PROJECT REPORT

Computer Science and Engineering

BRAIN TUMOR DETECTION USING DEEP LEARNING

(MENTORED BY: DR. TRILOK CHAND ASERI)

Submitted By:

20103001: Karan Dhar

20103035: Ankit Rupal

20103034: Adil Vinayak

20103051: Lalit Kumar

Declaration

We, hereby certify that the work embodied in this project report entitled “BRAIN TUMOR DETECTION USING DEEP LEARNING” is an authentic record of our own work carried out under the mentorship of Dr. Trilok Chand (Professor, PEC).

NAME	**SID**	**SIGNATURE**
Karan Dhar	20103001	
Ankit Rupal	20103035	
Adil Vinayak	20103034	
Lalit Kumar	20103051	

This is to certify that the above statement made by the candidates is correct to the best of our knowledge and belief. However, responsibility for any plagiarism related issue solely stands with the students.

Signature of the Mentor:

Name with Designation: Dr. Trilok Chand

ABSTRACT

Nowadays brain tumor is the second most leading cause of cancer. Due to cancer, large number 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 will be able to 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 brain tumor patients. A brain 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 mobile_net Transfer learning for detect the brain tumor. The task of model is to predict if tumor is present or not in image. If the tumor is present it return yes otherwise return no.

ACKNOWLEDGMENTS

This project report is based on research work conducted for “Brain Tumor Detection using deep learning techniques”. This work would not be possible without the people whose contributions can’t be ignored.

We consider it an honor to work under the mentorship of Dr. Trilok Chand. This project report is on brain tumor detection. We were provided with much needed guidance and freedom to work on this project.

We are grateful to Punjab Engineering College for providing everything from faculty guides to resources for this work. We would like to thank all other people who have directly or indirectly helped us to realize this work.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 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 tumor. 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 Brain Cancer is very critical disease which causes deaths of many individuals. The brain tumor detection and classification system is 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 tumor blocks and classify the type of tumor 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 tumor detection in the MRI images of the cancer-affected patients.

Detecting Brain tumor using Image Processing techniques its 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 tumor 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 [4].

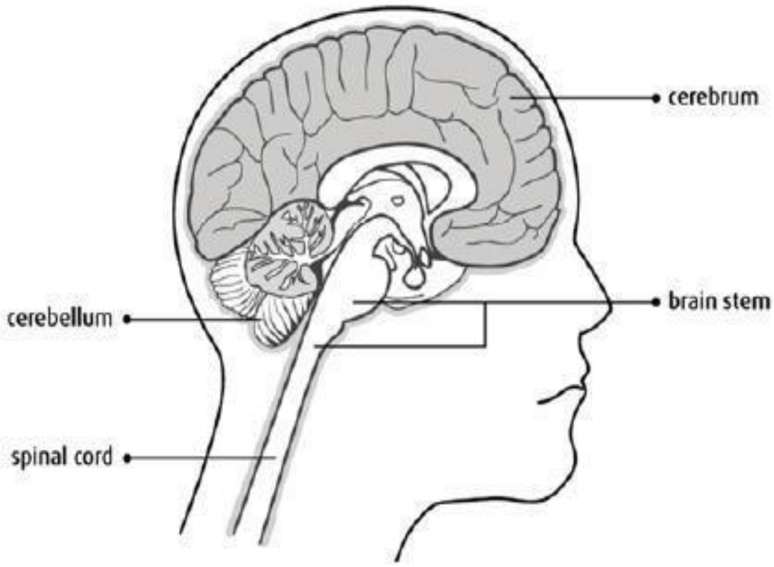

Fig.1: Basic Structure of human brain [5]

The brain tumors are classified into mainly two types: Primary brain tumor (benign tumor) and secondary brain tumor (malignant tumor).The benign tumor is one type of cell grows slowly in the brain and type of brain tumor is gliomas. It originates from non neuronal brain cells called astrocytes. Basically primary tumors are less aggressive but these tumors have much pressure on the brain and because of that, brain stops working properly [6]. The secondary tumors are more aggressive and more quick to spread into other tissue. Secondary brain tumor originates through other part of the body. These type of tumor have a cancer cell in the body that is metastatic which spread into different areas of the body like brain, lungs etc. Secondary brain tumor is very malignant. The reason of secondary brain tumor cause is mainly due to lungs cancer, kidney cancer, bladder cancer etc [7].

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 tumor in human body. There are different images of MRI for mapping tumor induced Change including T1 weighted, T2 weighted and FLAIR (Fluid attenuated inversion recovery) weighted shown in figure.

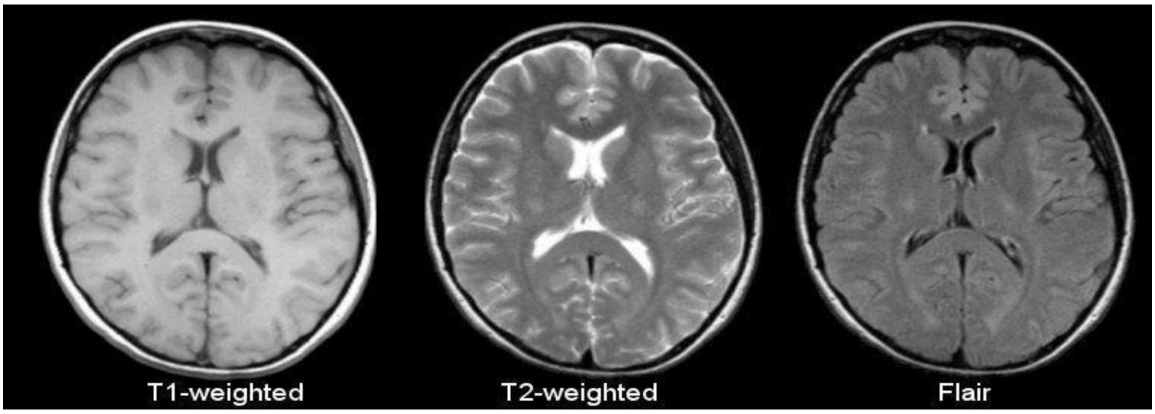

Fig 2: T1, T2 and Flair image [9]

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 an TR are the pulse sequence parameter and stand for repetition time and time to echo and it can be measured in millisecond(ms). 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 is shown in figure.

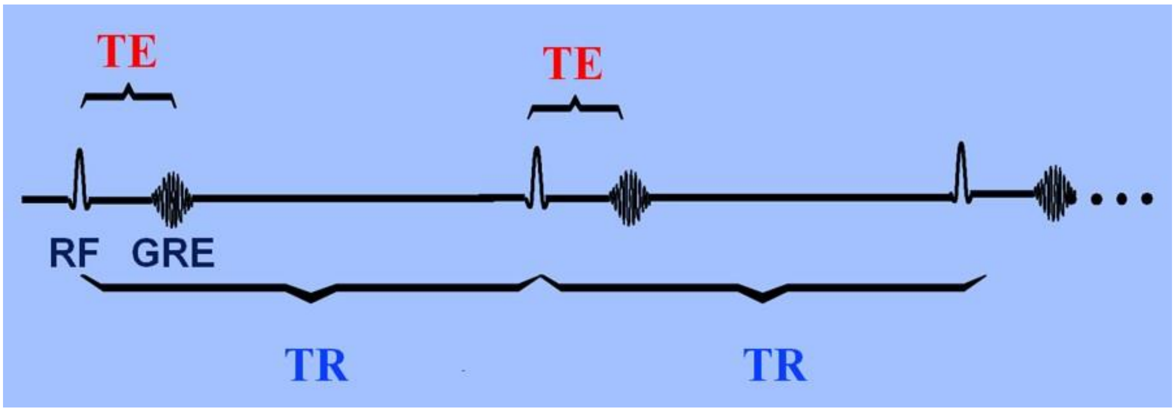

Fig. 3: Graph of TE and TR [10]

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. Their approximate TR and TE times are shown in table.

	TR (msec)	TE (msec)
T1-Weighted (short TR and TE)	500	14
T2-Weighted (long TR and TE)	4000	90
Flair (very long TR and TE	9000	114

Fig.4: Table of TR and TE time [9]

1.2 APPLICATION

- The main aim of the applications is tumor 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.

1.3 OBJECTIVE

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

1.4 MOTIVATION

The main motivation behind Brain tumor detection is to only detect tumor. So it can be useful in cases such as we have to know if the tumor is positive or negative, it can detect tumor from image and return the result tumor if positive or not. This project deals with such a system, which uses computer based procedures to detect tumor blocks and classify the type of tumor using Convolution Neural Network Algorithm for MRI images of different patients.

1.5 ORGANIZATION OF REPORT

Chapter 1 gives the brief introduction of Brain Tumor Detection and Classification using Deep Learning, its applications, objective of the system and motivation.

Chapter 2 contains literature survey that provide summary of individual paper.

Chapter 3 provides overview of existing work for Brain tumor detection and classification that has been done using done using CNN.

Chapter 4 presents Implementation and its results, tools and technology used to achieve this and dataset detail.

Chapter 5 contains conclusion about Brain tumor detection using deep learning.

2. LITERATURE SURVEY

Paper-1: Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM

- **Publication Year:** 6 March 2017
- **Author**: Nilesh Bhaskarrao Bahadure, Arun Kumar Ray and Har Pal Thethi
- **Journal Name:** Hindawi International Journal of Biomedical Imaging
- **Summary:** In this paper using MR images of the brain, we segmented brain tissues into normal tissues such as white matter, gray matter, cerebrospinal fluid (background), and tumor-infected tissues. We used pre-processing to improve the signal-to-noise ratio and to eliminate the effect of unwanted noise. We can used the skull stripping algorithm its based on threshold technique for improve the skull stripping performance.

Paper-2: A Survey on Brain Tumor Detection Using Image Processing Techniques

- **Publication Year:** 2017
- **Author:** Luxit Kapoor, Sanjeev Thakur
- **Journal Name:** IEEE 7th International Conference on Cloud Computing, Data Science & Engineering
- **Summary:** This paper surveys the various techniques that are part of Medical Image Processing and are prominently used in discovering brain tumors from MRI Images. Based on that research this Paper was written listing the various techniques in use. A brief description of each technique is also provided. Also of All the various steps involved in the process of detecting Tumors, Segmentation is the most significant.

Paper-3: Identification of Brain Tumor using Image Processing Techniques

- **Publication Year:** 11 September 2017
- **Author:** Praveen Gamage
- **Journal Name:** Research gate
- **Summary:** This paper survey of Identifying brain tumors through MRI images can be categorized into four different sections; pre-processing, image segmentation, Feature extraction and image classification.

Paper-4: Review of Brain Tumor Detection from MRI Images

- **Publication Year:** 2016
- **Author:** Deepa, Akansha Singh
- **Journal Name:** IEEE International Conference on Computing for Sustainable Global Development
- **Summary:** In this paper, some of the recent research work done on the Brain tumor detection and segmentation is reviewed. Different Techniques used by various researchers to detect the brain Tumor from the MRI images are described. By this review we found that automation of brain tumor detection and Segmentation from the MRI images is one of the most active Research areas.

Paper-5: An efficient Brain Tumor Detection from MRI Images using Entropy Measures

- **Publication Year:** December 23-25, 2016
- **Author:** Devendra Somwanshi , Ashutosh Kumar, Pratima Sharma, Deepika Joshi
- **Journal Name:** IEEE International Conference on Recent Advances and Innovations in Engineering
- **Summary:** In this paper, we have investigated the different Entropy functions for tumor segmentation and its detection from various MRI images. The different threshold values are obtained depend on the particular definition of the entropy. The threshold values are dependent on the different entropy function which in turn affects the segmented results.

3. EXISTING WORK & PROPOSED WORKFLOW

3.1 OVERVIEW OF EXISTING WORK

- In the first stage, there is a computer-based procedures to detect tumor blocks and classify the type of tumor using Artificial Neural Network Algorithm for MRI images of different patients.
- The second stage involves the use of different image pre-processing techniques such as histogram equalization, image segmentation, image enhancement, morphological operations and feature extraction are used for brain tumor detection in the MRI images for the cancer-affected patients.
- This work is introduced one automatic brain tumor detection method to increase the accuracy and decrease the diagnosis time.
- **Image Preprocessing:** As input for this system is MRI, scanned image and it contain noise. Therefore, our first aim is to remove noise from input image.
- **Size:** We resize the image sizes to (224,224) as it the image size we will give to our model to train it.
- **Conversion:** We convert the high definition images to greyscale to make it easier for our model to be trained.
- **Data Split:** We split our data into training and testing data set.
- **Mobile_net:** We initiate our model i.e. mobilenet_v2.
- **Tumor Identification:** In this phase, we are having dataset previously collected brain MRIs. Knowledge base is created for comparison by training our convolutional neural network.

IMAGE DESCRIPTION AND PROCESSING

- In the first step we can take image as input. In the image we used tumor in the image and only fat and water tissues in the images.
- In the second step convert image to grayscale
    - Signal to noise
    - Complexity of the code
    - Learning image processing
    - Difficulty of visualization
    - Color is complex
- Then we convert image to binary image by thresholding.

    Thresholding is the simplest method of image segmentation and the most common way to convert a grayscale image to binary image.

    In thresholding we select threshold value and then gray level value .below the selected threshold value is classified as 0.and equal and greater then the threshold value are classified as 1.

3.2 PROPOSED WORKFLOW

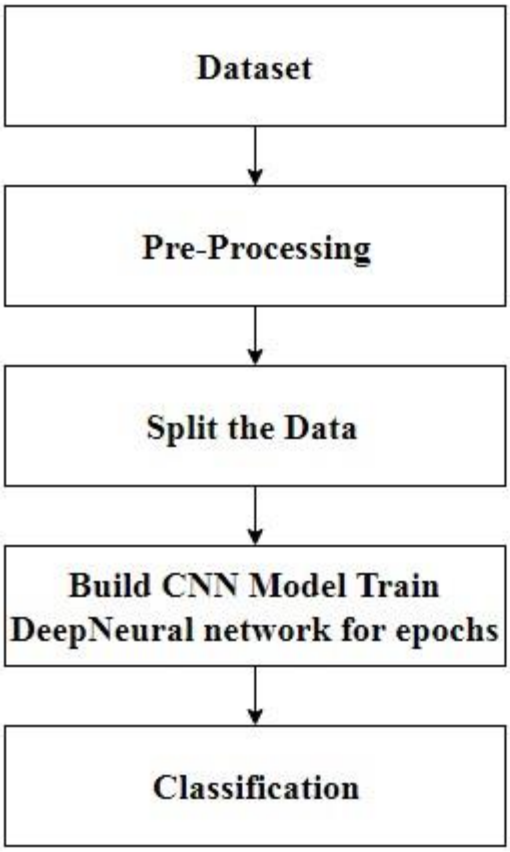

Fig. 7. Proposed work flow of brain tumor detection

The proposed system has mainly five modules. Dataset, Pre-processing, Split the data, Build CNN model train Deep Neural network for epochs, and classification. In dataset we can take multiple MRI images and take one as input image. In pre-processing image to encoded the label and resize the image. In split the data we set the image as 78% Training Data and 22% Testing Data. Then build CNN model train deep neural network for epochs. Then classified the image as yes or no if tumor is positive then it returns yes and the tumor is negative the it returns no.

3.2.1 Working of CNN model

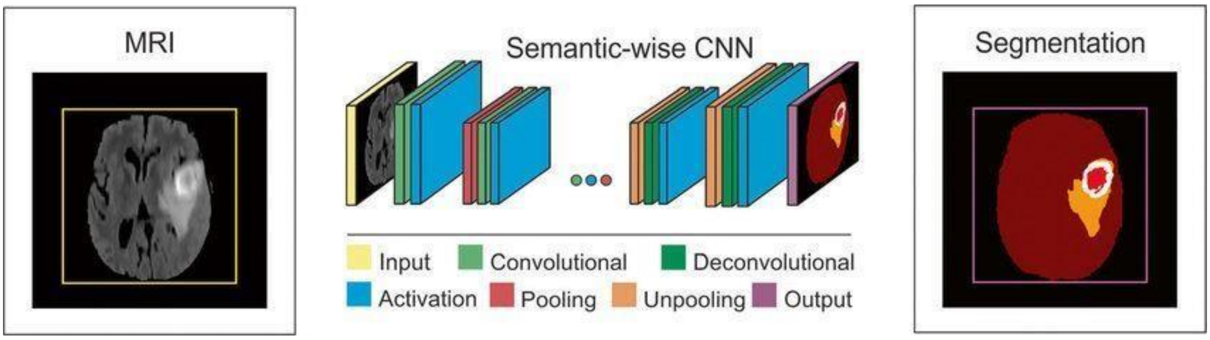

Fig.8.Working of CNN model for brain tumor detection

➢ **Layer of CNN model:** o Convolution 2D o MAX Pooling2D o Dropout o Flatten o Dense o Activation

Convolution 2D	In the Convolution 2D extract the featured from input image. It given the output in matrix form.
MAX Pooling 2D	In the MAX polling 2D it takes the largest element from rectified feature map.
Dropout	Dropout is randomly selected neurons are ignored during training.
Flatten	Flatten feed output into fully connected layer. It gives data in list form.
Dense	A Linear operation in which every input is connected to every output by weight. It followed by nonlinear activation function.
Activation	It used Sigmoid function and predict the probability 0 and 1.

- COMPILE MODEL: We used binary cross entropy because of two layers 0 and 1.
- OPTIMER: Adam optimizer in compile model.

  Adam: - Adaptive moment estimation. It used for non-convex optimization problem**.**
  - Computationally efficient.
  - Little memory requirement.

All the above layers are included in the mobilenet_v2.

3.2.2 Working of mobilenet_v2 model

Description

MobileNet-v2 is a convolutional neural network that is 53 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

Syntax

```
net = mobilenetv2
net = mobilenetv2('Weights','imagenet')
lgraph = mobilenetv2('Weights','none')
```

.

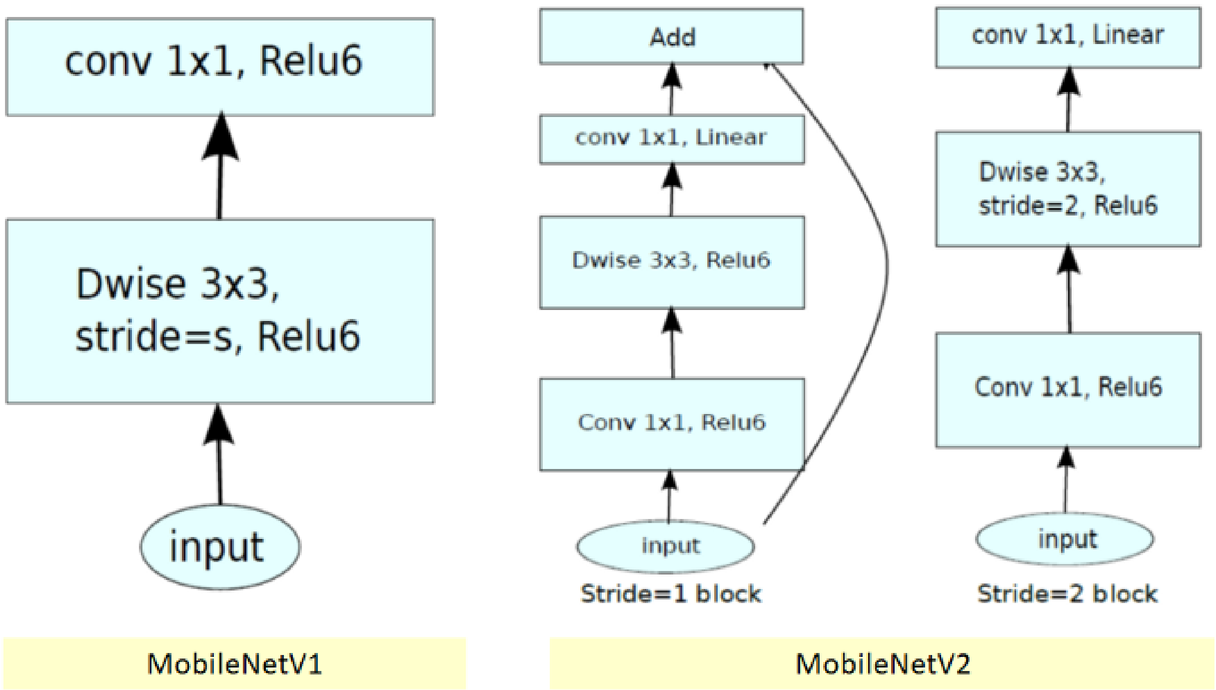

Fig.9. Mobilenet_v1 and Mobilenet_v2 architecture.

4. DATASET, IMPLEMENTATION AND RESULT

4.1: DATASET DETAIL

The training dataset has 3000 images with two classes of images belonging to 'no' and 'yes' i.e., images having brain MRI scans having no brain tumor and images having brain MRI scans having brain tumor. As well as, the Testing dataset has 253 MRI scans with binary classification.

1. **TRAINING DATASET:** https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection

2. **TESTING DATASET:** https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

4.2: TOOLS & TECHNOLOGY USED

- **Python:** Python was the language of selection for this project. This was a straightforward call for many reasons.
  1. Python as a language has a vast community behind it. Any problems which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on the positioning that makes it very likely there will be straight answer to any question
  2. Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas and SciPy area unit freely available and well documented. Packages like these will dramatically scale back, and change the code required to write a given program.This makes iteration fast.
  3. Python as a language is forgiving and permits for program that appear as if pseudo code. This can be helpful once pseudo code given in tutorial papers must be enforced and tested. Using python this step is sometimes fairly trivial. However, Python is not without its errors. The language is dynamically written and packages are area unit infamous for Duck writing. This may be frustrating once a package technique returns one thing that, for instance, looks like an array instead of being an actual array. Plus the actual fact that standard Python documentation does not clearly state the return type of a method, this can lead to a lot of trials and error testing that will not otherwise happen in a powerfully

written language. This is a problem that produces learning to use a replacement Python package or library more difficult than it otherwise may be.

- **Jupiter Notebook:** The Jupyter Notebook is an open-source web application that enables you to make and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.
- **Ngrok:** Ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed. Similar functionality can be achieved with Reverse SSH Tunneling, but this requires more setup as well as hosting of your own remote server.

  Ngrok is able to bypass NAT Mapping and firewall restrictions by creating a long-lived TCP tunnel from a randomly generated subdomain on ngrok.com (e.g., 3gf892ks.ngrok.com) to the local machine. After specifying the port that your web server listens on, the ngrok client program initiates a secure connection to the ngrok server and then anyone can make requests to your local server with the unique ngrok tunnel address.
- **Streamlit :** Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc. With Streamlit, no callbacks are needed since widgets are treated as variables. Data caching simplifies and speeds up computation pipelines. Streamlit watches for changes on updates of the linked Git repository and the application will be deployed automatically in the shared link.

- **Give the Label of Image:**

DATA PREPROCESSING

```
# GETING DATASET
gdrive_path = './dataset/'
# DATA SEGMENTATION
# PREPROCESSING THE TRAINING SET
datagen = ImageDataGenerator(rescale=1./255,validation_split=0.22)
training_set = datagen.flow_from_directory(directory=gdrive_path,
                                           target_size=(224,224),
                                           color_mode="rgb",
                                           subset="training",
                                           class_mode="binary",
                                           batch_size=32,
                                           shuffle=True)
```

```
Found 2340 images belonging to 2 classes.
```

```
valid_generator=datagen.flow_from_directory( directory=gdrive_path,
                                             target_size=(224,224),
                                             color_mode="rgb",
                                             subset="validation",
                                             class_mode="binary",
                                             batch_size=32,
                                             shuffle=True)
```

```
Found 660 images belonging to 2 classes.
```

- **Train Data**

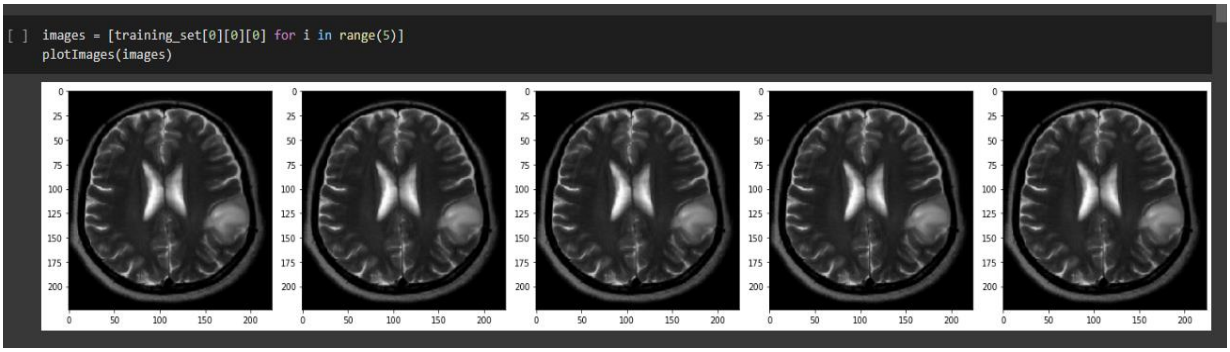

Fig.14.Test CNN image Data

- **Test Data**

GETTING THE TESTING DATASET

```
path = './archive/'
datagen = ImageDataGenerator(rescale=1./255,validation_split=0.0)
testing_set = datagen.flow_from_directory(directory=path,
                                          target_size=(224,224),
                                          color_mode="rgb",
                                          subset="training",
                                          class_mode="binary",
                                          batch_size=32,
                                          shuffle=True)
```

```
Found 253 images belonging to 2 classes.
```

Fig.16.Test image Data

Fig 16. Consist output of testing accuracy score 98.8%

- **Implementation: CNN model summary**

```
[ ]  model = tf.keras.models.Sequential([
         mobile_net,
         ### ann layer
         tf.keras.layers.Dense(1, activation='sigmoid') #[0, 1] or [1, 0]
         ])
```

```
[ ]  model.summary()

     Model: "sequential"
     _________________________________________________________________
      Layer (type)                Output Shape              Param #
     =================================================================
      keras_layer (KerasLayer)    (None, 1280)              2257984

      dense (Dense)               (None, 1)                 1281

     =================================================================
     Total params: 2,259,265
     Trainable params: 1,281
     Non-trainable params: 2,257,984
     _________________________________________________________________
```

Mobile Net Model Summary

IMPLEMENTATION: mobilenet_v2 Training

```
history = model.fit(training_set, epochs=45, validation_data = valid_generator)

Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/45
74/74 [==============================] - 88s 1s/step - loss: 0.4778 - accuracy: 0.7842 - val_loss: 0.3430 - val_accuracy: 0.8712
Epoch 2/45
74/74 [==============================] - 84s 1s/step - loss: 0.2970 - accuracy: 0.9000 - val_loss: 0.3016 - val_accuracy: 0.8606
Epoch 3/45
74/74 [==============================] - 84s 1s/step - loss: 0.2375 - accuracy: 0.9256 - val_loss: 0.2456 - val_accuracy: 0.9227
Epoch 4/45
74/74 [==============================] - 84s 1s/step - loss: 0.2057 - accuracy: 0.9363 - val_loss: 0.2204 - val_accuracy: 0.9167
Epoch 5/45
74/74 [==============================] - 85s 1s/step - loss: 0.1801 - accuracy: 0.9457 - val_loss: 0.2028 - val_accuracy: 0.9242
Epoch 6/45
74/74 [==============================] - 85s 1s/step - loss: 0.1645 - accuracy: 0.9551 - val_loss: 0.1881 - val_accuracy: 0.9273
Epoch 7/45
74/74 [==============================] - 85s 1s/step - loss: 0.1517 - accuracy: 0.9526 - val_loss: 0.1804 - val_accuracy: 0.9303
Epoch 8/45
74/74 [==============================] - 85s 1s/step - loss: 0.1414 - accuracy: 0.9573 - val_loss: 0.1655 - val_accuracy: 0.9394
Epoch 9/45
74/74 [==============================] - 84s 1s/step - loss: 0.1318 - accuracy: 0.9641 - val_loss: 0.1682 - val_accuracy: 0.9379
Epoch 10/45
74/74 [==============================] - 85s 1s/step - loss: 0.1237 - accuracy: 0.9679 - val_loss: 0.1529 - val_accuracy: 0.9424
Epoch 11/45
74/74 [==============================] - 84s 1s/step - loss: 0.1172 - accuracy: 0.9705 - val_loss: 0.1566 - val_accuracy: 0.9394
Epoch 12/45
74/74 [==============================] - 83s 1s/step - loss: 0.1113 - accuracy: 0.9714 - val_loss: 0.1327 - val_accuracy: 0.9561
Epoch 13/45
...
Epoch 44/45
74/74 [==============================] - 85s 1s/step - loss: 0.0376 - accuracy: 0.9927 - val_loss: 0.0742 - val_accuracy: 0.9773
Epoch 45/45
74/74 [==============================] - 85s 1s/step - loss: 0.0378 - accuracy: 0.9919 - val_loss: 0.0719 - val_accuracy: 0.9788
```

Mobile Net Model Training

• IMPLEMENTATION: Streamlit app creation

```
[ ] %%writefile app.py
    import streamlit as st
    import tensorflow as tf
    import streamlit as st
    import tensorflow_hub as hub

    #URL = "https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4"
    #mobile_net = hub.KerasLayer(URL, input_shape=(IMG_SHAPE, IMG_SHAPE, 3))


    @st.cache(allow_output_mutation=True)
    def load_model():
      #IMG_SHAPE = 224
      #mobile_net = hub.KerasLayer(URL, input_shape=(IMG_SHAPE, IMG_SHAPE, 3))
      #mobile_net.trainable = False
      #model= tf.keras.models.Sequential([
      #  mobile_net,
        ### ann layer
      #  tf.keras.layers.Dense(1, activation='sigmoid') #[0, 1] or [1, 0]
      #  ])
      model=tf.keras.models.load_model('/content/my_model.hdf5',custom_objects={'KerasLayer':hub.KerasLayer})
      return model
    with st.spinner('Model is being loaded..'):
      model=load_model()

    st.write("""
             # Brain Tumor Detection
```

IMPLEMENTATION: Deployment of streamlit app using Ngrok

```
!streamlit run /content/app.py & npx localtunnel --p 8501

[.................] | fetchMetadata: sill resolveWithNewModule localtunnel@2.0
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.


  You can now view your Streamlit app in your browser.

  Network URL: http://172.28.0.12:8501
  External URL: http://35.204.191.165:8501

npx: installed 22 in 5.858s
your url is: https://gold-eggs-smash-35-204-191-165.loca.lt
2022-12-18 05:57:03.164093: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2022-12-18 05:57:03.173 Using /tmp/tfhub_modules to cache modules.
1/1 [==============================] - 0s 492ms/step
1/1 [==============================] - 0s 324ms/step
1/1 [==============================] - 0s 317ms/step
1/1 [==============================] - 0s 328ms/step
1/1 [==============================] - 0s 307ms/step
  Stopping...
^C
```

5. CONCLUSION

In brain tumor detection we have studied about feature based existing work. In feature based we have study about image processing techniques likes image pre-processing, image segmentation, features extraction, classification. And also study about deep learning techniques CNN and mobilenet_v2. In this system we have detect the tumor is present or not if the tumor is present then model return's yes otherwise it returns no. and we have constructed CNN with the mobilenet_v2 Model. The accuracy our model achievers is over 85 %. However, not every task is said to be perfect in this development field even more improvement may be possible in this application. We have learned so many things and gained a lot of knowledge about neural network development field.

References

[1] L.Guo,L.Zhao,Y.Wu,Y.Li,G.Xu,andQ.Yan, “Tumor detection in MR images using oneclass immune feature weighted SVMs,” IEEE Transactions on Magnetics, vol. 47, no. 10, pp. 3849–3852,2011.

[2] R.Kumari,“SVMclassificationanapproachondetectingabnormalityinbrainMRIimages,”Inter nationalJournalofEngineeringResearchandApplications,vol.3,pp.1686–1690,2013.

[3] DICOM Samples Image Sets, http://www.osirix-viewer.com/.

[4] Brainweb: http://brainweb.bic.mni.mcgill.ca/cgi/brainweb1.

[5] Obtainable Online: *www.cancer.ca/~/media/CCE* 10/08/2015.

[6] J. C. Buckner, P. D. Brown, B. P. O’Neill, F. B. Meyer , C. J. Wetmore,J. H Uhm, "Central nervous system tumors." In *Mayo Clinic Proceedings*,Vol. 82, No. 10, pp. 12711286, October 2007.

[7] Deepa , Singh Akansha. (2016). - Review of Brain Tumor Detection from tomography. *International Conference on Computing for Sustainable Global Development (INDIACom)*

[8] R. A. Novellines, M. D. - *Squire's fundamentals of radiology*; Six Edition; UPR, 2004.

[9] Preston, D. c. (2006). Magnetic Resonance Imaging (MRI) of the Brain and Spine from Basics. *casemed.case.edu* .

[10] Hendrik RE. (2005) Glossary of MR Terms from *American College of Radiology* .

.

[11] A. Demirhan, M. Toru, and I. Guler, “Segmentation of tumor and edema along with healthy tissues of brain using wavelets and neural networks,” IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 4, pp. 1451–1458, 2015.

[12] Nilesh Bhaskarrao Bahadure, A.K. (2017, March 6). Retrieved from https://www.hindawi.com/journals/ijbi/2017/9749108/.

[13] S. Mohsin, S. Sajjad, Z. Malik, and A. H. Abdullah, “Efficient way of skull stripping in MRI to detect brain tumor by applying morphological operations, after detection of false background,” *International Journal of Information and Education Technology*, vol. 2, no. 4, pp. 335–337, 2012.

[14] Gavale, P. M., Aher, P. V., & Wani, D. V. (2017, April 4). Retrieved from https://www.irjet.net/archives/V4/i4/IRJET-V4I462.pdf.

[15] N. Gordillo, E. Montseny, and P. Sobrevilla, “State of the art survey on MRI brain tumor segmentation,”*Magnetic Resonance Imaging*, vol. 31, no. 8, pp. 1426–1438, 2013.

[16] Samantaray, M.(2016, November 3). Retrieved from http://ieeexplore.ieee.org/document/7727089/ .

[17] Nandi, A. (2016, April 11) Retrieved from http://ieeexplore.ieee.org/document/7449892/

[18] C. C. Benson and V. L. Lajish, “Morphology based enhancement and skull stripping of MRI brain images,” in *Proceedings of the international Conference on Intelligent Computing Applications (ICICA ’14)*, pp. 254–257, Tamilnadu, India, March 2014.

[19] S. Z. Oo and A. S. Khaing, “Brain tumor detection and segmentation using watershed segmentation and morphological operation,” *International Journal of Research in Engineering and Technology*, vol. 3, no. 3, pp. 367–374, 2014.

[20] R. Roslan, N. Jamil, and R. Mahmud, “Skull stripping magnetic resonance images brain images: region growing versus mathematical morphology,” *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 150– 158, 2011.

APPENDIX A ABBRAVIATION

Sr. No.	ABBREVIATION	MEANING
1	CNN	Convolutional neural network
2	MRI	Magnetic resonance imaging
3	FLAIR	Fluid attenuated in version recovery weighted MRI
4	TR	Time repetition
5	TE	Pulse sequence parameter
6	VGG 16	Visual Geometry Group
7	FC	Fully connected layer
8	ReLU	Rectified linear unit
9	LRN	Local response normalization
10	SVM	Support vector machine
11	KMM	K nearest neighbor