**Brain Tumor Detection Using Deep Learning**

**END-TERM REPORT**

By

1. R. Suresh Ram 11908772(B56)
2. M. Karthik 11908893(B58)



Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

November-2022

**Student Declaration**

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We ever that if any part of the report is found to be copied, we are shall take full responsibility for it.

R. Suresh Ram

11908772

M. Karthik

11908893

Place: phagwara

Date: 07-11-2022

**BONAFIDE CERTIFICATE**

Certified that this project report” Brain Tumor Detection Using Deep Learning” is the bonafide work of M. Karthik, R. Suresh Ram, who carried out the project work under Dr. Dhanpratap Singh supervision.

Dr. Dhanpratap Singh

Computer science and engineering

**ABSTRACT**

Now a day’s tumor is second leading cause of cancer. Due to cancer large no of patients are in danger. The medical field needs fast, automated, efficient and reliable technique to detect tumor like brain tumor. Detection plays very important role in treatment. If proper detection of tumor is possible then doctors keep a patient out of danger. Various image processing techniques are used in this application. Using this application doctors provide proper treatment and save a number of tumor patients. A tumor is nothing but excess cells growing in an uncontrolled manner. Brain tumor cells grow in a way that they eventually take up all the nutrients meant for the healthy cells and tissues, which results in brain failure. Currently, doctors locate the position and the area of brain tumor by looking at the MR Images of the brain of the patient manually. This results in inaccurate detection of the tumor and is considered very time consuming. A tumor is a mass of tissue it grows out of control. We can use a Deep Learning architectures CNN (Convolution Neural Network) generally known as NN (Neural Network) and VGG 16(visual geometry group) Transfer learning for detect the brain tumor. The performance of model is predict image tumor is present or not in image. If the tumor is present it return yes otherwise return no.

**MOTIVATION FOR THE WORK**:

A brain tumor is defined as abnormal growth of cells within the brain or central spinal canal. Some tumors can be cancerous thus they need to be detected and cured in time. The exact cause of brain tumors is not clear and neither is exact set of symptoms defined, thus, people may be suffering from it without realizing the danger. Primary brain tumors can be either malignant (contain cancer cells) or benign (do not contain cancer cells).

Brain tumor occurred when the cells were dividing and growing abnormally. It is appearing to be a solid mass when it diagnosed with diagnostic medical imaging techniques. There are two types of brain tumor which is primary brain tumor and metastatic brain tumor. Primary brain tumor is the condition when the tumor is formed in the brain and tended to stay there while the metastatic brain tumor is the tumor that is formed elsewhere in the body and spread through the brain.

The MRI is the most regularly utilized strategy for imaging brain tumors and the identification of its vicinity. The conventional technique for CT and MR image classification and detection of tumor cells remains largely supported for the human reviewing apart from different other methods. MR images are mainly used because there are non-destructive and non-ionizing. MR imaging offers high-definition pictures that are extensively utilized in discovering brain tumors. MRI has diverse schemes such as flair, T1-weighted, T2-weighted images. There are many image processing techniques such as pre-processing, segmentation of images, image improvements, feature extraction, and classifiers.

**INTRODUCTION**

**BRAIN TUMOR DETECTION SYSTEM**

The human body is made up of many organs and brain is the most critical and vital organ of them all. One of the common reasons for dysfunction of brain is brain tumour. A tumour is nothing but excess cells growing in an uncontrolled manner. Brain tumour cells grow in a way that they eventually take up all the nutrients meant for the healthy cells and tissues, which results in brain failure. Currently, doctors locate the position and the area of brain tumour by looking at the MR Images of the brain of the patient manually. This results in inaccurate detection of the tumour and is considered very time consuming.

A Brain Cancer is very critical disease which causes deaths of many individuals. The brain tumour detection and classification system are available so that it can be diagnosed at early stages. Cancer classification is the most challenging tasks in clinical diagnosis.

This project deals with such a system, which uses computer, based procedures to detect tumour blocks and classify the type of tumour using Convolution Neural Network Algorithm for MRI images of different patients.

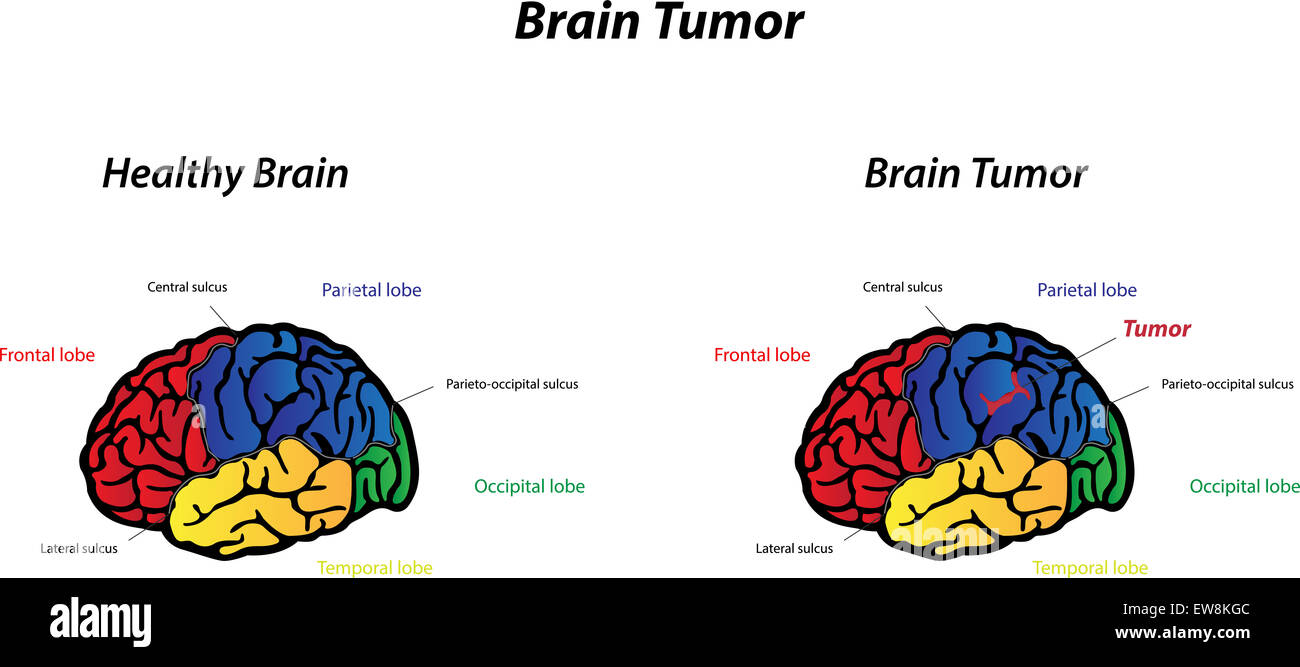
Different types of image processing techniques like image segmentation, image enhancement and feature extraction are used for the brain tumour detection in the MRI images of the cancer-affected patients.

Detecting Brain tumour using Image Processing techniques it involves the four stages is Image Pre-Processing, Image segmentation, Feature Extraction, and Classification.

Image processing and neural network techniques are used for improve the performance of detecting and classifying brain tumour in MRI images.

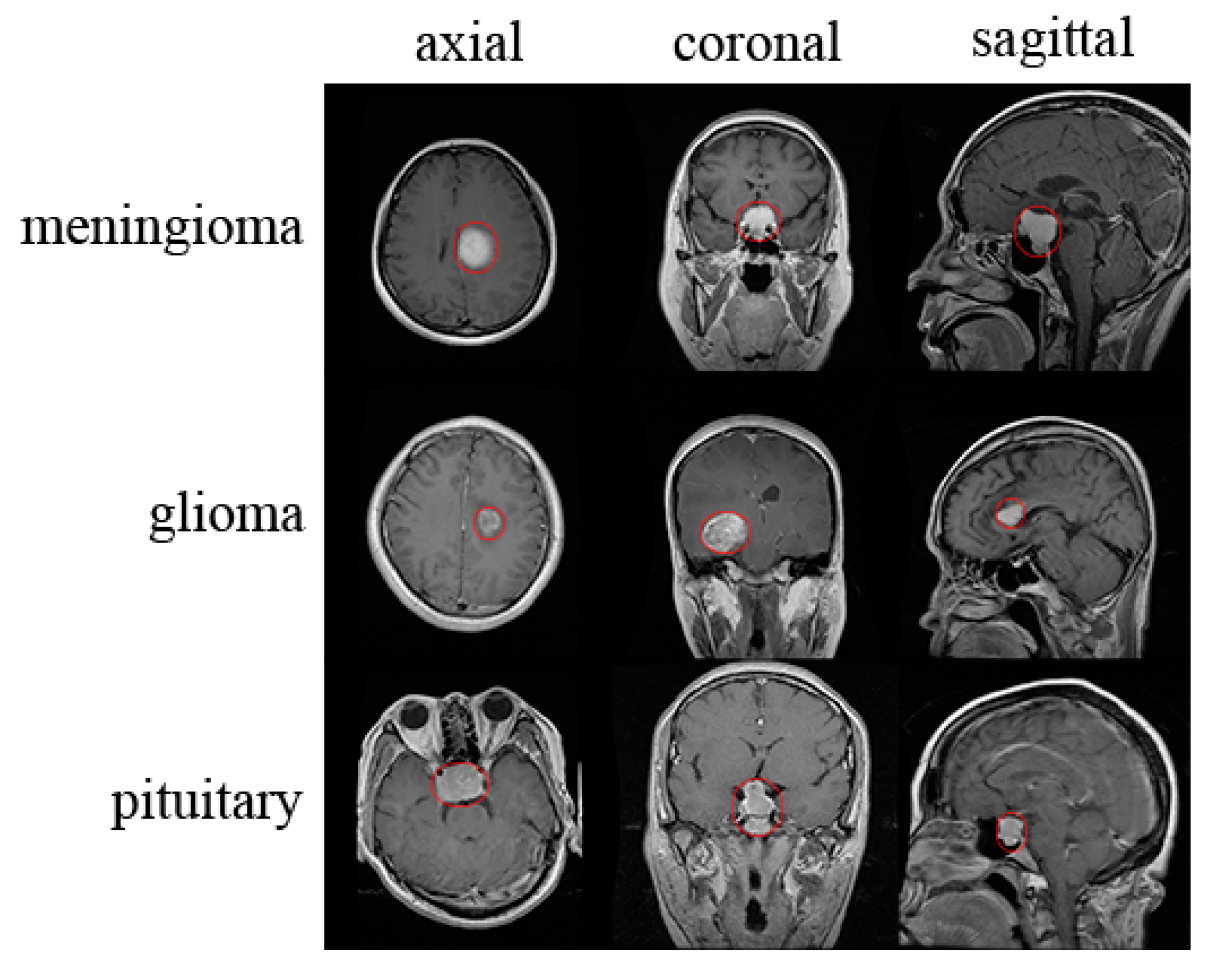
**OVERVIEW OF BRAIN AND BRAIN TUMOR**

Main part in human nervous system is human brain. It is located in human head and it is covered by the skull. The function of human brain is to control all the parts of human body. It is one kind of organ that allows human to accept and endure all type of environmental condition. The human brain enables humans to do the action and share the thoughts and feeling. In this section we describe the structure of the brain for understanding the basic things.



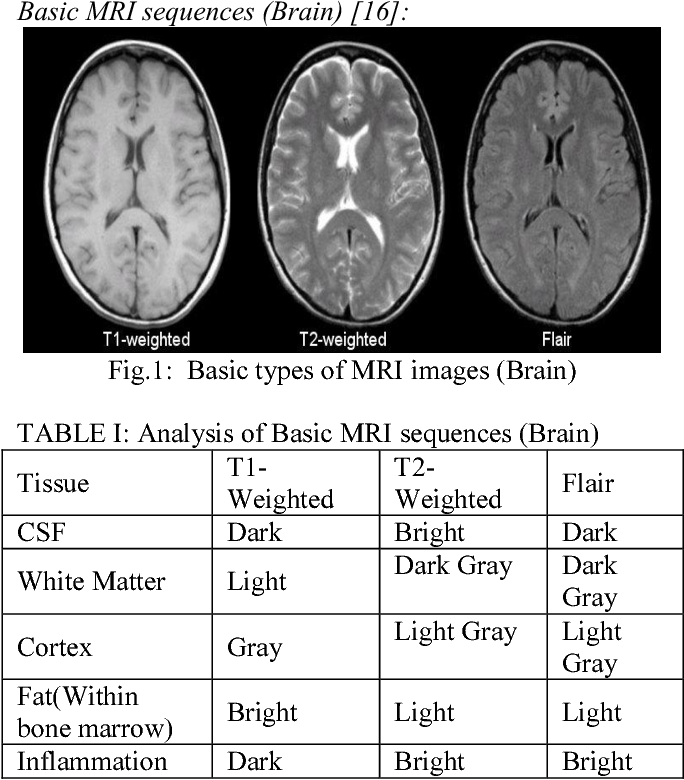
The brain tumours are classified into mainly two types: Primary brain tumour and secondary brain tumour. The benign tumour is one type of cell grows slowly in the brain and type of brain tumour is gliomas. It originates from non-neuronal brain cells called astrocytes. Basically, primary tumours are less aggressive but these tumours have much pressure on the brain and because of that, brain stops working properly.

The secondary tumours are more aggressive and quicker to spread into other tissue. Secondary brain tumour originates through other part of the body. These types of tumours have a cancer cell in the body that is metastatic which spread into different areas of the body like brain, lungs etc. Secondary brain tumour is very malignant. The reason of secondary brain tumour cause is mainly due to lungs cancer, kidney cancer, bladder cancer etc.

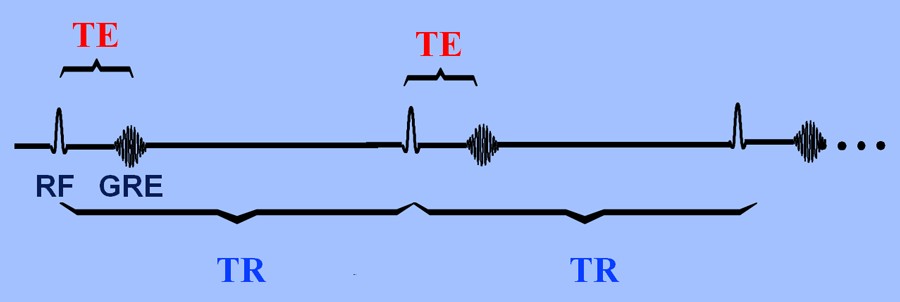


**MAGNETIC RESONANCE IMAGING (MRI)**

Raymond v. Damadian invented the first magnetic image in 1969. In 1977 the first MRI image were invented for human body and the most perfect technique. Because of MRI we are able to visualize the details of internal structure of brain and from that we can observe the different types of tissues of human body. MRI images have a better quality as compared to other medical imaging techniques like X-ray and computer tomography. MRI is good technique for knowing the brain tumour in human body. There are different images of MRI for mapping tumour induced Change including T1 weighted, T2 weighted and FLAIR (Fluid attenuated inversion recovery) weighted.



The most common MRI sequence is T1 weighted and T2 weighted. In T1 weighted only one tissue type is bright FAT and in T2 weighted two tissue types are Bright FAT and Water both. In T1 weighted the repetition time (TR) is short in T2 weighted the TE and TR is long. The TE a TR are the pulse sequence parameter and stand for repetition time and time to echo and it can be measured in milliseconds. The echo time represented time from the centre of the RF pulse to the centre of the echo and TR is the length of time between the TE repeating series of pulse and echo.



The third commonly used sequence in the FLAIR. The Flair sequence is almost same as T2-weighted image. The only difference is TE and TR time are very long.

**APPLICATION**

* The main aim of the applications is tumour identification.
* The main reason behind the development of this application is to provide proper treatment as soon as possible and protect the human life which is in danger.
* This application is helpful to doctors as well as patient.
* The manual identification is not so fast, more accurate and efficient for user. To overcome those problem this application is design.
* It is user friendly application.

**OBJECTIVE**

* To provide doctors good software to identify tumour and their causes.
* Save patient’s time.
* Provide a solution appropriately at early stages.
* Get timely consultation.

**CONVOLUTIONAL NEURAL NETWORK**

An automatic and reliable classification method Convolutional Neural Network (CNN) will be used since it is robust in structure which helps in identifying every minute detail. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance to various aspects/objects in the image and be able to differentiate one from the other.

The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNet have the ability to learn these filters/characteristics.

A ConvNet is able to successfully capture the spatial and temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

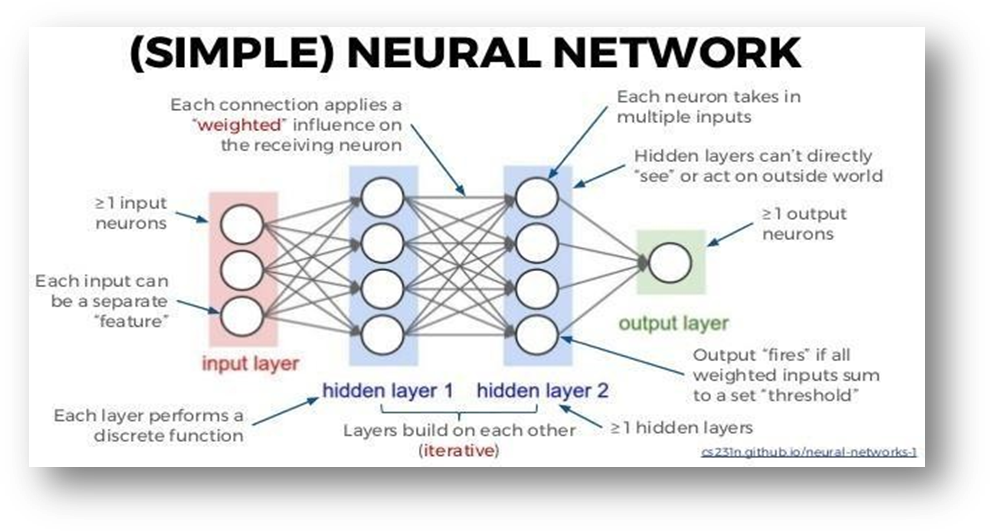
For this step we need to import Keras and other packages that we’re going to use in building the CNN.

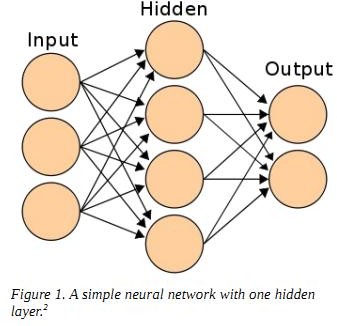
Import the following packages:

* Sequential is used to initialize the neural network.
* Convolution2D is used to make the convolutional network that deals with the images.
* MaxPooling2D layer is used to add the pooling layers.
* Flatten is the function that converts the pooled feature map to a single column that is passed to the fully connected layer.
* Dense adds the fully connected layer to the neural network.

**Basic Operation of Neural Networks:**

Neural Networks (NN) form the base of deep learning, a subfield of machine learning where the algorithms are inspired by the structure of the human brain. NN take in data, train themselves to recognize the patterns in this data and then predict the outputs for a new set of similar data. NN are made up of layers of neurons. These neurons are the core processing units of the network. First we have the input layer which receives the input; the output layer predicts our final output. In between, exist the hidden layers which perform most of the computations required by our network.





Input from medical professionals or users

Results are shown on IoT based devices or Web-based applications

Weights(w)

Nodes

EXPERIMENTAL ANALYSIS AND RESULTS

Python:

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991; Python's design philosophy emphasizes code Readability with its notable use of significant Whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

PIP:

It is the package management system used to install and manage software packages written in Python.

NumPy:

NumPy is a general-purpose array-processing package. It provides high performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

• A powerful N-dimensional array object

• Sophisticated (broadcasting) functions

• Tools for integrating C/C++ and Fortran code

• Useful linear algebra, Fourier transform, and random number capabilities

Pandas:

Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python.

We can analyze data in pandas with

1. Series

2. Data frames

Jupyter Notebook:

Anaconda distribution comes with 1,500 packages selected from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text mathematics, plots and rich media, usually ending with the “. ipynb" extension.

Tensor Flow:

Tensor flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Keras:

Keras is an open-source neural-network library written in Python. It is capable of running on top of Tensor Flow, Microsoft Cognitive Toolkit, R, Theano, or Plaid ML.

Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

OpenCV:

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability.

**DATASET:**

The dataset was taken from the website called “Kaggle”.

Website: “[brain tumor detection dataset](https://www.kaggle.com/code/darshkhanna007/brain-tumor-classification-and-detection/data)”

The brain tumor detection dataset consists of two folders:

1. Training folder

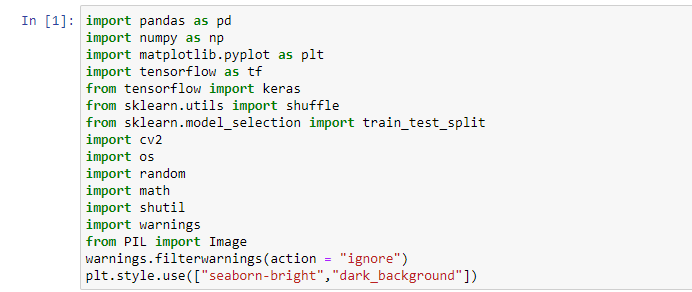
2. Testing folder

Each folder consists of 4 files namely glioma\_tumor, meningioma\_tumor, no\_tumor, pituitary\_tumor.

The each Training folder files contains nearly 100 jpg file images.

The each Testing folder files contains nearly 330 jpg file images.

**SAMPLE CODE AND RESULTS**

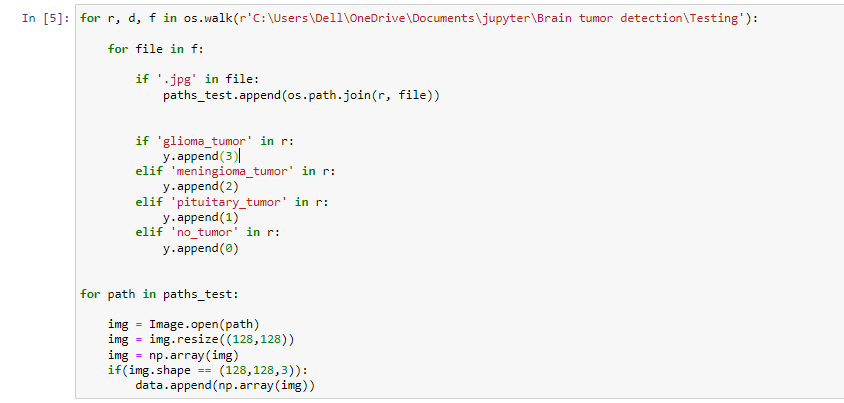


1. matplotlib is used for displaying images.
2. sklearn.utils.shuffle (\*arrays, \*\*options) Shuffle arrays or sparse matrices in a consistent way. This is a convenience alias to resample (\*arrays, replace=False) to do random permutations of the collections.
3. The train\_test\_split function of the sklearn. model\_selection package in Python splits arrays or matrices into random subsets for train and test data, respectively.
4. Python “os” system function allows us to run a command in the Python script, just like if I was running it in my shell.
5. The shutil module offers a number of high-level operations on files and collections of files.
6. PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The Image module provides a class with the same name which is used to represent a PIL image.

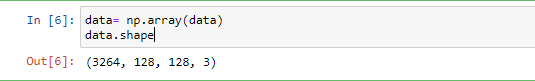


In this I have imported the training file from the brain tumor detection dataset.

1. OS. walk() generate the file names in a directory tree by walking the tree either top-down or bottom-up. For each directory in the tree rooted at directory top (including top itself), it yields a 3-tuple (dirpath, dirnames, filenames).
2. JPG is a digital image format which contains compressed image data. With a 10:1 compression ratio JPG images are very compact. JPG format contains important image details.
3. os. path. join combines path names into one complete path. This means that you can merge multiple parts of a path into one, instead of hard-coding every path name manually.
4. Image.open() Opens and identifies the given image file. This is a lazy operation; this function identifies the file, but the file remains open and the actual image data is not read from the file until you try to process the data (or call the load() method).
5. In image shape we have taken 128,128,3 as image dimension.In that we are adding the images one after another.



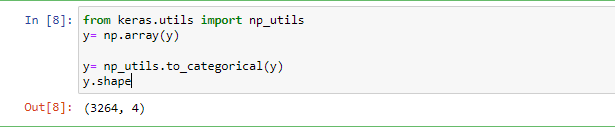
* In this we have imported the testing file from brain tumor detection dataset.
* After importing we have done same as what we have done in Training dataset.



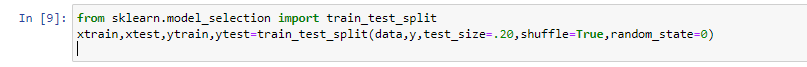
* Here we have the total shape of the dataset.
* Both training and testing files having 3264 jpg file images with 128\*128\*3 dimension.



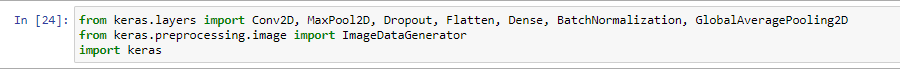
* In this we are casting the datatype to float type.
* Here we divide data by 255 because the numbers will be small and the computation becomes easier and faster.



* np\_utils. to\_categorical is used to convert array of labelled data(from 0 to nb\_classes - 1 ) to one-hot vector.
* We use to\_categorical to transform our training data before we pass it to we model. If our training data uses classes as numbers, to\_categorical will transform those numbers in proper vectors for using with models.



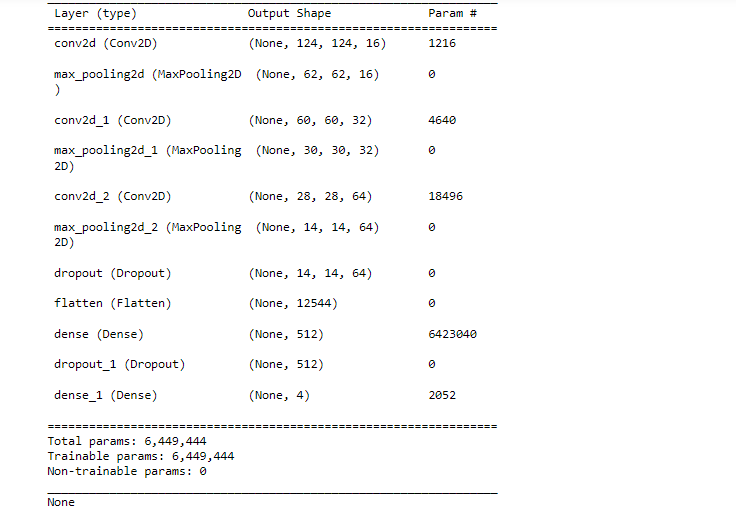
* We are giving 20% dataset for testing.
* Shuffle=true means we are shuffling the dataset before it’s splitting.
* Random\_state=0 means it will tell how the data is shuffled before it is splitted to training and testing.



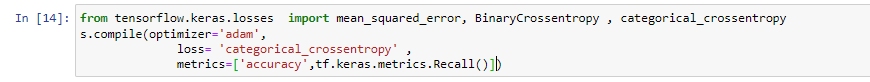
Here we are importing some keras layers which are helpful for us to building the CNN model.

Keras Conv2D is 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.

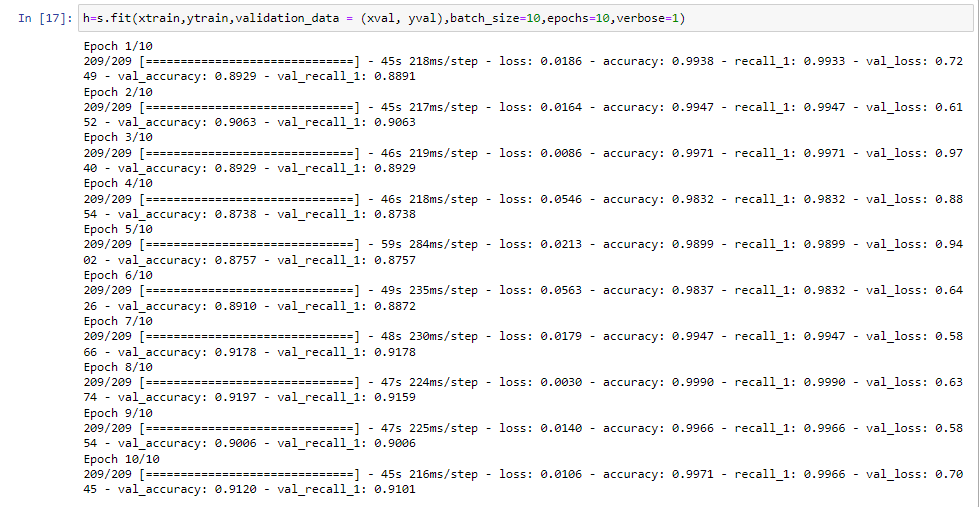




* Here we are creating a convolutional kernel that is convolved with the layer input to produce a tensor of outputs.
* Activation=’relu’. The Rectified Linear Unit function returns 0 if it receives any negative input, but for any positive value x it returns that value back.
* Max pooling selects the brighter pixels from the image. It is useful when the background of the image is dark and we are interested in only the lighter pixels of the image.
* The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.
* Dense Layer is used to classify image based on output from convolutional layers.



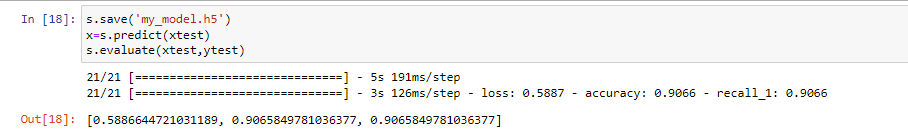
* Adam=optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
* categorical\_crossentropy: Used as a loss function for multi-class classification model where there are two or more output labels.
* Accuracy class = Calculates how often predictions equal labels. This metric creates two local variables, total and count that are used to compute the frequency with which y\_pred matches y\_true.
* Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made.



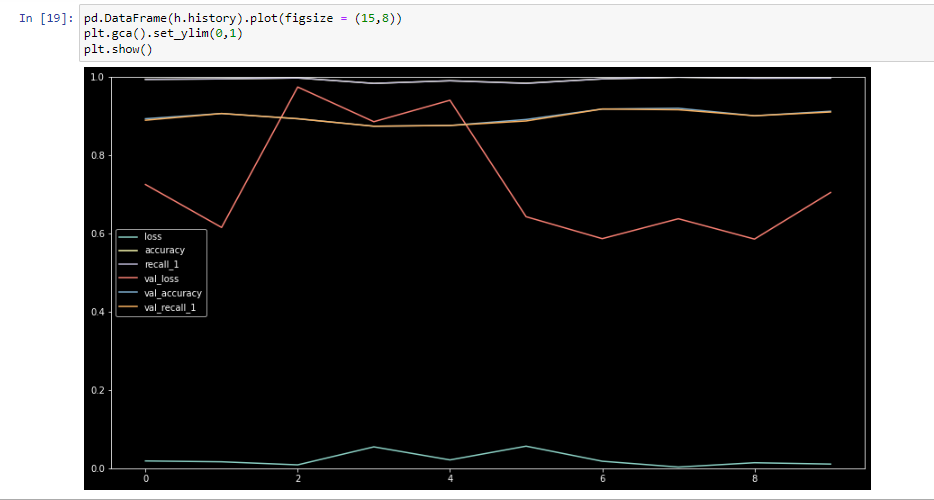
* We use fit() to pass the whole dataset at once.
* batch\_size=10 means we are passing 10 images at one time.
* The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.
* verbose=0 will show you nothing (silent) verbose=1 will show you an animated progress bar. verbose=2 will just mention the number of epochs.

For every epoch we are getting an average

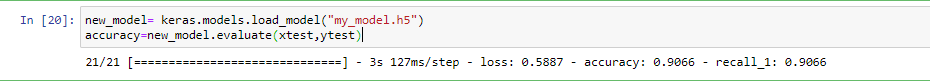
* Loss = 0.2682
* accuracy = 99.5
* recall = 95.5
* val\_loss = 0.8575
* val\_accuracy = 0.899
* val\_recall = 0.905



* H5 is a file format to store structured data, it's not a model by itself. Keras saves models in this format as it can easily store the weights and model configuration in a single file.
* model. predict() : given a trained model, predict the label of a new set of data.



Here we used the plot graph to identify the loss, accuracy, recall, val\_loss, val\_accuracy, val\_recall.



Here we load the model using H5

Finally we evaluated the testing dataset and we got the

Loss = 0.5887

accuracy = 0.9066

recall = 0.9066