experimental results show that the CNN archives rate of 97.5% accuracy with low complexity and compared with the all other state of arts methods (Seetha J and Selvakumar Raja S, 2018).

Brain tumor classification is a challenging task in the field of medical image processing. The present study proposes a hybrid method using Neutrosophy and Convolutional Neural Network (NS-CNN). It aims to classify tumor region areas that are segmented from brain images as benign and malignant. In the first stage, MRI images were segmented using the neutrosophic set – expert maximum fuzzy-sure entropy (NS-EMFSE) approach. The features of the segmented brain images in the classification stage were obtained by CNN and classified using SVM and KNN classifiers. Experimental evaluation was carried out based on 5-fold cross-validation on 80 of benign tumors and 80 of malignant tumors. The findings demonstrated that the CNN features displayed a high classification performance with different classifiers. Experimental results indicate that CNN features displayed a better classification performance with SVM as simulation results validated output data with an average success of 95.62% (Fatih Özyurt, Eser Sert, Engin Avci and Esin Dogantekin, 2019).

Gliomas are the most infiltrative and life-threatening brain tumors with exceptionally quick development. Gliomas segmentation using computer-aided diagnosis is a challenging task, due to irregular shape and diffused boundaries of tumor with the surrounding area. Magnetic resonance imaging (MRI) is the most widely used method for imaging structures of interest in human brain. In this study, a deep learning-based method that uses different modalities of MRI is presented for the segmentation of brain tumor. The proposed hybrid convolutional neural network architecture uses patch-based approach and takes both local and contextual information into account, while predicting output label. The proposed network deals with over-fitting problem by utilizing dropout regularizes alongside batch normalization, whereas data imbalance problem is dealt with by using two-phase training procedure. The proposed method contains a preprocessing step, in which images are normalized and bias field corrected, a feed-forward pass through a CNN and a post-processing step, which is used to remove small false positives around the skull portion. The proposed method is validated on BRATS 2013 dataset, where it achieves scores of 0.86, 0.86 and 0.91 in terms of dice score, sensitivity and specificity for whole tumor region, improving results

compared to the state-of-the-art techniques (Sidra Sajid, Saddam Hussain and Amna Sarwar, 2019).

The perilous disease in the worldwide now a days is brain tumor. Tumor affects the brain by damaging healthy tissues or intensifying intra cranial pressure. Hence, rapid growth in tumor cells may lead to death. Therefore, early brain tumor diagnosis is a more momentous task that can save patient from adverse effects. In the proposed work, the Grab cut method is applied for accurate segmentation of actual lesion symptoms while Transfer learning model visual geometry group (VGG-19) is fine-tuned to acquire the features which are then concatenated with hand crafted (shape and texture) features through serial based method. These features are optimized through entropy for accurate and fast classification and fused vector is supplied to classifiers. The presented model is tested on top medical image computing and computer-assisted intervention (MICCAI) challenge databases including multimodal brain tumor segmentation (BRATS) 2015, 2016, and 2017 respectively. The testing results with dice similarity coefficient (DSC) achieve 0.99 on BRATS 2015, 1.00 on BRATS 2015 and 0.99 on BRATS 2017 respectively (Tanzila Saba, Ahmed Sameh, and Mohammed Elaffendi, 2019).

A brain tumor is an abnormal growth of tissue in the brain that results from uncontrolled multiplication of cells. It can disrupt normal brain functioning and must be detected early on using MRI or CT scan images while it is still small, as it may potentially lead to cancer. This research focuses on detecting and localizing the region of the brain where the tumor is present using a proposed methodology that utilizes MRI images of the patient. To achieve this, the first stage involves preprocessing the original image by converting it into grayscale and eliminating any noise that may have been introduced. The image is then subjected to edge detection using algorithms such as Sobel, Prewitt, and Canny, along with image enhancement techniques. Finally, segmentation is applied to clearly display the affected region of the MRI image where the tumor is present (Animesh Hazra, Ankit Dey and Sujit Kumar Gupta, 2017)

The focus of this paper is to examine the different Medical Image Processing techniques that are commonly utilized to identify brain tumors from MRI images. Through thorough research, the paper presents a comprehensive list of techniques that are currently employed in this field, along with a brief description of each method. Out of all the different steps involved in detecting tumors, segmentation is considered to be the most important, as it helps to precisely identify the affected region in the MRI image (Luxit Kapoor and Sanjeev Thakur, 2017)

The objective of this paper is to explore the different entropy functions that can be utilized for tumor segmentation and detection from MRI images. The study investigates the impact of the various definitions of entropy on the threshold values that are obtained. These threshold values are significant as they influence the results of the segmentation process, and are determined based on the particular entropy function that is being utilized (Devendra Somwanshi, Ashutosh Kuma, Pratima Sharma and Deepika Joshi, 2016)

In the field of medical image processing, image segmentation plays a crucial role as it is the initial step in the process. This involves partitioning the image into distinct regions. The processing of MRI scan images is particularly important in the medical field. The focus of this project is to utilize the patient's MRI scan image for detecting brain tumors using a MATLAB-based GUI. Prior to detection, noise is removed through the implementation of removal functions, and basic image processing concepts such as segmentation and morphological operations are applied. The proposed method accurately detects the location of the tumor in the MRI image, and can be applied to different MRI scan images. In this model, the segmentation algorithm is employed to detect the tumor region. Following this, features are extracted using DWT and then reduced in dimensions through PCA. Finally, SVM, CNN, and Lazy IBK classifiers are utilized to accurately detect the type of tumor with a high degree of accuracy (Rabia Ahmad, Asma Khalid, Hammeedur Rahman, 2020)

In the present day, detecting brain tumors at an early stage is crucial since many people die due to a lack of awareness about their condition. Meanwhile, the influence of machine learning is increasing in our society, and artificial intelligence is expected to play an important role in medical diagnosis and support for doctors and surgeons. The primary focus of this paper is to review previous studies that involve the segmentation, detection, and classification of brain tumors. Typically, an algorithm that aims to classify brain tumors on fMRI or MRI scans will preprocess the image by removing noise, segment the image to identify regions that may be brain tumors, and finally classify features such as intensity, shape, and texture of the region. Several machine learning approaches have been made towards the detection of brain tumors. However, this research topic still requires attention as early detection remains crucial for effective treatment, and machine learning has the potential to improve the accuracy of diagnosis and support the work of medical professionals (Tom Philip PRIES, Roshan Jahan, Preetam Suman, 2017)

CHAPTER 3

AIM AND SCOPE

3.1 AIM

- To develop a reliable and accurate model that can predict the presence of brain tumors in patients using medical imaging data, in order to assist medical professionals in making timely and informed decisions for patient treatment and management.
- To incorporate advanced machine learning algorithms and techniques to analyze and interpret medical imaging data, including MRI and CT scans, to accurately detect and classify brain tumors.
- To minimize the risk of misdiagnosis and ensure timely and appropriate treatment for patients.
- The goal of the brain tumor prediction project is to improve patient outcomes, enhance the efficiency of healthcare systems, and contribute to ongoing efforts in the field of medical research and innovation.

3.2 OBJECTIVE

- To develop high-quality software that can accurately identify brain tumors and their causes, providing doctors with a reliable tool to assist in early diagnosis and treatment.
- By using advanced algorithms and techniques to analyze medical imaging data, such as MRI and CT scans, the software will be able to identify tumors quickly and efficiently, saving valuable time for patients.
- Early detection is critical in the successful treatment of brain tumors, and this software will provide doctors with the necessary information to provide appropriate treatment options and timely consultation.

3.3 MOTIVATION

The primary motivation behind the Brain tumor detection project is to identify the presence of a tumor. This would be especially beneficial in cases where it is necessary to determine if the tumor is positive or negative. By analyzing medical images and using computer-based procedures, the system can accurately detect tumor blocks and classify the type of tumor present. The project focuses on utilizing the Convolution Neural Network Algorithm to analyze MRI images from various patients, thereby allowing for efficient and reliable identification of tumors. This system has the potential to be highly effective in aiding medical professionals in making informed decisions for the treatment and management of patients with brain tumors.

3.4 EXISTING METHOD

Detecting brain tumors automatically in MR images involves extracting features and classifying them using machine learning algorithms. The feature set created using the above-mentioned method is then applied to Multi-Layer Perceptron (MLP) and Naive Bayes classifiers. This approach requires only a small amount of training data to estimate the parameters necessary for classification. Primary brain tumors occur when healthy cells in the brain develop mutations in their DNA, causing them to grow and divide at a faster rate than normal cells. These mutations also enable them to survive when healthy cells would die, resulting in the formation of an abnormal mass of cells known as a tumor. Early detection and prediction of brain tumors can aid in the timely diagnosis of benign tumors and potentially save a person's life. The automated detection of brain tumors using MR images and machine learning algorithms has the potential to improve the accuracy and speed of diagnosis, enabling doctors to take the necessary steps to provide patients with the best possible care.

3.4.1 Disadvantages

- The first disadvantage is that the Naive Bayes classifier makes a very strong assumption about the shape of your data distribution.
- Another problem happens due to data scarcity.

- A third problem arises for continuous features. It is common to use a binning procedure to make them discrete, but if you are not careful you can throw away a lot of information.
- The existing system used different algorithms to predict the disease, but accuracy is low compared to our model.
- Complexity is high.
- Training and Testing the model is used the same algorithm, but we provide different methods.

3.5 PROPOSED METHOD

Our proposed system for brain tumor detection employs the use of Dense Layer in Convolutional Neural Network (CNN) Algorithm, which is a key component in the field of Deep Learning. The CNN is trained on a dataset using the Dense Layer, which allows each layer to receive additional inputs from all preceding layers and transmit its own feature-maps to all subsequent layers. The Dense Layer is designed to incorporate features of all complexity levels, thereby producing more precise and accurate decision boundaries. This approach results in a more efficient and effective detection system, allowing for the rapid and accurate identification of brain tumors. The use of Dense Layer in the CNN Algorithm represents a significant advancement in the field of brain tumor detection, offering an innovative and reliable solution for medical professionals.

3.5.1 Advantages

- Easy detection of the Brain Tumor with the concluded technique.
- Save patient's time.
- The best accuracy Model helps in better treatment as early.
- Detection of the best Model will quick the treatment which is life-saving

CHAPTER 4

MATERIALS AND METHODS

4.1 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make computers "smart". As machines become increasingly capable, mental facilities once thought to require intelligence are removed from the definition. AI is an area of computer sciences that emphasizes the creation of intelligent machines that work and reacts like humans. Some of the activities computers with artificial intelligence are designed for include: Face recognition, Learning, Planning, Decision making etc., Artificial intelligence is the use of computer science programming to imitate human thought and action by analysing data and surroundings, solving or anticipating problems and learning or self-teaching to adapt to a variety of tasks.

4.2 MACHINE LEARNING

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more. Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as: "Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed".

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps

to build a better model which predicts the output more accurately. Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:

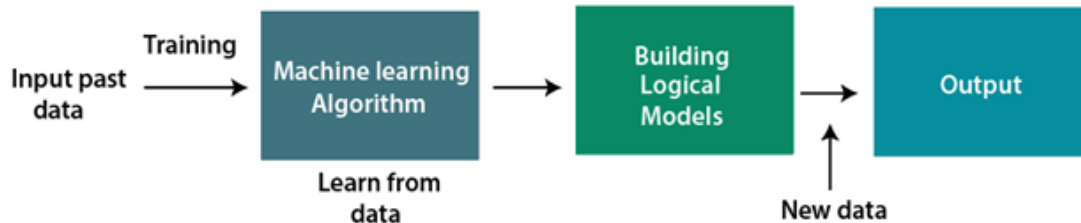


Fig 4.1 Working of machine learning algorithm

4.2.1 Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with a huge amount of data.

4.2.2 Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing

are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering. Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of algorithms:

- Clustering
- Association

4.3 DEEP LEARNING

Deep learning is a branch of machine learning which is completely based on Artificial Neural Networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture. In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousands of

their neighbors. The question here is how we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for-output value and in between, there may be lots of neurons interconnected in the hidden layer.

4.3.1 Architecture of deep learning:

1. *Deep Neural Network*: It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.
2. *Deep Belief Network (DBN)*: It is a class of Deep Neural Network. It is multi-layer belief networks.

Steps for performing DBN:

- a. Learn a layer of features from visible units using Contrastive Divergence algorithm.
 - b. Treat activations of previously trained features as visible units and then learn features of features.
 - c. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.
3. *Recurrent (perform same task for every element of a sequence) Neural Network*: Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

4.4 CONVOLUTIONAL NEURAL NETWORK (CNN):

When it comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used in various classification task like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural

Network. In this blog, we are going to build basic building block for CNN. Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network there are three types of layers:

1. *Input Layers*: It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels in case of an image).
2. *Hidden Layer*: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layers can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.
3. *Output Layer*: The output from the hidden layer is then fed into a logistic function like sigmoid or SoftMax which converts the output of each class into probability score of each class.

The data is then fed into the model and output from each layer is obtained this step is called feed forward, we then calculate the error using an error function, some common error functions are cross entropy, square loss error etc. After that, we back propagate into the model by calculating the derivatives. This step is called Back propagation which basically is used to minimize the loss. Here's the basic python code for a neural network with random inputs and two hidden layers. Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image) and height (as image generally have red, green, and blue channels).

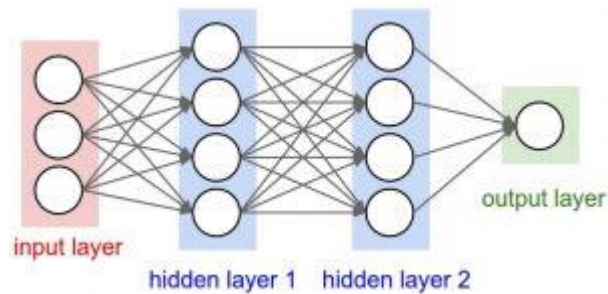


Fig 4.2 Layers in CNN

Now let's talk about a bit of mathematics which is involved in the whole convolution process.

- Convolution layers consist of a set of learnable filters (patch in the above image). Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimension $34 \times 34 \times 3$. Possible size of filters can be $a \times a \times 3$, where 'a' can be 3, 5, 7, etc. but small as compared to image dimension.
- During forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have value 2 or 3 or even 4 for high dimensional images) and compute the dot product between the weights of filters and patch from input volume.
- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

4.4.1 Types of layers:

Let's take an example by running a convnets on of image of dimension $32 \times 32 \times 3$.

1. *Input Layer:* This layer holds the raw input of image with width 32, height 32 and depth 3.
2. *Convolution Layer:* This layer computes the output volume by computing dot product between all filters and image patch. Suppose we use total 12 filters for this layer we'll get output volume of dimension $32 \times 32 \times 12$.

3. *Activation Function Layer:* This layer will apply element wise activation function to the output of convolution layer. Some common activation functions are RELU: $\max(0, x)$, Sigmoid: $1/(1+e^{-x})$, Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimensions $32 \times 32 \times 12$.
4. *Pool Layer:* This layer is periodically inserted in the convnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents from overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2×2 filters and stride 2, the resultant volume will be of dimension $16 \times 16 \times 12$.
5. *Fully-Connected Layer:* This layer is regular neural network layer which takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

4.5 Dense Layer:

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output. Dense layers add an interesting non-linearity property; thus, they can model any mathematical function. However, they are still limited in the sense that for the same input vector we get always the same output vector. They can't detect repetition in time, or produce different answers on the same input. Each neuron in a layer receives an input from all the neurons present in the previous layer—thus, they're densely connected. In other words, the dense layer is a fully connected layer, meaning all the neurons in a layer are connected to those in the next layer. Original implementation was using the Tanh function for the activation, it is now more frequent to use the ReLU, and it is leading to faster training and lower probability of vanishing gradient. There are two convolutional layers based on 3×3 filters with average pooling.

4.6 SYSTEM REQUIREMENTS

Table 4.6.1 system Requirements

Operating System	Windows 7 or later
Simulation Tool	Anaconda (Jupyter notebook)
Documentation	MS – Office

4.7 REQUIREMENT ANALYSIS

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system's purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

4.7.1 PYTHON

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language. For example, `x=10`. Here, `x` can be anything such as String, int, etc. Python is an interpreted, object-oriented programming language similar to PERL that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users. There are many features in Python, some of which are discussed below

- Easy to code
- Free and Open Source
- Object-Oriented Language

- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted Language

4.7.2 ANACONDA

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail. In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2, 0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the users own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for

Windows 32/64 bit, Linux 64 bit and Mac OS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

4.7.3 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux. The following applications are available by default in Navigator:

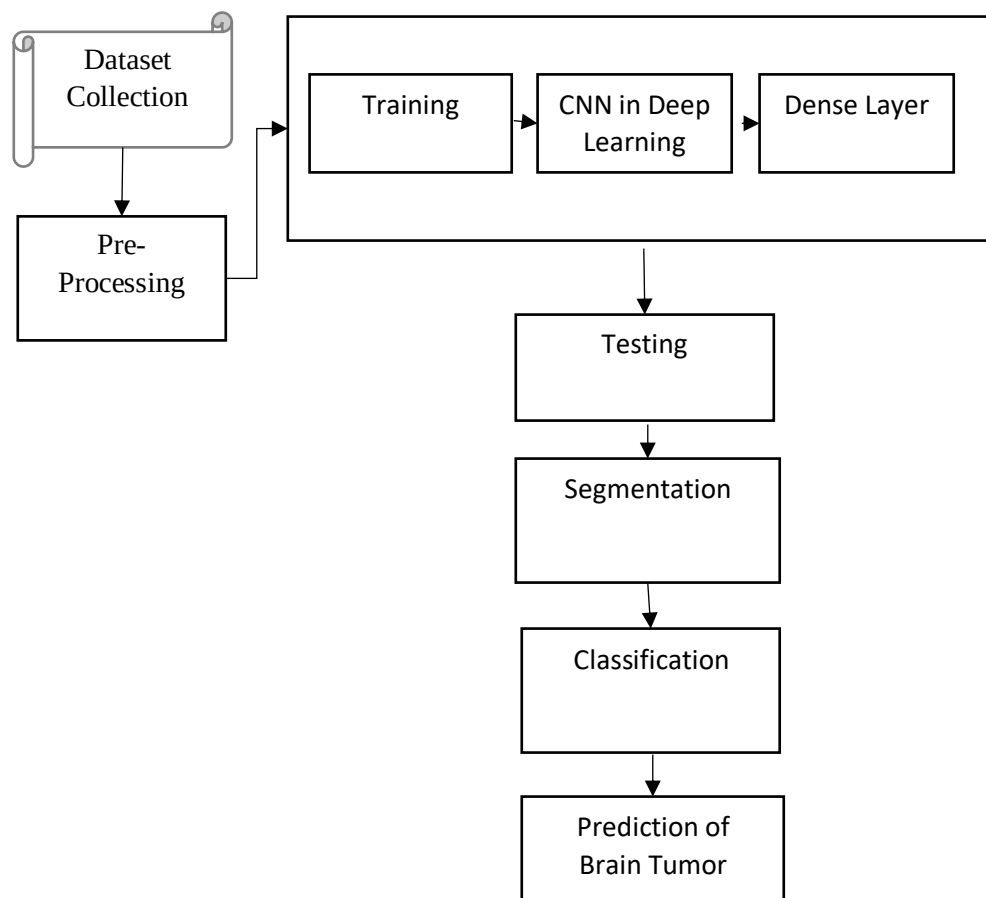
- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

4.7.4 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the

Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension. Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell. The Notebook interface was added to IPython in the 0.12 release (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. According to *The Atlantic*, Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

4.8 SYSTEM ARCHITECTURE



4.9 SYSTEM MODULES:

- Module 1: Image Processing
- Module 2: Pre-Processing
- Module 3: Segmentation
- Module 4: Classification

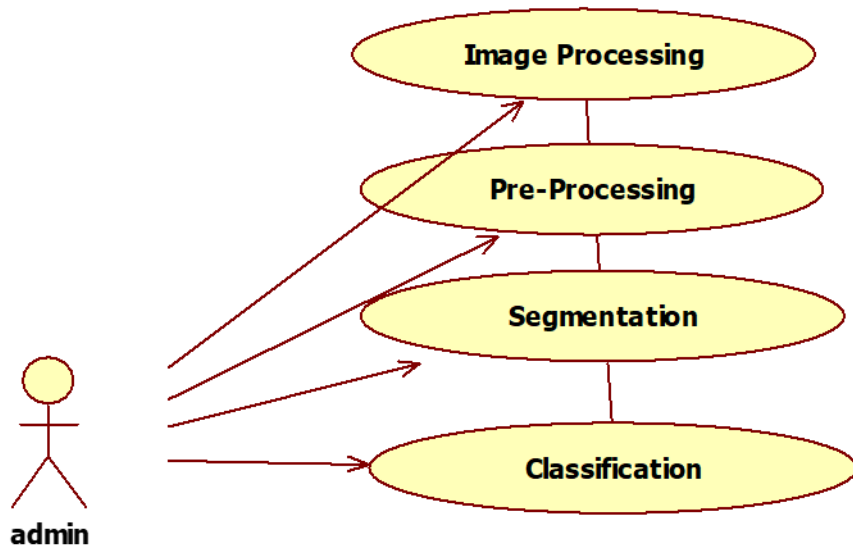


Fig 4.3 System Modules

4.9.1 Module 1: Dataset Collection

A dataset (or data set) is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object. Each value is known as a datum.

We have chosen to use a publicly-available Healthcare dataset which contains a relatively small number of inputs and cases. The data is arranged in such a way that will allow those trained in medical disciplines to easily draw parallels between familiar statistical and novel ML techniques. Additionally, the compact dataset enables short computational times on almost all modern computers.

4.9.2 Module 2: preprocessing

The sklearn. Preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers,

transformers, and normalizers on a dataset containing marginal outliers is highlighted in Compare the effect of different scalers on data with outliers. Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance. In practice we often ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation. For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered on zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected. An alternative standardization is scaling features to lie between a given minimum and maximum value, often between zeros and one, or so that the maximum absolute value of each feature is scaled to unit size. This can be achieved using MinMaxScaler or MaxAbsScaler, respectively. The motivation to use this scaling include robustness to very small standard deviations of features and preserving zero entries in sparse data. MaxAbsScaler works in a very similar fashion, but scales in a way that the training data lies within the range $[-1, 1]$ by dividing through the largest maximum value in each feature. It is meant for data that is already centered at zero or sparse data. Normalization is the process of scaling individual samples to have unit norm. This process can be useful if you plan to use a quadratic form such as the dot-product or any other kernel to quantify the similarity of any pair of samples. This assumption is the base of the Vector Space Model often used in text classification and clustering contexts.

4.9.3 Module 3: Segmentation

Image segmentation is the process of dividing the image into non- overlapping meaningful regions. The main objective if an image segmentation is to divide an image into many sections for the further analysis, so we can get the only necessary

or a segment of information. We use various image segmentation algorithms to split and group a certain set of pixels together from the image. By doing so, we are actually assigning labels to pixels and the pixels with the same label fall under a category where they have some or the other thing common in them.

Using these labels, we can specify boundaries, draw lines, and separate the most required objects in an image from the rest of the not-so-important ones. In the below example, from a main image on the left, we try to get the major components, e.g. chair, table etc. and hence all the chairs are colored uniformly. In the next tab, we have detected instances, which talk about individual objects, and hence the all the chairs have different colors. This is how different methods of segmentation of images work in varying degrees of complexity and yield different levels of outputs.

4.9.4 Module 4: Classification

Image classification is to identify and portray, as a unique gray level (or color), the features occurring in an image in terms of the object or type of land cover these features actually represent on the ground. Image classification is perhaps the most important part of digital image analysis.

- *K-Nearest Neighbours*: Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.
- *Support Vector Machine*: Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

4.10 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to

ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- i. Economic Feasibility
- ii. Technical Feasibility
- iii. Social Feasibility

4.10.1 Types of Feasibility Study:

- *Economic Feasibility:* This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.
- *Technical Feasibility:* This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.
- *Social Feasibility:* The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.11 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub – assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

4.11.1 TYPES OF TESTS

- *UNIT TESTING*: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.
- *INTEGRATION TESTING*: Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- *FUNCTIONAL TEST*: Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

- *SYSTEM TEST*: System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.
- *WHITE BOX TESTING*: White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.
- *BLACK BOX TESTING*: Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test

provides inputs and responds to outputs without considering how the software works.

- *UNIT TESTING*: Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

4.12 IMPLEMENTATION

4.12.1 Python

Below is a step by step process on how to download and install Python on Windows:

Step 1: Go to <https://www.python.org/downloads>. The most recent version of Python will always appear on the "Download" button near the top of the page.

Step 2: Click Download Python. If this does not start the download immediately, click Save on the pop-up window to start it.

Step 3: Run the installer. You can do so by double-clicking python-<version>.exe in your Downloads folder.

Step 4: Click Install button. It's at the bottom of the window.

Step 5: Click Yes to confirm. This installs Python on your PC. Once the installation is complete, you'll see a "Setup was successful" window.

Step 6: Click Close to exit the installer.

4.12.2 Jupyter Notebook

Below is a step by step process on how to download and install Jupyter Notebook on windows:

Step 1: Please go to the [Anaconda.com/downloads](https://anaconda.com/downloads) site to install Anaconda

Step 2: Select the respective platform: Windows/Mac/Linux

Step 3: Download the .exe installer

Step 4: Open and execute the .exe installer

Step 5: Launch Anaconda Navigator

Step 6: Click on the Install Jupyter Notebook Button

Step 7: Beginning the Installation

Step 8: Loading Packages

Step 9: Finish Installation

Installing Jupyter Notebook using pip command: PIP stands for the package management system which is used to install and manage software packages/libraries. These libraries and the packages are written in Python. These files are stored in a large “on-line repository” termed as Python Package Index (PyPI). pip uses PyPI as the default source for packages and their dependencies. Before we start installing pip, we have to check the version of the pip command. If the version of the pip command is not updated then we need to update the pip in our system.

Update PIP command:

```
python3 -m pip install --upgrade pip
```

Then after updating the pip version we need to follow the upcoming process to install Jupyter.

- Command to install Jupyter: pip3 install Jupyter
- Begin Installation
- Collect Files and Data
- Download Packages
- Run Installation
- Finish Installation

Now Launch the Jupyter:

Use the command to launch Jupyter using command-line:

```
jupyter notebook
```

4.12.3 Anaconda

Below is a step by step process on how to download and install Python on Windows:

Step 1: Go to the Anaconda Website and choose a Python 3.x graphical installer (A) or a Python 2.x graphical installer (B). If you aren't sure which Python version you want to install, choose Python 3. Do not choose both.

Step 2: Locate your download and double click it.

Step 3: Read the license agreement and click on I Agree.

Step 4: Click on Next.

Step 5: Note your installation location and then click Next.

Step 6: This is an important part of the installation process. The recommended approach is to not check the box to add Anaconda to your path. This means you will have to use Anaconda Navigator or the Anaconda Command Prompt (located in the Start Menu under "Anaconda") when you wish to use Anaconda (you can always add Anaconda to your PATH later if you don't check the box). If you want to be able to use Anaconda in your command prompt (or git bash, cmd, PowerShell etc.), please use the alternative approach and check the box

Step 7: Click on Next.

Step 8: You can install Microsoft VSCode if you wish, but it is optional.

Step 9: Click on Finish.

CHAPTER 5

RESULTS

5.1 COLLECTED DATASET

We have chosen to use a publicly-available Healthcare dataset which contains a relatively small number of inputs and cases. The data is arranged in such a way that will allow those trained in medical disciplines to easily draw parallels between familiar statistical and novel ML techniques. Additionally, the compact dataset enables short computational times on almost all modern computers.

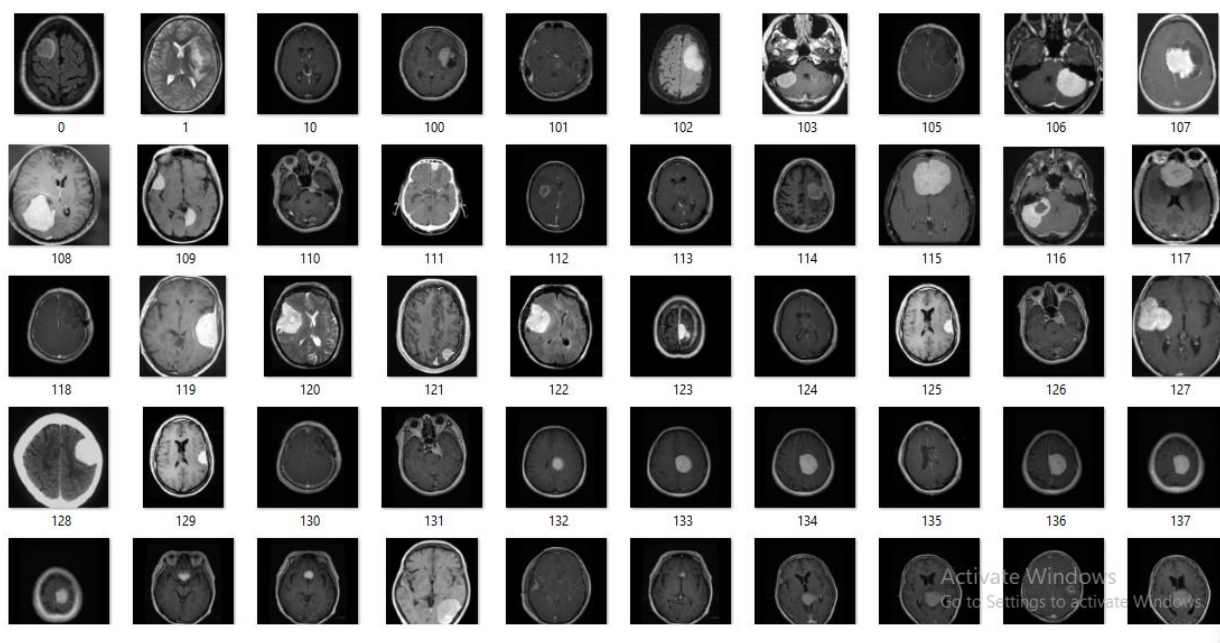


Fig 5.1 Training dataset

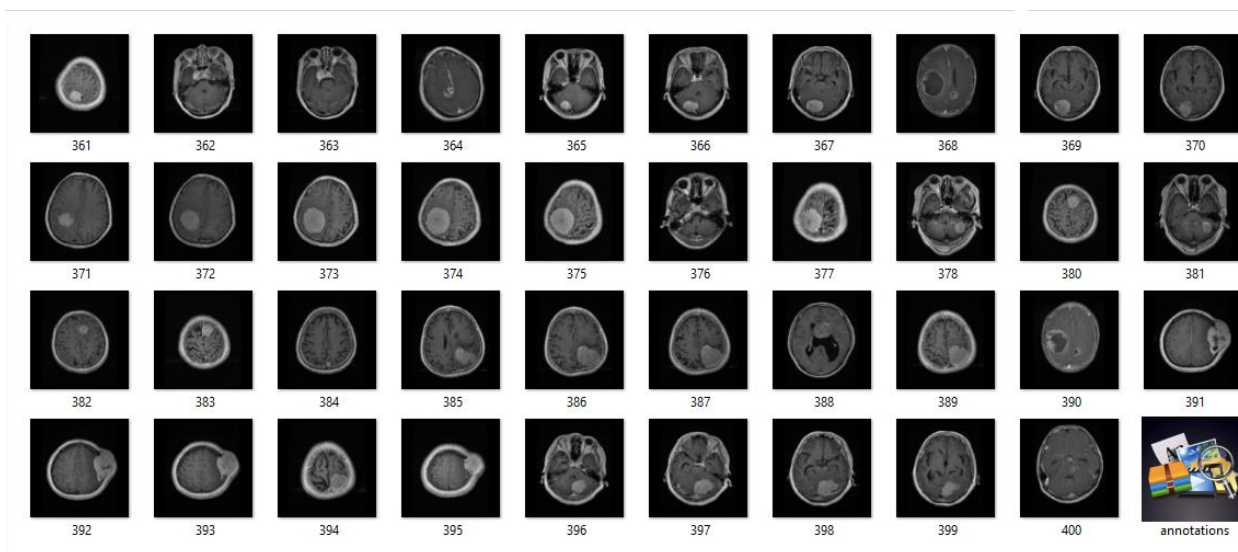


Fig 5.2 Testing dataset

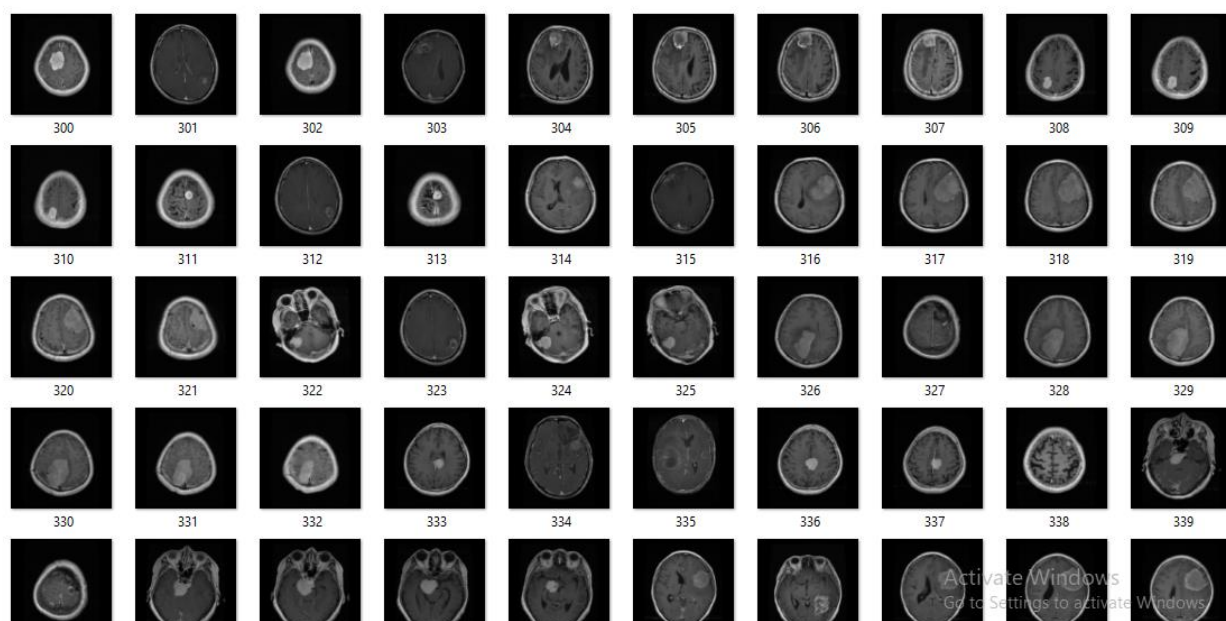


Fig 5.3 Validation dataset

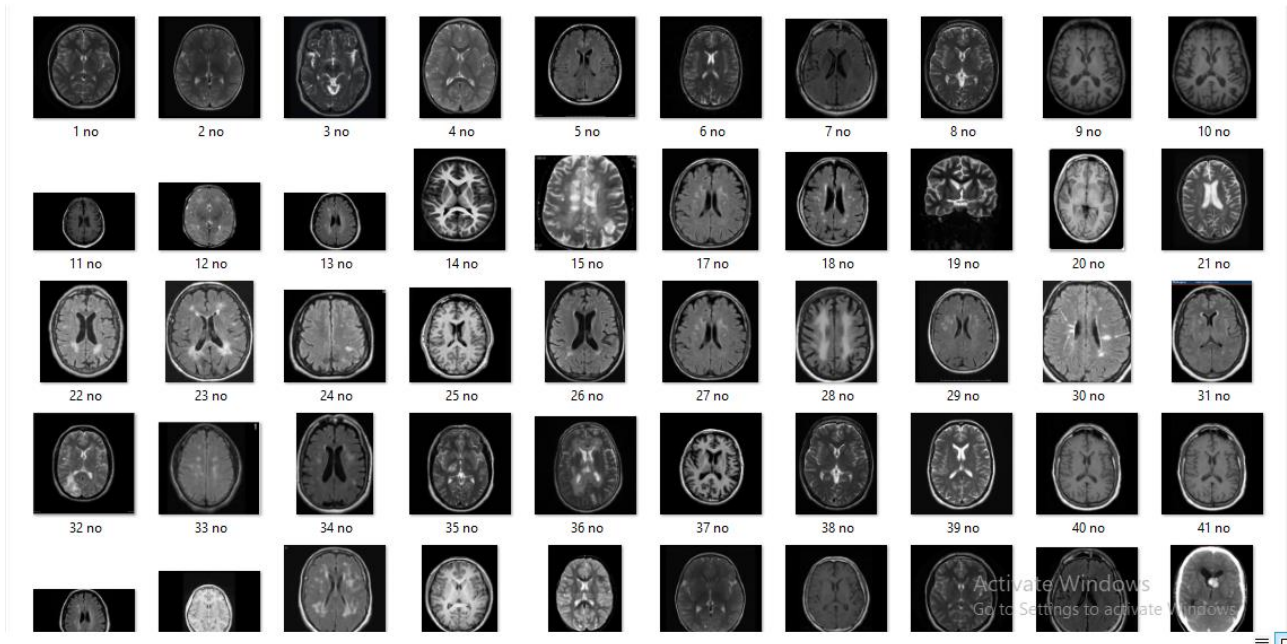


Fig 5.4 Sample Negative Brain Tumor MRI Scans

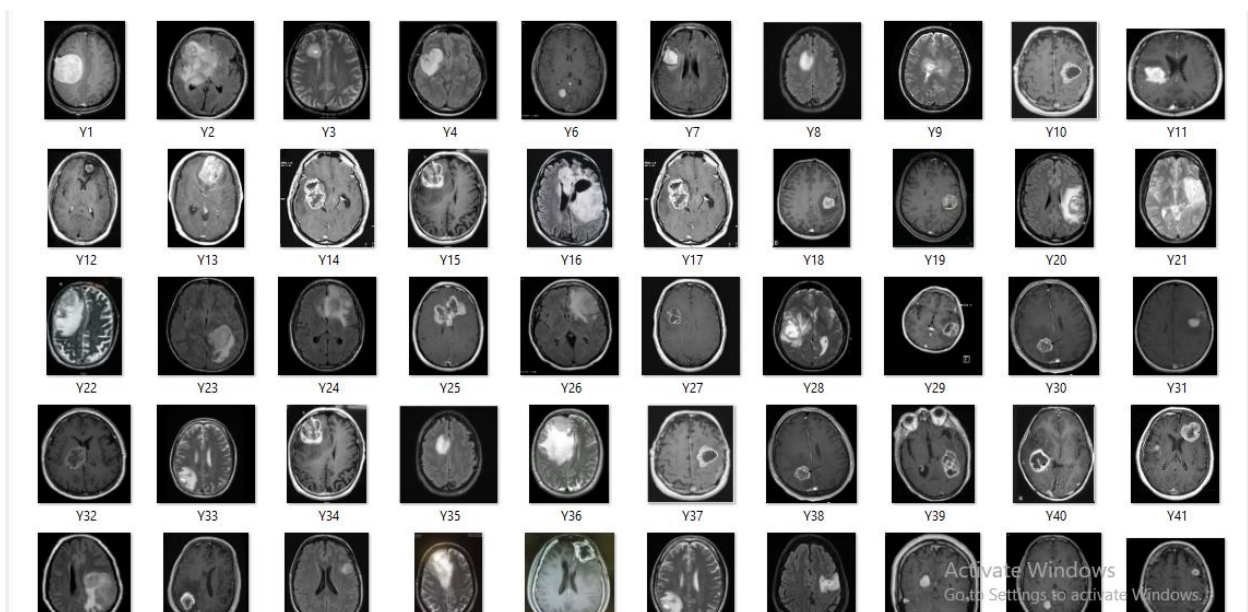


Fig 5.5 Sample Positive Brain Tumor MRI Scans

5.2 SEGMENTATION

```
# Segmentation

In [14]: import json
import math
import random
import imutils
import itertools
import matplotlib
import numpy as np
import skimage.draw
import pandas as pd
from PIL import Image
from tqdm import tqdm
import matplotlib.cm as cm
from skimage.io import imread
import matplotlib.pyplot as plt
import PIL.ImageDraw as ImageDraw
from imgaug import augmenters as iaa

from IPython.display import clear_output

In [13]: !pip install imutils
!pip install imgaug

Collecting imgaug
  Downloading imgaug-0.4.0-py2.py3-none-any.whl (948 kB)
-----
Requirement already satisfied: Pillow in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (9.2.0)
Requirement already satisfied: imageio in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (2.19.3)
Collecting Shapely
  Downloading Shapely-1.8.5.post1-cp39-cp39-win_amd64.whl (1.3 MB)
-----
Requirement already satisfied: six in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (1.16.0)
Requirement already satisfied: opencv-python in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (4.6.0.66)
Requirement already satisfied: numpy>=1.15 in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (1.21.5)
Requirement already satisfied: matplotlib in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (3.5.2)
Requirement already satisfied: scikit-image>=0.14.2 in c:\users\lenovo\anaconda3\lib\site-packages (from imgaug) (0.19.2)
Requirement already satisfied: networkx>=2.2 in c:\users\lenovo\anaconda3\lib\site-packages (from scikit-image>=0.14.2->imgaug) (2.8.4)
```

Fig 5.6 Segmentation Process

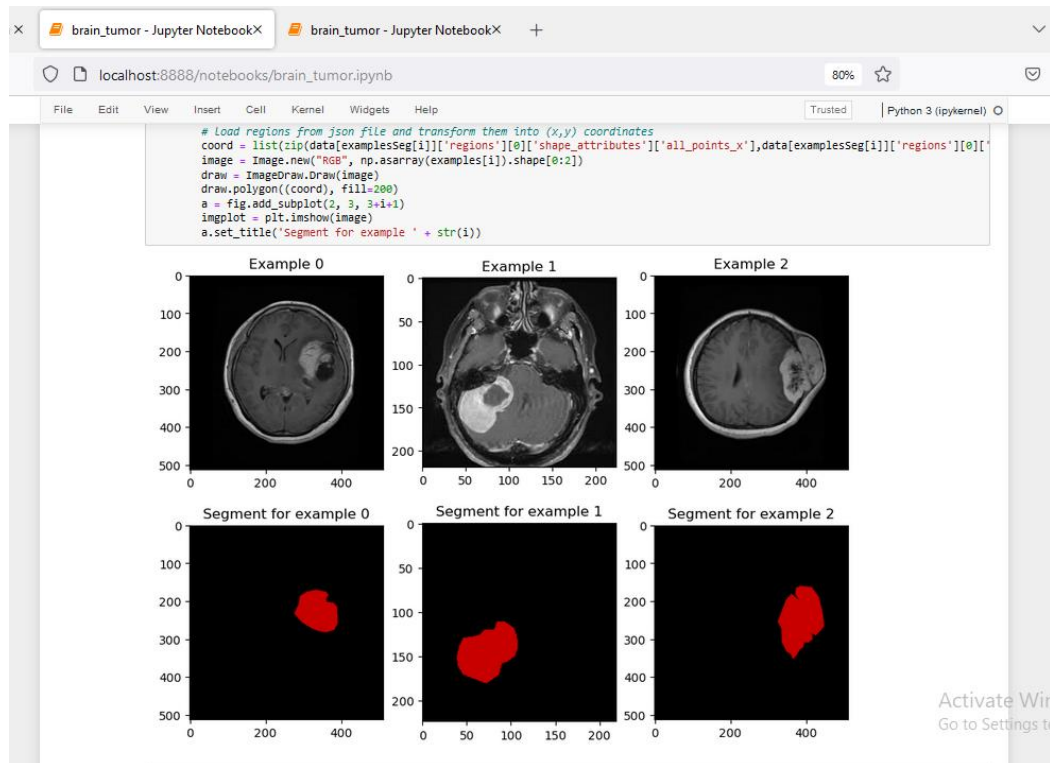


Fig 5.7 Segmentation Output

		T1	T1C	T2	FLAIR	Result from [8]	Result from [6]	Result by JLF	Ground Truth
Patient1	Image								
	Label map by using label fusion								
Patient2	Image								
	Label map by using label fusion								

Fig 5.8 Example Longitudinal Tumor Segmentation Results

5.3 TUMOR PREDICTION OUTPUT

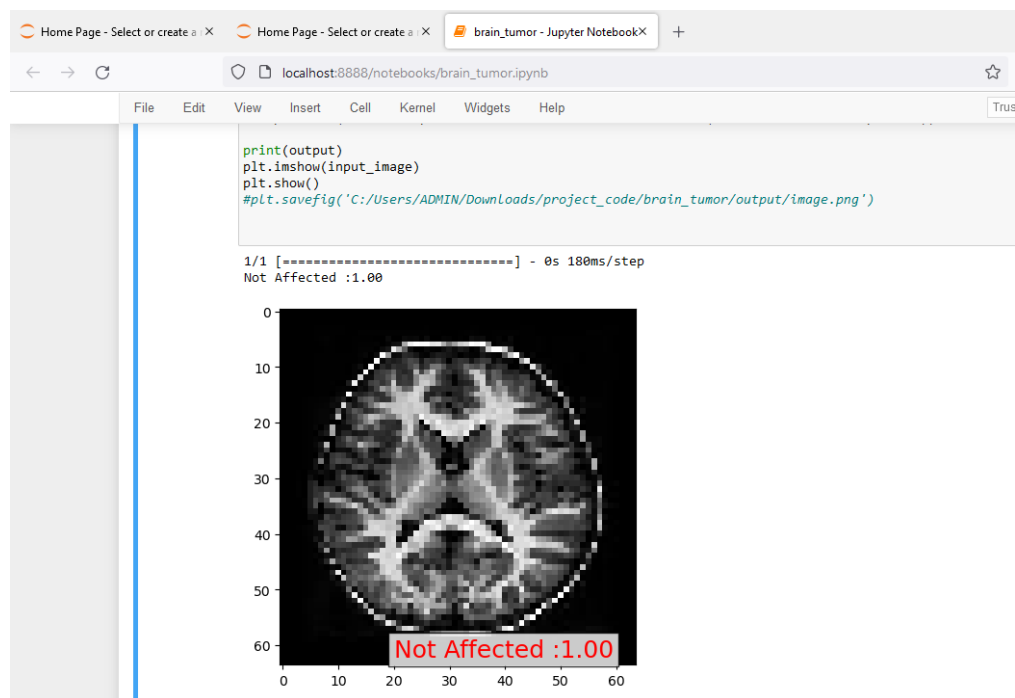


Fig 5.9 Output for Normal Brain MRI (Not Affected)

When we give a normal brain MRI scan as the input, the output shows “Not Affected”

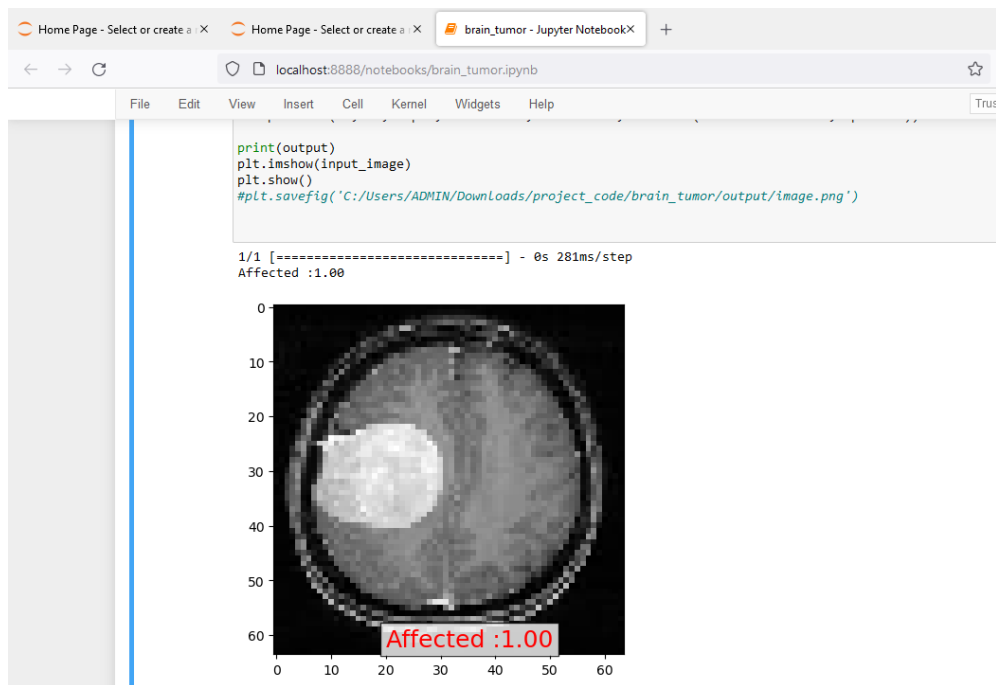


Fig 5.10 Output for Abnormal Brain MRI (Affected)

When we give an abnormal brain MRI scan as the input, the output shows “Affected”

CHAPTER 6

CONCLUSION

The main goal of this project is to design efficient automatic brain tumor classification with high accuracy, performance and low complexity. In the conventional brain tumor classification is performed by using Fuzzy C Means (FCM) based segmentation, texture and shape feature extraction and SVM and DNN based classification are carried out. The complexity is low. But the computation time is high meanwhile accuracy is low. Further to improve the accuracy and to reduce the computation time, a convolution neural network based classification is introduced in the proposed scheme. Also the classification results are given as tumor or normal brain images. CNN is one of the deep learning methods, which contains sequence of feed forward layers. Also python language is used for implementation. Image net database is used for classification. It is one of the pre-trained models. So the training is performed for only final layer. Also raw pixel value with depth, width and height feature value are extracted from CNN. Finally, the Gradient decent based loss function is applied to achieve high accuracy. The training accuracy, validation accuracy and validation loss are calculated. The training accuracy is 97.5%. Similarly, the validation accuracy is high and validation loss is very low.

REFERENCE

- [1] Angulakshmi M, Lakshmi Priya G, Automated brain tumour segmentation techniques – a review, *Int. J. Imaging Syst. Technol* 27 (2017) Pg.no 66–77.
- [2] Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, Freymann JB, Farahani K, Davatzikos C, Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features, *Nat. Sci. Data* 4 (2017).
- [3] Bakas S, Reyes M, Menze B, Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge (2018).
- [4] Connor Shorten and Taghi M Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 60, (2019).
- [5] Chen L., Bentley P., Mori K., Misawa K., Fujiwara M., Rueckert D, Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis* 58, 101539 (2019).
- [6] Dai L., Li T., Shu H., Zhong L., Shen H., Zhu H, Automatic brain tumor segmentation with domain adaptation. In: *International MICCAI Brainlesion Work-shop*. pp. 380–392. Springer (2018).
- [7] Deepak S., and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning." *Computers in biology and medicine* 111 (2019): 103345.
- [8] Ezhilarasi R., and P. Varalakshmi, "Tumor detection in the brain using faster R-CNN." In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2018 2nd International Conference on, pp. 388-392. IEEE, 2018.

- [9] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019).
- [10] Hossain, Tonmoy, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, and Faisal Muhammad Shah. "Brain Tumor Detection Using Convolutional Neural Network." In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-6. IEEE, 2019.
- [11] Huang L., Ruan, S., Denoeux, T, Belief function-based semi-supervised learning for brain tumor segmentation. In: International Symposium on Biomedical Imaging. pp. 160–164. IEEE (2021).
- [12] Huo Y., Xu, Z., Xiong, Y., Aboud, K., Parvathaneni, P., Bao, S., Bermudez, C., Resnick, S.M., Cutting, L.E., Landman, B.A.: 3d whole brain segmentation using spatially localized atlas network tiles. *NeuroImage* 194, 105–119 (2019).
- [13] Isensee F., Jäger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H, nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. pp. 1–9. *Nature Methods* (2020).
- [14] Isensee F., Jäger, P.F., Full, P.M., Vollmuth, P., Maier-Hein, K.H, nnU-Net for Brain Tumor Segmentation. In: Crimi, A., Bakas, S. (eds.) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. pp. 118–132. Springer International Publishing, Cham (2021).
- [15] Jakub Nalepa, Michal Marcinkiewicz, and Michal Kawulok, "Data augmentation for brain-tumor segmentation: A review," *Frontiers in Computational Neuroscience*, vol. 13, (2019).

- [16] Jiachi Zhang, Xiaolei Shen, Tianqi Zhuo, and Hong Zhou, Brain tumor segmentation based on refined fully convolutional neural networks with a hierarchical dice loss (2017).
- [17] Jiang Z., Ding, C., Liu, M., Tao, D, Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In: International MICCAI Brainlesion Workshop. pp. 231–241. Springer (2019).
- [18] Sajid, Sidra, Saddam Hussain, and Amna Sarwar, "Brain tumor detection and segmentation in MR images using deep learning." Arabian Journal for Science and Engineering 44, no. 11 (2019): 9249-9261.
- [19] Seetha, J., and S. Selvakumar Raja, "Brain tumor classification using convolutional neural networks." Biomedical & Pharmacology Journal 11, no. 3 (2018): 1457.
- [20] Siar, Masoumeh, and Mohammad Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm." In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 363-368. IEEE, 2019.
- [21] Toğaçar, Mesut, Burhan Ergen, and Zafer Cömert, "BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model." Medical hypotheses 134 (2020): 109531.
- [22] Wang H and P. A. Yushkevich, "Multi-Atlas Segmentation without Registration: A Supervoxel-based Approach, " in Med Image Comput Comput Assist Interv., PMC, (2014).
- [23] Wang W., Chen, C., Ding, M., Yu, H., Zha, S., Li, J, Transbts: Multimodal brain tumor segmentation using transformer. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. pp. 109–119 (2021).

- [24] Zhang J., Jiang, Z., Dong, J., Hou, Y., Liu, B.: Attention gate resu-net for automatic mri brain tumor segmentation. *IEEE Access* 8, 58533–58545 (2020).
- [25] Zhao Y.X., Zhang, Y.M., Liu, C.L.: Bag of tricks for 3d mri brain tumor segmentation. In: *International MICCAI Brainlesion Workshop*. pp. 210–220. Springer (2019).
- [26] Zhong L., Li, T., Shu, H., Huang, C., Johnson, J.M., Schomer, D.F., Liu, H.L.,Feng, Q., Yang, W., Zhu, H, (ts)2wm: Tumor segmentation and tract statistics for assessing white matter integrity with applications to glioblastoma patients.*NeuroImage* 223, 117368 (2020).

APPENDIXES

```
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.callbacks import TensorBoard

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense

import tensorflow
from warnings import filterwarnings

classifier = Sequential()
classifier.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation = 'relu'))
classifier.add(MaxPooling2D(pool_size=(2,2),strides=2))
classifier.add(Conv2D(32,(3,3),activation = 'relu'))
classifier.add(MaxPooling2D(pool_size=(2,2),strides=2))
classifier.add(Flatten())

classifier.add(Dense(units=128,activation='relu'))
classifier.add(Dense(units=1,activation='sigmoid'))
adam = tensorflow.keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999,
epsilon=None, decay=0.0, amsgrad=False)
classifier.compile(optimizer=adam,loss='binary_crossentropy',metrics=['accuracy'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator
datagenerate_train = ImageDataGenerator(rescale=1./255,
                                         shear_range=0.1,
                                         zoom_range=0.1,
                                         horizontal_flip=True)
```

```
datagenerate_test = ImageDataGenerator(rescale=1./255)
```

```
print(datagenerate_train)
```

```
print(datagenerate_test)
```

```
<tensorflow.python.keras.preprocessing.image.ImageDataGenerator object at  
0x000001EABFB86700>
```

```
<tensorflow.python.keras.preprocessing.image.ImageDataGenerator object at  
0x000001EABFB86A90>
```

```
#Training Set
```

```
training_set =
```

```
datagenerate_train.flow_from_directory('C:/Users/ADMIN/Downloads/project_code/br  
ain_tumor/dataset',
```

```
target_size=(64,64),
```

```
batch_size=32,
```

```
class_mode='binary')
```

```
#Validation Set
```

```
Validation_set =
```

```
datagenerate_test.flow_from_directory('C:/Users/ADMIN/Downloads/project_code/bra  
in_tumor/dataset',
```

```
target_size=(64,64),
```

```
batch_size = 32,
```

```
class_mode='binary',
```

```
shuffle=False)
```

```
model_hist=classifier.fit_generator(training_set,
```

```
steps_per_epoch=5,
```

```
epochs = 25,
```

```
validation_data = Validation_set,
```

```

        validation_steps = 20,
    );

#Save the weight file
classifier.save('C:/Users/ADMIN/Downloads/project_code/brain_tumor/brain_tumer.h5')

#Classification using CNN
#Load the model
from tensorflow.keras.models import load_model
classifier_model=
load_model('C:/Users/ADMIN/Downloads/project_code/brain_tumor/brain_tumer.h5')

#Image Prediction
import tensorflow
import cv2
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np

input_image=
image.load_img('C:/Users/ADMIN/Downloads/project_code/brain_tumor/predict/2_no.
jpeg', target_size=(64, 64))
img_1= image.img_to_array(input_image)
img_1 = img_1/255

img_1 = np.expand_dims(img_1, axis=0)
prediction = classifier_model.predict(img_1, batch_size=None, steps=1)
if(prediction[:,>0.5):

```

```

        output = 'Affetecd :%1.2f'%(prediction[0,0])
        plt.text(20,
62,output,color='red',fontsize=18,bbox=dict(facecolor='white',alpha=0.8))
    else:
        output = 'Not Affected :%1.2f'%(1.0-prediction[0,0])
        plt.text(20,
62,output,color='red',fontsize=18,bbox=dict(facecolor='white',alpha=0.8))

print(output)
plt.imshow(input_image)
plt.show()
#plt.savefig('C:/Users/ADMIN/Downloads/project_code/brain_tumor/output/image.png
')
```

```

#segmentation
import json
import math
import random
import imutils
import itertools
import matplotlib
import numpy as np
import skimage.draw
import pandas as pd
from PIL import Image
from tqdm import tqdm
import matplotlib.cm as cm
from skimage.io import imread
import matplotlib.pyplot as plt
import PIL.ImageDraw as ImageDraw
```

```

from imgaug import augmenters as iaa

from IPython.display import clear_output

DATASET_DIR      ='C:/Users/ADMIN/Downloads/project_code/brain_tumor/New
folder/brain-tumor-segmentation-main/brain_tumor_data/'

examples          =          [Image.open(DATASET_DIR          +
'train/100.jpg'),Image.open(DATASET_DIR          +
'train/116.jpg'),Image.open(DATASET_DIR+'train/221.jpg')]
examplesSeg = ['100.jpg20477','116.jpg10596','221.jpg19584']

## print some example images from train directory
fig = plt.figure(figsize=(10,14))

for i in range(0, len(examples)):
    a = fig.add_subplot(1, 3, i+1)
    imgplot = plt.imshow(examples[i])
    a.set_title('Example '+str(i))

# print segment examples for images from json annotations file
with open(DATASET_DIR+'train/annotations.json') as json_file:
    data = json.load(json_file)
    for i in range(0,len(examplesSeg)):
        # load regions from json file and transform them into (x,y) coordinates
        coord
        =
list(zip(data[examplesSeg[i]]['regions'][0]['shape_attributes']['all_points_x'],data[examplesSeg[i]]['regions'][0]['shape_attributes']['all_points_y']))
        image = Image.new("RGB", np.asarray(examples[i]).shape[0:2])
        draw = ImageDraw.Draw(image)
        draw.polygon((coord), fill=200)

```

```
a = fig.add_subplot(2, 3, 3+i+1)
imgplot = plt.imshow(image)
a.set_title('Segment for example ' + str(i))
```