BRAIN TUMOR IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

A Project Report

Submitted in the partial fulfilment of the requirements for

the award of the degree of

Bachelor of Technology in

Department of Computer Science and Engineering

By BATCH 333

Vatrapu Sri Lekha Reddy	**(180030193)**
Sarigala Shainy	**(180030200)**
Bolla Lahya	**(180031128)**
Uppala Bhuvana Priya	**(180031177)**

Under the supervision of

Dr. Chayan Paul (Asst. Professor)

K L E F, Green Fields,

Vaddeswaram-522502, Guntur (Dist.), Andhra Pradesh, India.

NOV 2021

DECLARATION

The Project Report entitled **"BRAIN TUMOR IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK"** is a record of bonafide work of **180030193, 180030200, 180031128, 180031177,** submitted in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to the **K L University**. The results embodied in this report have not been copied from any other Department/University/Institute.

Vatrapu Sri Lekha Reddy - 180030193

Sarigala Shainy - 180030200

Bolla Lahya - 180031128

Uppala Bhuvana Priya - 180031177

CERTIFICATE

This is to certify that the Project Report entitled **"BRAIN TUMOR IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK"** is being submitted by **180030193, 180030200, 180031128, 180031177,** submitted in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to the **K L University** is a record of bonafide work carried out under guidance and supervision.

The result embodied in this report have not been copied from any other Department/universities/institutes.

Signature of the Supervisor

Dr. Chayan Paul (Asst. Professor)

Signature of the HOD

Signature of External Examiner

ACKNOWLEDGEMENTS

A successful and satisfactory completion of any significant task is the outcome of invaluable aggregate combinations of different people in the radial direction, explicitly and implicitly.

We would therefore take this opportunity to thank and express our gratitude to all those without whom the completion of project would not be possible.

We extend our sincere and heartfelt thanks to our esteemed project supervisor, **Dr. Chayan Paul**, associate professor in Computer Science and Engineering, for providing us with the guidance and advice at the crucial junctures and for showing us the right way.

We express our sincere thanks to our respected Head of the Department, **Dr. Hari Kiran Vege** for allowing us to use the facilities available.

We would like to thank the other faculty members also, at this occasion for the support and encouragement they have given us during the course of our work.

ABSTRACT

The brain tumors, are the most widely recognized and forceful sickness, prompting a short future in their most elevated evaluation. Brain tumor is the growth of large mass of abnormal cells that effects the functioning of the nervous system. Brain tumor is examined by many medical procedures like MRI,CT Scan etc. The objective of this project is to reduce the errors and difficulty in manual classification of brain tumor images by using Deep learning techniques. Here, Using Convolutional Neural Networks (CNN) brain tumors are identified. It was performed on a continuous dataset having pictures with changed tumor factors, for example, picture power, shape and size. We classify the X-ray pictures into benevolent and harmful cerebrum tissues.

The main purpose of using CNN is that the important features are detected without being monitored by anyone. That is why CNN can be an ideal solution for computer vision problems and image categories.

In our project we will use the Convolution Neural Networks, to detect tumors in the brain. If the output is 0 we can say that the brain image has no tumors if the output is 1 we can say that the person has Brain tumor.

Table of contents:

Chapter-1. Introduction 01

Chapter-2. Literature Survey 03

2.1 Research Paper1 03

2.2 Research Paper2 04

2.3 Research Paper3 05

2.4 Research Paper4 06

2.5 Research Paper5 07

2.6 Research Paper6 08

2.7 Research Paper7 09

2.8 Research Paper8 10

2.9 Research Paper9 11

2.10 Research Paper10 12

2.11 Research Paper11 12

2.12 Research Paper12 13

2.13 Research Paper13 14

Chapter-3. Theoretical Analysis 15

3.1 Challenges 15

3.2 Problem Statement 15

3.3 Proposed System 15

3.4 Objective 16

Chapter-4. Experimental Investigations 17

4.1 System Requirements 17

4.1.1 Hardware Requirements 17

4.1.2 Software Requirements 17

4.2 Software Description 17

4.3 Design 18

Chapter -5. Experimental Results 24

5.1 Neural Network 24
5.2 Convolutional Neural Network..... 26
5.3 Architecture.....28
5.4 Code Implementation..... 37
Chapter -6. Discussion of Results 44
Chapter-7 Summary and conclusion
7.1 Conclusion..... 60
Chapter -8 References 63

CHAPTER 1. INTRODUCTION

1. INTRODUCTION

Growth of huge mass of abnormal cells in the brain is known as brain tumor. Different types of brain tumors exist. Brain tumors have an effect on the functioning of the nervous system which is determined by the location and expansion rate of brain tumor.

Symptoms can be caused by a tumor that compresses the arteries or damages a part of the brain. Also, they can be caused when the tumor blocks the flow of fluid around and around the brain, or when the brain becomes inflamed due to the accumulation of fluid. Some of the common symptoms of brain tumor are: A new onset or change in pattern of headaches , nausea, vomiting , behaviour changes, memory problem, walking problem etc.,

If an individual have any of the above symptoms that suggest a brain tumor, he/she must have to go through one or more of the following tests:

• MRI: A large magnetic field connected to a computer is used to make detailed images of the areas in your head. Sometimes a special (comparable) dye is injected into a vein in your arm or hand to help differentiate between brain tissue. Pictures may show unusual areas, such as a plant.

• CT scan: A computer-assisted x-ray machine captures a series of detailed images of your head. You can get something different by injecting blood into your arm or hand. Comparison objects make unfamiliar areas easily visible. Your doctor may request further tests:

• Angiogram: If there is a tumor, an x-ray may show the tumor or blood vessels eating the tumor by performing cerebral angiogram.

• Spinal touch: Initially, Cerebrospinal fluid sample is extracted .This procedure is performed with local anesthesia. The doctor uses a long, thin needle to remove fluid from the lower part of the spinal column. Spinal tapping takes about 30 minutes. You should lie down for a few hours later so as not to get a headache. The laboratory tests fluid for cancer cells or other signs of problems.

• Biopsy: A biopsy can show cancer, tissue changes that may cause cancer, and other conditions. A biopsy is the only reliable way to diagnose a tumor in the brain, learn what stage it is, and plan treatment. Surgeons can find tissue to diagnose tumor cells in two ways:

o Biopsy at the same time as treatment: The surgeon takes a tissue sample during surgery to remove part or all of the tumor. See the Surgery section.

o Stereotactic biopsy: You can get local or general anesthesia and wear a strong scalp for this procedure. The surgeon makes a small hole in the skin and drills a small hole in the skull. CT or MRI is used to direct the needle through the burr hole to the implant site. The surgeon pulls out a tissue sample with a needle. Needle biopsy can be used when the tumor is deep inside the brain or in a part of the brain that cannot be operated on.

So, for a neurologist to detect the brain tumor accurately a patient must go through all the above medical procedures. It takes a lot of effort and time, so reduce the the difficulty in detection of brain tumor we can use CNN. And sometimes the tumor may be very minute i.e. in the first stage, then it may be difficult even for the medical experts to detect it , so even then we can use CNN to reduce manual errors.

A dataset of 1518 images has been taken of which 926 are classified into tumors and remaining into non-tumorous. The use of deep learning techniques for automation segmentation is very beneficial for reducing the time and effort involved in manual task. **(Aswathy, 2014)**

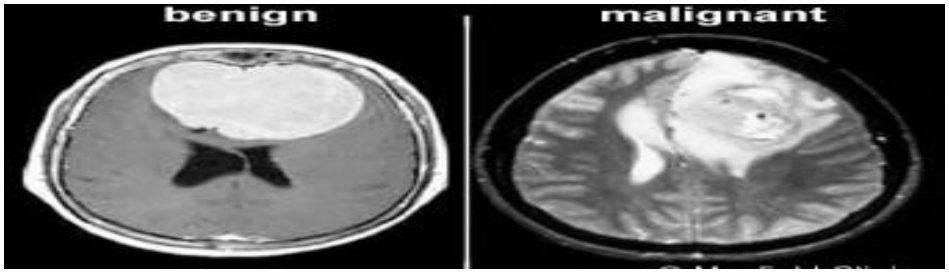


Figure 1.1: Benign and Malignant MRI scans of the brain

CHAPTER 2. LITERATURE SURVEY

2.1 Title: Detection and analysis of brain tumor from MRI by Integrated Thresholding and Morphological Process with Histogram based method. Authors: Harshini Badisa, Madhavi Polireddy, Aslam Mohammed.

This paper depicts a framework that may distinguish brain tumors all of the extra efficaciously and examination the diverse highlights of the tumor. Our framework proposed a laptop helped image-making ready primarily based method to that gives progressed exactness tempo of the cerebrum tumor identification alongside the estimation of the tumor length (floor location of the tumor) and its vicinity. It likewise furnishes with the facts that assist with figuring out if the tumor is harmful or not. The framework, we depicting right now, brain tumors from MRI by using coordinated thresholding and morphological process with histogram-based totally approach and offers careful research. In our evaluation, we've utilized BRATS database of appealing reverberation pictures. The tempo of fruitful discovery of our method is 86.84%.

CT sweeps and X-ray finding are applied to apprehend cerebrum tumor. X-ray makes use of appealing fields, in preference to x-beams, to create point via factor photos of the body. specific identity is very big in cerebrum tumor identity. The pace of precision may be expanded by making use of laptop supported framework. This framework can assist the radiologist to become aware of thoughts tumor all the extra certainly. Our check offers a communication about every other computer helped technique to enhance the exactness pace of the invention of thoughts tumor along with the estimation of the tumor size and its place. Aside from it assists with finding out if the tumor is dangerous or no longer. In addition, when an apprehensive system specialist chooses to do an activity, the tumor's width, stature, vicinity, or thickness matters. This paper offers this kind of route via which we can procedure the floor territory of the tumor. it's far truly a critical factor in scientific procedure.

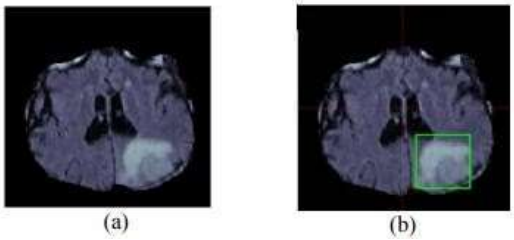


Fig 2.1 (a) Input FLAIR Image, (b) Generated Output Image

2.2 Title: Automatic Classification of Brain Tumor and Alzheimer's Disease in MRI.

Author: Bashayer Fouad Marghalania, Muhammad Arifb

PC vision (CV) and photo dealing with systems awareness on the quick development of clinical pics examine field. because the grasp units apart a long attempt to research one X-ray photographs, CV strategies, and AI calculations make the system faster than the manual manner, and those techniques spare time and exertion. right now, built up a savvy manner for the discovery and characterization of cerebrum pathologies like tumors, Alzheimer's malady (advertising), or usual brain pics. The proposed calculation envelops 4 phases: attractive Reverberation Imaging (MRI) photo acquiring, pre-handling, include extraction, and grouping. Right now, Sack of Highlights module has been utilized for the order of the X-ray of brain with tumor and X-ray of cerebrum of Alzheimer's sickness patients from ordinary brain X-ray. Normal arrangement exactness of 97% is accomplished for each of the three classes.

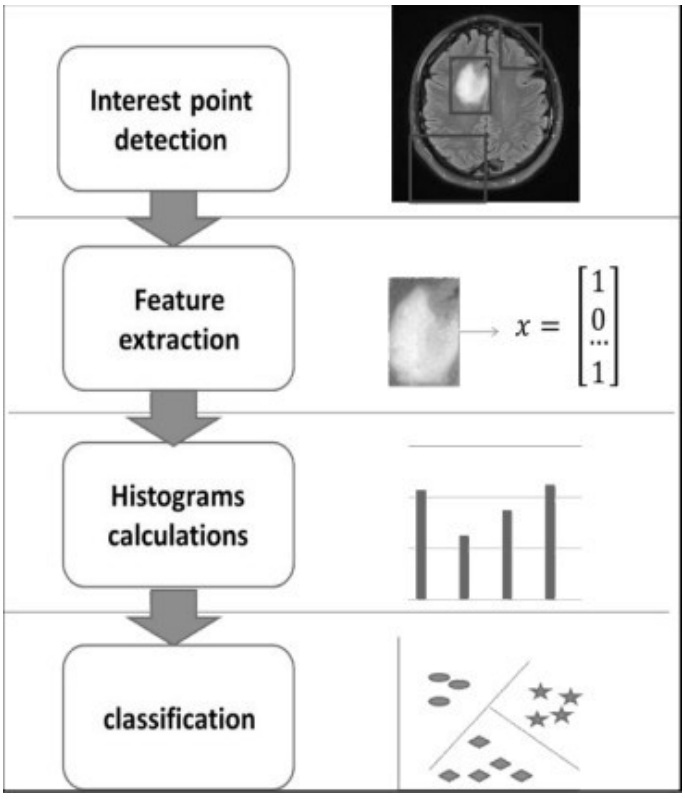


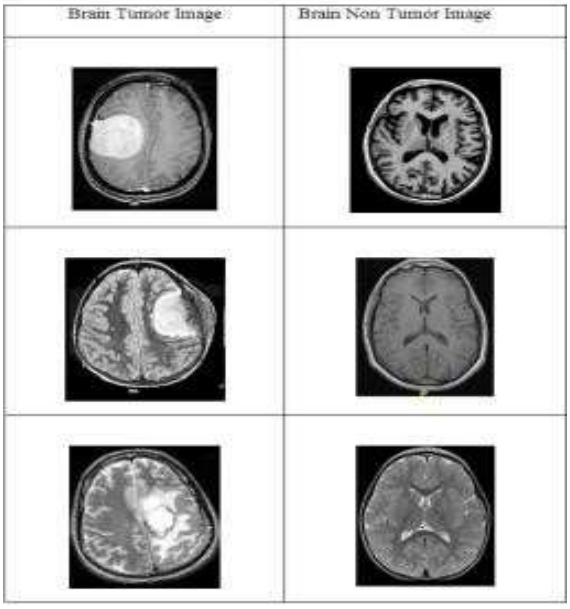
Fig 2.2 Flow of Bag of Features model

2.3 Title: Brain Tumor Classification Using Convolutional Neural Networks.

Author: J. Seetha and S. Selvakumar Raja.

The brain tumors, are the most widely recognized and forceful infection, prompting a short future in their most elevated evaluation. Consequently, treatment arranging is a key stage to improve the personal satisfaction of patients. For the most part, different picture methods, for example, Figured Tomography (CT), Attractive Reverberation Imaging (MRI)and ultrasound picture are utilized to assess the tumor in a cerebrum, lung, liver, bosom, prostate… and so on. Particularly, right now pictures are utilized to analyze tumor in the brain. Anyway, the colossal measure of information created by X-ray filter upsets manual characterization of tumor versus non-tumor in a specific time. Be that as it may, it having some restriction (i.e) exact quantitative estimations is accommodated predetermined number of pictures. Thus, trusted and programmed characterization conspire are fundamental to forestall the passing pace of human. The programmed cerebrum tumor order is testing task in enormous spatial and basic inconstancy of encompassing locale of brain tumor. Right now, cerebrum tumor identification is proposed by utilizing Convolutional Neural Systems (CNN) characterization. The more intense engineering configuration is performed by utilizing little parts. Trial results show that the CNN documents pace of 97.5% precision with low intricacy and contrasted and the all-other condition of expressions techniques. (Chahal, 2020)

Fig 2.3 CNN based classified results

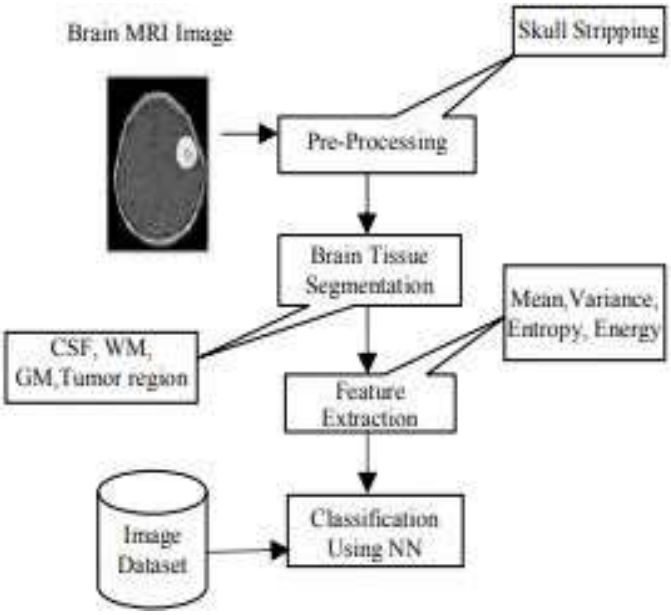


2.4 Title: Combining Tissue Segmentation and Neural Network for Brain Tumor Detection.

Author: Selvaraj Damodharan and Dhanasekaran Raghavan.

The definitive arrangement in an enormous number of pictures preparing applications is to take out the critical highlights from picture information, in which a portrayal, translation, or comprehension of the scene can be given by the machine Separation of the brain tumor from attractive retrieval images (MR) is an important, yet tedious task performed by medical professionals. Right now, have introduced a compelling cerebrum tumor location method dependent on Neural System (NN) and our recently planned brain tissue division. This strategy hits the objective with the guide of the accompanying significant advances, which incorporates: Pre-preparing of the brain pictures, division of neurotic tissues (Tumor), typical tissues, extraction of the applicable highlights from each fragmented tissues and arrangement of the tumor pictures with NN. Too, the test results and investigation is assessed by methods for Quality Rate (QR) with typical and the unusual Attractive Reverberation Imaging (X-ray) pictures. The presentation of the proposed method is been approved and contrasted and the standard assessment measurements, for example, affectability, explicitness and exactness esteems for NN, K-NN arrangement and bayesian order systems. The acquired outcomes delineate that the grouping results yields better outcomes in NNs when contrasted and different procedures.

Fig 2.4 Block diagram of the proposed technique.



2.5 Title: Convolutional Neural Network for Brain Tumor Analysis Using

MRI Images.

Author: Sourabh Hanwat, Chandra J.

Brain Malignant growth is one of the riskiest issues today. Cerebrum Tumor is controlled development of harmful or non-dangerous undesirable cells in the brain. In the current world cerebrum tumor are a risky infection and the fundamental explanation behind numerous passing. Attractive Reverberation Imaging is generally utilized the clinical picture for the brain tumor investigation. The primary goal of the paper is to characterize the brain tumor different stages utilizing Convolutional Neural System algorithm based on Cerebrum X-ray images. Brain tumor investigation is finished with the assistance of Convolutional Neural System and the work is additionally contrasted and another well-known AI classifier like Arbitrary Woods and K Closest Neighbors. During the correlation, the Convolutional Neural System is considered as truly outstanding classifiers for ordering the different phases of a brain tumor. The normal precision of the cerebrum tumor characterization with the assistance of Convolutional Neural System classifier is 98% with cross-entropy is 0.097 and approval exactness is 71% so the Convolutional Neural System is seen as one of the proficient techniques for performing various phases of cerebrum tumor order. (Thillaikkarasi, 2019)

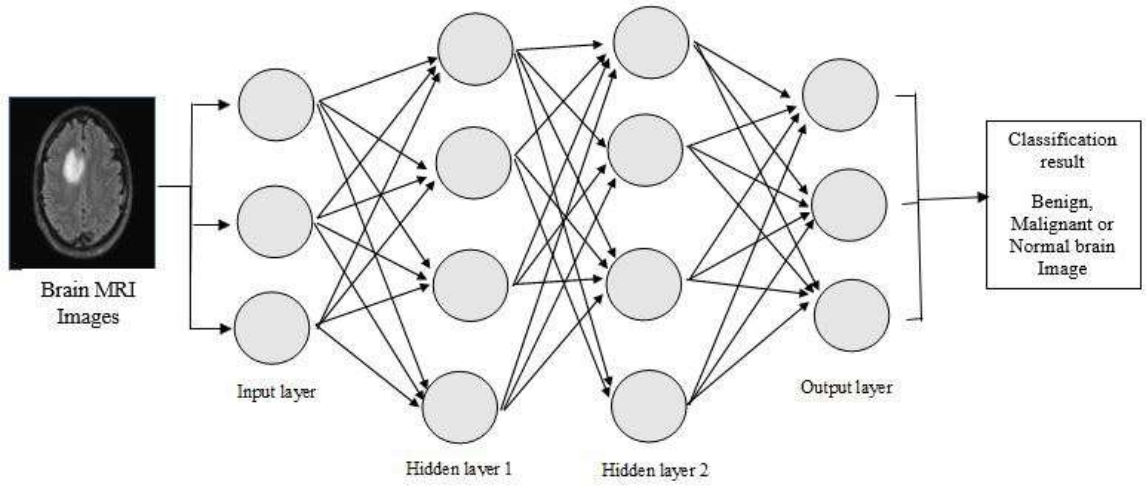


Fig 2.5 Structure of a Convolution Neural Network

2.6 Title: Brain Tumor Detection Using Convolutional Neural Network. Author: Tonmoy Hossain, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, Faisal Muhammad Shah.

Cerebrum Tumor division is one of the most essential and difficult assignments in the territory of clinical picture handling as a human-helped manual order can bring about off base forecast and determination. In addition, it is an irritating undertaking when there is a lot of information present to be helped. Brain tumors have high assorted variety in appearance and there is a closeness among tumor and typical tissues. In this manner the extraction of tumor varies from pictures to picture and gets relentless. Right now, proposed a technique to separate cerebrum tumor from 2D Attractive Reverberation brain Pictures (X-ray) by Fluffy C-Means grouping calculation which was trailed by customary classifiers and convolutional neural system. The exploratory examination was carried on an ongoing dataset with assorted tumor sizes, areas, shapes, and diverse picture powers. In conventional classifier part, we applied six customary classifiers specifically Bolster Vector Machine (SVM), K-Closest Neighbor (KNN), Multilayer Perceptron (MLP), Calculated Relapse, Innocent Bayes and Arbitrary Woods which was actualized in scikit-learn. A short time later, we proceeded onward to Convolutional Neural System (CNN) which is executed utilizing Keras and TensorFlow on the grounds that it respects a superior presentation than the customary ones. In our work, CNN increased an exactness of 97.87%, which is exceptionally convincing. The fundamental point of this paper is to recognize ordinary and irregular pixels, in light of surface based and factual based highlights. (Begum, 2020)

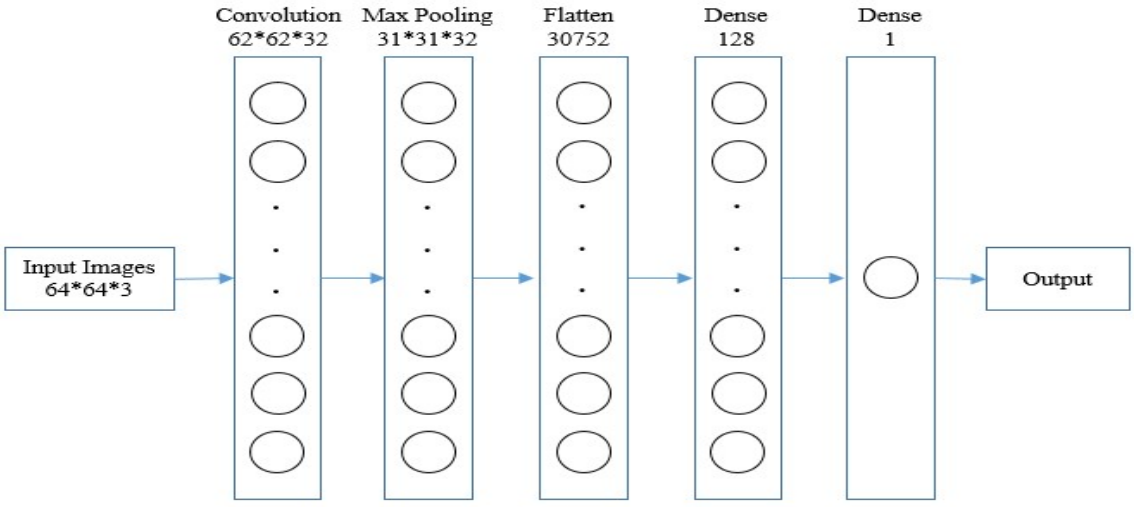


Fig 2.6 Proposed Methodology for tumor detection using 5-Layer Convolutional Neural Network

2.7 Title: Brain tumor detection based on a Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy.

Author: Fatih Ozyurt, Eser Sert, Engin Avci, Esin Dogantekin.

Brain tumor order is a difficult errand in the field of clinical picture preparing. In This examination we proposed a crossover strategy using Neutrosophy and Convolutional Neural System (NS-CNN). It expects to characterize tumor district regions that are fragmented from cerebrum pictures as generous and harmful. In the first organize, X-ray pictures were sectioned using neutrosophic set - master greatest fluffy sure entropy (NS-EMFSE) approach. The best images of cerebrum images divided into programs were obtained by CNN and ordered using SVM and KNN separators. Test assessment was completed dependent on 5-crease cross-approval on 80 of generous tumors and 80 of defame tumors. The discoveries exhibited that the CNN highlights showed a high grouping presentation with various classifiers. Test results demonstrate that CNN highlights showed a superior arrangement execution with SVM as reproduction results approved yield information with a normal accomplishment of 95.62%.

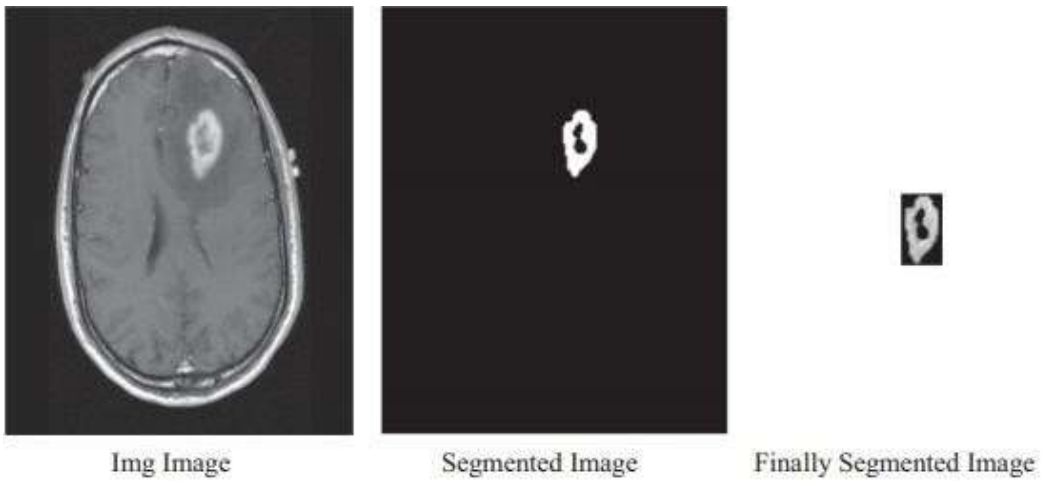


Fig 2.7 The segmentation result with using NS-EMFSE.

2.8 Title: Brain Tumor Detection using Magnetic Resonance Images with a Novel Convolutional Neural Network Model.

Author: Burhan ERGEN, Zafer COMERT.

A cerebrum tumor is a mass that develops unevenly in the brain and straightforwardly influences human life. This mass happens immediately as a result of the tissues encompassing the cerebrum or the skull. Brain tumors can be cured by using careful techniques. As of late, models of profound learning in the conclusion and treatment of illnesses in the biomedical field have increased serious premium. Right now, propose another convolutional neural system model named Brain MR Net.

This design is based on consideration modules and hypercolumn method; it has a remaining system. Initially, picture is preprocessed in Brain MR Net. At that point, this progression is moved to consideration modules utilizing picture enlargement strategies for each picture. Consideration modules select significant regions of the picture and the picture is moved to convolutional layers. One of the most significant procedures that the Brain MR Net model uses in the convolutional layers is hypercolumn. With the assistance of this method, the highlights separated from each layer of the Brain MR Net model are held by the exhibit structure in the last layer. The point is to choose the best and the most effective highlights among the highlights kept up in the cluster.

Open attractive reverberation pictures were utilized to distinguish brain tumor with the Brain MR Net model. Brain MR Net model is more fruitful than the pre-prepared convolutional neural system models (AlexNet, GoogleNet, VGG-16) utilized right now. The order achievement accomplished with the BrainMRNet model was 96.05%.

2.9 Title: Detection of Brain Tumor from MRI images by using Segmentation SVM.

Author: Swapnil R. Telrandhe, Amit Pimpalkar, Ankita Kendhe.

Right now, propose versatile brain tumor recognition, Picture handling is utilized in the clinical instruments for identification of tumor, just X-ray pictures are not ready to recognize the tumorous locale right now are utilizing K-Means division with preprocessing of picture. Which contains denoising by Middle channel and skull concealing is utilized. Additionally, we are utilizing object naming for progressively nitty gritty data of tumor district. To make this framework a versatile we are utilizing SVM (Bolster Vector Machine), SVM is utilized in unaided way which will use to make and keep up the example for some time later. Additionally, for designs we need to discover the component to prepare SVM. For that here we have discovered the surface element and shading highlights. It is normal that the exploratory consequences of the proposed framework will give better bring about correlation with other existing frameworks.

It is an essential to discover tumor from X-ray pictures however it is fairly tedious and troublesome errand at some point performed physically by clinical specialists. Enormous measure of time was spent by radiologist and specialists for ID of tumor and sectioning it from other cerebrum tissues. In any case, accurate naming cerebrum tumors is a tedious assignment, and impressive variety is seen between specialists [2]. Along these lines, in the course of the most recent decade, from different research results it is being seen that it is very tedious strategy however it will get quicker on the off chance that we use picture handling procedures. Essential cerebrum tumors do not spread to other parts of the body and can be dangerous or malignant and helpful brain cells remain dangerous. Harmful tumor is more perilous and dangerous than favorable tumor.

2.10 Authors: S. K. Shil, F.P. Polly, M. A. Hossain, M. s. Ifthehar, M. n. Uddin,

Title: An Improved Brain Tumor Detection and Classification Mechanism, 2017

S. K. Shil, F.P. Polly, M. A. Hossain, M. s. Ifthehar, M. n. Uddin used Otsu binarization followed by K means clustering for the segmentation of the brain tumor images. Discrete Wavelet Transform then followed by Principle Component Analysis was to extract features and reduce the dimensions of the features. The classification was done by Support Vector Machine.

2.11 Author: Moi Hoon Yap, Gerard Pons, Joan Marti, Sergi Ganau, Melcior Sentis, Reyer Zwiggelaar, Adrian K. Davison, Robert Mart

Title: Automated Breast Ultrasound Lesions Detection Using the Convolutional Neural Networks.

Based on in-depth study methods for breast ultrasound diagnosis and they came up with three different methods: Patch-based LeNet, U-Net and a pre-trained FCN-AlexNet-trained transmission method. Their performance was compared to the four wound detection algorithms namely, Radial Gradient Index, Multifractal Filtering, Rule-based Region Ranking and models of the paralyzed components. Among the three different methods, Transfer Learning FCN-AlexNet produced the best results. However, it was found that the reversal of the use of such methods was that they required a training process and negative images in the study.

2.12 Title: Automated Breast Ultrasound Lesions Detection Using CNN.

Authors: Moi Hoon Yap, Gerard Pons, Joan Marti, Sergi Ganau, Melcior Sentis, Reyer Zwiggelaar, Adrian K. Davison, Robert Mart.

Bosom sore location utilizing ultrasound imaging is viewed as a significant advance of PC Helped Conclusion frameworks. Over the previous decade, specialists have shown the conceivable outcomes to robotize the underlying sore discovery. Be that as it may, the absence of a typical dataset blocks inquiries about when looking at the presentation of such calculations. This paper proposes the use of in-depth study techniques for the detection of chest ulcers and explores three unique techniques: Fix based LeNet, U-Net, and the pre-trained FCN-AlexNet-trained learning method.Their presentation is looked at against four best in class sore discovery calculations (for example Outspread Slope Record). What's more, this paper thoroughly analyzes two customary ultrasound picture datasets obtained from two diverse ultrasound frameworks. Dataset An involves 306 (60 dangerous and 246 favorable) pictures and Dataset B contains 163 (53 harmful and 110 considerate) pictures. To beat the absence of open datasets right now, B will be made accessible for investigate purposes. The outcomes shows us a general improvement by profound learning approaches when they were evaluated on both datasets regarding Genuine Positive Part, Bogus Positives per picture, and F-measure.

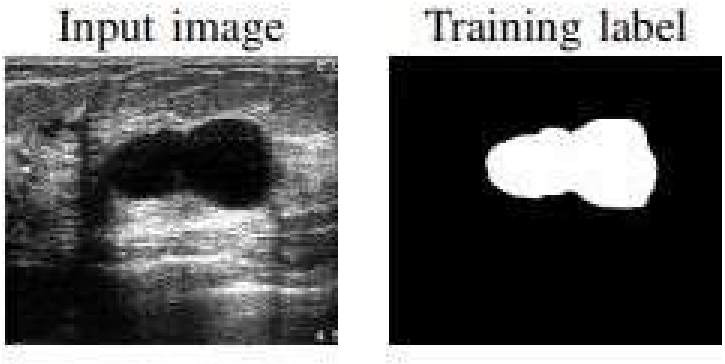


Fig 2.12 An example input image for training the network, with the training label used for U-Net and FCN-AlexNet.

2.13 Title: Computerized Detection of Mass Lesions in Digital Breast Tomosynthesis Images Using Two and Three Dimensional Radial Gradient Index Segmentation.

Author: I. Reiser,R. M. Nishikawa,M. L. Giger, T. Wu, E. Rafferty, R. H. Moore, D. B. Kopans.

Starting outcomes for an automated mass injury discovery plot for advanced bosom tomosynthesis (DBT) pictures are introduced. The calculation utilizes an outspread slope record include for the underlying injury recognition and for division of sore up-and-comers.

A lot of highlights are removed for each portioned segment. Execution of two-and three-dimensional highlights was analyzed. For inclination includes, the extra measurement gave no improvement in order execution. For shape highlights, grouping utilizing 3D highlights was improved contrasted with the 2D proportional highlights. The primer by and large execution was 76% affectability at 11 bogus positives for every test, assessed dependent on DBT picture information of 21 masses. A bigger database will take into account further advancement and improvement in our PC helped discovery plot.

Advanced Bosom Tomosynthesis (DBT) has as of late rose as a potential new device in bosom imaging (1, 2), inferable from the advancement of enormous territory, quick readout identifiers for mammography (3). In DBT, the bosom volume is remade from a progression of projection pictures taken at a scope of source edges. DBT gives cuts of the three- dimensional bosom volume at a dividing on the request for 1 mm. Not at all like in traditional mammography, overlaying tissue structures are settled. Quiet portion conveyed during DBT picture procurement is tantamount or not exactly that of a mammographic test (comprising of two screen-film projection pictures)

The presence of DBT bosom cuts is like that of traditional mammograms, permitting radiologists to utilize aptitudes gained in the mammography practice. A pilot concentrate by
E. Rafferty demonstrated that sore conspicuity was expanded essentially when seen as DBT pictures instead of survey the comparing screen-film mammograms (4). Anyway this comes at the expense of the radiologist exploring 50 to 90 pictures for each bosom, contingent upon bosom thickness, contrasted with the two perspectives on a standard mammographic test. Accordingly, we are creating mechanized mass injury discovery for DBT to give a perusing help to the radiologist.

CHAPTER-3. THEORETICAL ANALYSIS

3.1 Challenges:

The brain tumors, are the most widely recognized and forceful sickness, prompting a short future in their most elevated evaluation. In the field of medicine, different parts of the image are used for the analysis of different structures and tissues, the distribution of functional areas and pathologic regions. MRI scans are very different and therefore make it difficult to separate them from cancer-free tissue. The automatic segmentation need overcome the problems of manual segmentation, it not only requires less time from human experts, but can also provide fewer variable results.

3.2 Problem Statement:

The cerebrum tumor discovery is testing task in enormous spatial and auxiliary changeability of encompassing district of brain tumor. Here, tumor location is identified by utilizing Convolutional Neural Networks (CNN) arrangement. It was performed on a continuous dataset having pictures with changed tumor factors, for example, picture power, shape and size. The point of this project is to classify the X-ray pictures into benevolent and harmful cerebrum tissues.

3.3 Proposed System:

The human brain is modeled by using the design and implementation of neural networks. The convolution neural network is employed for brain tumor detection. The brain image dataset is taken from the Kaggle repository is one of the pre-trained models. In the convolution layer, the given input image is segregated into various small regions. Element wise activation function is carried out in ReLU layer and the pooling layer is used for down sampling. In the final layer fully, the connected layer is used to generate the class score. In the proposed CNN, we will train the last layer in python implementation. We don’t want to train all the layers. So, computation time is less meanwhile the performance is high within the proposed brain tumor detection scheme.

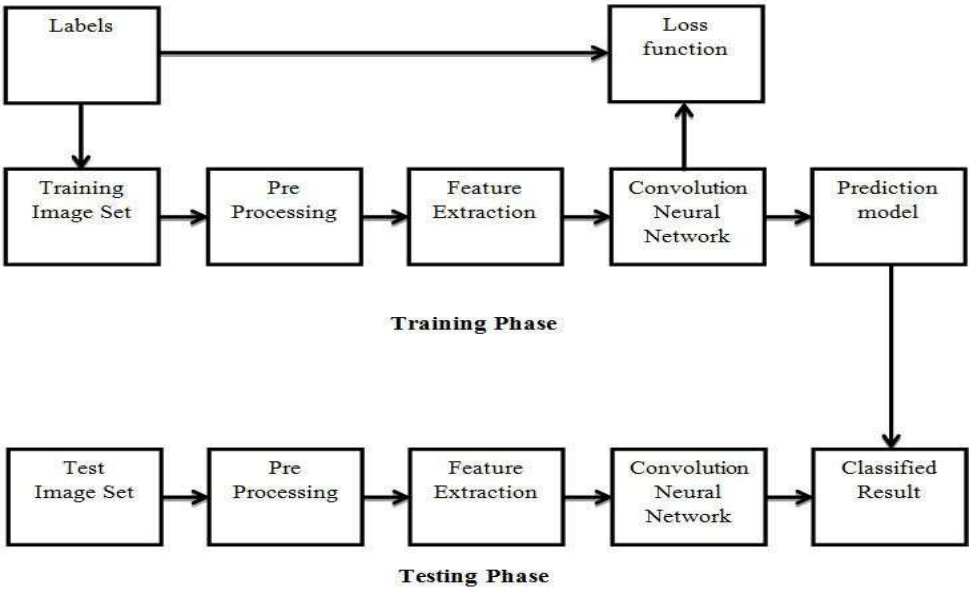


Figure 1: Block diagram of proposed brain tumor classification using CNN

3.3.1 Block diagram of CNN

3.4 Objective:

The objective of this project is to reduce the errors and difficulty in manual classification of brain tumor images by using Deep learning techniques. Convolutional Neural Networks has been used for feature extraction and image classification.

CHAPTER 4 EXPERIMENTAL ANALYSIS

4.1 SYSTEM SPECIFICATION:

Hardware Requirements:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor : 14’ Color Monitor.
- Mouse : Optical Mouse.
- Ram : 512 Mb.

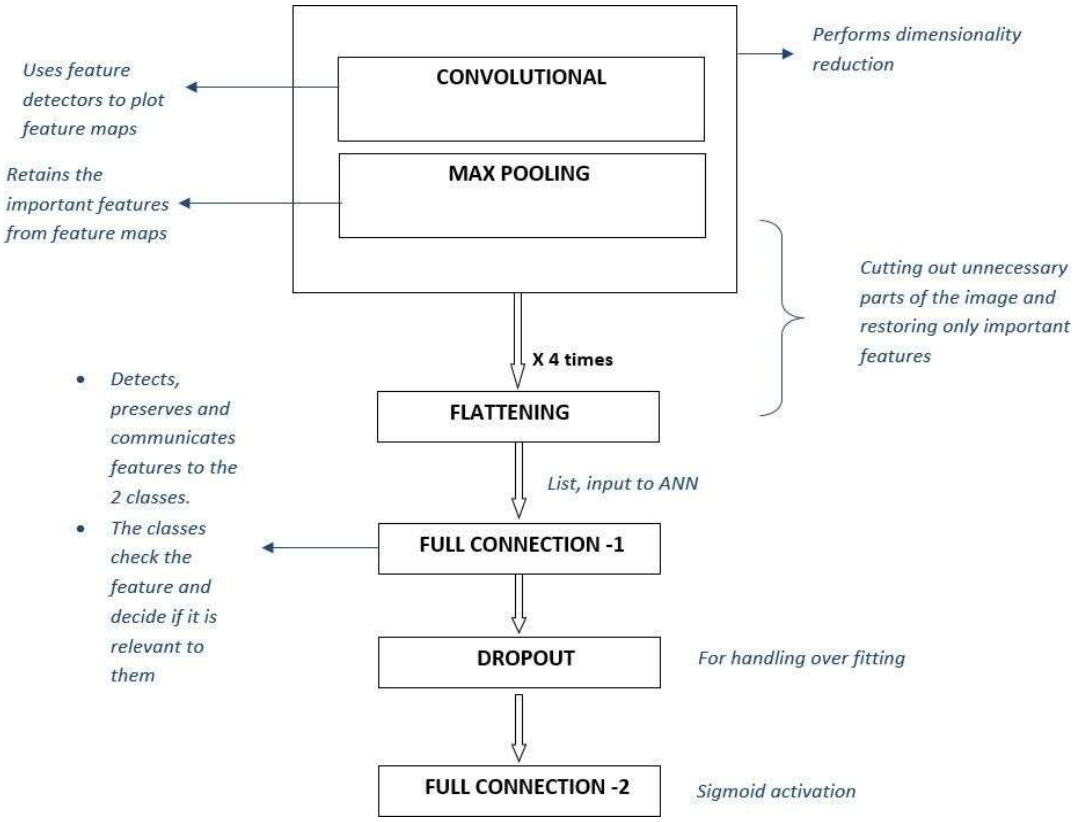
Software Requirements:

- Operating system : Windows
- Coding Language : Python
- Software Tool : Anaconda Navigator Jupyter notebook

4.2 SOFTWARE DESCRIPTION

The software tool we used is Anaconda Navigator which is an intuitive user interface for the anaconda distribution that will allow us to launch applications and manage conda packages, location, and channels without using command line commands. The Jupyter notebook application allows us to create and edit documents that display the input and output of the Python and R languages automatically installed. This allows us to create and share documents that contain live code, statistics, visuals, and text.

4.3 DESIGN (Flowchart)



4.3.1 Brain tumor

Under certain conditions, brain cells grow and multiply uncontrollably because for a few reasons, the mechanism that control normal cells is unable to manage the expansion of the brain cells. The abnormal mass of brain tissue is that the brain tumour that occupies space within the skull and interrupts the traditional functions of brain and creates an increasing pressure within the brain. Due to increased pressure on the brain, some brain tissues are shifted, pushed against the skull or are liable for the damage of the nerves of the opposite healthy brain tissues. Scientists have classified brain tumour consistent with the situation of the tumor, sort of tissue involved, whether or not they are noncancerous or cancerous. The site of the origin) and other factors involved. (Bahadure, 2017)

World Health Organization (WHO) classified brain tumour into 120 types. This classification is completed on the idea of the cell origin and therefore the behaviour of the cell from less aggressive to more aggressive behaviour. Even, some tumor types are graded starting from grade I (less malignant) to grade IV (more malignant). This signifies the speed of the expansion despite of variations in grading systems which depends on the sort of the tumor. Primary brain

tumors are the tumors that originated within the brain and are named for the cell types from which they originated. They can be benign (non-cancerous) and malignant (cancerous). Benign tumors grow slowly and don't spread elsewhere or invade the encompassing tissues. However, occupying a quick space, even the less aggressive tumor can exercise much pressure on the brain and makes it dysfunctional. Otherwise, more violent tumors can grow with in no time and spread to other tissues. Each of those tumors has unique clinical, radiographic and biological characteristics. Secondary brain tumors originate from another a part of the body. These tumors consist of cancer cells somewhere else in the body that have spread to the brain. The most common explanation for secondary brain tumors are: carcinoma , carcinoma , melanoma, kidney cancer, bladder cancer, certain sarcomas, and testicular and reproductive cell tumors.

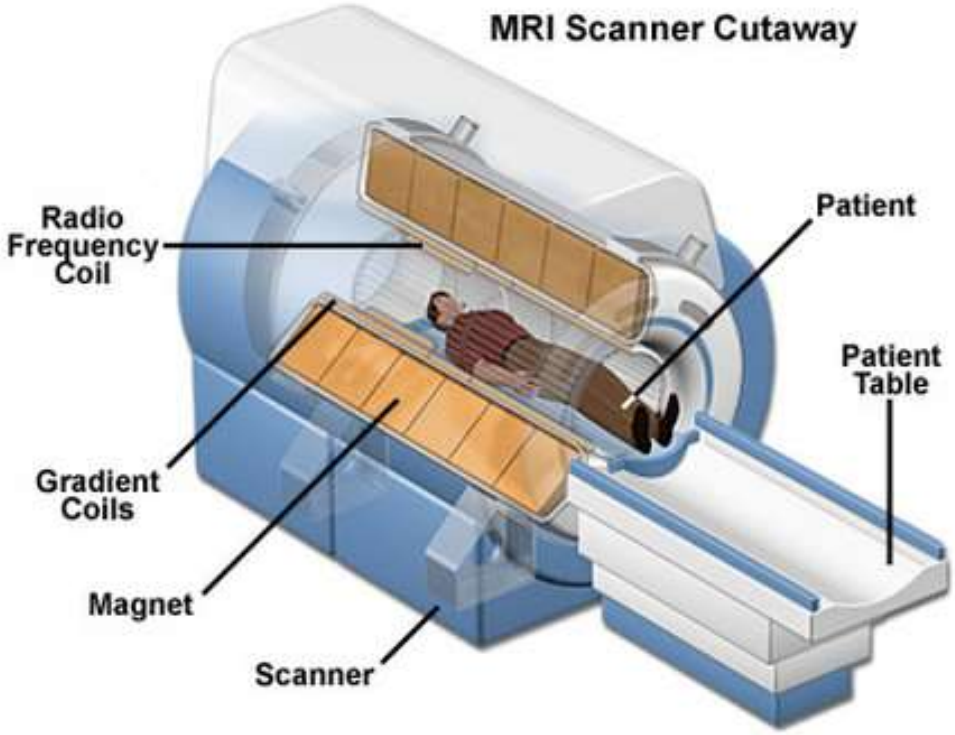
4.3.2 Under certain conditions, brain cells grow and multiply uncontrollably because for a few reasons, the mechanism that control normal cells is unable to manage the expansion of the brain cells. The abnormal mass of brain tissue is that the brain tumour that occupies space within the skull and interrupts the traditional functions of brain and creates an increasing pressure within the brain. Due to increased pressure on the brain, some brain tissues are shifted, pushed against the skull or are liable for the damage of the nerves of the opposite healthy brain tissues. Scientists have classified brain tumour consistent with the situation of the tumor, sort of tissue involved, whether or not they are noncancerous or cancerous. The site of the origin) and other factors involved.

4.3.3 World Health Organization (WHO) classified brain tumour into 120 types. This classification is completed on the thought of the cell origin and thus the behaviour of the cell from less aggressive to more aggressive behaviour. Even, some tumor types are graded starting from grade I (less malignant) to grade IV (more malignant). This signifies the speed of the expansion despite of variations in grading systems which depends on the sort of the tumor. Primary brain tumors are the tumors that originated within the brain and are named for the cell types from which they originated. they're going to be benign (non-cancerous) and malignant (cancerous). Benign tumors grow slowly and don't spread elsewhere or invade the encircling tissues. However, occupying a quick space, even the less aggressive tumor can exercise much pressure on the brain and makes it dysfunctional. Otherwise, more violent tumors can grow with in no time and spread to other tissues. Each of those tumors has unique clinical, radiographic and biological characteristics. Secondary brain tumors originate from another an area of the body. These tumors contains cancer cells elsewhere within the body that have spread

to the brain. the foremost common explanation for secondary brain tumors are: carcinoma , carcinoma , melanoma, kidney cancer, bladder cancer, certain sarcomas, and testicular and reproductive cell tumors.MR imaging (MRI): Raymond V. Damadian invented MRI in 1969 and was the first person to use MRI to research the human body . Eventually, MRI became the foremost preferred imaging technique in radiology because MRI enabled internal structures be visualized in some detail. With MRI, good contrast between different soft tissues of the body are often observed. This makes MRI suitable for providing better quality images for the brain, the muscles, the center and cancerous tissues compared with other medical imaging techniques, like computed tomography (CT) or X-rays.

4.3.4 MRI Brain Imaging and Characteristics of Brain Tumors:

There is a proliferation of imaginative techniques commonly used to study brain tumors, such as: resonance imaging (MRI), computerized tomography (CT), positron emission tomography (PET), and single photon emission computer tomography (SPECT) imaging and -cerebral angiography. In recent years, CT and MR imaging are the most widely used methods, due to their wide availability and their ability to provide high-resolution images of common anatomic structures and pathological tissue. Magnetic resonance imaging (MRI) may be a common way to visualize pathological changes or other biological tissue and is often used to scan a brain tumor for the following reasons: It does not use ionizing radiation such as CT, SPECT and PET Correction its of diversity. superior to other techniques mentioned above The ability of MRI devices to produce 3D space images enables them to locate a higher tumor's ability to detect both functional and anatomical information about the tumor during the same scan. Before discussing the MR image features of brain tumors, it is important to explain the applicable law of MR imaging. During MR imaging, the patient is placed during a dynamic magnetic field that causes the protons within the body molecule to align or parallel (low power) or anti-parallel (high power) with magnetic flux. Then a radiofrequency pulse is introduced that forces the rotating protons out of balance. When the frequency pulse is stopped, the protons return to normal and produce a sinusoidal signal with a frequency connected to the local magnetic flux. Radio frequency coils Coils coils inside the scanner receive signal and create an image.



4.3.5 Magnetic-resonance imaging (MRI) is an imaging technique used primarily in medical settings to supply top quality images of the within of the physical body . A MRI is analogous to CT, but it doesn't use X-rays. Instead, a strong, magnetic flux is employed to affect the orientation of protons, which behave like miniature magnets and have a tendency to align themselves with the external field.

4.3.6 Difficulties in detection of brain MRI:

Problems related to MRI sound. Random audio connected to the MR thinking system is a problem. The intensity of homogeneity is also referred to as the bias field or artifact shading: the difference between the frequency (RF) field during data collection results in a blurring effect. Partial volume effect: during this type of effect one type of class or tissue takes one pixel or voxel image. Pixels or voxels are called mixels. Separation of MRI results is usually done by a doctor and requires time-consuming procedures. because tumor tissue images from different patients contain many different shapes and intensity of gray matter and generally look like normal tissue, the automatic method of classifying MRI results faces many challenges. one of these challenges is overcome by using prior knowledge about normal brain appearance when making a distinction from a set of multi-dimensional volume feature. Self-diagnosis and analysis of tumor images in MR brain radiologists are performed in almost all hospitals at present. (Bandar, 2020)

The reliability of the categories depends on the knowledge and skill of the radiologists. However, this strategy is manual, tedious, time-consuming, extremely popular and does not work in modern medical imaging where multiple patient photographs are taken. As a result, there is a strong need for automated plant detection and segregation process. However, there are many efforts and promising results in the medical photography community.

There are still some challenges such as precise and repetitive classification and aberration of abnormal signals using intelligent algorithms due to variations in shape, location and intensity of images of various brain tumors. This project focuses on developing an automated system for brain tumor detection and classification. this may enhance the detection and visualization of brain tumors from the output of MRI scans. Sample MRI scan images of Brain Tumor

1. Cancerous

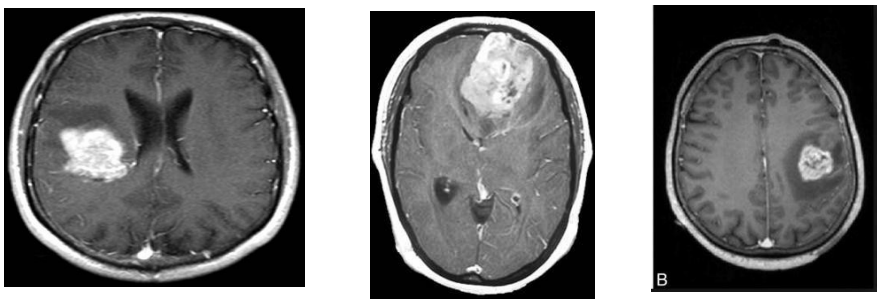


Fig 4.3.4.1 MRI scan of cancerous tumor

2. Non-Cancerous

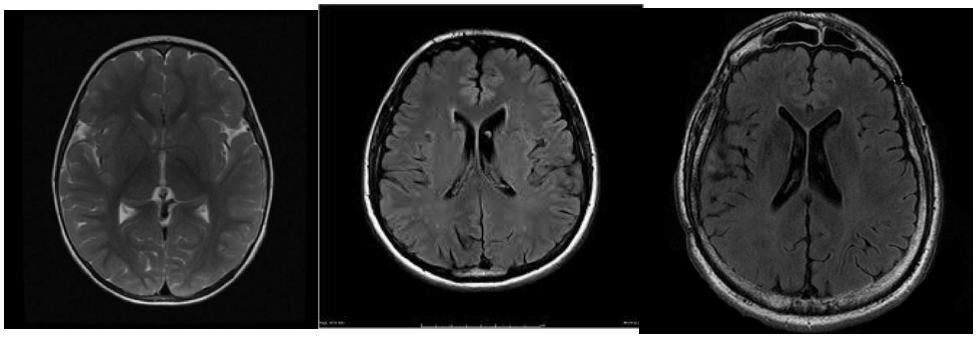


Fig 4.3.4.1 MRI scan of non cancerous tumor

CHAPTER 5. EXPERIMENTAL RESULTS

Convolutional neural networks

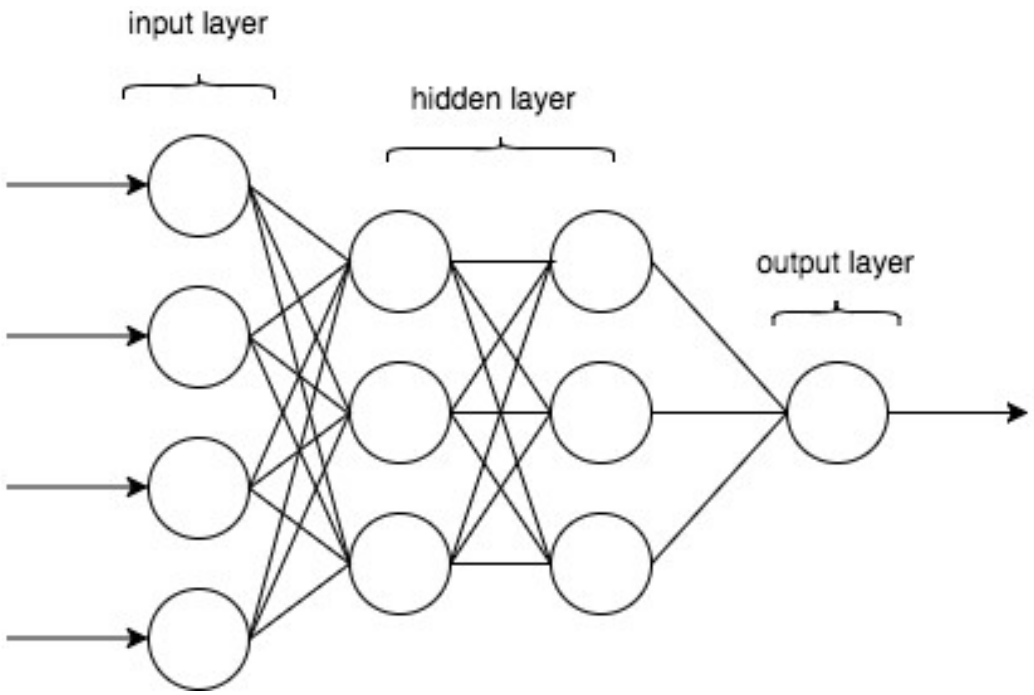
5.1 Neural networks:

The neural network is split into three types supported their interconnections. Feedback, feed forward and recurrent network are the three types of neural networks. The Feed Forward Neural network is again divided into two, a single layer network and a multilayer network. Within the single layer network, the hidden layer isn't presented. But it contains only input and output layer. However, the multilayer network consists of an input layer, a hidden layer and an output layer. The closed loop-based feedback network is named as recurrent network. Within the normal neural network, image cannot scalable. But in convolution neural network, image can scalable, it'll take 3D input volume to 3D output volume (length, width, height). The Convolution Neural Network (CNN) consists of input layer, convolution layer, Rectified linear measure (ReLU) layer, pooling layer and fully connected layer. within the convolution layer, the given input image is separated into various small regions. Element wise activation function is administered in ReLU layer. Pooling layer is optional. the pooling layer is especially used for down sampling. within the final layer (i.e.) fully connected layer is employed to get the category score or label score.

FEED FORWARD LAYER

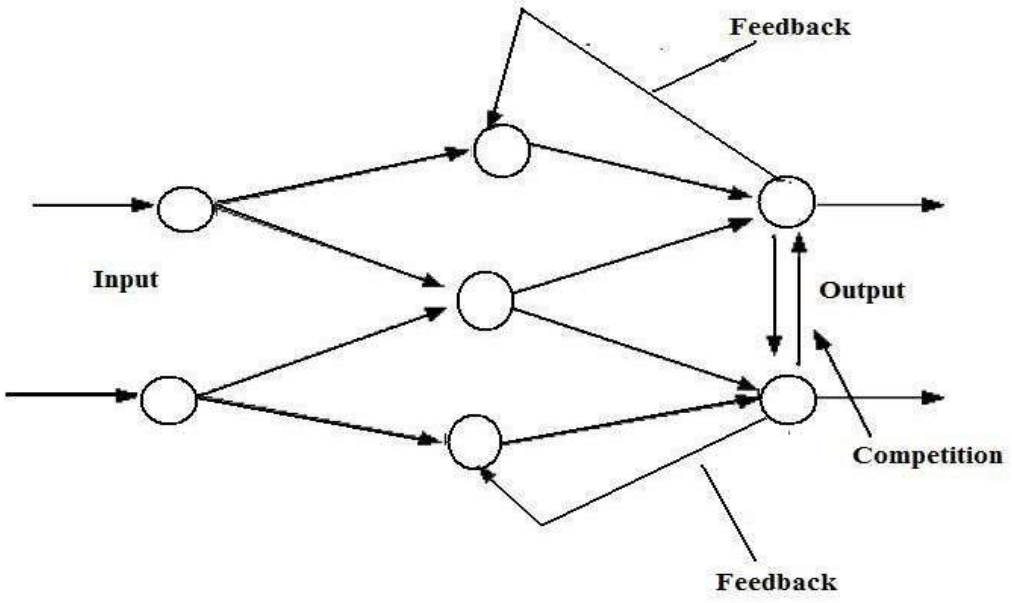
Deep Feedforward networks or also known multilayer perceptions are the inspiration of most deep learning models. Networks like CNNs and RNNs are just a few special cases of Feedforward networks. We use these networks mostly for supervised machine learning tasks since we already know the target function i.e. the result we would like our network to realize and are extremely important for practicing machine learning and form the idea of the many commercial applications, areas like computer vision and NLP were greatly suffering from the presence of those networks. for example, a regression function $y = f^*(x)$ maps an input x to a worth y. A feedforward network defines a mapping $y = f(x; \theta)$ and learns the worth of the parameters θ that end in the simplest function approximation.

The reason these networks are called feedforward is that the flow of data takes place within the forward direction, as x is employed to calculate some intermediate function within the hidden layer which successively is employed to calculate y. In this, if we add feedback from the last hidden layer to the primary hidden layer it might represent a recurrent neural network.

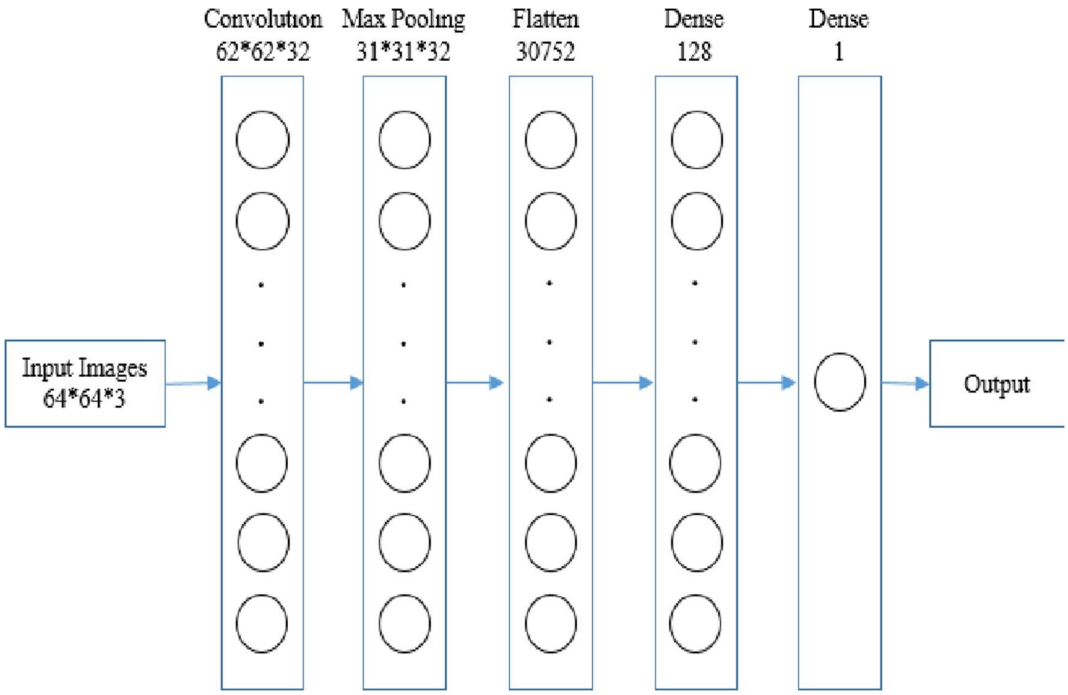


5.1.1 FEEDBACK NETWORK:

The state in such network keep changing until it reaches an equilibrium point. Until the input changes and a new equilibrium is found they remain at the equilibrium point. Feedback neural specification is additionally referred to as interactive or recurrent, though the latter term is usually wont to denote feedback connections in single-layer organizations. Feedback loops are allowed in such networks. They are used in content addressable memories.



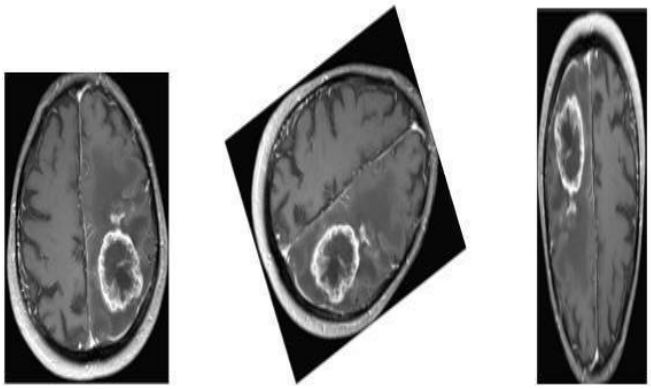
5.2 CONVOLUTIONAL NEURAL NETWORK



The beginning layer in cnn process. Converting all the pictures into 64*64*3 homogeneous dimension Convolutional kernel of three 2 convolutional filters of size 3*3 with the support of 3 tensor channels. The activation function we used is RELU function. CONV layer parameters contain a group of readable filters. Each filter has a small area but extends to the full depth of the input volume. For example, a standard filter in the main ConvNet layer may require a size 5x5x3 (i.e., 5 pixels wide and long, as well as 3 because the images have a depth of 3, color channels). During aerial, we move (more accurately, including) each filter over the width and length of the input volume and calculate the dot products between the filter inserts and therefore the input in any position. As we move the filter over the width and length of the input volume, a 2-dimensional activation map is generated and provides answers to that filter in all areas. In particular, the network will learn filters that work when they detect a certain type of visual element such as a slight shrinkage or a blotch of a particular color in the main layer, or wheel-like patterns or ultimately the entire honeycomb layer of the network layer. Now, we will have a whole set of filters in each CONV layer (e.g., 12 filters), and they will all produce a separate 2dimensional activation map.These opening maps are packaged in smart size and output volume is generated.

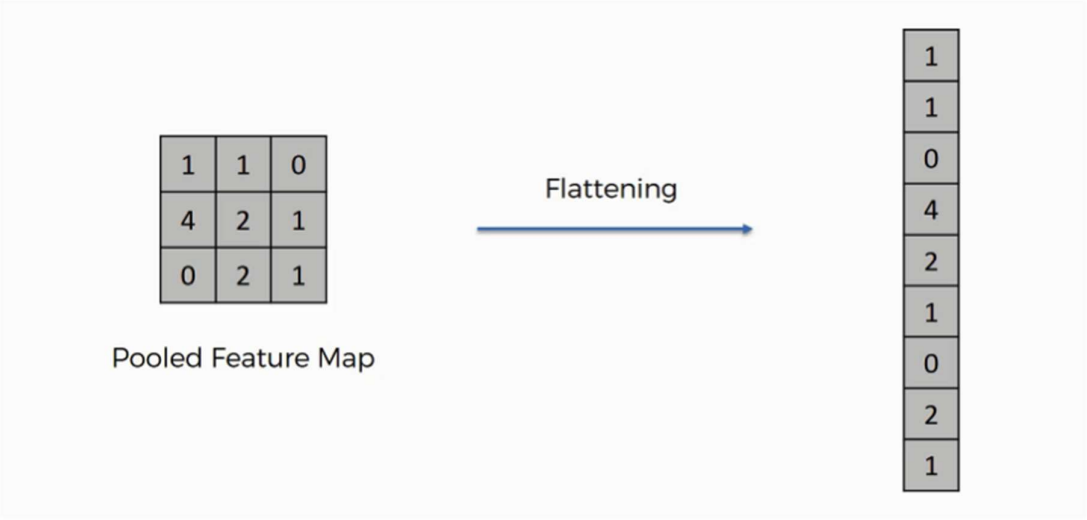
5.2.2 Pooling layer:

The function of pooling layer is to progressively reduce the spatial size of the representation to scale back the amount of parameters and computation within the network, and hence to also control overfitting. On every depth slice of the input the pooling layer operates independently and using the MAX operation, resizes it spatially. The purpose of max pooling is enabling the convolutional neural network to detect the brain tumour when presented with the image in any manner. (B.V., 2020)

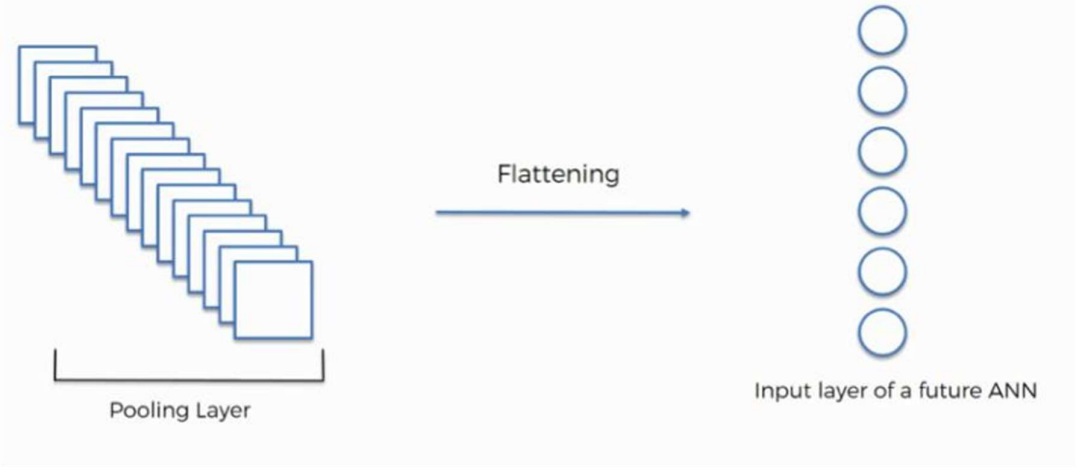


5.2.3 Flattening layer:

In this layer we flatten the pooled feature map into a column.



The reason behind the step is that we are going to insert the data into an artifificial neural network



5.2.3. multiple pooled feature maps

5.2.4 Fully connected layer:

The vector data created during the flattening step is contained within the input layer. The features that are distilled throughout the previous steps are encoded in this vector. Taking the results of the convolution/pooling process and using them in classifying the image into a label is the objective of a fully connected layer. The image of a brain features representing things like tumorous vs nontumorous should have high probabilities for the label brain.

5.3 ARCHITECTURE:

The Architecture of a network is the structure of the network i.e., the number of hidden layers and the number of hidden units present in each layer together are called as the architecture of a network. (Reddy, 2018)

5.3.1 LeNet-5 (1998)

LeNet-5 is probably the least difficult engineering. It has 2 convolutional and 3 completely associated layers (henceforth "5" — it is exceptionally regular for the names of neural systems to be gotten from the quantity of convolutional and completely associated layers that they have). The normal pooling layer as we probably are aware it presently was known as a sub-testing layer and it had trainable loads (which isn't the present act of structuring CNNs these days). This design has around 60,000 parameters.

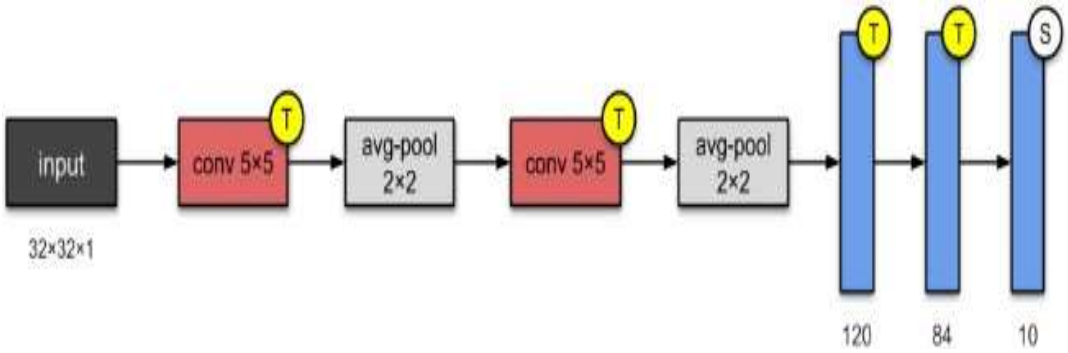


Fig5.3.1 LeNet-5 architecture, based on their paper

This engineering has become the standard 'format': stacking convolutions and pooling layers, and consummation the system with at least one completely associated layers.

5.3.2 AlexNet (2012)

With 60M parameters, AlexNet has 8 layers — 5 convolutional and 3 completely associated. AlexNet simply stacked a couple of more layers onto LeNet-5. At the purpose of distribution, the creators called attention to that their engineering was "one of the biggest convolutional neural systems to date on the subsets of ImageNet."

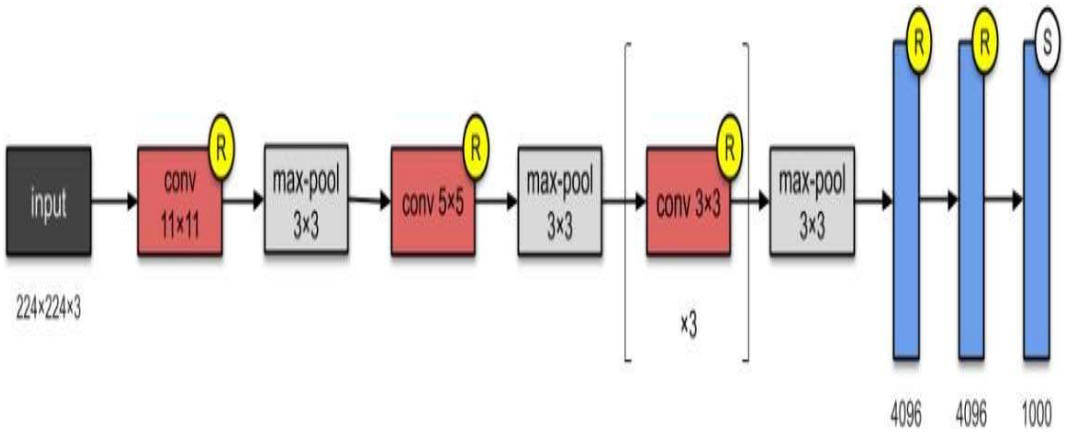


Fig 5.3.2 AlexNet architecture, based on their paper.

They were the first to actualize Rectified Linear Units (ReLUs) as initiation capacities.

5.3.3 VGG-16 (2014)

At this point you would've just seen that CNNs were beginning to get further and more profound. This is on the grounds that the most direct method for improving execution of profound neural systems is by expanding their size (Szegedy et. al). The people at Visual Geometry Group (VGG) imagined the VGG-16 which has 13 convolutional and 3 completely associated layers, conveying with them the ReLU convention from AlexNet. This system stacks more layers onto AlexNet, and utilize littler size channels (2×2 and 3×3). It comprises of 138M parameters and takes up about 500MB of extra room. They additionally structured a more profound variation, VGG-19.

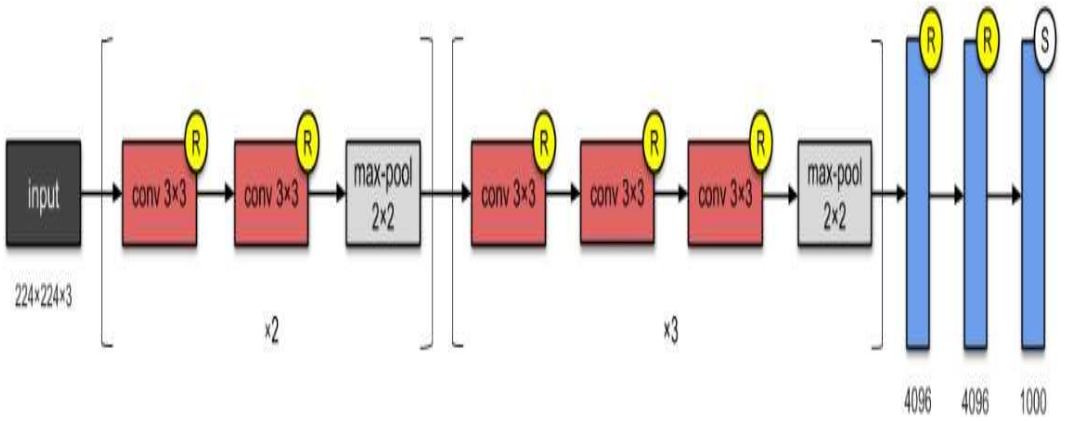


Fig 5.3.3VGG-16 architecture, based on their paper.

5.3.4 Inception-v1 (2014)

This 22-layer design with 5M parameters is known as the Inception-v1. Here, the Network In Network (see Appendix) approach is vigorously utilized, as referenced in the paper. This is finished by methods for 'Beginning modules'. The plan of the design of an Inception module is a result of research on approximating meager structures (read paper for additional!). Every module presents 3 thoughts:

1. Having equal towers of convolutions with various channels, trailed by link, catches various highlights at 1×1, 3×3 and 5×5, in this way 'bunching' them. This thought is spurred by Arora et al. in the paper Provable limits for learning some profound portrayals, proposing a layer-by layer development in which one ought to examine the connection measurements of the last layer and bunch them into gatherings of units with high relationship.
2. 2.1×1 convolutions are utilized for dimensionality decrease to expel computational bottlenecks.
3. Because of the actuation work from 1×1 convolution, its option additionally includes nonlinearity. This thought depends on the Network in Network paper. See reference section here.
4. The creators likewise acquainted two helper classifiers with support segregation in the lower phases of the classifier, to build the inclination signal that gets spread back, and to give extra regularization. The helper arranges (the branches that are associated with the assistant classifier) are disposed of at surmising time.

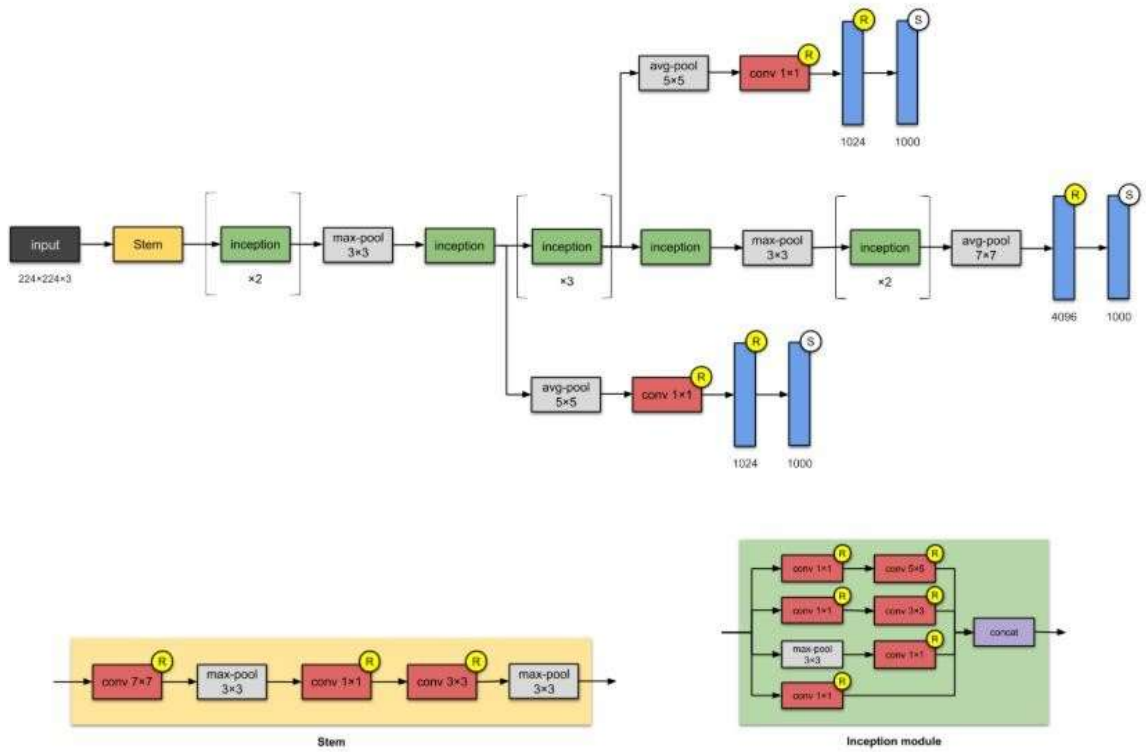


Fig 5.3.4 Inception-v1 architecture. This CNN has two auxiliary networks (which are discarded at inference time). Architecture is based on Figure 3 in the paper.

5.3.5 Inception-v3 (2015)

Initiation v3 is a successor to Inception-v1, with 24M parameters. Hold up where's Inception-v2? Try not to stress over it — it's a previous model of v3 subsequently it's fundamentally the same as v3 however not normally utilized. At the point when the creators turned out with Inception-v2, they ran numerous examinations on it, and recorded some effective changes. Commencement v3 is the system that consolidates these (changes to the optimizer, misfortune work and adding cluster standardization to the assistant layers in the helper organize). The inspiration for Inception-v2 and Inception-v3 is to stay away from illustrative bottlenecks (this implies definitely decreasing the information measurements of the following layer) and have progressively productive calculations by utilizing factorization strategies.

5.3.6 ResNet-50 (2015)

Truly, it's the response to the inquiry you see on the highest point of the article here ("what engineering is this?").

From the previous few CNNs, we have seen only an expanding number of layers in the plan, and accomplishing better execution. Be that as it may, "with the system profundity expanding, precision gets soaked (which may be obvious) and afterward debases quickly." The people from Microsoft Research tended to this issue with ResNet utilizing skip associations (a.k.a. alternate route associations, residuals), while building further models. ResNet is one of the early adopters of clump standardization (the cluster standard paper created by Ioffe and Szegedy was submitted to ICML in 2015). Appeared above is ResNet- 50, with 26M parameters.

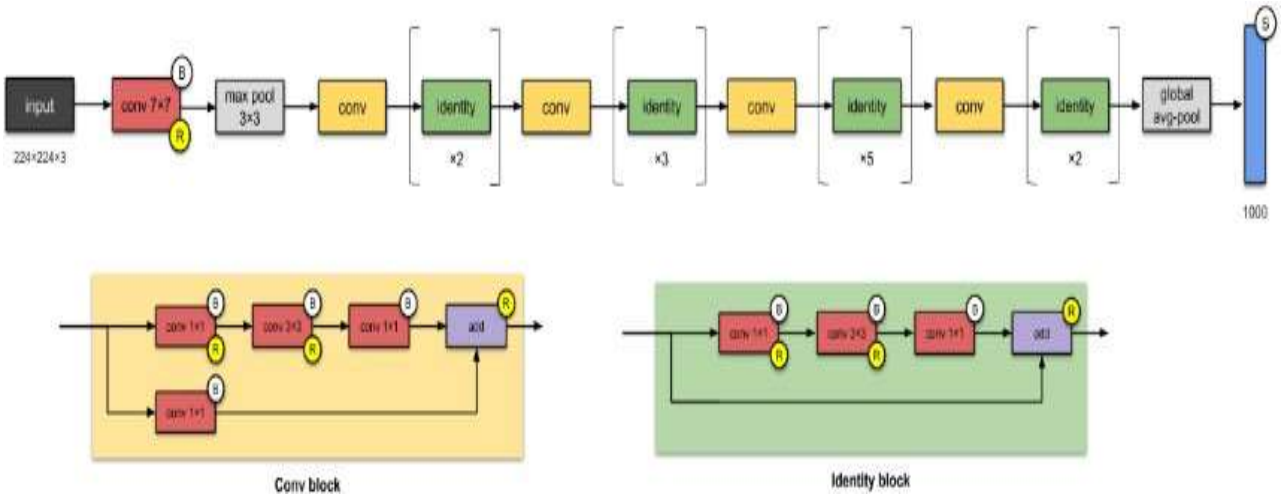


Fig 5.3.6 ResNet-50 architecture, based on the GitHub code from keras-team.

5.3.7 Xception (2016)

Xception is an adaptation from Inception, where the Inception modules have been replaced with depth wise separable convolutions. It also has approximately the same number of parameters as Inception-v1

The Inception hypothesis is taken by the Xception to an extreme (hence the name). What's the Inception hypothesis again? Thank goodness this was explicitly and concisely mentioned in this paper

- Firstly, cross-channel (or cross-feature map) correlations are captured by 1×1 convolution.

- Consequently, spatial correlations within each channel are captured via the regular 3×3 or 5×5 convolutions.

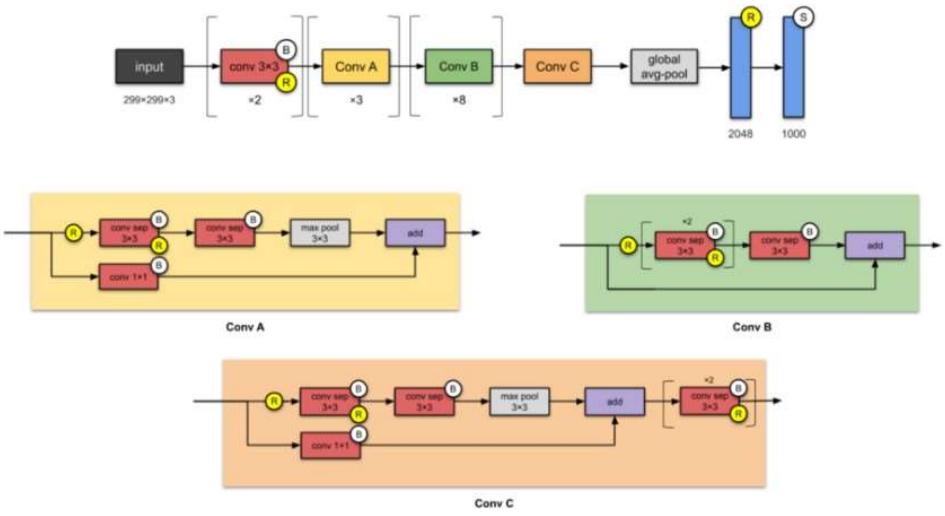


Fig 5.3.7 Inception-v4 architecture. This CNN has an auxiliary network (which is discarded at inference time). *Note: All convolutional layers are followed by batch norm and ReLU activation. Architecture is based on their GitHub code.

5.3.8 Inception-ResNet-V2 (2016)

In a similar paper as Inception-v4, similar creators additionally presented Inception-ResNets — a group of Inception-ResNet-v1 and Inception-ResNet-v2. The last individual from the family has 56M parameters.

5.3.9 Inception-v4 (2016)

The people from Google hit again with Inception-v4, 43M parameters. Once more, this is an improvement from Inception-v3. The primary contrast is the Stem gathering and some minor changes in the Inception-C module. The creators additionally "settled on uniform decisions for the Inception hinders for every network size." They likewise referenced that having "lingering associations prompts significantly improved preparing speed."

All things considered, note that it was referenced that Inception-v4 works better in light of expanded model size.

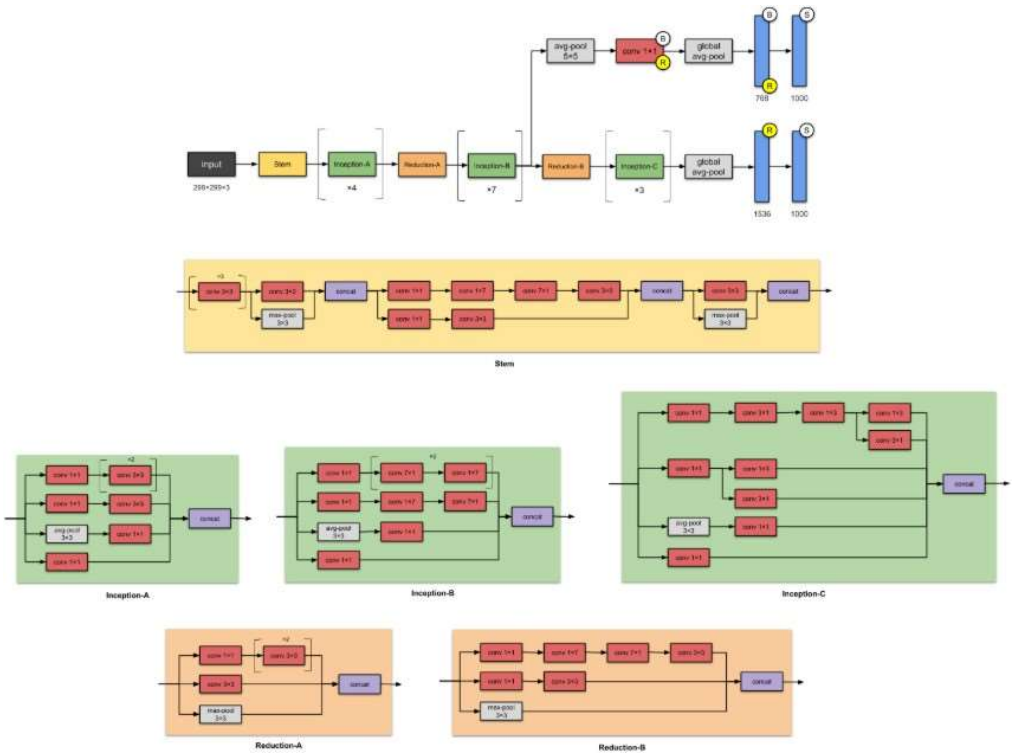


Fig 5.3.8 Inception-v4 architecture. This CNN has an auxiliary network (which is discarded at inference time). *Note: All convolutional layers are followed by batch norm and ReLU activation. Architecture is based on their GitHub

5.3.10 ResNeXt-50 (2017)

In case you're pondering ResNets, truly, they are connected. ResNeXt-50 has 25M parameters (ResNet-50 has 25.5M). What's distinctive about ResNeXts is the including of equal towers/branches/ways inside every module, as observed above showed by 'complete 32 towers.

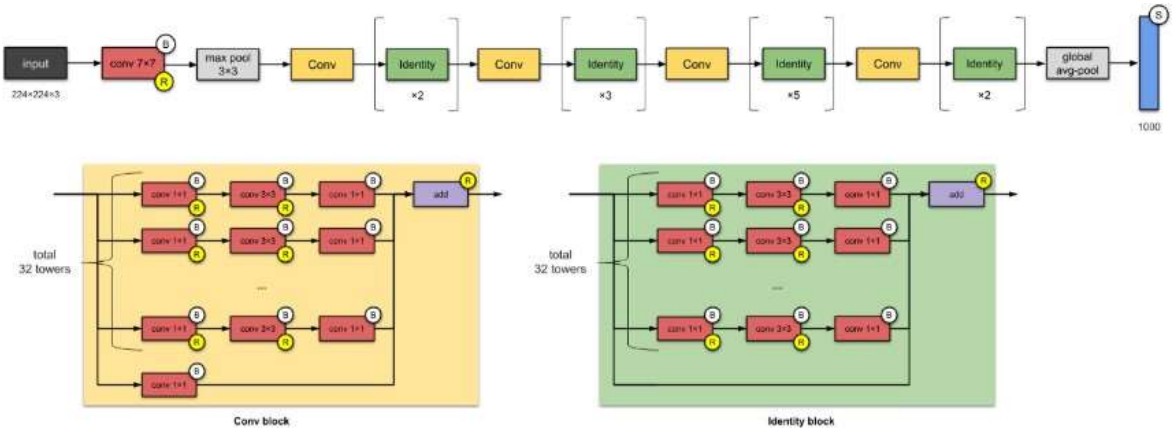


Fig 5.3.10 ResNeXt architecture, based on their paper.

5.4 CODE IMPLEMENTATION

Source Code:

```
!pip install imutils
import numpy as np
import pandas as pd
import os
from os import listdir
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import sys
pip install opencv-python
import cv2
import imutils
import itertools
from sklearn.metrics import accuracy_score,confusion_matrix
from tensorflow.keras.models import Model,load_model
from tensorflow.keras.layers import
Conv2D,Input,ZeroPadding2D,BatchNormalization,Flatten,Activation,Dense,MaxPooling2D
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle

images="C:\\Users\\Lenovo\\Desktop\\Project\\Data"
os.makedirs('C:\\Users\\Lenovo\\Desktop\\Project\\Output')
os.makedirs('C:\\Users\\Lenovo\\Desktop\\Project\\Output\\Yes')
os.makedirs('C:\\Users\\Lenovo\\Desktop\\Project\\Output\\No')


def augment_data(file_dir,n_generated_samples,save_to_dir):

data_gen=ImageDataGenerator(rotation_range=10,width_shift_range=0.1,height_shift_range=
```

```
0.1,shear_range=0.1,

brightness_range=(0.3,1.0),horizontal_flip=True,vertical_flip=True,fill_mode='nearest')
    for filename in listdir(file_dir):
        image=cv2.imread(file_dir+'/'+filename)
        image=image.reshape((1,)+image.shape)
        save_prefix='aug_'+filename[:-4]
        i=0
        for batch in
data_gen.flow(x=image,batch_size=1,save_to_dir=save_to_dir,save_prefix=save_prefix,save_
format='jpg'):
                i+=1
                if i>n_generated_samples:
                    break

augmented_data_path='C:\\Users\\Lenovo\\Desktop\\Project\\Output'
augment_data(file_dir=images+'yes',n_generated_samples=6,save_to_dir=augmented_data_pa
th+'yes')
augment_data(file_dir=images+'no',n_generated_samples=9,save_to_dir=augmented_data_pat
h+'no')

def crop_brain_contour(image,plot=False):
    gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    gray=cv2.GaussianBlur(gray,(5,5),0)
    thresh=cv2.threshold(gray,45,255,cv2.THRESH_BINARY)[1]
    thresh=cv2.erode(thresh,None,iterations=2)
    thresh=cv2.dilate(thresh,None,iterations=2)

cnts=cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPL
E)
    cnts=imutils.grab_contours(cnts)
    c=max(cnts,key=cv2.contourArea)
    extLeft=tuple(c[c[:, :, 0].argmin()][0])
    extRight=tuple(c[c[:, :, 0].argmax()][0])
    extTop=tuple(c[c[:, :, 1].argmin()][0])
```

```
    extBot=tuple(c[c[:, :, 1].argmax()][0])
    new_image=image[extTop[1]:extBot[1], extLeft[0]:extRight[0]]
    if plot:
        plt.figure()
        plt.subplot(1, 2, 1)
        plt.imshow(image)

plt.tick_params(axis='both',which='both',top=False,bottom=False,left=False,right=False,labelb
ottom=False,labeltop=False,
                    labelleft=False,labelright=False)
        plt.title('Original Image')
        plt.subplot(1, 2, 2)
        plt.imshow(new_image)

plt.tick_params(axis='both',which='both',top=False,bottom=False,left=False,right=False,labelb
ottom=False,labeltop=False,
                    labelleft=False,labelright=False)
        plt.title('Cropped Image')
        plt.show()
    return new_image

ex_img=cv2.imread('C:\\Users\\Lenovo\\Desktop\\Project\\Data\\yes\\Y6.jpg')
ex_crop_img=crop_brain_contour(ex_img,True)

def load_data(dir_list,image_size):
    X=[]
    Y=[]
    image_width,image_height=image_size
    for directory in dir_list:
        for filename in listdir(directory):
            image=cv2.imread(directory+'/'+filename)
            image=crop_brain_contour(image,plot=False)
            image=cv2.resize(image,
dsize=(image_width,image_height),interpolation=cv2.INTER_CUBIC)
            image=image/255.
```

```
            X.append(image)
            if directory[-3:]=='yes':
                Y.append([1])
            else:
                Y.append([0])
    X=np.array(X)
    Y=np.array(Y)
    X,Y=shuffle(X,Y)
    print(f'Number of examples is: {len(X)}')
    print(f'X shape is: {X.shape}')
    print(f'Y shape is: {Y.shape}')
    return X,Y

augmented_yes=augmented_data_path+'yes'
augmented_no=augmented_data_path+'no'
IMG_WIDTH,IMG_HEIGHT=(240,240)
X,Y=load_data([augmented_yes,augmented_no],(IMG_WIDTH,IMG_HEIGHT))

def plot_sample_images(X,Y,n=50):
    for label in [0,1]:
        images=X[np.argwhere(Y==label)]
        n_images=images[:n]
        columns_n=10
        rows_n=int(n/columns_n)
        plt.figure(figsize=(10,8))
        i=1
        for image in n_images:
            plt.subplot(rows_n,columns_n,i)
            plt.imshow(image[0])
            plt.tick_params(axis='both',which='both',top=False,
bottom=False,left=False,right=False,labelbottom=False,
                            labeltop=False,labelleft=False,labelright=False)
            i += 1
        label_to_str=lambda label: "Yes" if label==1 else "No"
        plt.suptitle(f"Brain Tumor: {label_to_str(label)}")
```

```
        plt.show()

plot_sample_images(X,Y)

def split_data(X,Y,test_size=0.2):
    X_train,X_test_val,Y_train,Y_test_val=train_test_split(X,Y,test_size=test_size)
    X_test,X_val,Y_test,Y_val=train_test_split(X_test_val,Y_test_val,test_size=0.5)
    return X_train,Y_train,X_val,Y_val,X_test,Y_test

X_train,Y_train,X_val,Y_val,X_test,Y_test=split_data(X,Y,test_size=0.3)

print("number of training examples = "+str(X_train.shape[0]))
print("number of validation examples = "+str(X_val.shape[0]))
print("number of test examples = "+str(X_test.shape[0]))

def build_model(input_shape):
    X_input=Input(input_shape)
    X=ZeroPadding2D((2, 2))(X_input)
    X=Conv2D(32,(7,7),strides=(1,1))(X)
    X=BatchNormalization(axis=3,name='bn0')(X)
    X=Activation('relu')(X)
    X=MaxPooling2D((4,4))(X)
    X=MaxPooling2D((4,4))(X)
    X=Flatten()(X)
    X=Dense(1,activation='sigmoid')(X)
    model=Model(inputs=X_input,outputs=X)
    return model

IMG_SHAPE=(IMG_WIDTH,IMG_HEIGHT,3)
model=build_model(IMG_SHAPE)
model.summary()

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
model.fit(x=X_train,y=Y_train,batch_size=32,epochs=22,validation_data=(X_val,Y_val))
```

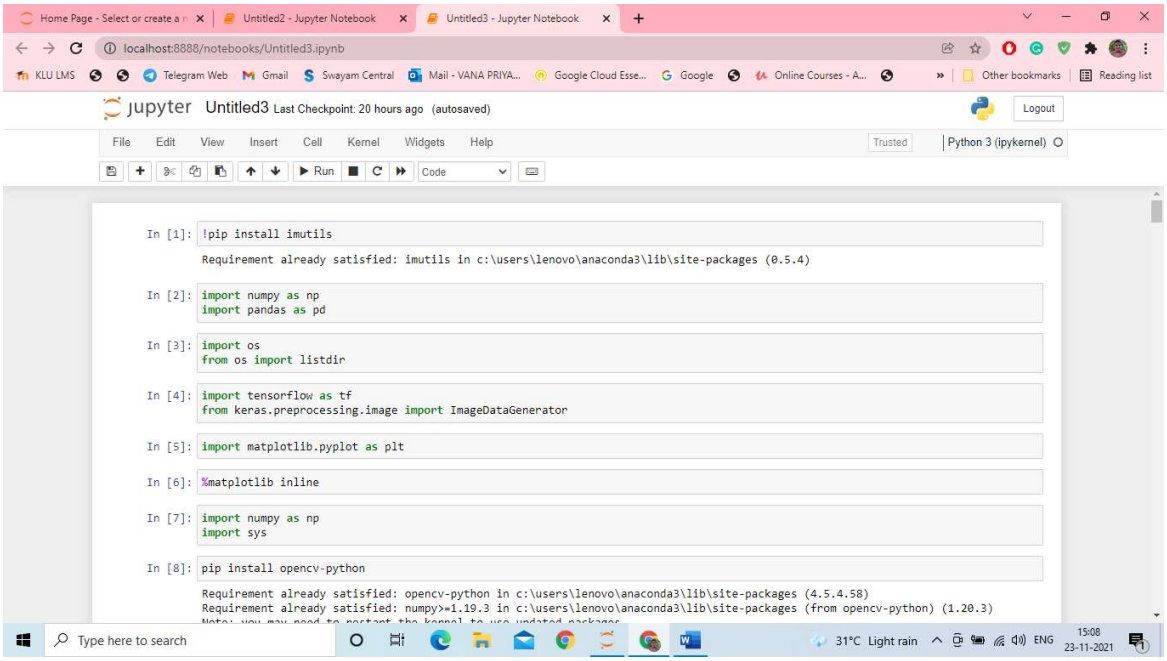
```
history = model.history.history

def plot_metrics(history):
    train_loss=history['loss']
    val_loss=history['val_loss']
    train_acc=history['accuracy']
    val_acc=history['val_accuracy']

    # Loss Graph
    plt.figure()
    plt.plot(train_loss,label='Training Loss')
    plt.plot(val_loss,label='Validation Loss')
    plt.title('Loss')
    plt.legend()
    plt.show()

    # Accuracy Graph
    plt.figure()
    plt.plot(train_acc,label='Training Accuracy')
    plt.plot(val_acc,label='Validation Accuracy')
    plt.title('Accuracy')
    plt.legend()
    plt.show()


plot_metrics(history)

def plot_confusion_matrix(cm,classes,normalize=False,title='Confusion
matrix',cmap=plt.cm.Blues):
    plt.figure(figsize=(6,6))
    plt.imshow(cm,interpolation='nearest',cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks=np.arange(len(classes))
    plt.xticks(tick_marks,classes,rotation=90)
```

```
    plt.yticks(tick_marks,classes)
    if normalize:
        cm=cm.astype('float')/cm.sum(axis=1)[:,np.newaxis]
    thresh=cm.max()/2.
    cm=np.round(cm,2)
    for i,j in itertools.product(range(cm.shape[0]),range(cm.shape[1])):
        plt.text(j,i,cm[i,j],horizontalalignment="center",color="white" if cm[i,j]>thresh else
"black")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()

labels=['yes','no']
predictions=model.predict(X_val)
predictions=[1 if x>0.5 else 0 for x in predictions]
accuracy=accuracy_score(Y_val,predictions)
print('Val Accuracy = %.2f' % accuracy)
confusion_mtx=confusion_matrix(Y_val,predictions)
cm=plot_confusion_matrix(confusion_mtx,classes=labels,normalize=False)

predictions=model.predict(X_test)
predictions=[1 if x>0.5 else 0 for x in predictions]
accuracy=accuracy_score(Y_test,predictions)
print('Val Accuracy = %.2f' % accuracy)
confusion_mtx=confusion_matrix(Y_test,predictions)
cm=plot_confusion_matrix(confusion_mtx,classes=labels,normalize=False)

for i in range(10):
    plt.figure()
    plt.imshow(X_test[i])
    plt.xticks([])
    plt.yticks([])
    plt.title(f'Actual class: {Y_test[i]}\nPredicted class: {predictions[i]}')
    plt.show()
```

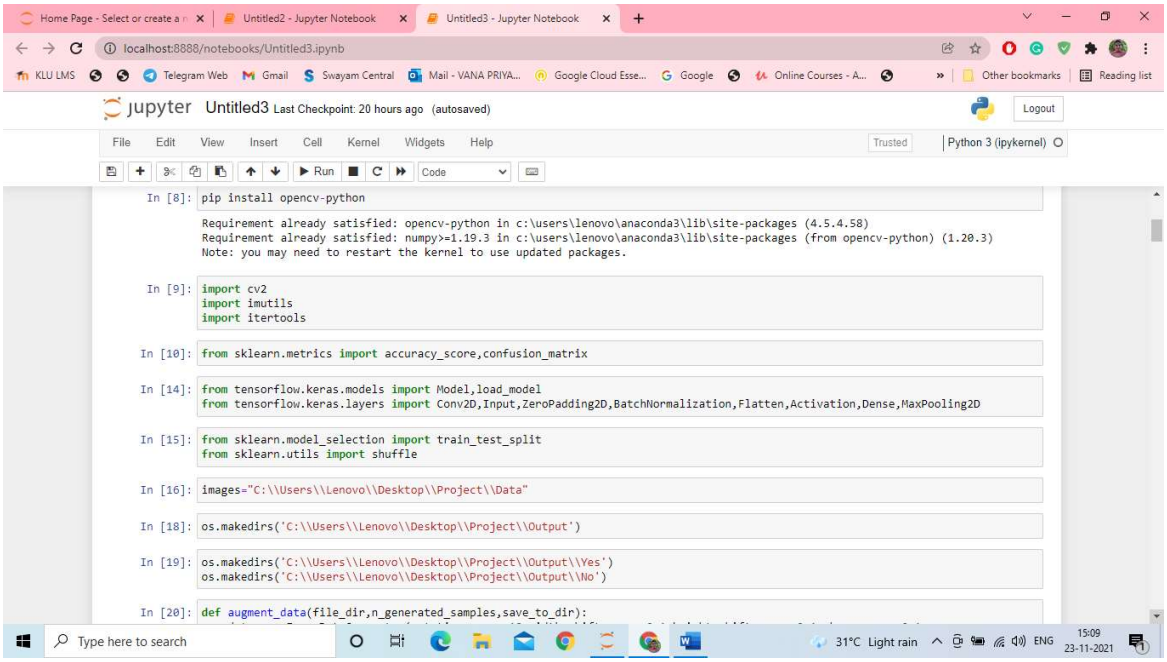
CHAPTER-6

DISCUSSION OF RESULTS

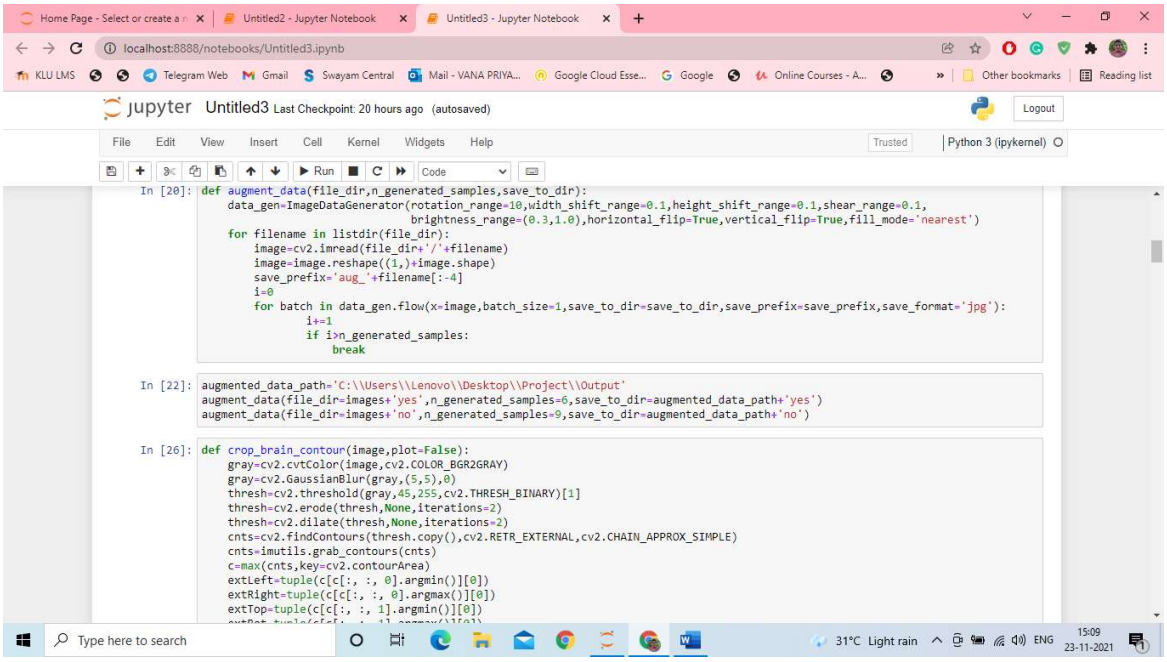
Code Screenshot-1



Code Screenshot-2

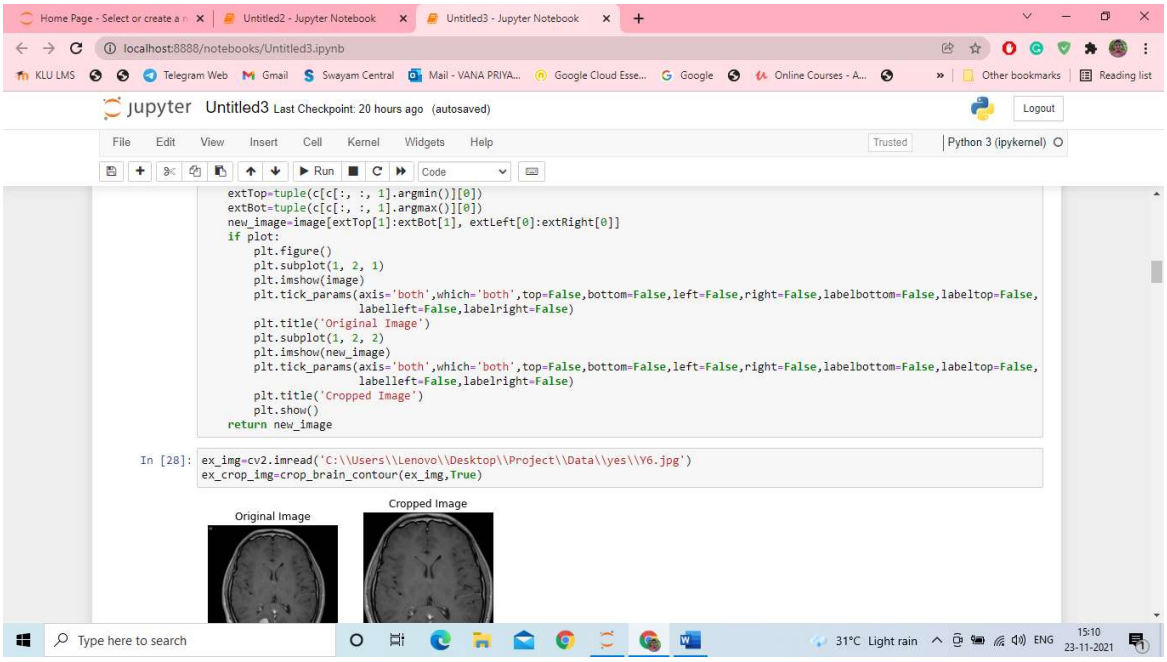


Code Screenshot-3



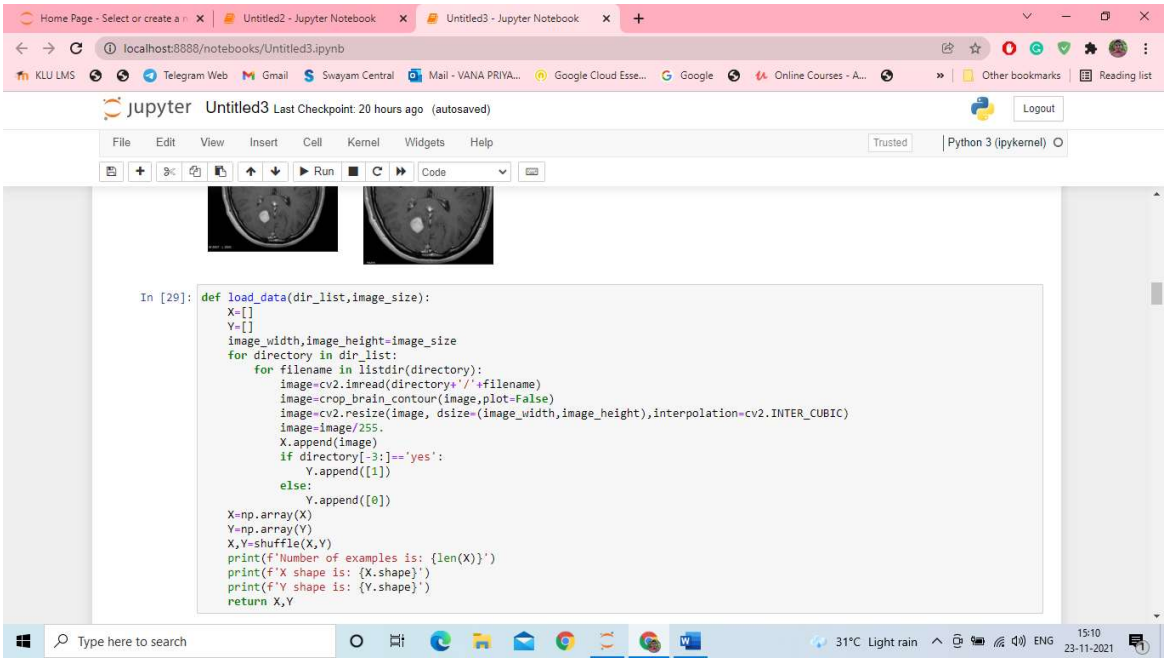
```
In [20]: def augment_data(file_dir,n_generated_samples,save_to_dir):
             data_gen=ImageDataGenerator(rotation_range=10,width_shift_range=0.1,height_shift_range=0.1,shear_range=0.1,
                                         brightness_range=(0.3,1.0),horizontal_flip=True,vertical_flip=True,fill_mode='nearest')
             for filename in listdir(file_dir):
                 image=cv2.imread(file_dir+'/'+filename)
                 image=image.reshape((1,)+image.shape)
                 save_prefix='aug_'+filename[:-4]
                 i=0
                 for batch in data_gen.flow(x=image,batch_size=1,save_to_dir=save_to_dir,save_prefix=save_prefix,save_format='jpg'):
                         i+=1
                         if i>n_generated_samples:
                             break

In [22]: augmented_data_path='C:\\Users\\Lenovo\\Desktop\\Project\\Output'
         augment_data(file_dir=images+'yes',n_generated_samples=6,save_to_dir=augmented_data_path+'yes')
         augment_data(file_dir=images+'no',n_generated_samples=9,save_to_dir=augmented_data_path+'no')

In [26]: def crop_brain_contour(image,plot=False):
             gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
             gray=cv2.GaussianBlur(gray,(5,5),0)
             thresh=cv2.threshold(gray,45,255,cv2.THRESH_BINARY)[1]
             thresh=cv2.erode(thresh,None,iterations=2)
             thresh=cv2.dilate(thresh,None,iterations=2)
             cnts=cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
             cnts=imutils.grab_contours(cnts)
             c=max(cnts,key=cv2.contourArea)
             extLeft=tuple(c[c[:, :, 0].argmin()][0])
             extRight=tuple(c[c[:, :, 0].argmax()][0])
             extTop=tuple(c[c[:, :, 1].argmin()][0])
```

Code Screenshot-4



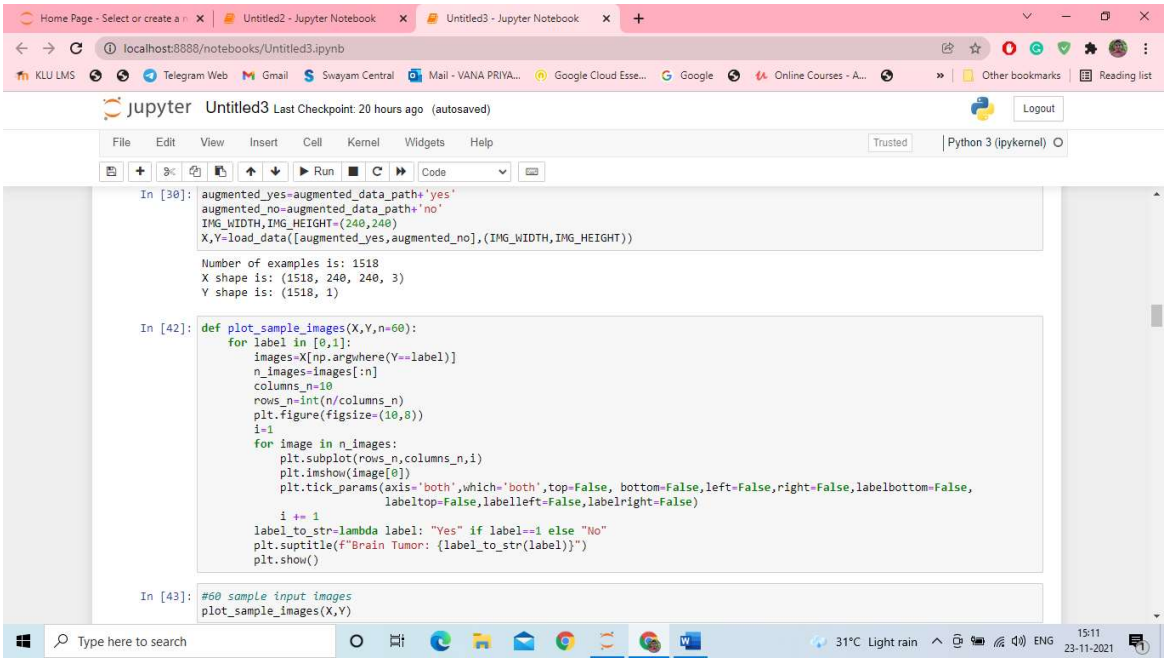
```
             extTop=tuple(c[c[:, :, 1].argmin()][0])
             extBot=tuple(c[c[:, :, 1].argmax()][0])
             new_image=image[extTop[1]:extBot[1], extLeft[0]:extRight[0]]
             if plot:
                 plt.figure()
                 plt.subplot(1, 2, 1)
                 plt.imshow(image)
                 plt.tick_params(axis='both',which='both',top=False,bottom=False,left=False,right=False,labelbottom=False,labeltop=False,
                                 labelleft=False,labelright=False)
                 plt.title('Original Image')
                 plt.subplot(1, 2, 2)
                 plt.imshow(new_image)
                 plt.tick_params(axis='both',which='both',top=False,bottom=False,left=False,right=False,labelbottom=False,labeltop=False,
                                 labelleft=False,labelright=False)
                 plt.title('Cropped Image')
                 plt.show()
             return new_image

In [28]: ex_img=cv2.imread('C:\\Users\\Lenovo\\Desktop\\Project\\Data\\yes\\Y6.jpg')
         ex_crop_img=crop_brain_contour(ex_img,True)
```

Code Screenshot-5



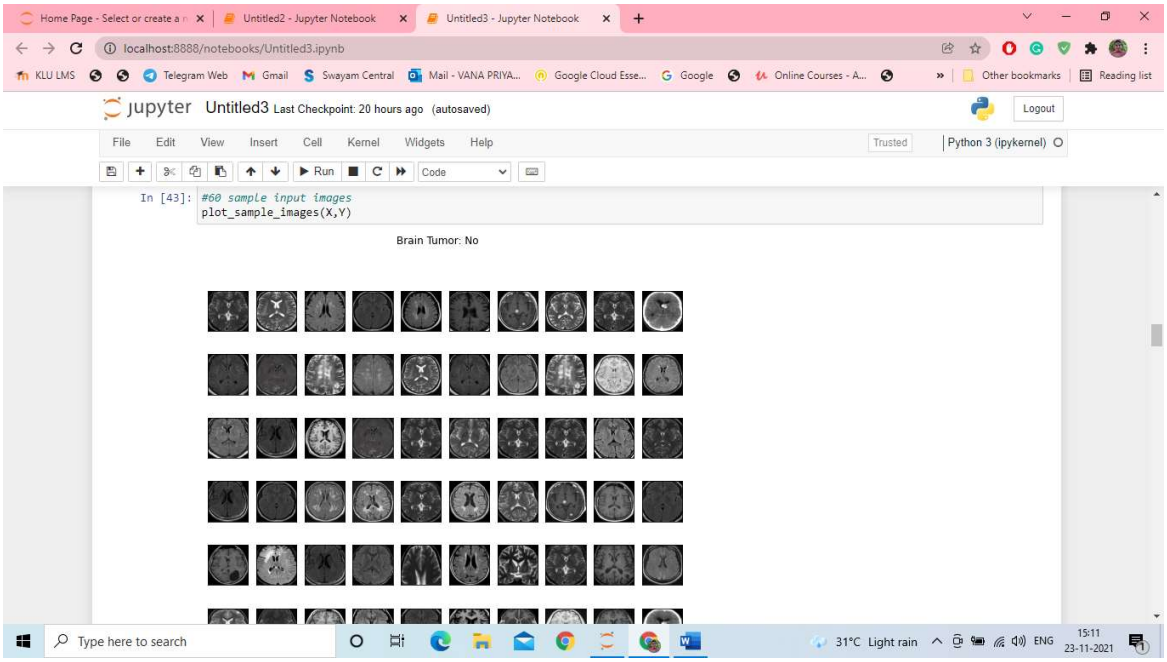
```
In [29]: def load_data(dir_list,image_size):
             X=[]
             Y=[]
             image_width,image_height=image_size
             for directory in dir_list:
                 for filename in listdir(directory):
                     image=cv2.imread(directory+'/'+filename)
                     image=crop_brain_contour(image,plot=False)
                     image=cv2.resize(image, dsize=(image_width,image_height),interpolation=cv2.INTER_CUBIC)
                     image=image/255.
                     X.append(image)
                     if directory[-3:]=='yes':
                         Y.append([1])
                     else:
                         Y.append([0])
             X=np.array(X)
             Y=np.array(Y)
             X,Y=shuffle(X,Y)
             print(f'Number of examples is: {len(X)}')
             print(f'X shape is: {X.shape}')
             print(f'Y shape is: {Y.shape}')
             return X,Y
```

Code Screenshot-6

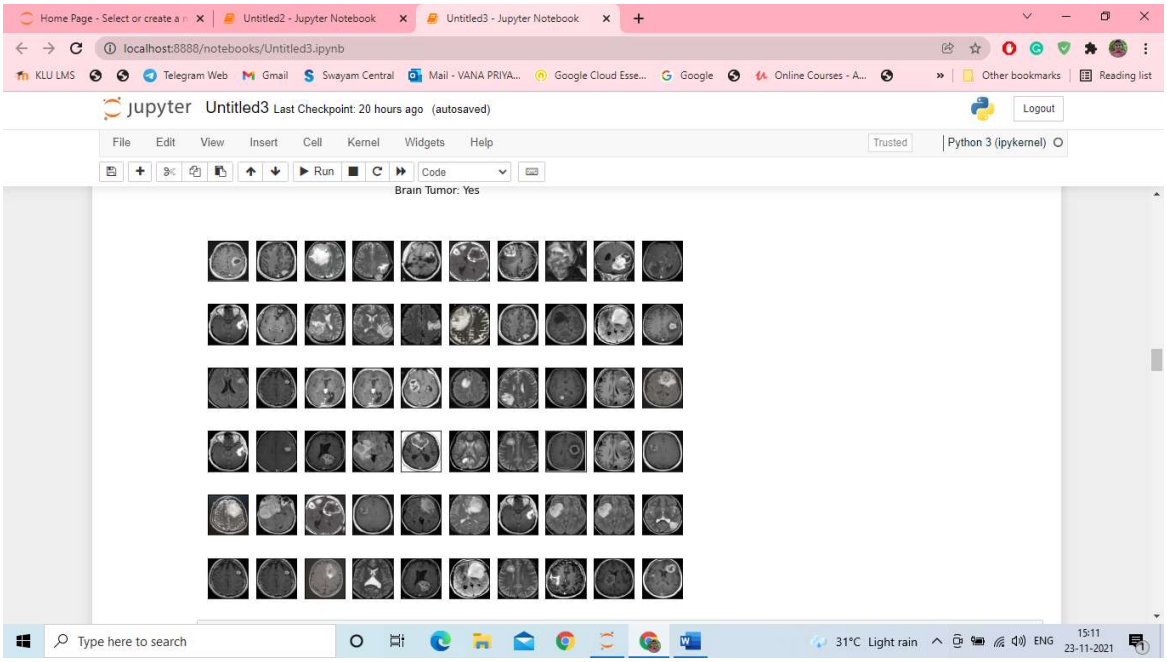


```
In [30]: augmented_yes=augmented_data_path+'yes'
         augmented_no=augmented_data_path+'no'
         IMG_WIDTH,IMG_HEIGHT=(240,240)
         X,Y=load_data([augmented_yes,augmented_no],(IMG_WIDTH,IMG_HEIGHT))

         Number of examples is: 1518
         X shape is: (1518, 240, 240, 3)
         Y shape is: (1518, 1)

In [42]: def plot_sample_images(X,Y,n=60):
             for label in [0,1]:
                 images=X[np.argwhere(Y==label)]
                 n_images=images[:n]
                 columns_n=10
                 rows_n=int(n/columns_n)
                 plt.figure(figsize=(10,8))
                 i=1
                 for image in n_images:
                     plt.subplot(rows_n,columns_n,i)
                     plt.imshow(image[0])
                     plt.tick_params(axis='both',which='both',top=False, bottom=False,left=False,right=False,labelbottom=False,
                                     labeltop=False,labelleft=False,labelright=False)
                     i += 1
                 label_to_str=lambda label: "Yes" if label==1 else "No"
                 plt.suptitle(f"Brain Tumor: {label_to_str(label)}")
                 plt.show()

In [43]: #60 sample input images
         plot_sample_images(X,Y)
```

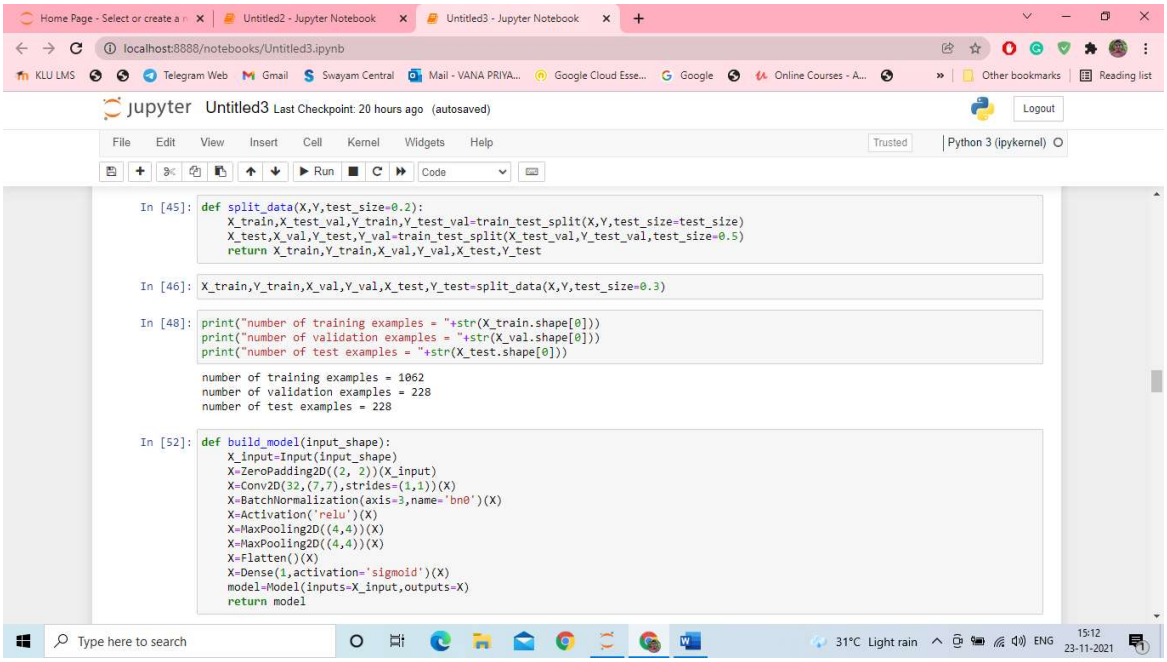
Code Screenshot-7



Code Screenshot-8

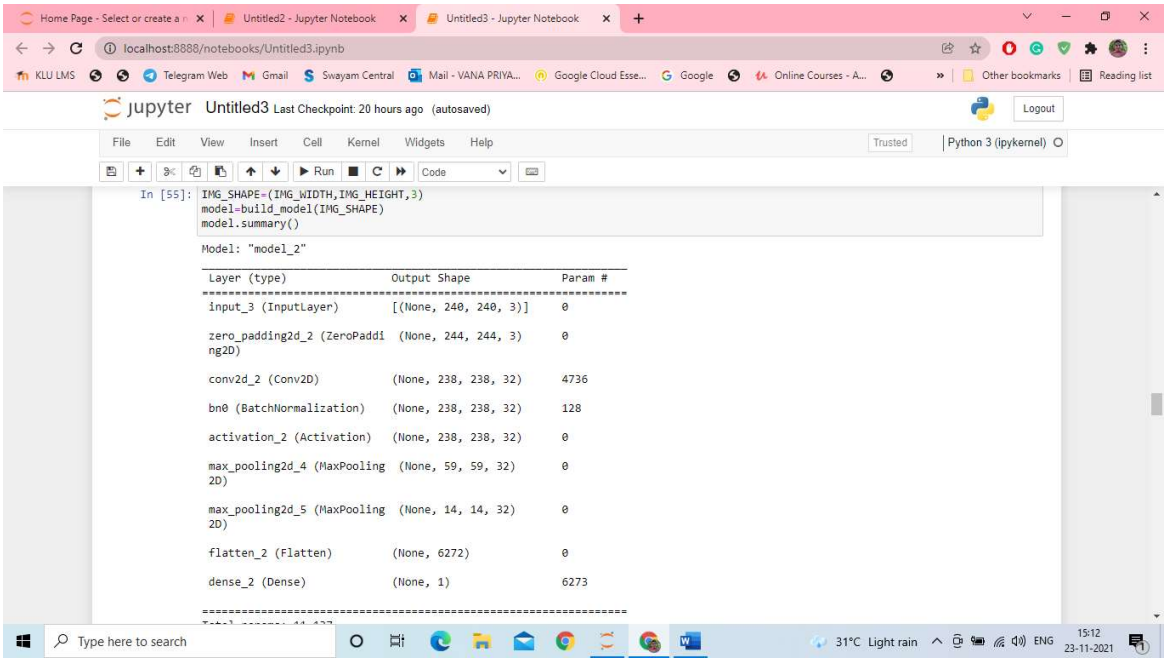


Code Screenshot-9



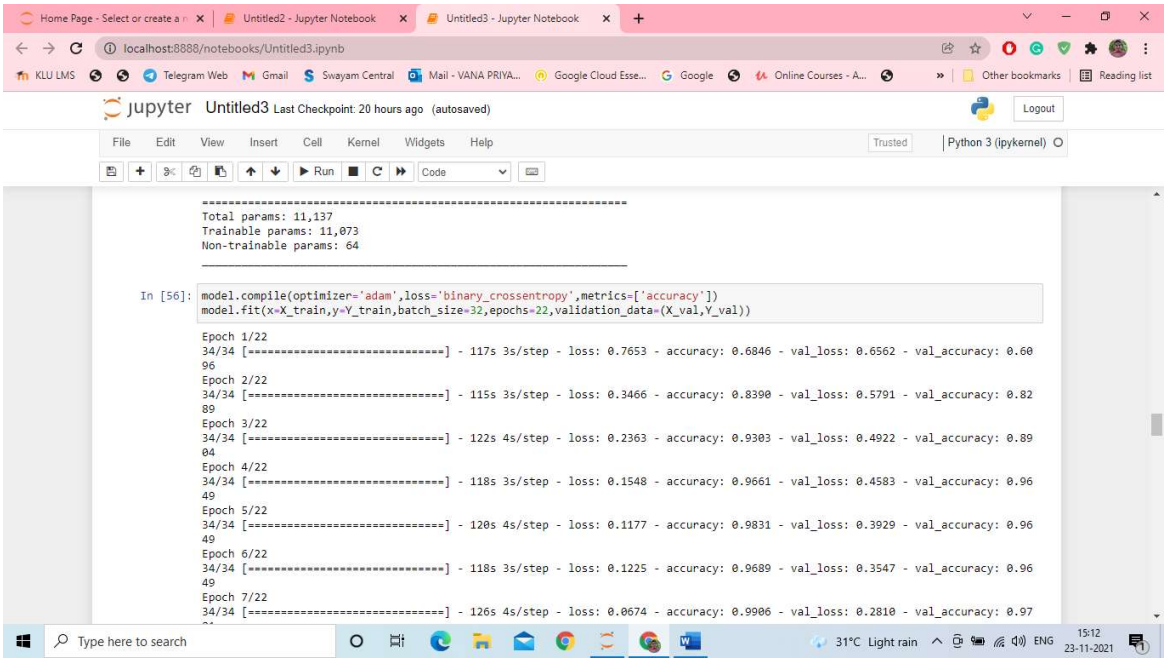
```
In [45]: def split_data(X,Y,test_size=0.2):
             X_train,X_test_val,Y_train,Y_test_val=train_test_split(X,Y,test_size=test_size)
             X_test,X_val,Y_test,Y_val=train_test_split(X_test_val,Y_test_val,test_size=0.5)
             return X_train,Y_train,X_val,Y_val,X_test,Y_test

In [46]: X_train,Y_train,X_val,Y_val,X_test,Y_test=split_data(X,Y,test_size=0.3)

In [48]: print("number of training examples = "+str(X_train.shape[0]))
         print("number of validation examples = "+str(X_val.shape[0]))
         print("number of test examples = "+str(X_test.shape[0]))

         number of training examples = 1062
         number of validation examples = 228
         number of test examples = 228

In [52]: def build_model(input_shape):
             X_input=Input(input_shape)
             X=ZeroPadding2D((2, 2))(X_input)
             X=Conv2D(32,(7,7),strides=(1,1))(X)
             X=BatchNormalization(axis=3,name='bn0')(X)
             X=Activation('relu')(X)
             X=MaxPooling2D((4,4))(X)
             X=MaxPooling2D((4,4))(X)
             X=Flatten()(X)
             X=Dense(1,activation='sigmoid')(X)
             model=Model(inputs=X_input,outputs=X)
             return model
```

Code Screenshot-10

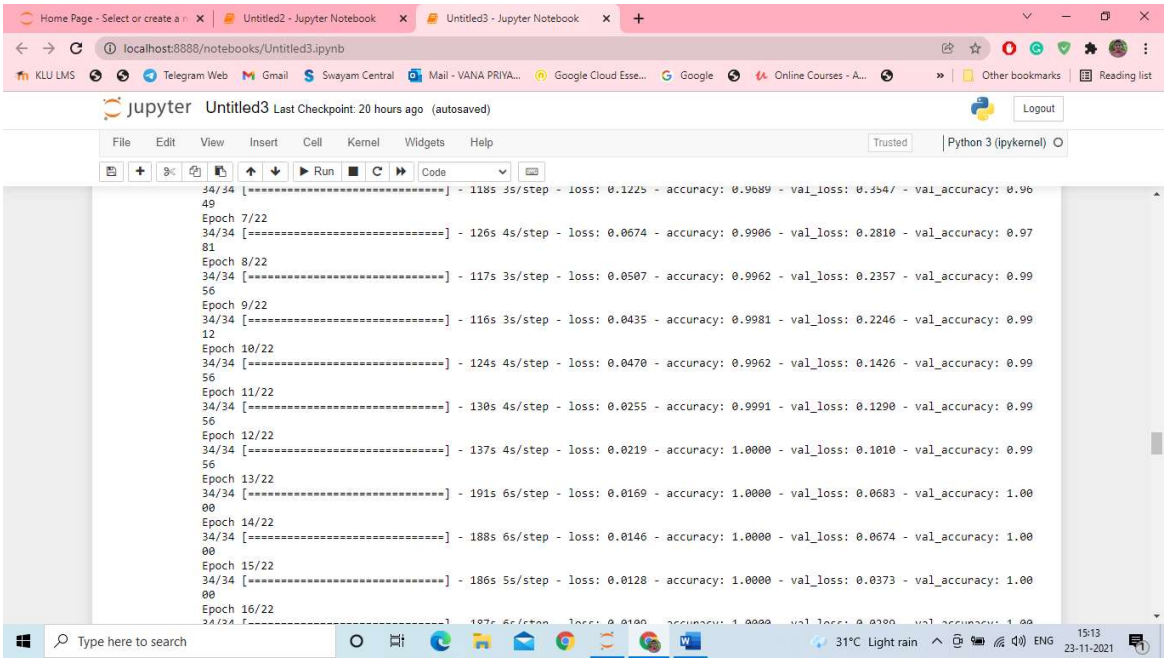


```
In [55]: IMG_SHAPE=(IMG_WIDTH,IMG_HEIGHT,3)
         model=build_model(IMG_SHAPE)
         model.summary()

         Model: "model_2"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         input_3 (InputLayer)         [(None, 240, 240, 3)]     0

         zero_padding2d_2 (ZeroPaddi  (None, 244, 244, 3)       0
         ng2D)

         conv2d_2 (Conv2D)            (None, 238, 238, 32)      4736

         bn0 (BatchNormalization)     (None, 238, 238, 32)      128

         activation_2 (Activation)    (None, 238, 238, 32)      0

         max_pooling2d_4 (MaxPooling  (None, 59, 59, 32)        0
         2D)

         max_pooling2d_5 (MaxPooling  (None, 14, 14, 32)        0
         2D)

         flatten_2 (Flatten)          (None, 6272)              0

         dense_2 (Dense)              (None, 1)                 6273

         =================================================================
```

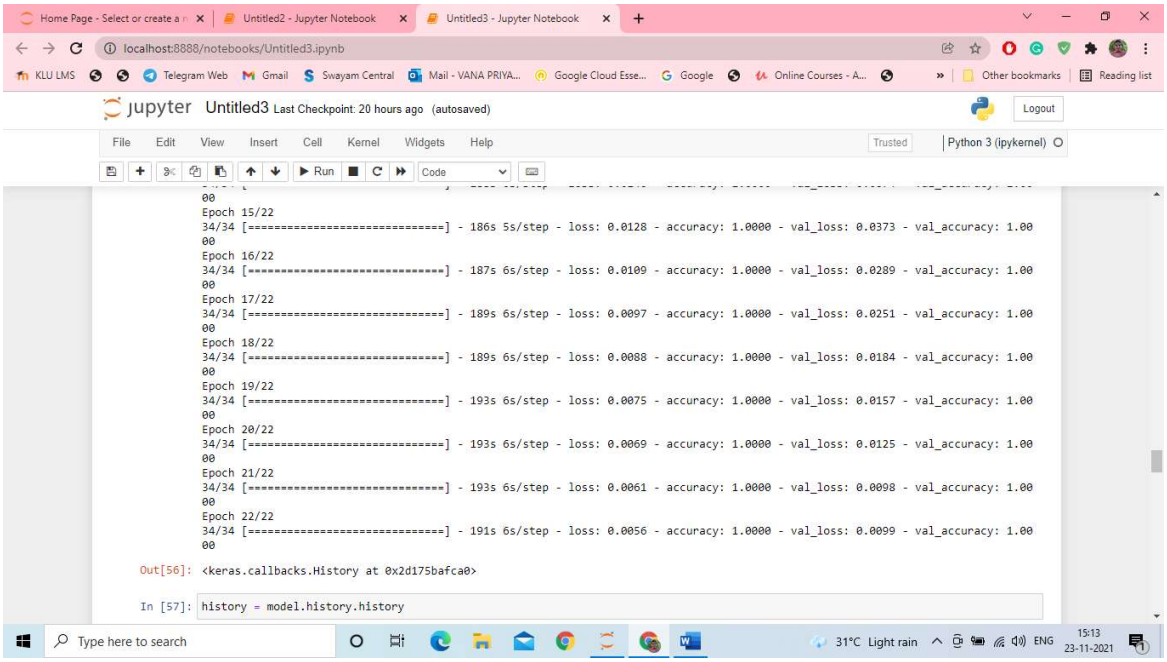
Code Screenshot-11



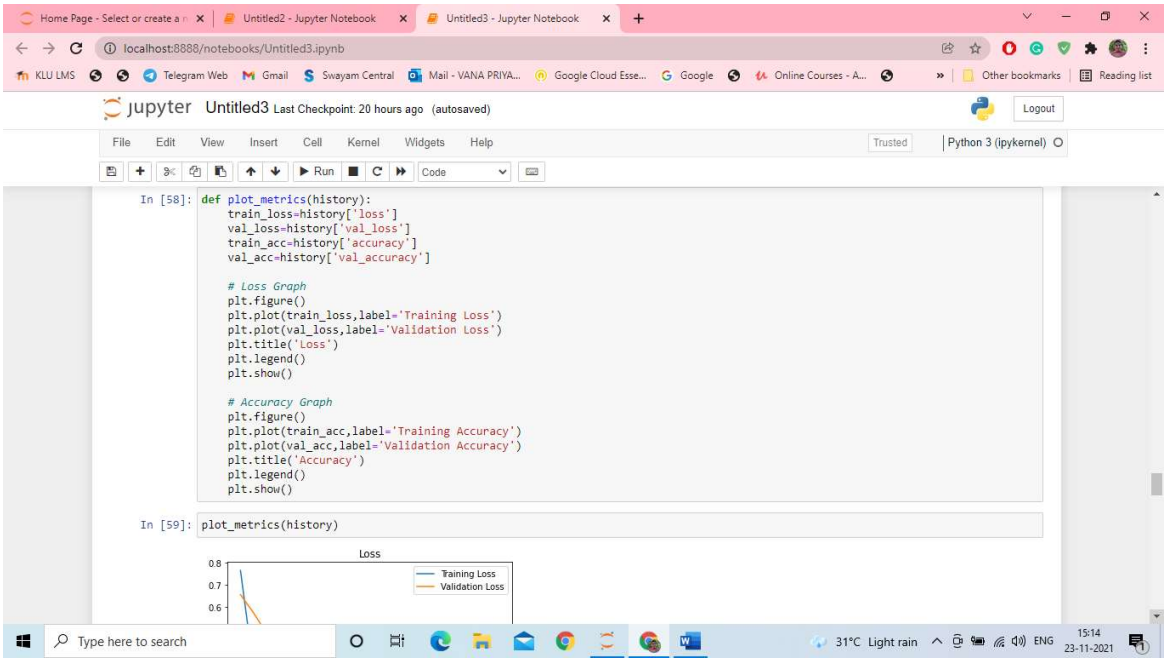
Code Screenshot-12



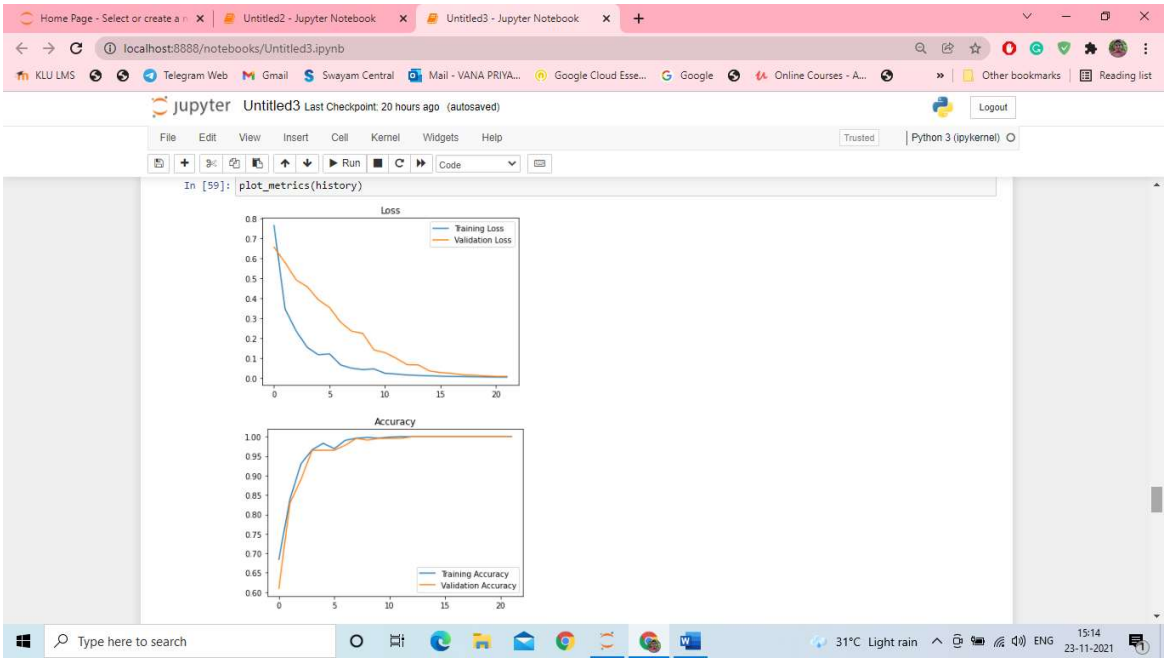
Code Screenshot-13



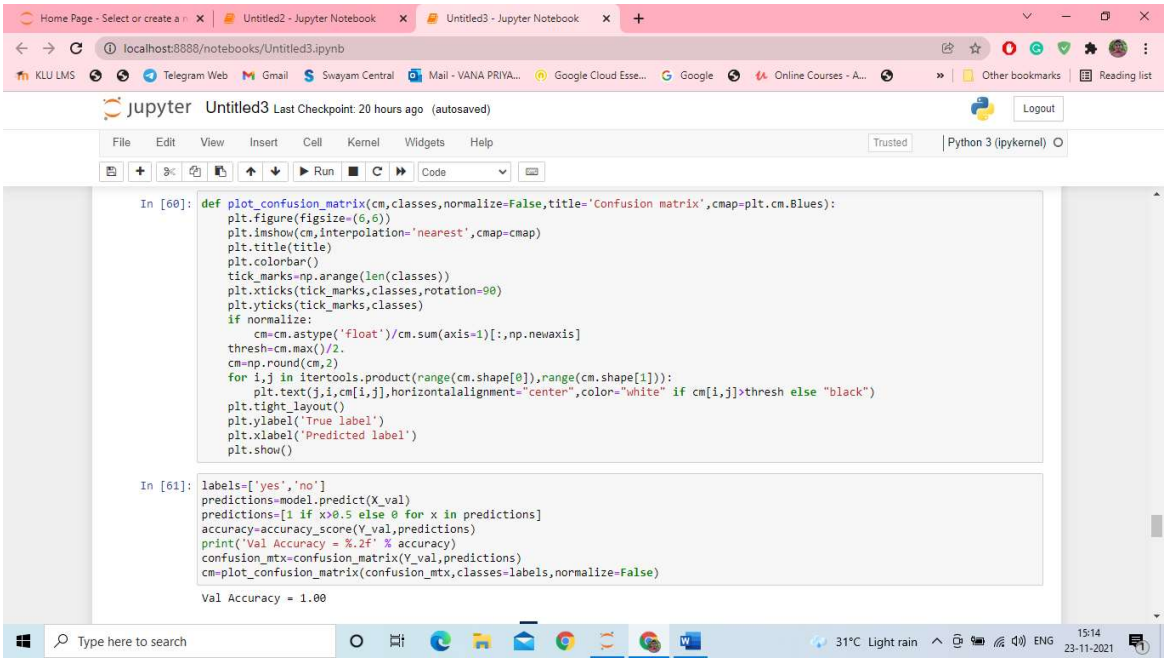
Code Screenshot-14



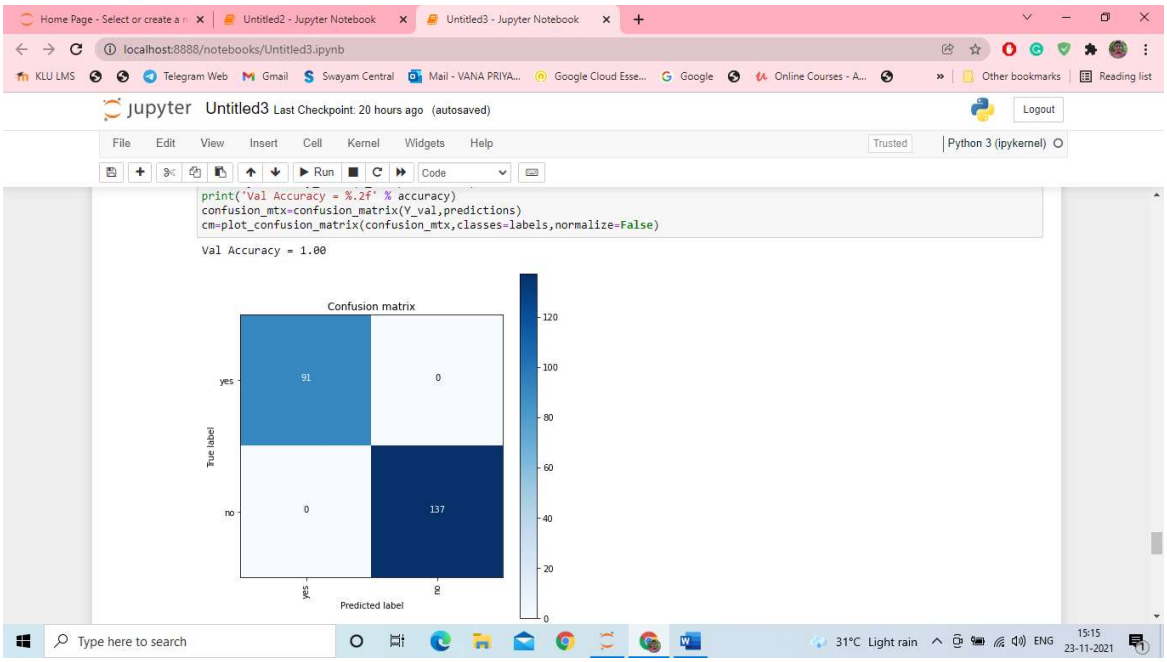
Code Screenshot-15



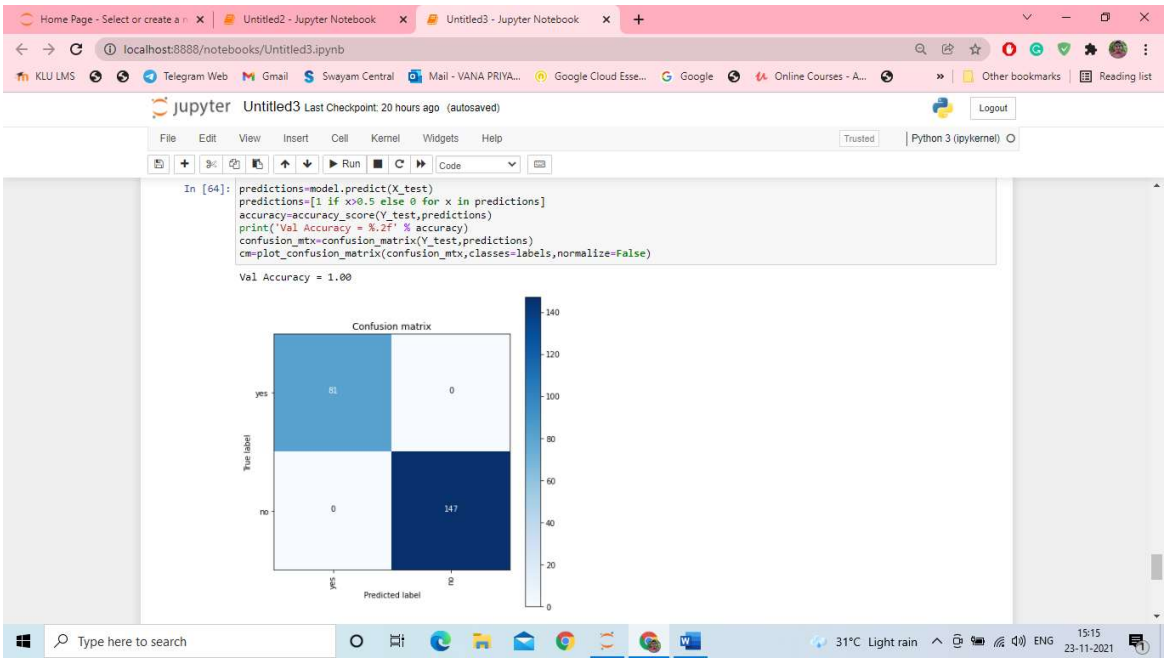
Code Screenshot-16



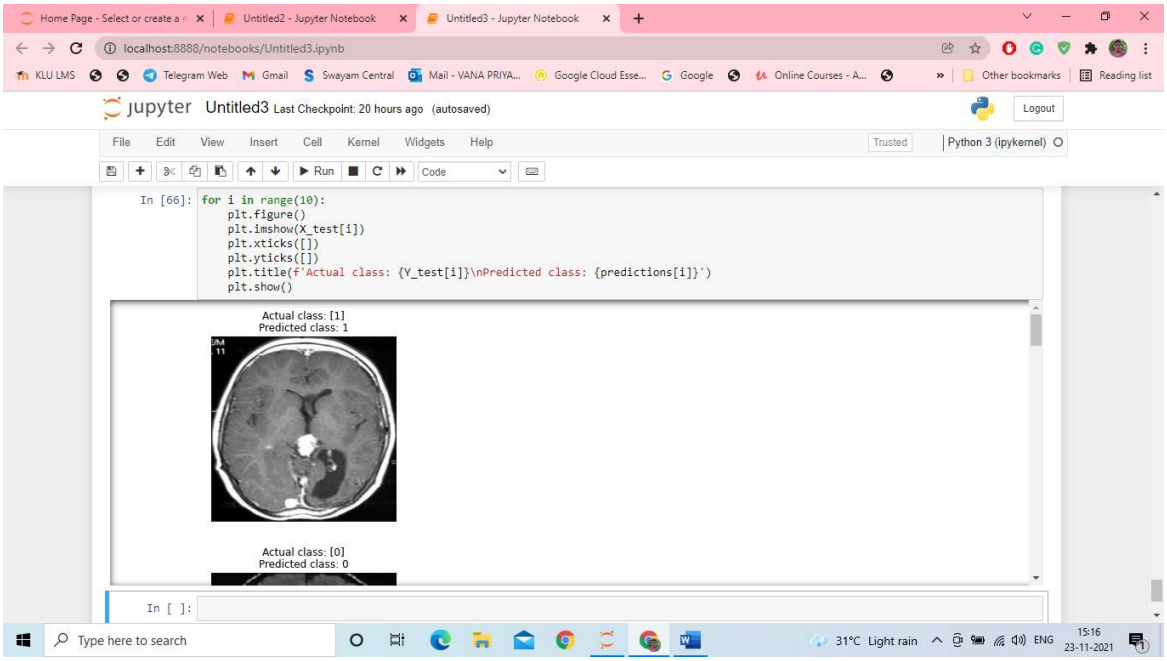
Code Screenshot-17



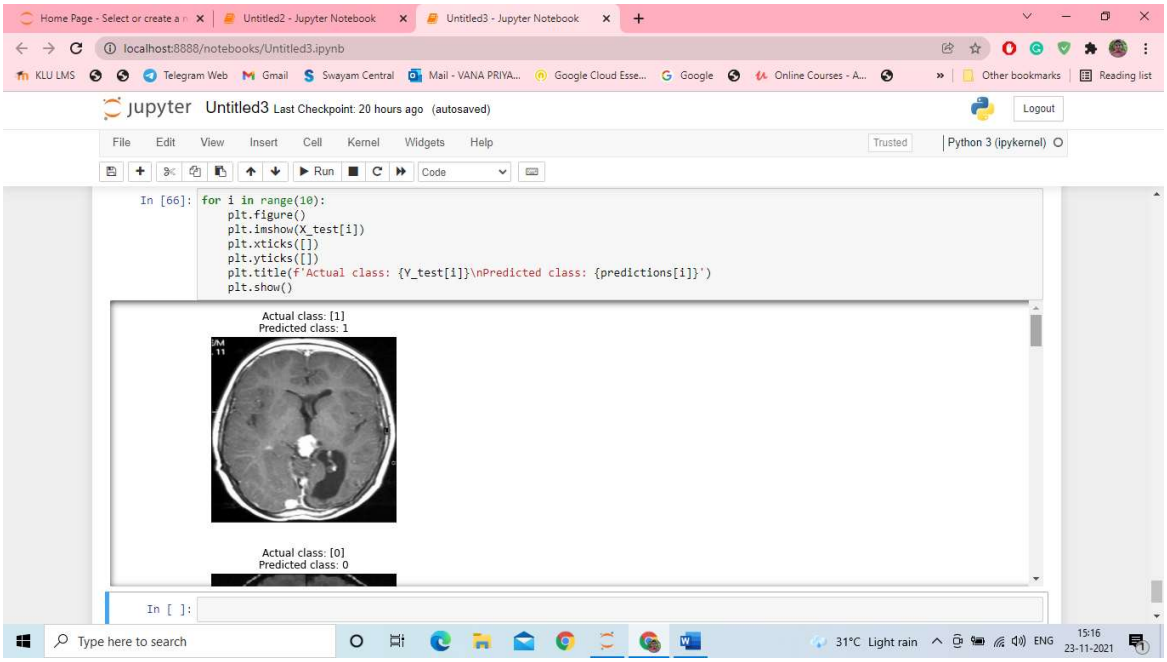
Code Screenshot-18



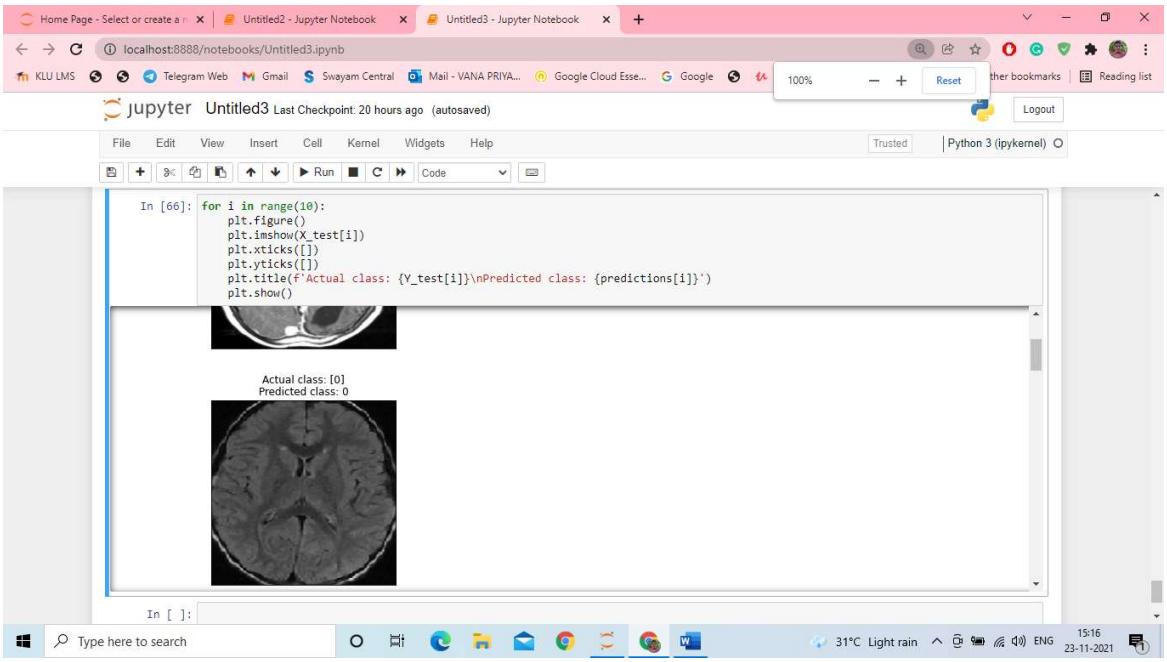
Code Screenshot-19



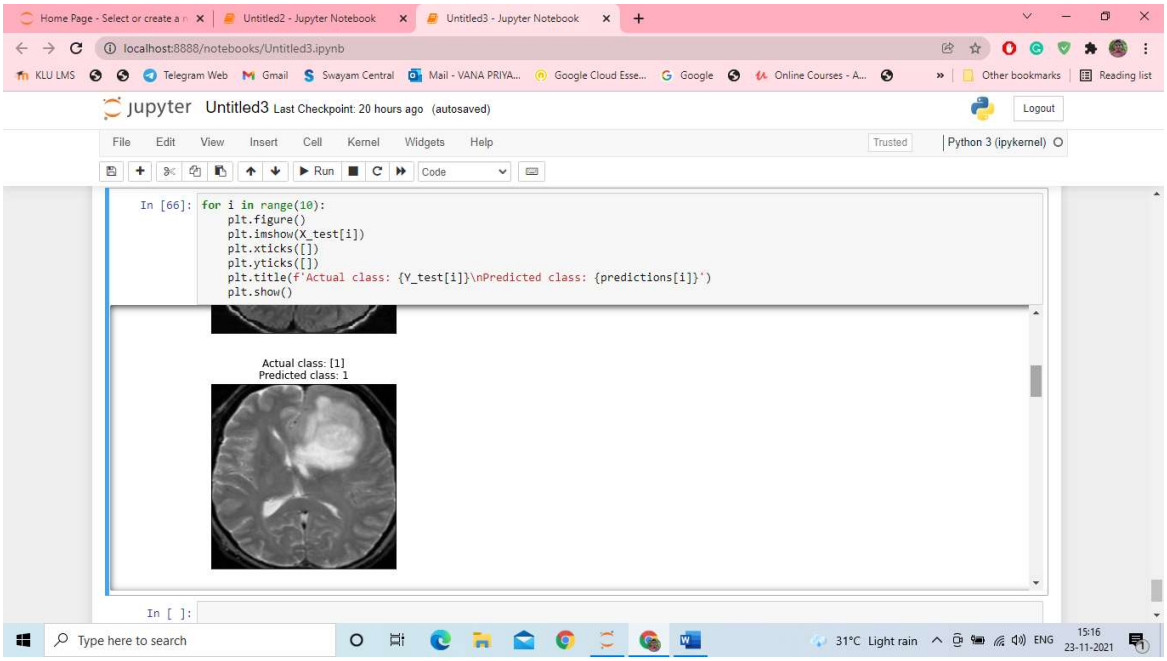
Output Screenshot-1



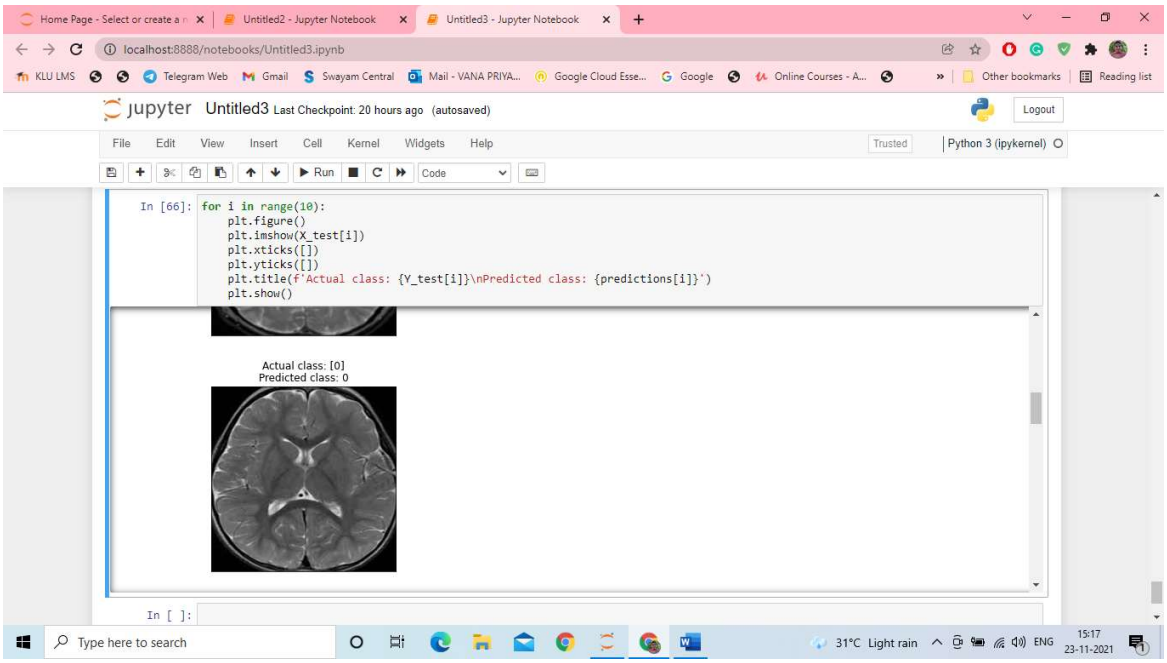
Output Screenshot-2



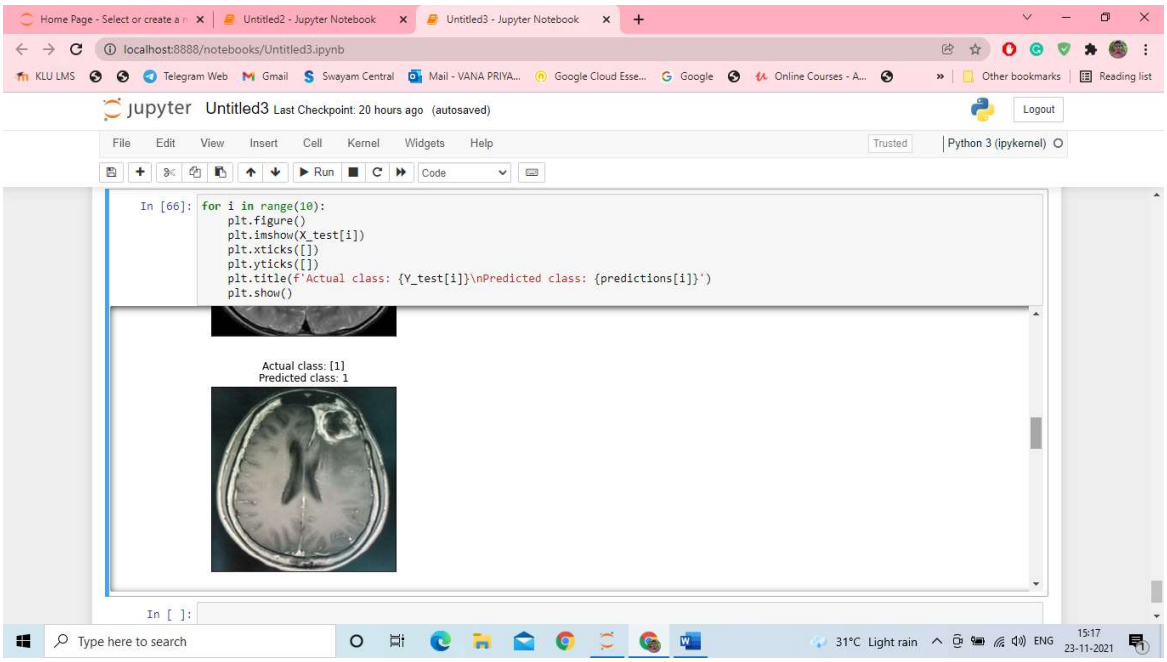
Output Screenshot-3



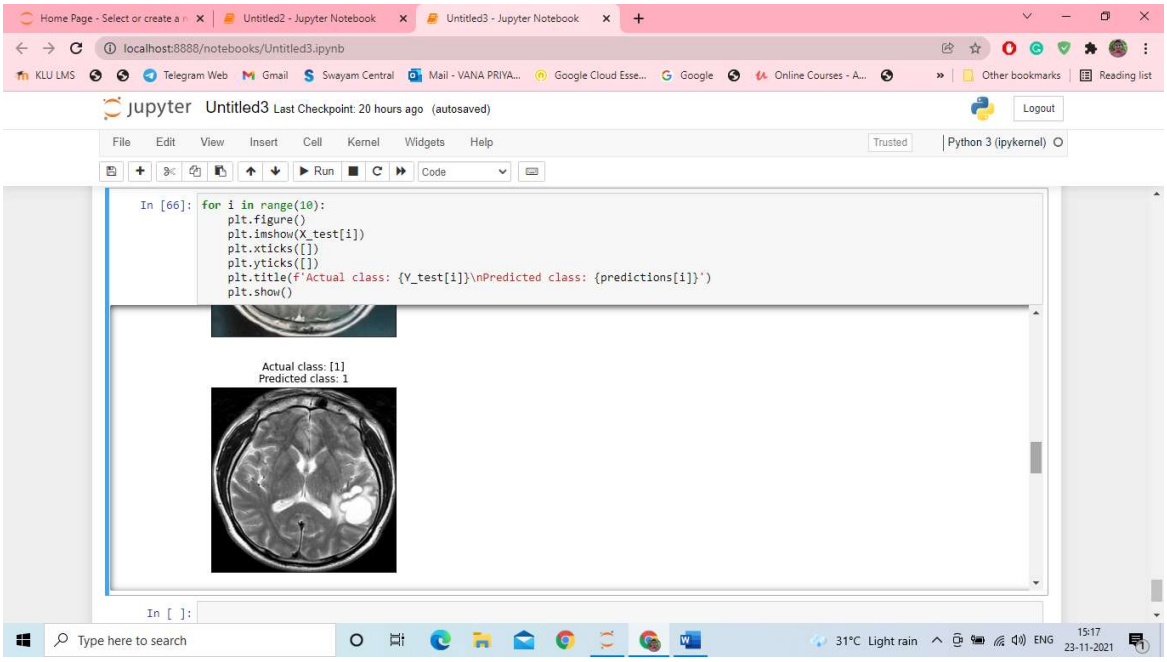
Output Screenshot-4



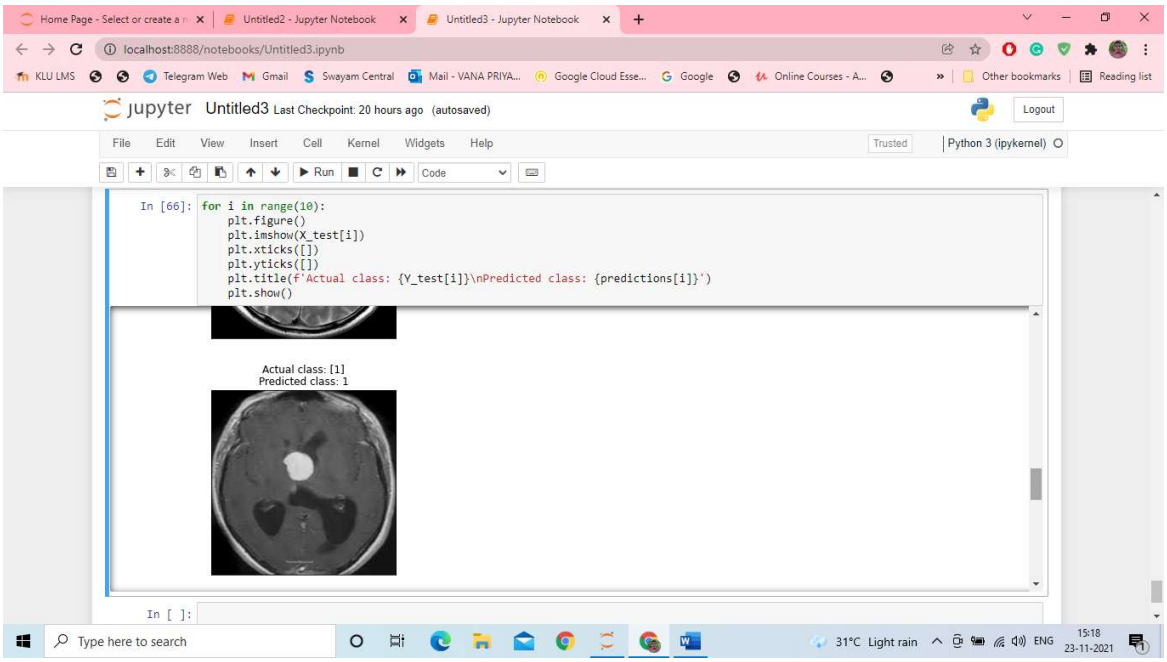
Output Screenshot-5



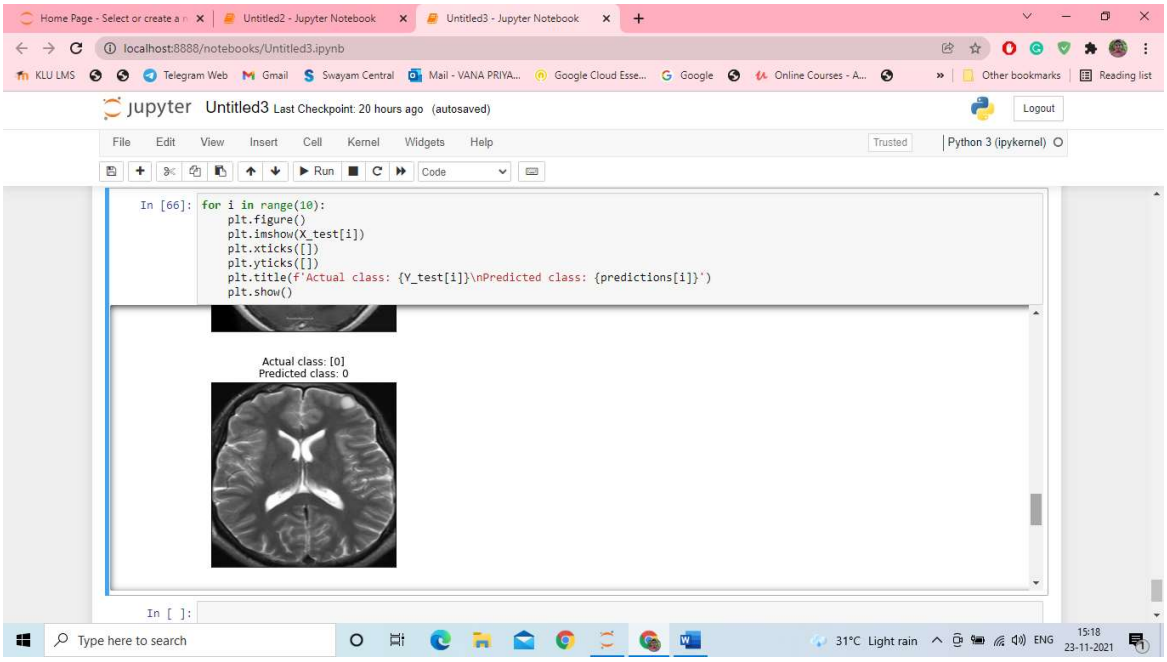
Output Screenshot-6



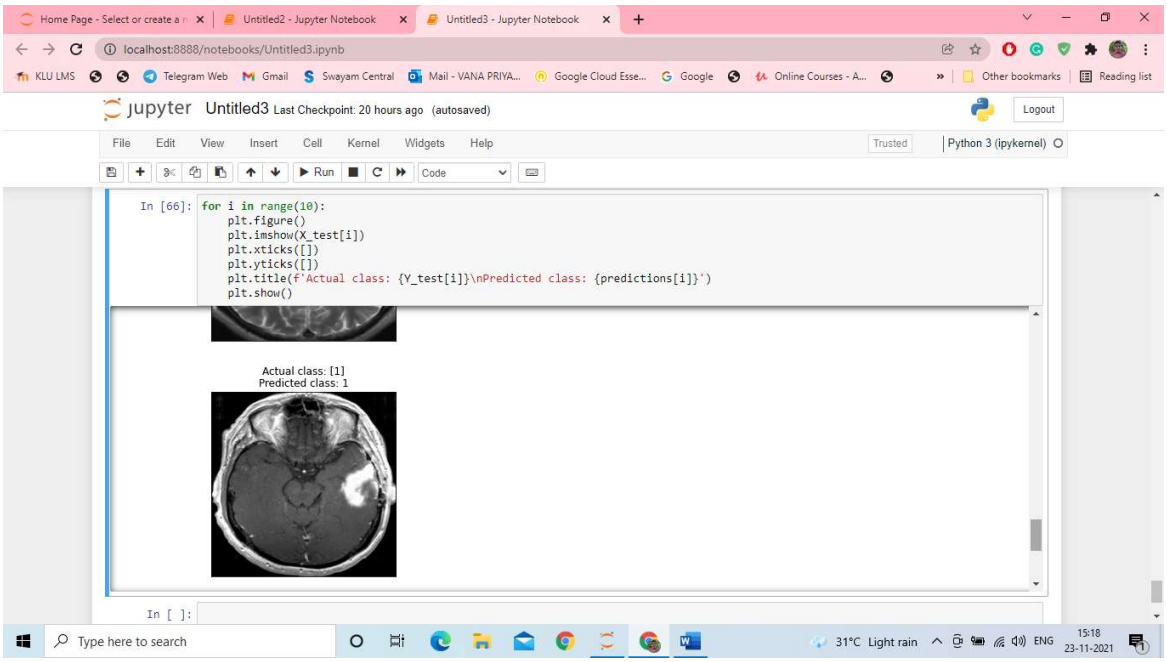
Output Screenshot-7



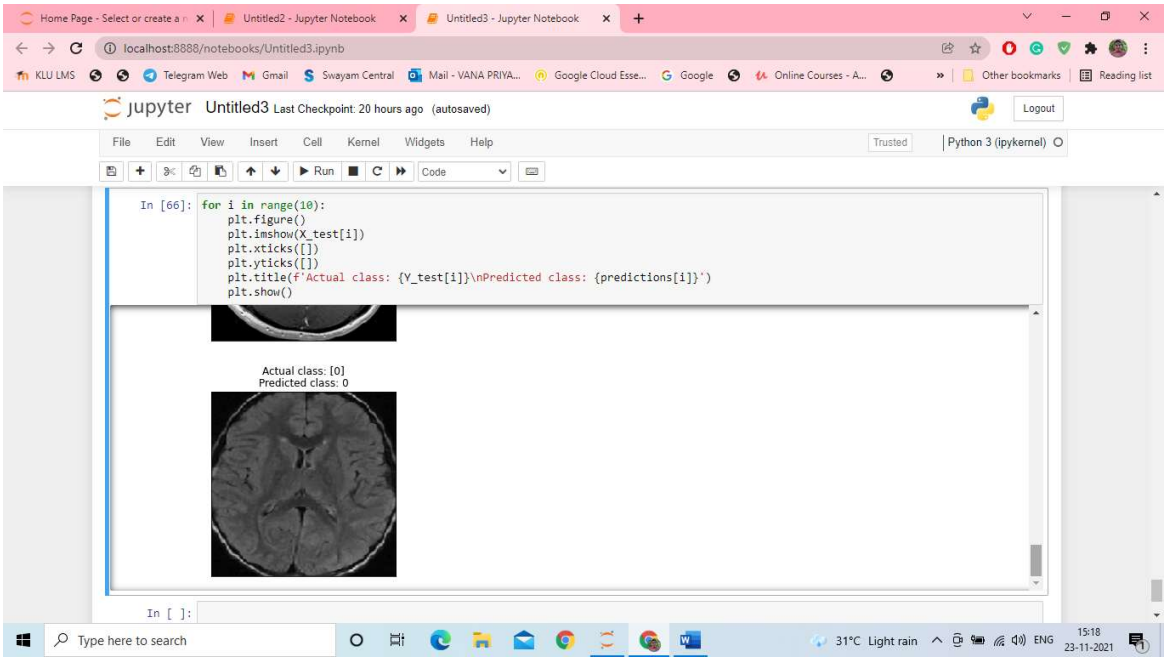
Output Screenshot-8



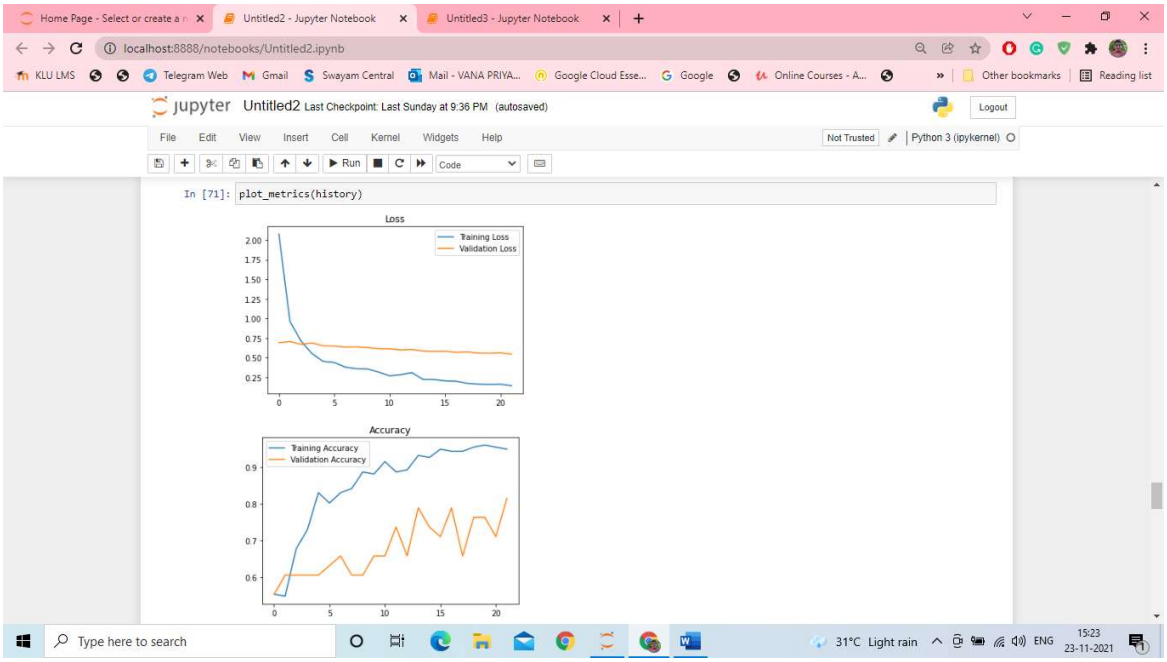
Output Screenshot-9



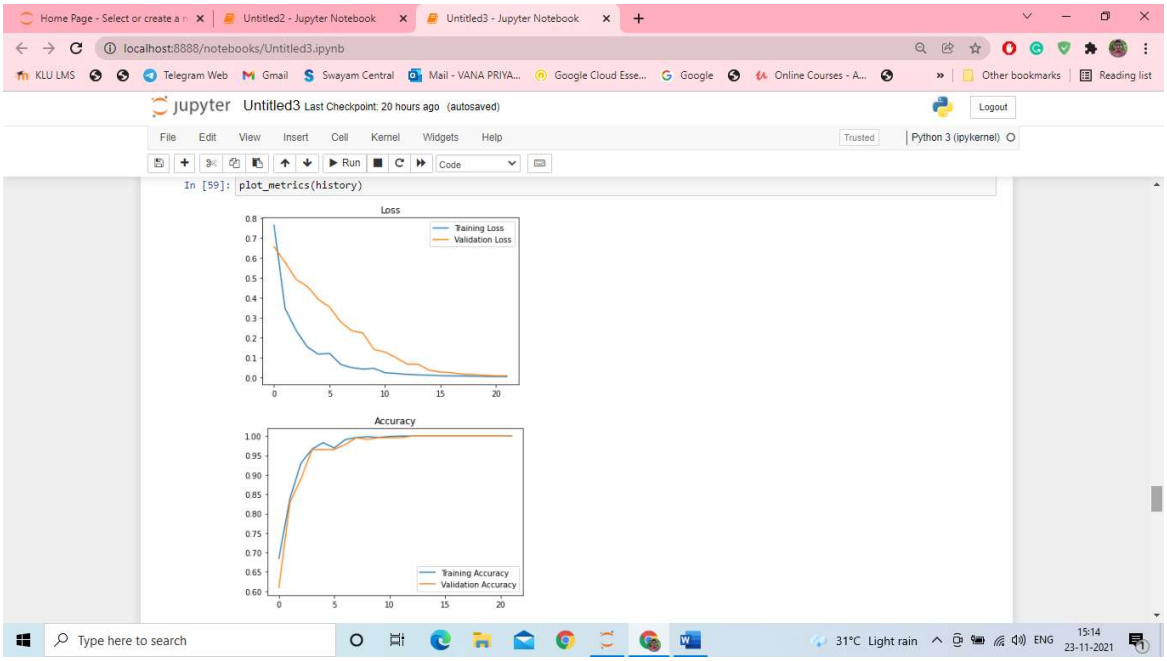
Output Screenshot-10



Graphs



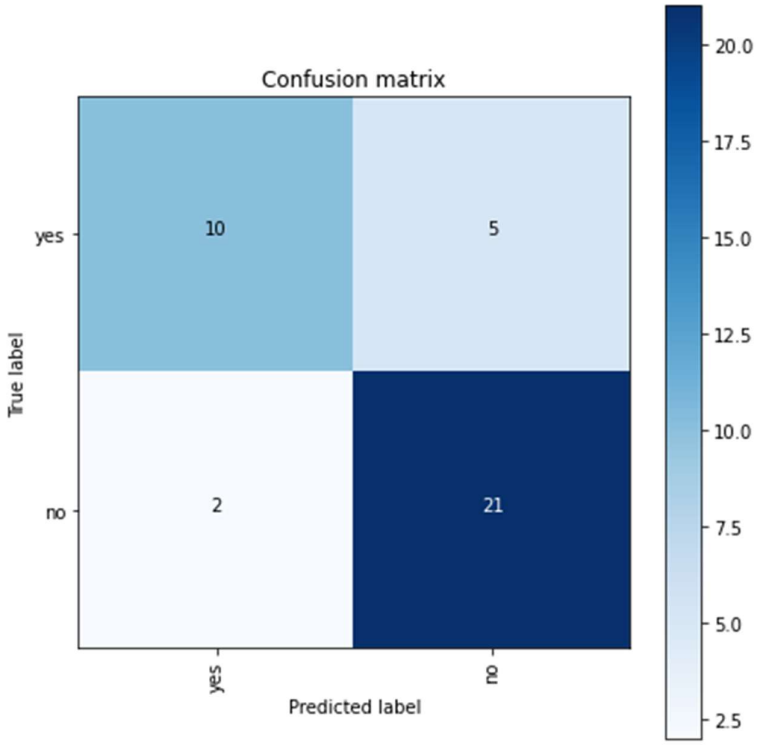
When tested, the accuracy of the first model is 82%. The above graphs represents the accuracy and the loss of first model.

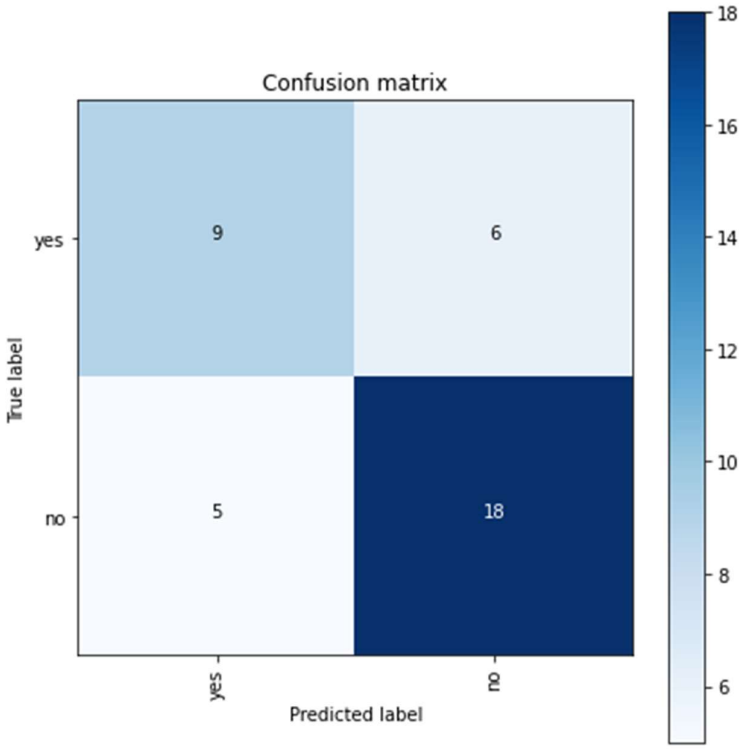


The above 2 graphs represent the accuracy and loss occurred while using the extended model. Here the accuracy is recorded as 100%.

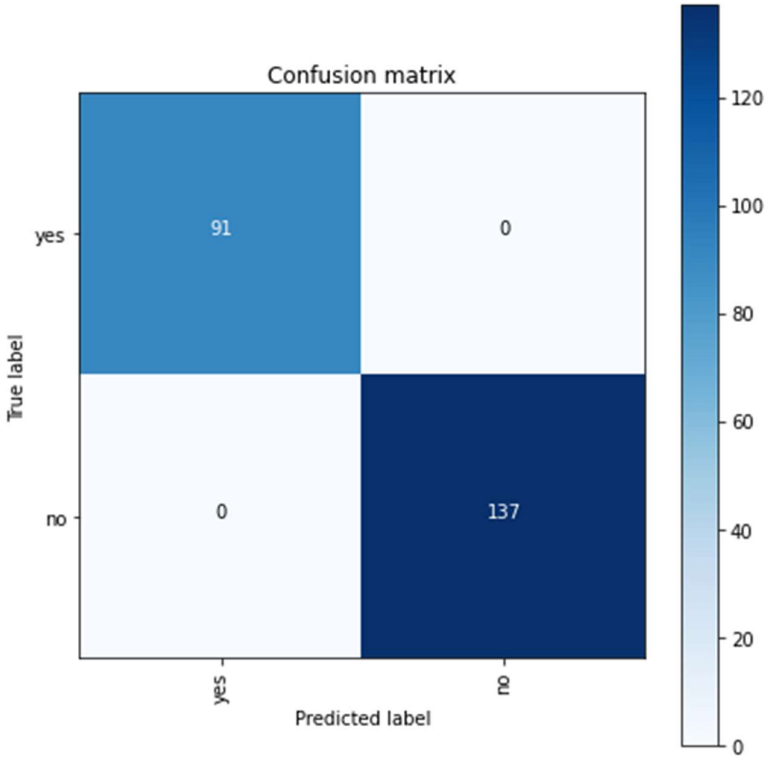
Chapter-7 Summary and conclusion

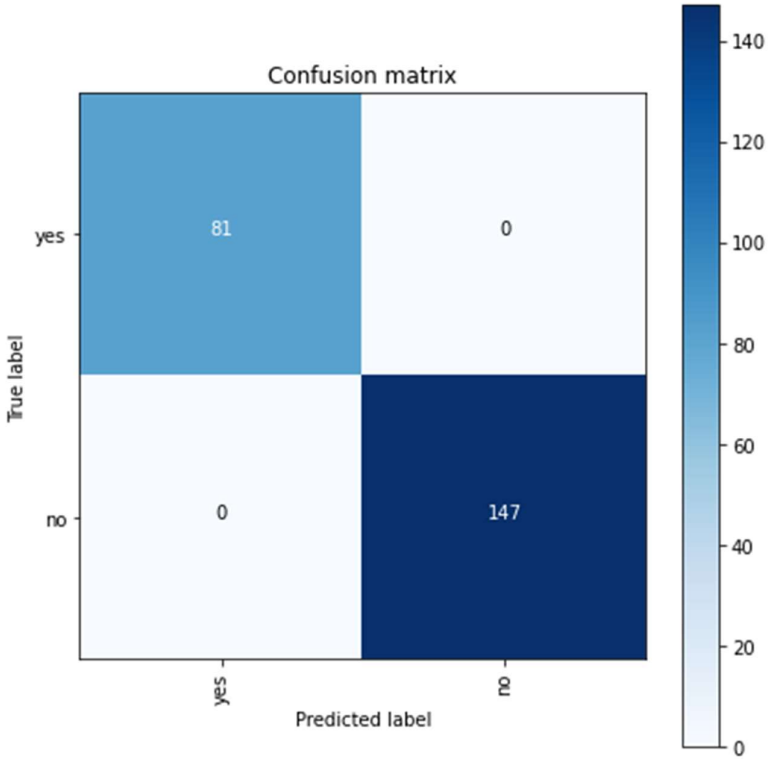
The below 2 pictures representing the confusion matrix for the existing model, with accuracy of 82%.





The below 2 pictures representing the confusion matrix for the proposed model, with accuracy of 100%.





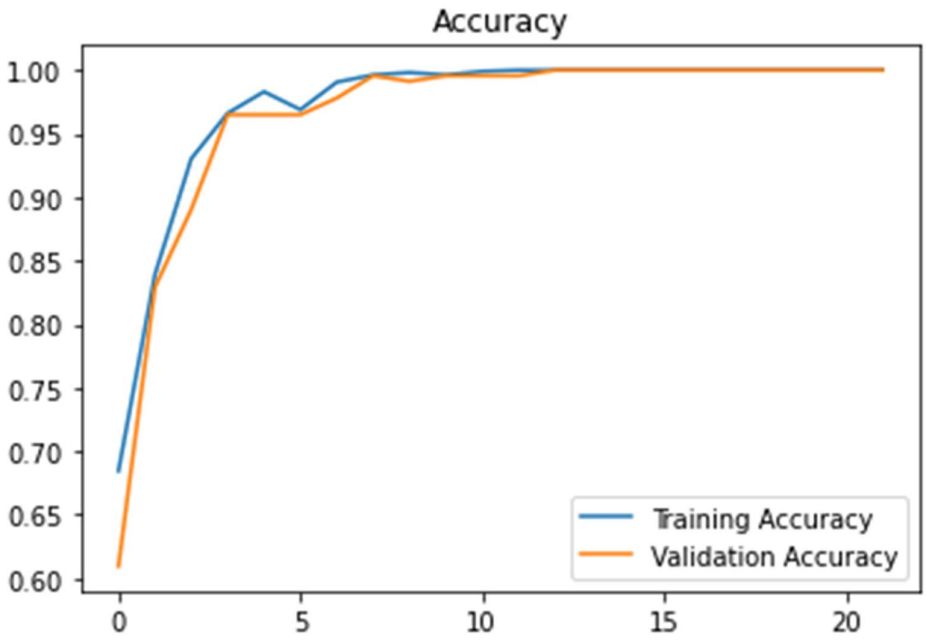


Figure 7.1 Accuracy Graph

Brain tumor detection is done using CNN with usage of layers multiple layer segmentation is done by shape, size, and texture. Data set we have taken consists of images of brain classified into 2 types. i.e., tumorous and non-tumorous. We have taken a data set consisting of images each. The data set is divided into 78% training data and 22% testing data. The training data set has undergone 22 epochs for accurate feature extraction and thereby prediction is done using testing dataset. The results are 100% accurate and more reliable compared to manual testing.

CHAPTER 8:

References

Aswathy, S. (2014). A survey on detection of brain tumor from MRI brain images. *IEEE Xplore*, 10-16.

B.V., E. (2020). Integrating uncertainty in deep neural networks for MRI based stroke analysis. *ScienceDirect*, 1-3.

Bahadure, N. B. (2017). Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. *International Journal of Biomedical Imaging*, 6-10.

Begum, S. S. (2020). Combining optimal wavelet statistical texture and recurrent neural network for tumour detection and classification over MRI. *SpringerLink*, 5-8.

Bhandari, A. (2020). Convolutional neural networks for brain tumour segmentation. *Springer*, 2-6.

Chahal, P. K. (2020). A survey on brain tumor detection techniques for MR images. *SpringerLink*, 1-4.

Reddy, D. (2018). Brain Tumor Detection Using Image Segmentation Techniques. *IEEE Xplore*, 3-6.

Thillaikkarasi, R. (2019). An Enhancement of Deep Learning Algorithm for Brain Tumor Segmentation Using Kernel Based CNN with M-SVM. *SpringlerLink*, 3-8.