DETECTION OF BRAIN TUMOR FROM MRI SCAN

Project Report

Raghav Agrawal (19BDS0170)
Shrishail Ravi Terni (19BCE0445)
Singamsetty Aditya sai (19BCI0077)

Brief Description about the project :-

- **Abstract :-** According to the National Brain Tumor Society today, an estimated 700,000 people in the United States are living with a primary brain tumor, and approximately 88,970 more will be diagnosed in 2022. Brain tumors can be deadly, significantly impact quality of life, and change everything for a patient and their loved ones. They do not discriminate, inflicting men, women, and children of all races and ethnicities.

- **Objective and Methodology :-**To create a model which takes MRI scans of the brain as input and can accurately predict whether the corresponding person has a brain tumor or not. The model will be a novel approach in which we will try to enhance the accuracy scores using both Machine learning and Deep Learning. A detailed comparison/analysis will be done.After rigorous research from numerous research papers, reliable websites we sum up to a conclusion that CNN is the best performing model on image data**.**

- **Problem Definition :-** Brain Tumor in an early stage called Grade I. These tumors are slow growing and unlikely to spread. They can often be cured with surgery. So, detecting them at an early stage is a crucial task. We took this crucial task as our problem statement and will fit a model which can predict whether the tumor is malignant or benign with high accuracy by utilizing Deep learning**.**

- **Objective :-**
  1. To create a model which takes MRI scans of the brain as input and can accurately predict whether the corresponding person has a brain tumor or not.
  2. The model will be a novel approach in which we will try to enhance the accuracy scores using both Machine learning and Deep Learning.
  3. A detailed comparison/analysis will be done.

Motivation :-

Brain Tumor is one of the deadliest diseases in the world. It kills around 0.3 million people every year. A person diagnosed with Brain Tumor if detected in the early stage can live for a considerably longer period. Only 12% of the patients suffering with Brain Tumor are diagnosed.

The Brain Tumor Foundation estimates the cost of treating a glioblastoma at more than $450,000 and says costs of treating a brain tumor can reach $700,000 in a lifetime. Early stage brain tumor detection not only ensures early recovery but also low cost to treat these tumors.

Novelty : -

- We are proposing to develop a novel Ensemble model which will compare with existing deep learning models.
- We have prepared a comparative analysis of all the models used like VGG, CNN, and rest all machine learning modes.
- We are marking the area in the uploaded image where the tumor is supposed to be found for the convenience of the user.

Literature review:-

Article	**Publication Details**	**Methodology Adopted**	**Dataset Used**	**Limitations**
Amjad et al. [1]	Wiley Publication	The clustering-based approach is used in MR tumor segmentation as it splits pixels into groups based on the same group’s highest similarity, that is, categorizing the pixels using certain features. In the proposed method, the concept of transfer	BraTS 2015 benchmark data set is used for the detection. However, the synthetic data augmentation technique is also	The algorithm focuses on random rotation of the MRI scans which

		learning is utilized in the VGG19 CNN model. The use of fine tuned CNN helps to minimize training time and increase predictive accuracy. The weights of the layers of the CNN model VGG19 classifies the tumors in two classes. In VGG19, the first and second convolution layers of the group of two layers are followed by max pooling layers, and the next eight convolution layers are a group of four layers followed by max-pooling with the same filter size. The last FC8 layer is passed as a features vector with 1,000 dimensions to the SoftMax function to predict brain tumor class. In the proposed technique, we incorporated VGG19, since in this model the filter size is minimum, that is, 3 × 3, which were same for all the convolution layers and with stride 1, while in other CNNmodels, filter size is 11 × 11 or 7 × 7 with 4–5 strides. The synthetic data	programmed to escalate the size of the labeled data set. Experiments are conducted with and without data augmentation and results are equated with each other and previously available benchmarks. https://www.smir.ch/BRATS/Start2015	can sometimes give some false positives and false negatives.

		augmentation techniques used in the proposed technique are; random rotation, zoom, geometric transformations, sharpening, noise addition and contrast.		
Javeria et al. [2]	Future Generation Computer Systems	The method used in the paper is a general Machine Learning process in which a benchmarked dataset is used for the training as well as testing of the model. In the process of data preprocessing, image splits are done in order to reach a performance measure stage of the trained CNN/DNN model. The DNN method is an NXN 2D patch with all MR modalities. The first DNN model is created using convolution layers which is termed as CNN model in general. In the model there are many layers stacked on top of each other. The next layers take the inputs from the previous layer. Images planes are supplied as an input stack and the output is a feature map. Finally, the	The research takes the MRI (Magnetic Resonance Images) of the brain of the BRATS (2012-2015) benchmarked dataset which can be classified as having a tumor or not. Also, ISLES (2015,2017) dataset. BRATS 2012 MRI dataset contains multimodal images total 80 input subjects are used.25 LGG (Low Grade Glioma) and 25 HGG (High Grade	The models achieved fairly well accurate results when compared to the standard deep learning models for brain tumor detection, but the computational time can be brought under 1 seconds so that the process can be speeded

		prediction is performed based on testing and training labels. In the intermediate steps normalization is done to assign label position to each pixel of the image	Glioma) are used in the training phase and 20 LGG and 10 HGG in the testing phase. In BRATS 2014 dataset 300 subjects were used as 200 for training and 100 for testing. ISLES 2017 dataset contains 75 input subjects, 43 training and 32 stroke testing subjects. https://www2.imm.dtu.dk/projects/BRATS20 12/	up and achieve better performance
Divyama ry et al. [3]	International Conference on Advanced Computing and Communication Systems	A 4-step process is used including image pre-processing, Morphological based segmentation, Feature extraction, Naive classifier fitting. In pre-processing step tasks like filtering, noise removal and removing	The BraTS 2018 dataset is used in this paper. It is a dataset which provides brain MRIs annotated by physicians. Annotations	Naive Bayes assumes that all predictors (or features) are independent so

	(ICACCS)	unnecessary part of images are performed. In segmentation the foreground of the image is segmented from background. In the feature extraction step Legendary level sets algorithm is used to extract features such as energy, brightness, intensity, homogeneity etc and discretization of these features was performed. Finally, a Naive Bayes model is fitted to the training data.	include whole tumor (WT), tumor core (TC), and enhancing tumor (ET). The data were collected from 19 institutions, using various MRI scanners. https://www.med.upenn.edu/sbia/brats2018/data.htm	sometimes it produces erroneous results. This algorithm faces the ‘zero frequency problem’ where it assigns zero probability to a categorical variable whose category in the test data set was not available in the training dataset
Faisal et al. [4]	International conference on advances in science, engineering, and robotics technology	Two different models for segmentation and detection of brain tumor: -> First model segmented the tumor by FCM and classified by traditional machine learning algorithms ->	BRATS Dataset https://www.smir.ch/BRATS/Start2015	*Segmentation by FCM takes more execution time. * Using

	(ICASERT)	the second model focused on deep learning for tumor detection. * A Five-Layer Convolutional Neural Network is introduced and implemented for tumor detection.		sigmoid function is complex
Ari, Hanbay [5]	Journal of Electrical Engineering and Computer Sciences	* Pre-processing: 1) nonlocal means and local smoothing methods 2) denoising and normalization operations were employed in order to prepare the input images for the next stage. * Processes: -> Watershed segmentation: segmentation of cranial MRI'S -> Morphological process: opening & closing * Convolution and pooling operations were applied to the images in the input layer. * ELM-LRF was used to classify the brain tumors as benign or malignant	DICOM Dataset http://www.osirixviewer.com/datasets	Watershed segmentation is extremely sensitive to gradient noise and may lead to over segmentation.
Sujihelen et al. [6]	international conference on trends in electronics and informatics (ICOEI)	*pre-processing: data cleaning, data transformation, data integration, data resizing, data reduction, normalization, and aggregation *Bilateral and average filtering processes are used * Pixel based segmentation	UCI Brain Tumor DataSet Link: https://archive.ics.uci.edu/ml/datasets/primary+tumor	*CNN does not encode the position and orientation of objects. *CNN

		is used * Classification of brain tumor is done by using CNN		requires lots of training data
Sharif et al. [7]	Cognitive Systems Research 59	*Segmentation: Grab cut algorithm is used to segment glioma using MRI. *Segmented images are supplied to the proposed 7 deep learning model (VGG19 model) and the deep features vector is obtained *Hand crafted (shape and texture) features and deep features are concatenated serially to classify the gliomas and healthy images	BRATS 2015-2017 DataSet Link:	*Used only to classify primary glioma tumors
Amin et al. [8]	Computer Methods and Programs in Biomedicine	Three automated methods are proposed to enhance the performance, Lesion Enhancement, Lesion Segmentation, Feature extraction. Then Decision Tree, Random Forest, SVM (Quadratic kernel function) and KNN.	The BRATS 2013, 2015 datasets are used in this paper. BRATS 2013 challenge dataset consists of thirty cases with ground truth annotations in which 20 belong to HG and 10 to LG tumors. BRATS 2015	The work can be stretched to deep learning approaches for segmentation of sub tumoral regions including complete, core and enhanced

			has 273 cases in which 54 LG and 220 HG gliomas are included. https://www.smir.ch/BRATS/Start2013 https://www.smir.ch/BRATS/Start2015	tumor.
Islam et al. [9]	Machine Learning with Applications (MLWA)	The input data is firstly sent for pre-processing, in this input data is converted in grayscale and removed additional noises using a filter (median filter). After that the data features are extracted in two ways one using normal pixels and another by perceptual grouping of pixels considering the similarity in close proximity called super pixels. And the approach used for feature extraction is SaPSO. And for feature extraction from superpixels SLIC is which has a fast computation time of feature extraction. After that using PCA reduces	To create the dataset, they have taken some critical images from kaggle.com open access data, and some images are taken locally. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor detection	The drawback of the paper is that it is implemented on a small dataset and does not use the standard classification system which is used with real clinical data. And its computation time increases

		the dimensionality of the data. After this the data is sent to the segregation layer where this is done to approximately locate the boundaries of objects for segmenting in a way that all the segments collectively form the whole image. After that this data is sent to a template-based Kmeans clustering algorithm instead of a K-means algorithm. In this model template is selected, based on segmentation that provides a better and robust result. This is the whole methodology used in this paper		as the data increases
Rehman et al. [10]	Wiley Publication	In this paper, a 3D convolutional neural network (CNN) architecture is designed at the first step to extract brain tumors and extracted tumors are passed to a pre-trained CNN model for feature extraction. The extracted features are transferred to the correlation based selection method and as the output, the best features are selected. These selected features are validated through a	In this paper, three BraTS datasets 2015, 2017, and 2018 are utilized for experiments, validation. These are the datasets which contain brain MRIs annotated by physicians. https://www.smir.ch/B	Requires high computational power, there is a huge risk of overfitting, exploding gradient and class imbalance has also a

		feedforward neural network for final classification	RATS/Start2015 https://paperswithcode.com/dataset/brats2015-1	major effect on the model performance

Specifications and Requirements:-

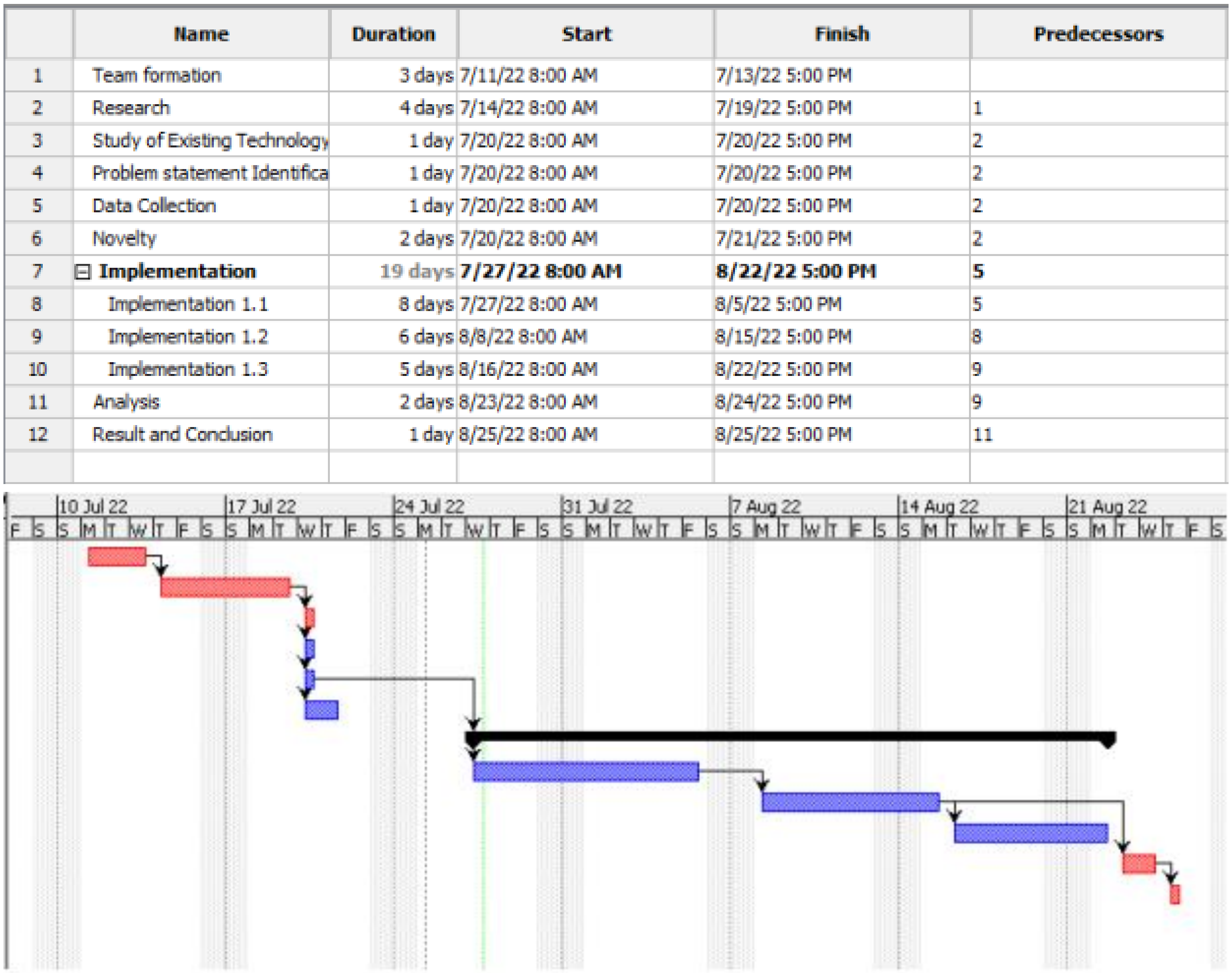
Machine Specifications:-

Table of Content	**Description**
Operating System	Windows 10 and Windows 11 64 bit Operating System
RAM	8GB and 16GB
Graphics	Intel(R), CORE(™), i5 8265 CPU @ 1.60 GHz, 1.8GHz
Editors	Jupyter Notebook, Spyder, Google Collab

Project Outcome:-

We have decided to develop an application which takes an MRI scan as an input and predicts whether the person has Brain Tumor. If the image is detected with a Tumor, we will highlight the portion of the image where the tumor is present.

Gantt Chart:-



	Name	Duration	Start	Finish	Predecessors
1	Team formation	3 days	7/11/22 8:00 AM	7/13/22 5:00 PM	
2	Research	4 days	7/14/22 8:00 AM	7/19/22 5:00 PM	1
3	Study of Existing Technology	1 day	7/20/22 8:00 AM	7/20/22 5:00 PM	2
4	Problem statement Identifica	1 day	7/20/22 8:00 AM	7/20/22 5:00 PM	2
5	Data Collection	1 day	7/20/22 8:00 AM	7/20/22 5:00 PM	2
6	Novelty	2 days	7/20/22 8:00 AM	7/21/22 5:00 PM	2
7	⊟ **Implementation**	19 days	**7/27/22 8:00 AM**	**8/22/22 5:00 PM**	**5**
8	Implementation 1.1	8 days	7/27/22 8:00 AM	8/5/22 5:00 PM	5
9	Implementation 1.2	6 days	8/8/22 8:00 AM	8/15/22 5:00 PM	8
10	Implementation 1.3	5 days	8/16/22 8:00 AM	8/22/22 5:00 PM	9
11	Analysis	2 days	8/23/22 8:00 AM	8/24/22 5:00 PM	9
12	Result and Conclusion	1 day	8/25/22 8:00 AM	8/25/22 5:00 PM	11

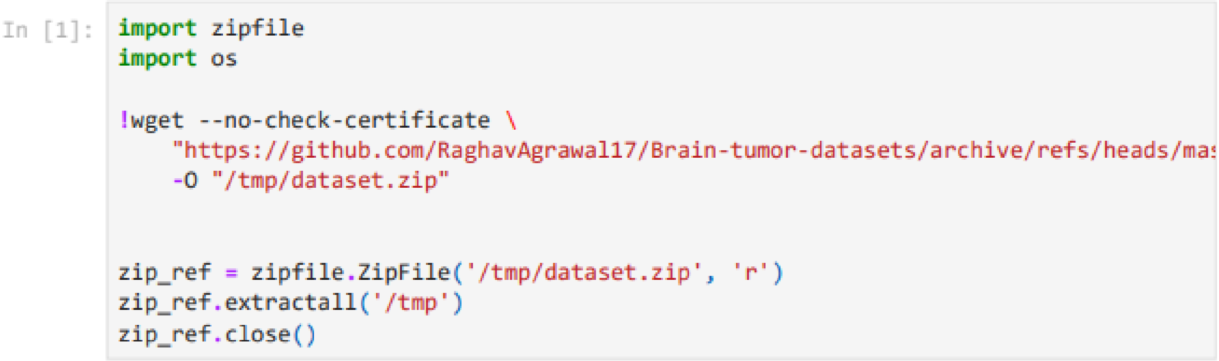
10 Jul 22
17 Jul 22
24 Jul 22
31 Jul 22
7 Aug 22
14 Aug 22
21 Aug 22
F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S

Fig 1:- WorkBreakdown Structure

Methodology Used :-

The proposed approach in our case focuses on gathering the data in order to achieve better output and accuracy scores from the model.We generally follow a series of standard data science steps where we gather the dataset. Then this gathered dataset is splitted in order to get the benign tumors (No Tumor) and malignant tumors(Tumors exists).The data then is passed through a series of Image Processing steps which are required for cleaning/scrubbing the data so that the data we get can be used for both feature extraction and subsequently for model training. This models will be trained by both Machine Learning and Deep Learning techniques so that the better performance and time complexities can be harnessed by the same. The novelty of our method will be the use of both feature extraction as well as the Deep Learning which will be an ensemble with different input parameters. After this step the models will be compared and properly analysed in order to find the best fit. This will be used and compete with the standard models.

a.Dataset Collection :- The dataset used, consists of Brain MRI images which are grouped into folders as either benign or malignant tumors.



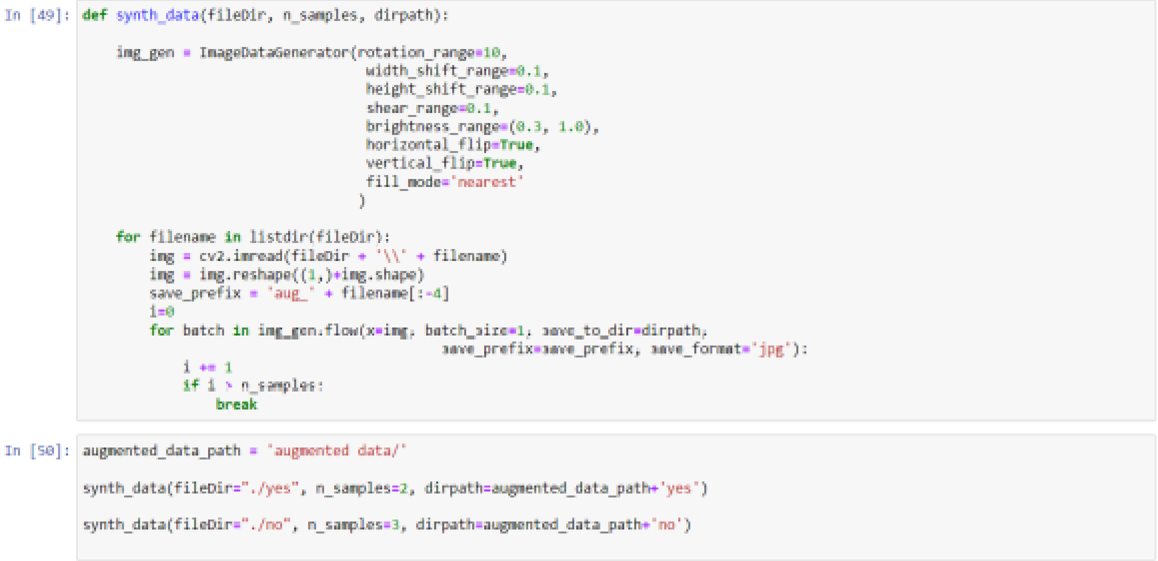
```
In [1]: import zipfile
        import os

        !wget --no-check-certificate \
            "https://github.com/RaghavAgrawal17/Brain-tumor-datasets/archive/refs/heads/mas
            -O "/tmp/dataset.zip"


        zip_ref = zipfile.ZipFile('/tmp/dataset.zip', 'r')
        zip_ref.extractall('/tmp')
        zip_ref.close()
```

Fig 2:- Dataset Directory

b.Data Synthesis :- Since the dataset we are using is not sufficient for effective training of the model, we are synthesizing data in order to create more images. Our approach states that this data must not be modified beyond a certain level of threshold as it will hamper our results. So a permissible level of synthesis is done. Citation :- [1]



```
In [49]: def synth_data(fileDir, n_samples, dirpath):

             img_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(fileDir):
                 img = cv2.imread(fileDir + '\\' + filename)
                 img = img.reshape((1,)+img.shape)
                 save_prefix = 'aug_' + filename[:-4]
                 i=0
                 for batch in img_gen.flow(x=img, batch_size=1, save_to_dir=dirpath,
                                           save_prefix=save_prefix, save_format='jpg'):
                     i += 1
                     if i > n_samples:
                         break
```

```
In [50]: augmented_data_path = 'augmented data/'

         synth_data(fileDir="./yes", n_samples=2, dirpath=augmented_data_path+'yes')

         synth_data(fileDir="./no", n_samples=3, dirpath=augmented_data_path+'no')
```

Fig 3 :- Image Synthesis

c. Validation Data :- The validation data or test data needs to be properly captured in the process such that model overfitting can be avoided as a precautionary step.

d. Data Processing/Image Processing :- In this phase we need to clean or preprocess our data, here the data are the MRI scan images which will be preprocessed with the steps such as Cropping, Resizing, Normalizing, Converting image to numpy array, Converting image data into a dataframe and finally Applying PCA to reduce the dimensions of the dataframe. These steps have many intermediate steps(such as thresholding, eroding, etc) which are used for 2D image processing in order to efficiently train our model.This is a very important step of our system design. Citation :- [2][3]

.1) **Converting image to gray-scale:-**

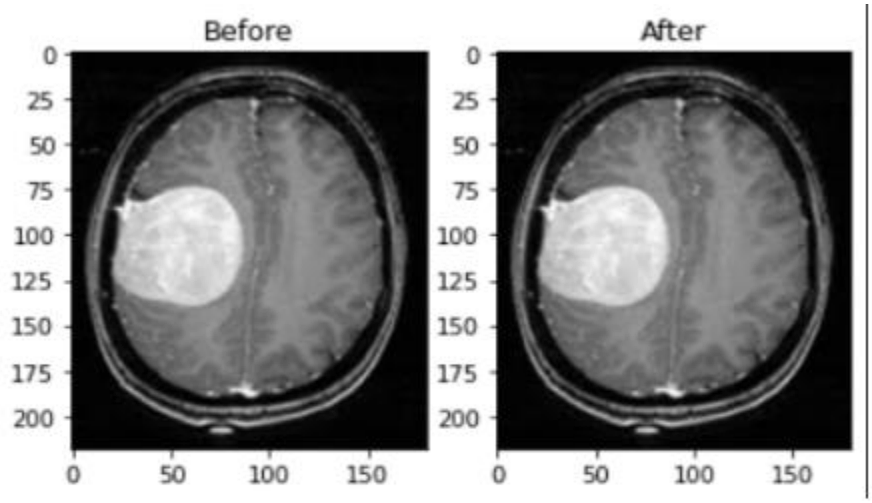


Fig 4:- Grayscaling

As MRI images are black and white converting it to gray-scale does not show much change in appearance

2) **Blurring it slightly :-**

Applying a low-pass blurring filter smooths edges and removes noise from an image

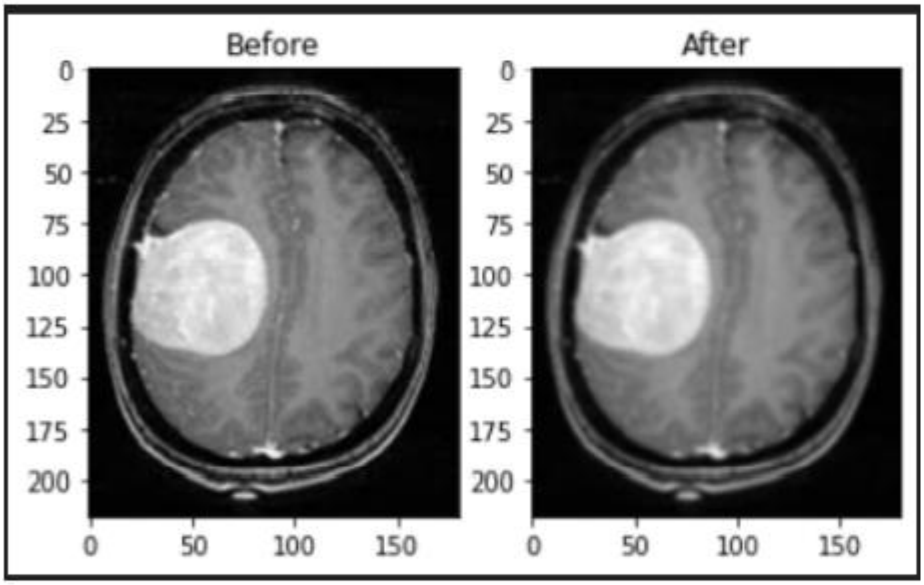


Fig 5:- Blurring

3) **Thresholding:-**

A method of image segmentation which segments foreground from background

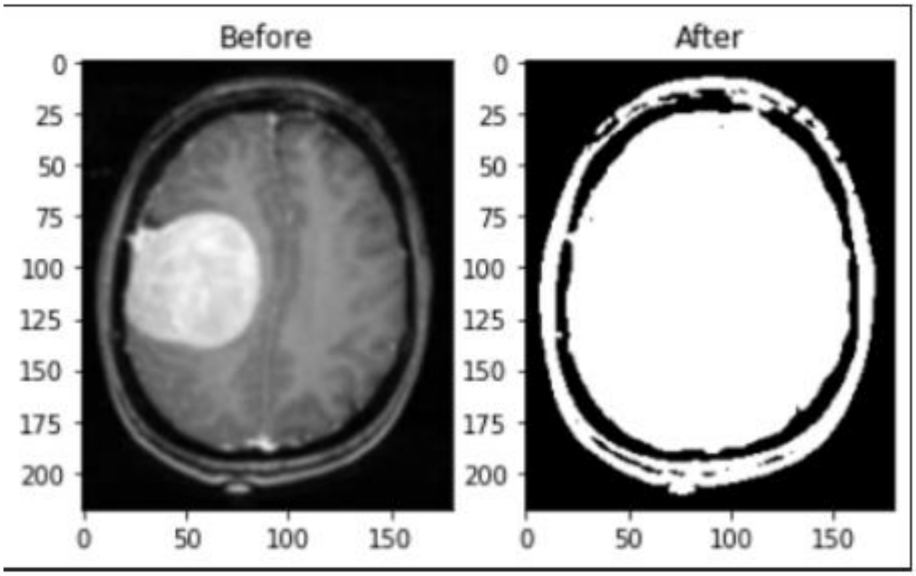


Fig 6:- Thresholding

4) **Eroding :-**

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object

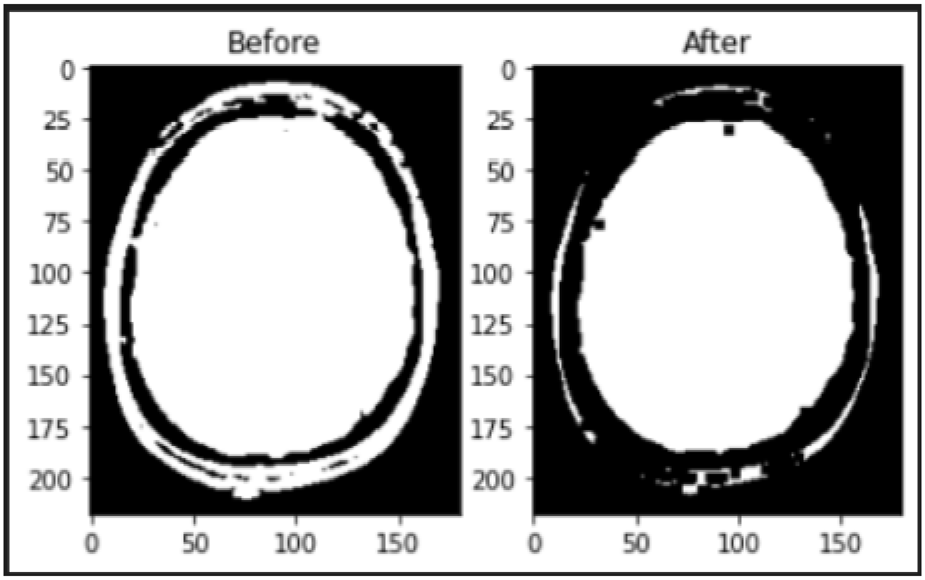


Fig 7:- Eroding

5) **Dilating (Steps 3,4,5 are called noise-reduction steps) :-**
Dilation adds pixels to object boundaries

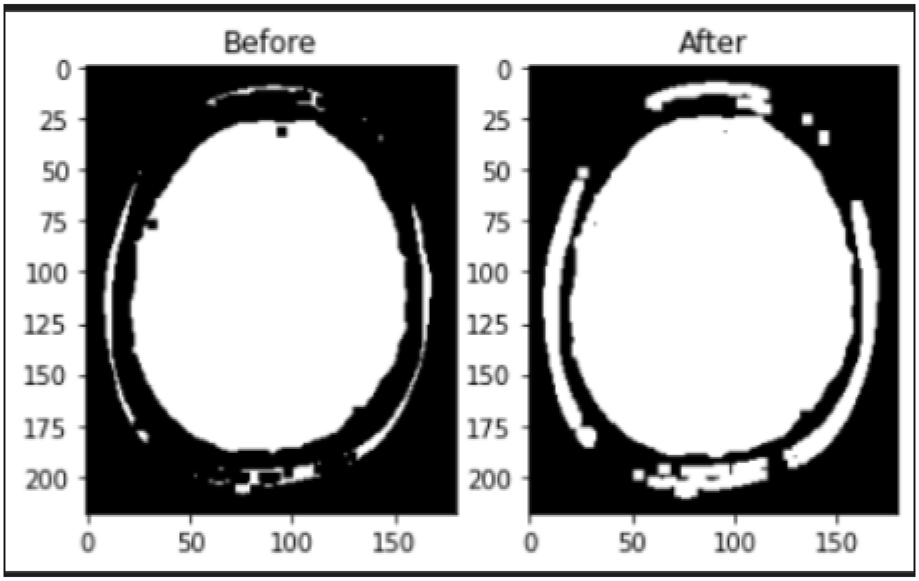


Fig 8:- Dilating

6) **Grabbing largest contours and finding extreme points:-**
These points will be used to extract the required image segment only from the original image

7) **Extracting the image located in these extreme points from the original image :-**

Now we have the image which contains the image of the brain without marginal length or padding

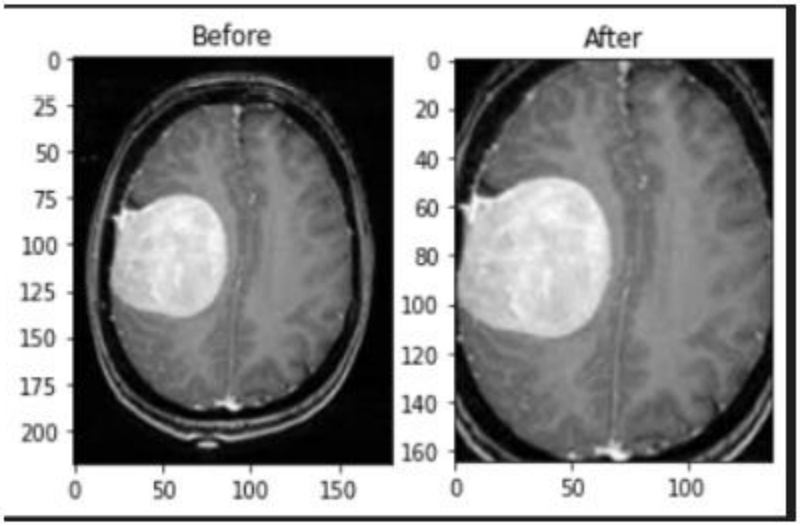


Fig 9:- Extracting

8)**Resizing image:-**

Resize the image (the images in the dataset come in different sizes (in the aspect of width, height and number of channels). So, we want all of our images to be (width=240,height=240,channels=3)

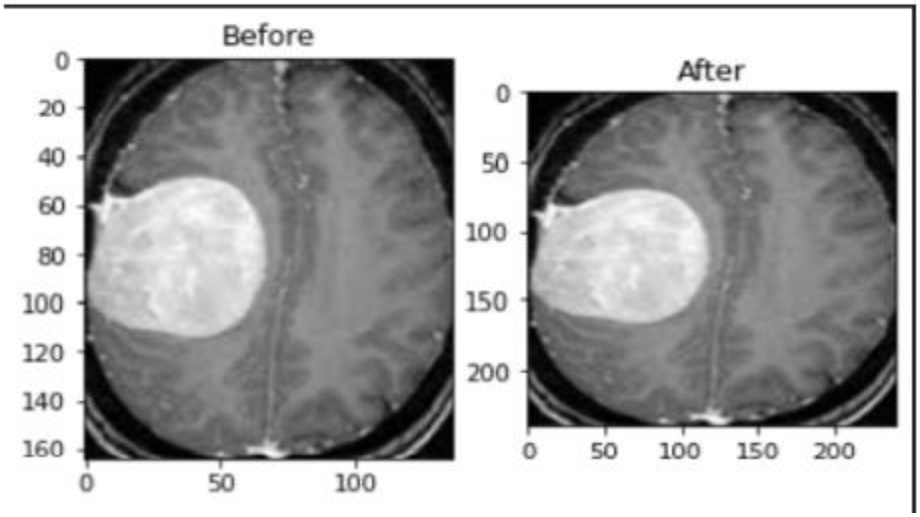


Fig 10:- Resizing

9) **Normalizing image:-**

Here normalization ensures that each input parameter (pixel) has a similar data distribution as Pixels intensity range from 0 to 255, divide them by 255 and it will be normalized to (0,1).

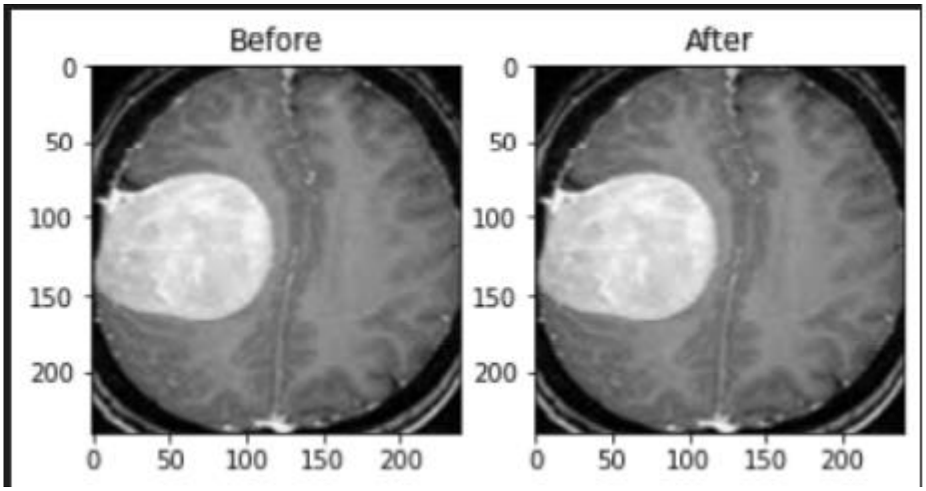
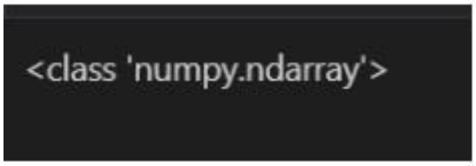


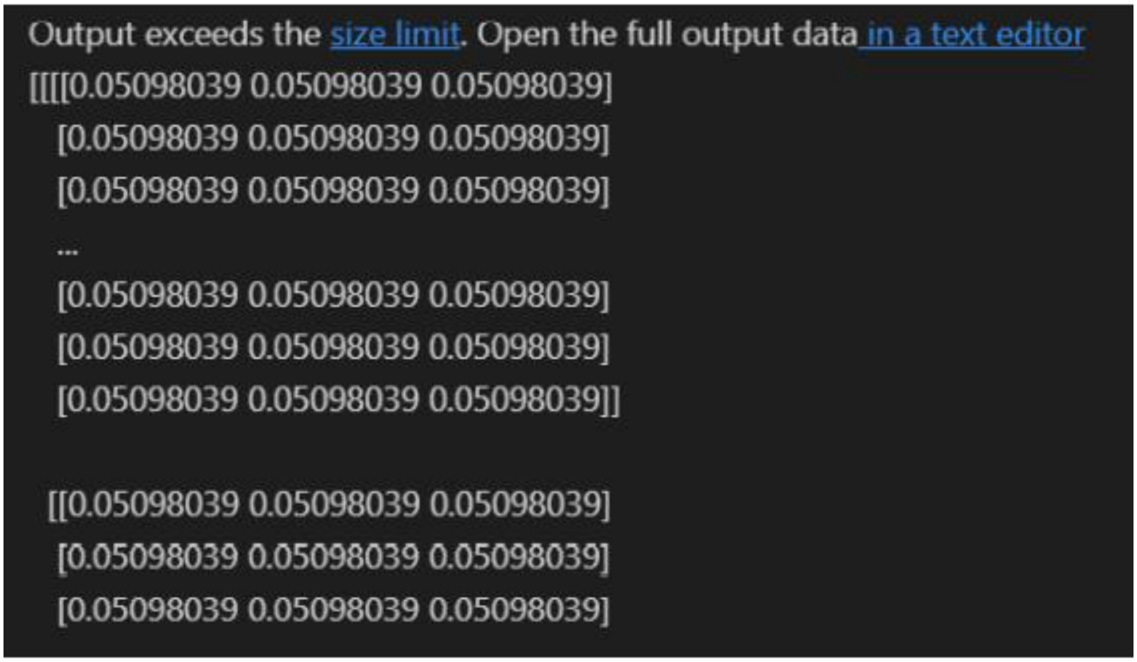
Fig 11:- Normalizing

9)**Converting the image into numpy array:-**

To use the vast functionality of numpy we have converted the image into numpy array



```
<class 'numpy.ndarray'>
```



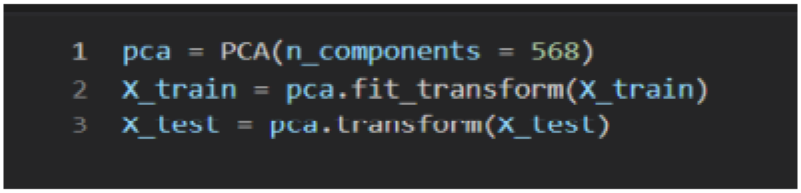
```
Output exceeds the size limit. Open the full output data in a text editor
[[[[0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]
   ...
   [0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]]

  [[0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]
   [0.05098039 0.05098039 0.05098039]
```

Fig 12:- Numpy Array

10)**Applying PCA to reduce the dimensions of dataframe:-**

As the dataframe we have created is high dimensional to process with any machine learning model we reduced its dimension as much as possible using PCA



```
pca = PCA(n_components = 568)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
```

Fig 13:- PCA

e. Balancing Dataset :- Since the number of images may not be having an equal number of labeled images, hence we have to balance the dataset after down sampling the records.

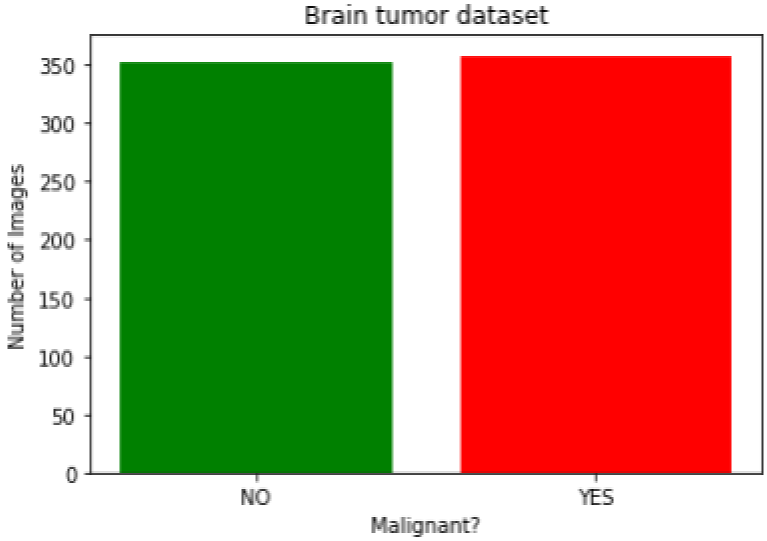
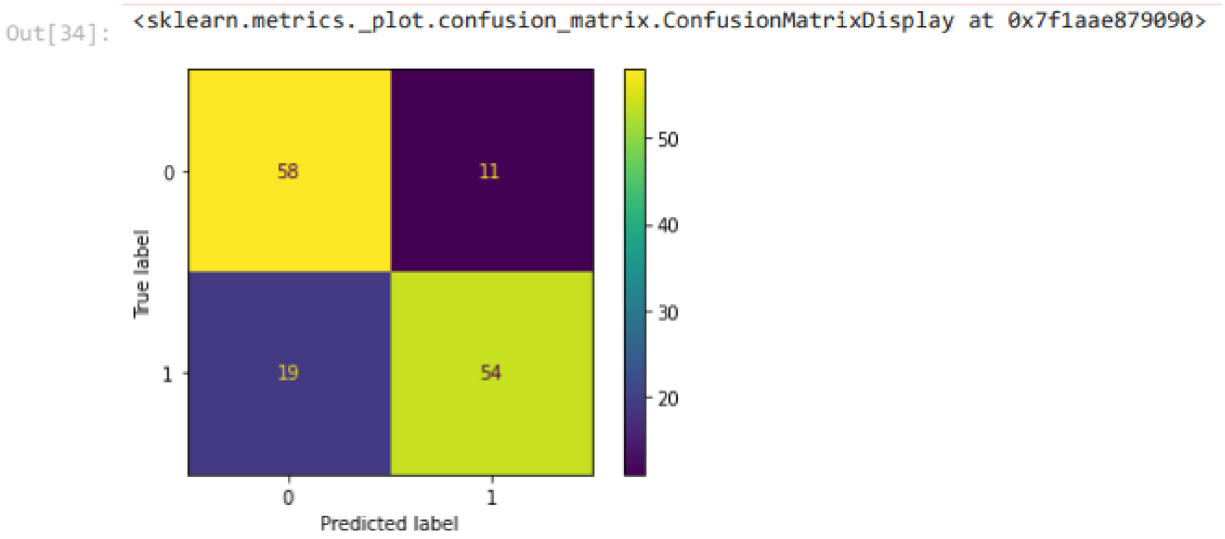


Fig 14:- Dataset Description

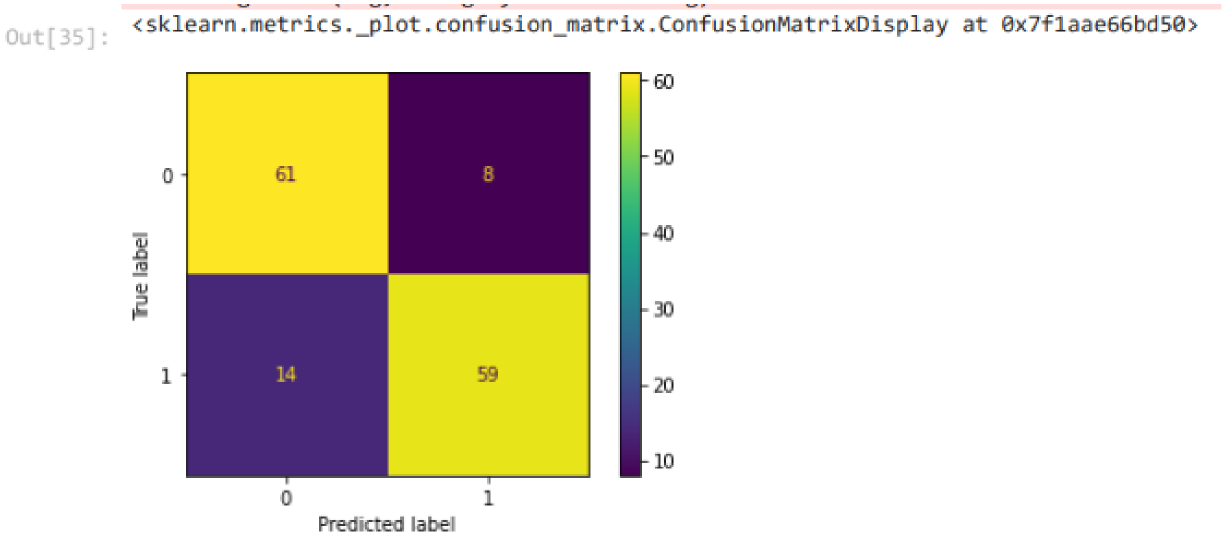
f. Model Creation :- Our data is supervised data and our problem is classification problem so we have utilized following classification models (including both deep learning as well as machine learning models)

- MLP classifier



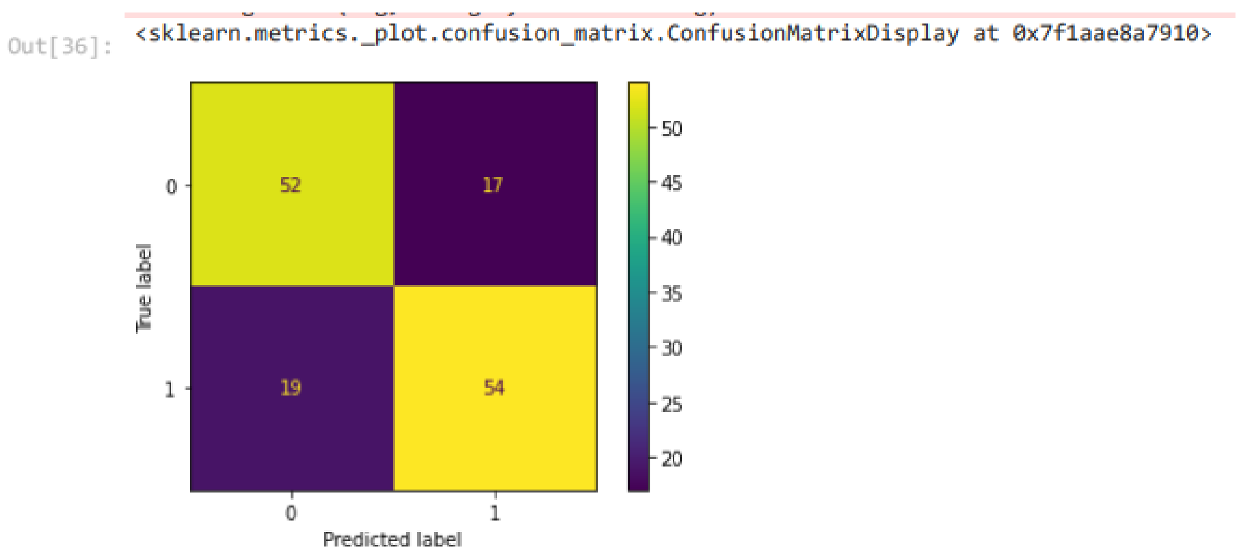
```
Out[34]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aae879090>
```

- SVM classifier



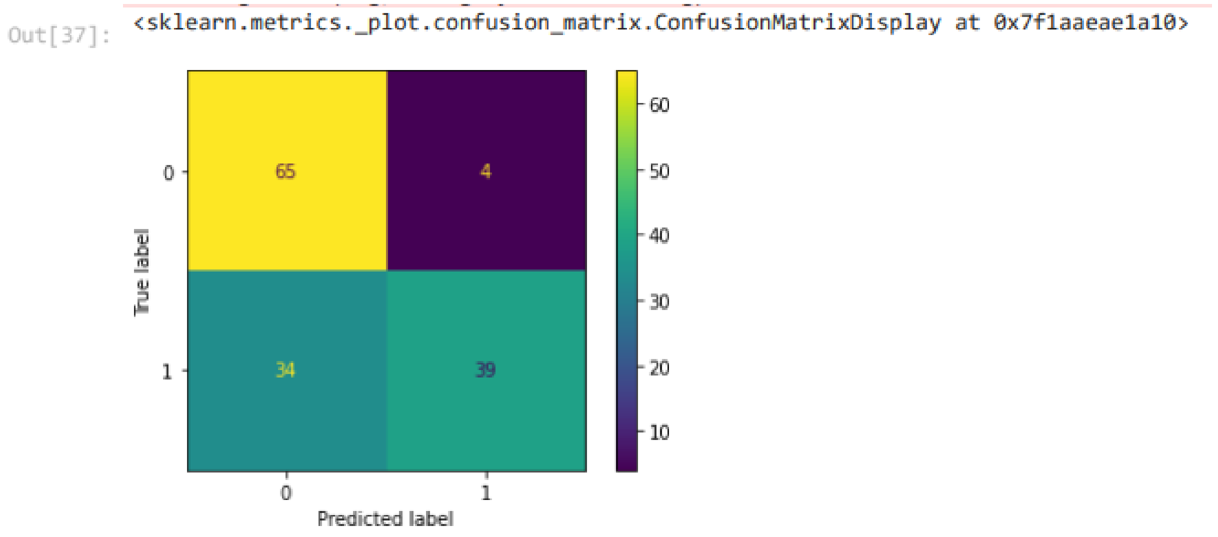
```
Out[35]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aae66bd50>
```

- Decision Tree



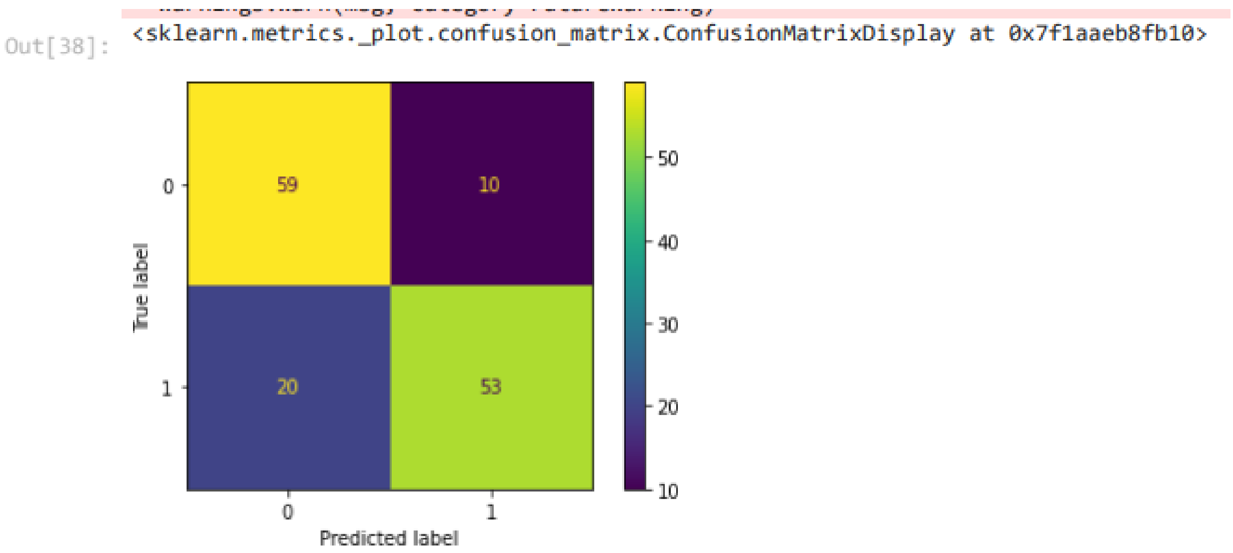
```
Out[36]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aae8a7910>
```

- Random Forest



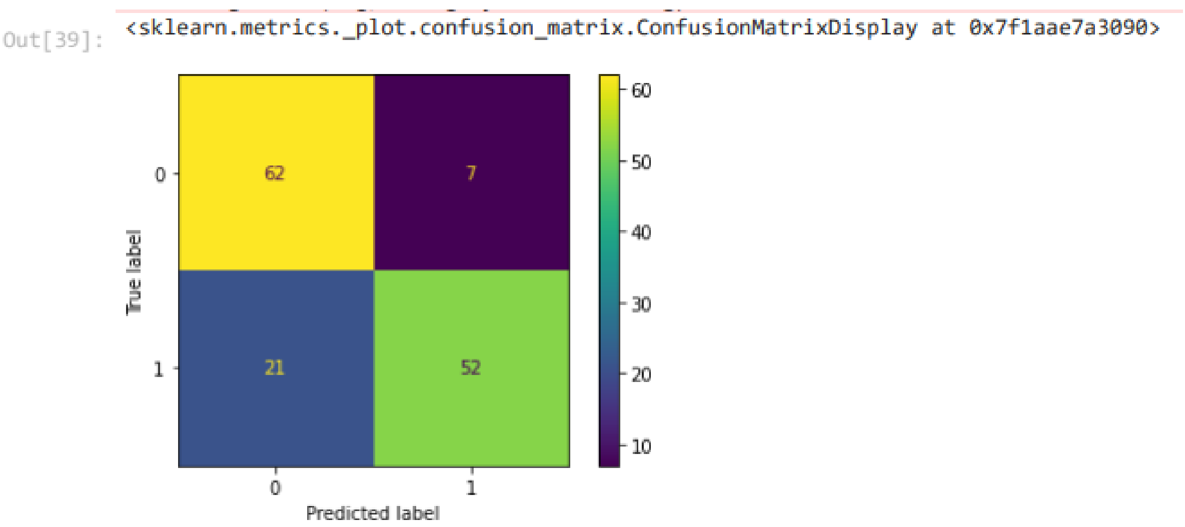
```
Out[37]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aaeae1a10>
```

- Logistic Regression



```
Out[38]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aaeb8fb10>
```

- XGB classifier



```
Out[39]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f1aae7a3090>
```

- CNN (Customized)

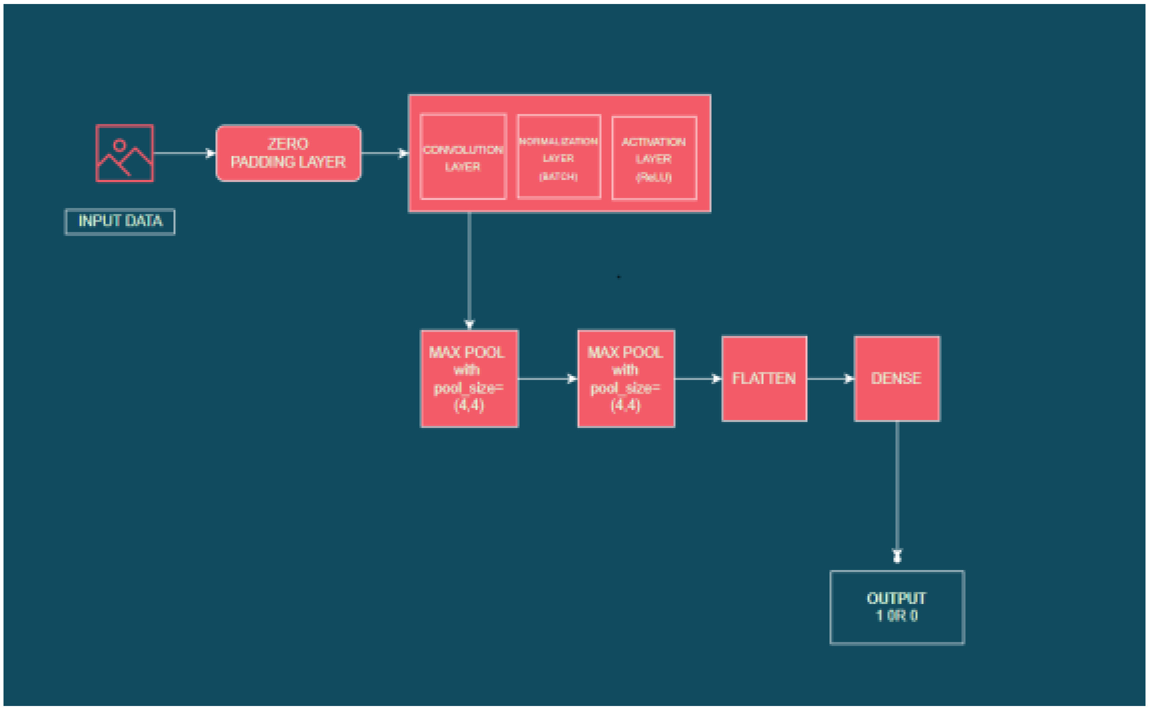


Fig 14:- CNN flow

- Ensemble learning model Citation :- [9][10]
- VGG16

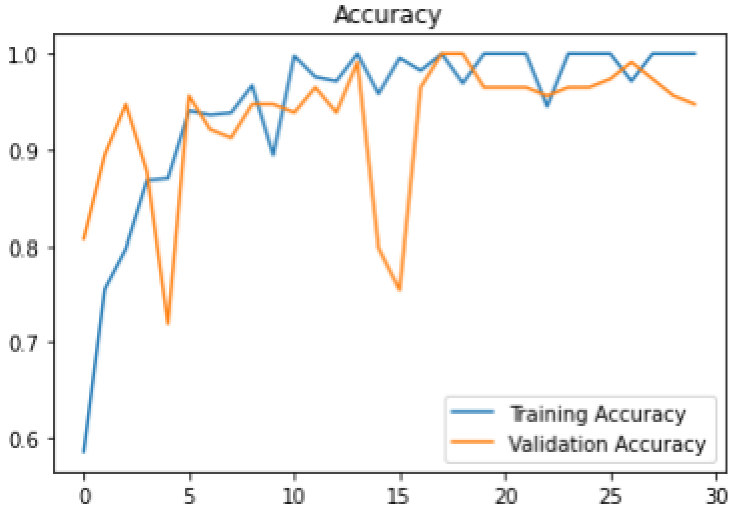
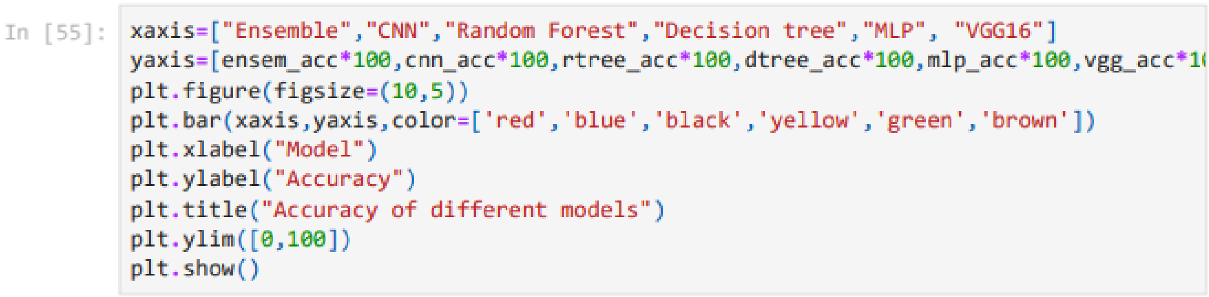


Fig 15:- VGG16 Accuracy

g. Comparison/Analysis :- After the model is created we need to use the validation data to compare the results with different models and finally analyze and conclude as a survey result and proposed approach.



```
In [55]: xaxis=["Ensemble","CNN","Random Forest","Decision tree","MLP", "VGG16"]
         yaxis=[ensem_acc*100,cnn_acc*100,rtree_acc*100,dtree_acc*100,mlp_acc*100,vgg_acc*1
         plt.figure(figsize=(10,5))
         plt.bar(xaxis,yaxis,color=['red','blue','black','yellow','green','brown'])
         plt.xlabel("Model")
         plt.ylabel("Accuracy")
         plt.title("Accuracy of different models")
         plt.ylim([0,100])
         plt.show()
```

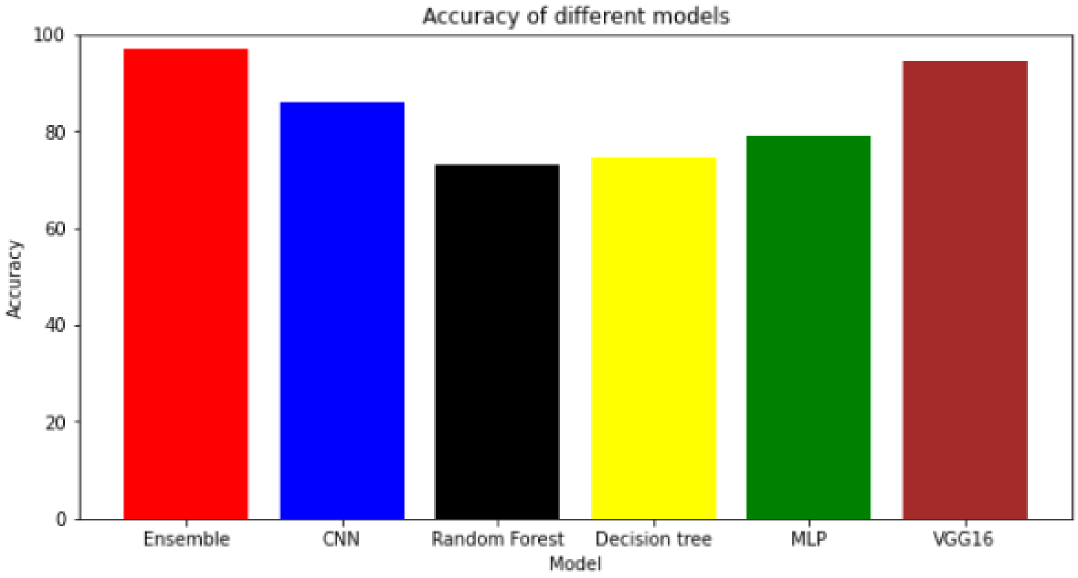


Fig 16:- Comparison of accuracy of models

Dataset Used :-

a. The dataset used, consists of Brain MRI images which are grouped into folders as either benign or malignant tumors.

b. The dataset contains 253 MRI images in which 155 brain MRIs are malignant and other 98 are benign.

c. This dataset is taken from Kaggle. You can find the dataset here.

d. The dataset is a BraTS 2015, 2018 dataset which is a benchmarked dataset which contains the MRI scan reports of various patients and is globally verified and makes the work easier to segregate.

e. There are two class labels in the dataset as Benign and Malicious or for our ease we can say that Yes and No.

The dataset looks like as follows:-

For labeled Data as “NO” :-

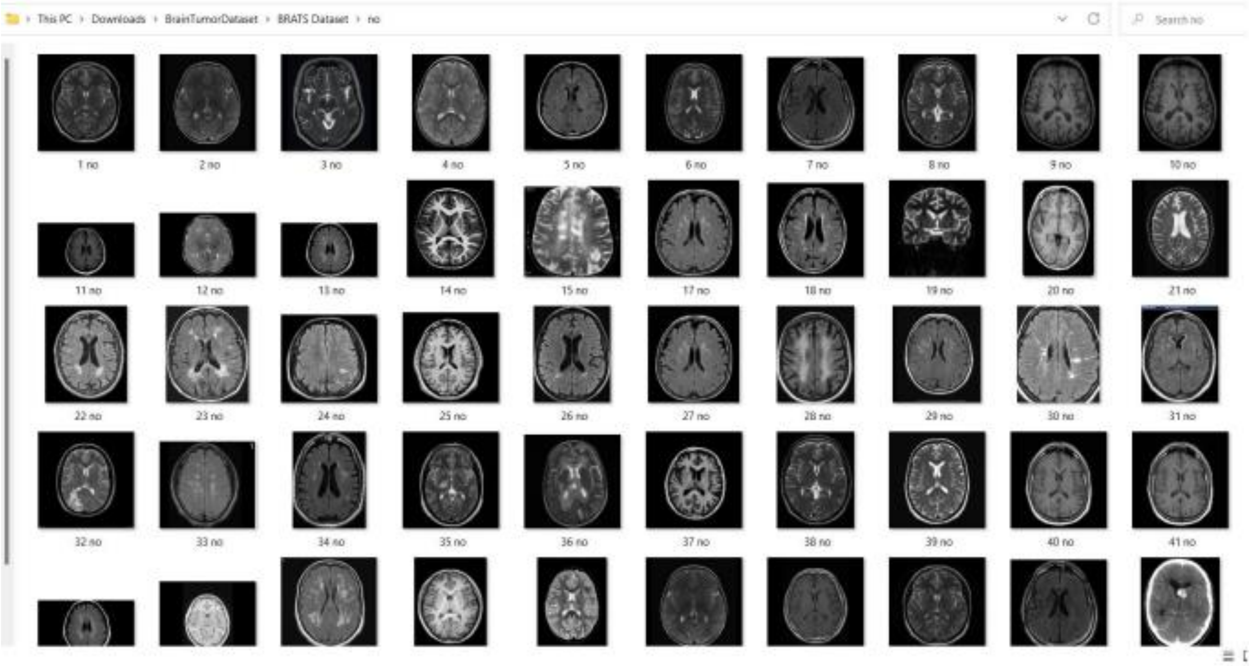


Fig 18:- Labelled data “NO”

Proposed Approach :-

The approach is depicted correctly below defining the complete design of the architecture diagram that will be required by our project. The project is focussing on the standard data science process which is assisted with the steps of Image Processing where the object focussing is to be done using python. The tech-stack of our project is :- Python, Computer Vision, Machine Learning, Deep Learning, Data Science. The initial stage of the project is focussing more on the data collection which is done using a benchmarked dataset which is collected from the kaggle. Then we started off with creating more data using the balanced synthesis technique so that it doesn't affect the overall model accuracy(i.e does not overfit). We proposed the Machine Learning and Deep Learning techniques which are compared after and we get good working classifiers as specified by our workflow

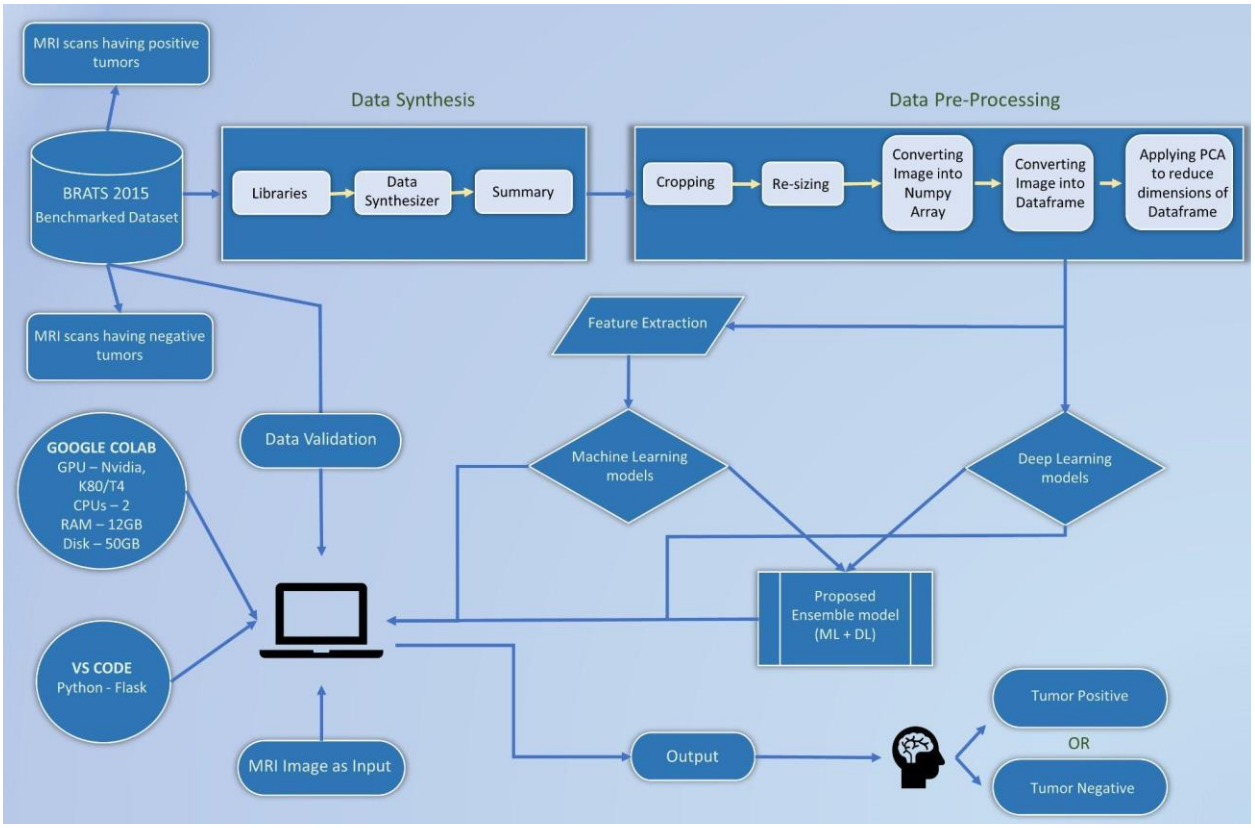
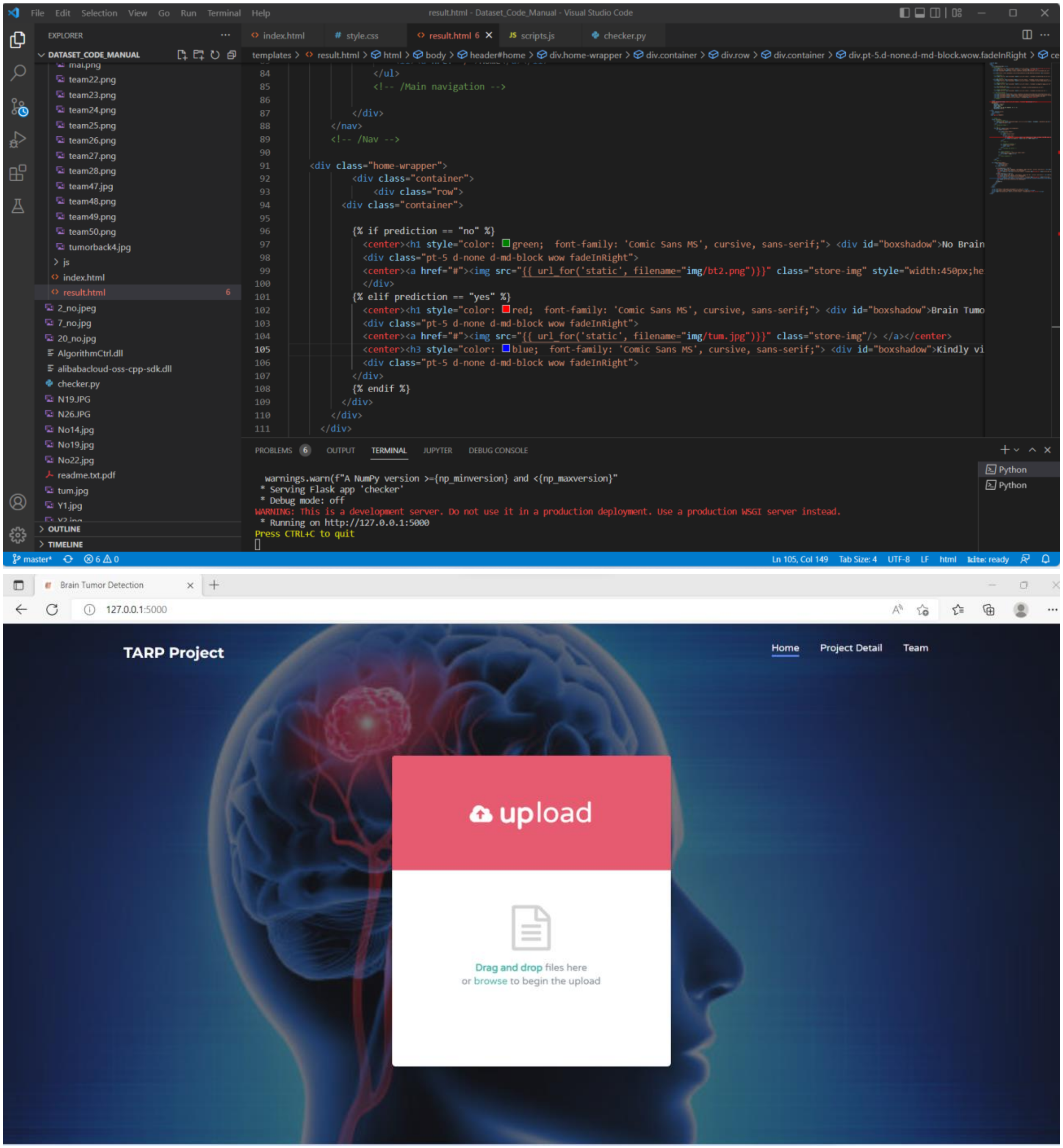


Fig 19:- Flowchart

Application Implementation:-

A GUI will be created using FLASK which will be a web application where the end user will be able to upload his MRI scans of his brain. The MRI scans will be sent to the model after applying the appropriate image processing techniques where the model will classify whether the user has or does not have Brain Tumor.



```
                </ul>
                <!-- /Main navigation -->

            </div>
        </nav>
        <!-- /Nav -->

    <div class="home-wrapper">
            <div class="container">
                <div class="row">
          <div class="container">

            {% if prediction == "no" %}
              <center><h1 style="color: 🟩green;  font-family: 'Comic Sans MS', cursive, sans-serif;"> <div id="boxshadow">No Brain
              <div class="pt-5 d-none d-md-block wow fadeInRight">
              <center><a href="#"><img src="{{ url_for('static', filename="img/bt2.png")}}" class="store-img" style="width:450px;he
              </div>
            {% elif prediction == "yes" %}
              <center><h1 style="color: 🟥red;  font-family: 'Comic Sans MS', cursive, sans-serif;"> <div id="boxshadow">Brain Tumo
              <div class="pt-5 d-none d-md-block wow fadeInRight">
              <center><a href="#"><img src="{{ url_for('static', filename="img/tum.jpg")}}" class="store-img"/> </a></center>
              <center><h3 style="color: 🟦blue;  font-family: 'Comic Sans MS', cursive, sans-serif;"> <div id="boxshadow">Kindly vi
              <div class="pt-5 d-none d-md-block wow fadeInRight">
            </div>
            {% endif %}
          </div>
        </div>
      </div>
```

```
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
 * Serving Flask app 'checker'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Fig 20:- Frontend Application

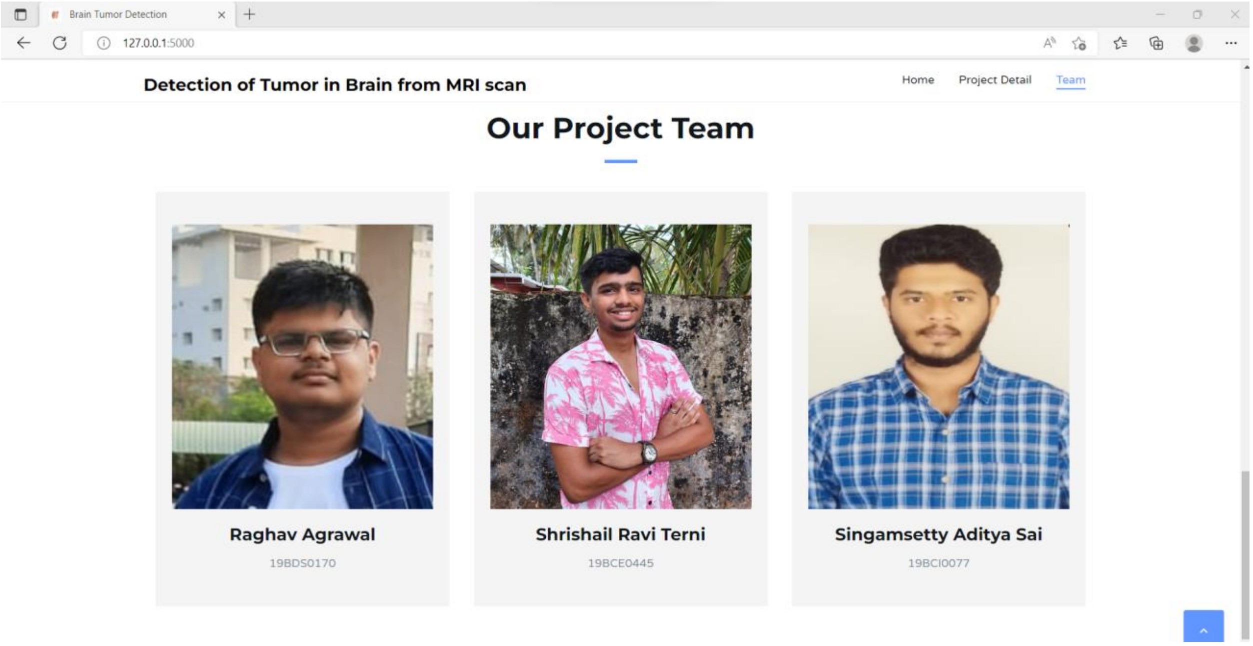


Fig 21:- Team members

Result Analysis in comparison to previous papers:-

The following table contains the result comparison of our work with the existing works

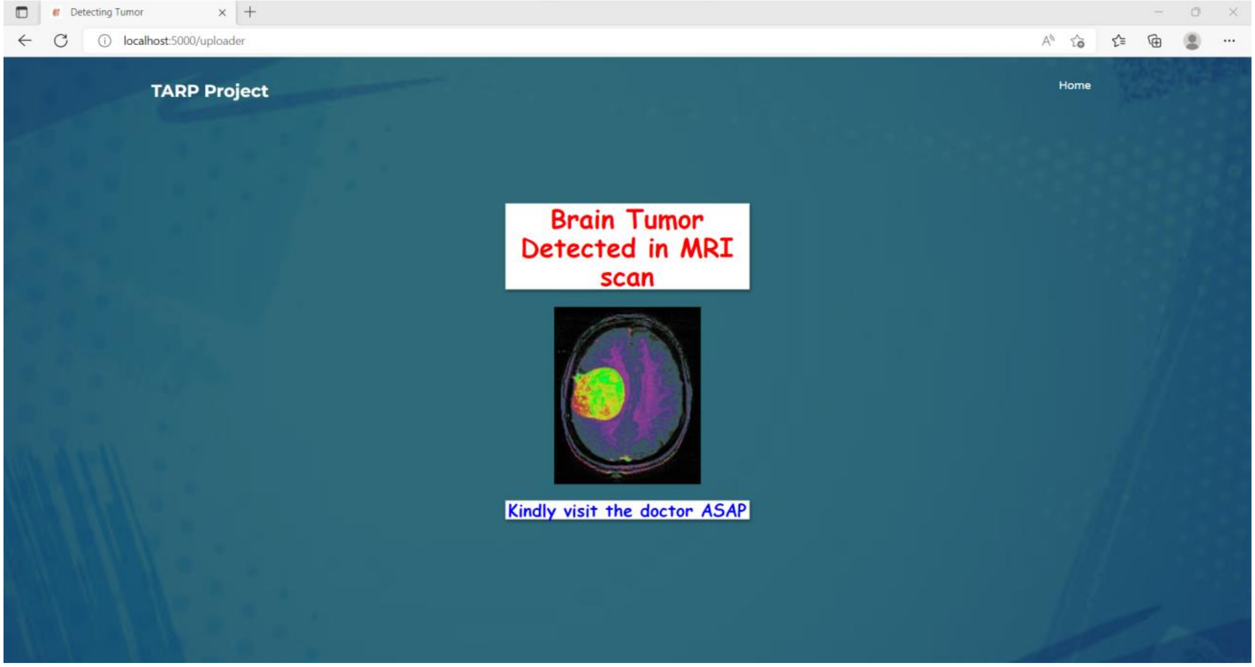
Author	**Approach**	**Resultant Accuracy (%)**
Our Work	In our work, we are trying to implement an Ensemble model which will be the combination of Deep Learning and Machine learning models and can be more accurate when compared to other existing models. This will create a proper critic that we should no blindly go with the deep learning models, i.e. same task can be	90 - 95% (For Now) accuracy obtained.

	done with Machine Learning models	
Amjad et al.	The proposed technique first converts each input MR modality to slices, and intensities are preprocessed using a statistical normalization approach. K-means clustering approach is implemented to segment brain tumors to focus ROI for precise feature extraction. Finally, to classify brain tumors into their two general classes (benign/malignant), a fine-tuned VGG-19 CNN model is trained perfectly using synthetic data augmentation techniques.	94.06 % accuracy obtained.
Divyamary et al.	In this paper, By using Naive classifier method, tumor is identified and also	84% accuracy obtained

	divided into normal and abnormal tumor. This paper Also focuses on the preprocessing of the particular brain image using morphological operations. At last, we are using the Naive classifier method to classify the pre-processing images, whether they contain the tumor or not.	
Rehman et al.	A pretrained CNN model VGG19 is employed for feature extraction and later selects the best feature by the proposed CbFNN approach. The selected features are validated through FNN. The tumor is extracted from MRI scans using proposed 3D CNN architecture while the classification process is done by transfer learning.	95.5% accuracy obtained.
Islam et al.	In this paper, the achievement of the proposed super pixels, PCA & TK-means scheme is better than other existing detection schemes. Super pixels and PCA played a vital	95% accuracy obtained.

	role in feature extraction that reduced the dimensions and complexity of the MR images. As a result, the brain tumors are detected accurately from MR images using our proposed scheme in a short execution time of 35–60 s.	

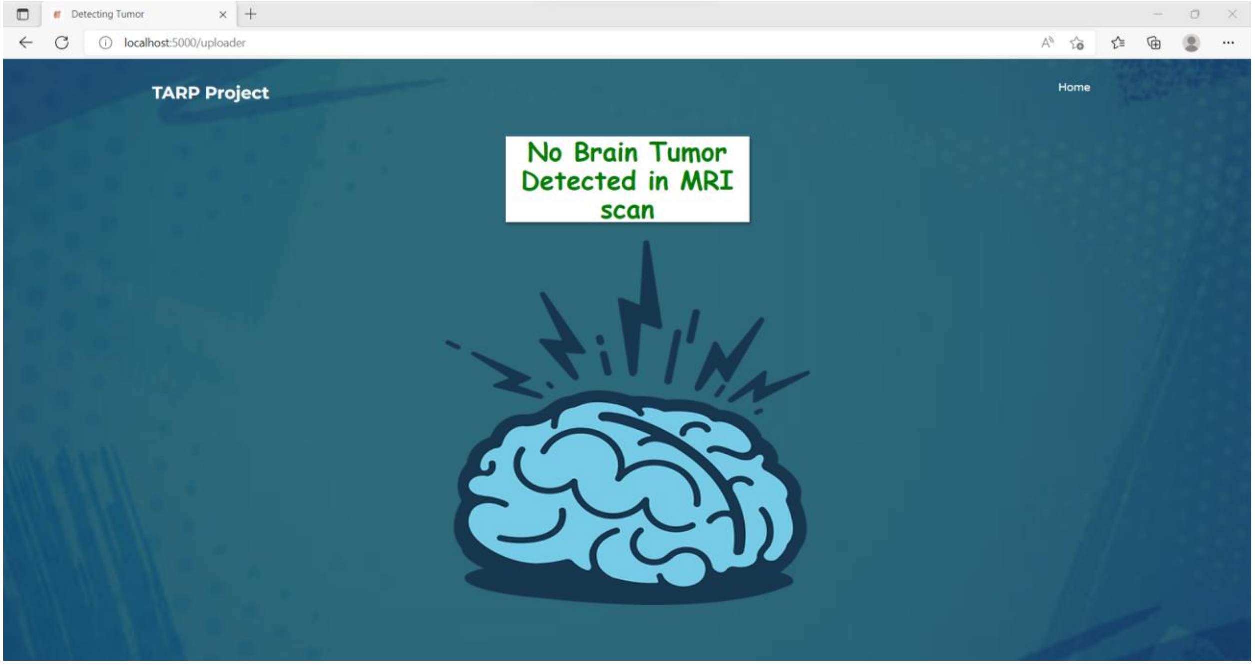
Sample screenshots of the result in the Application:-



Brain Tumor Detected in MRI scan

Kindly visit the doctor ASAP

Brain Tumor detected by our model



No Brain Tumor Detected in MRI scan

Brain Tumor is not detected by our model

References:-

[1] Khan, Amjad Rehman, et al. "Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification."Microscopy Research and Technique 84.7 (2021): 1389-1399.

[2] Amin, Javeria, et al. "Big data analysis for brain tumor detection: Deep convolutional neural networks." Future Generation Computer Systems 87 (2018): 290-297.

[3] Divyamary, D., et al. "Brain tumor detection from MRI images using Naive classifier." 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020.

[4] Hossain, Tonmoy, et al. "Brain tumor detection using convolutional neural networks." 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT). IEEE, 2019.

[5] Ari, Ali, and Davut Hanbay. "Deep learning based brain tumor classification and detection system." Turkish Journal of Electrical Engineering and Computer Sciences 26.5 (2018): 2275-2286.

[6] Hemanth, G., M. Janardhan, and L. Sujihelen. "Design and implement brain tumor detection using a machine learning approach." 2019 3rd international conference on trends in electronics and informatics (ICOEI). IEEE, 2019

[7] Saba, Tanzila, et al. "Brain tumor detection using fusion of hand crafted and deep learning features." Cognitive Systems Research 59 (2020): 221-230.

[8] Amin, Javaria, et al. "Brain tumor detection using statistical and machine learning methods." Computer methods and programs in biomedicine 177 (2019): 69-79.

[9] Islam, Md Khairul, et al. "Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm." Machine Learning with Applications 5 (2021): 100044.

[10] Rehman, Amjad, et al. "Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture." Microscopy Research and Technique 84.1 (2021): 133-149.

[11] Deb, Daizy, and Sudipta Roy. "Brain tumor detection based on hybrid deep neural network in MRI by adaptive squirrel search optimization." Multimedia tools and applications 80.2 (2021): 2621-2645.

[12] Singh, Garima, and M. A. Ansari. "Efficient detection of brain tumor from MRIs using K-means segmentation and normalized histogram." *2016 1st India international conference on information processing (IICIP)*. IEEE, 2016.

[13] Sazzad, TM Shahriar, et al. "Development of automated brain tumor identification using MRI images." *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019.

[14] Borole, Vipin Y., Sunil S. Nimbhore, and Dr Seema S. Kawthekar. "Image processing techniques for brain tumor detection: A review." *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 4.5 (2015): 2.

[15] Meera, R., and P. Anandhan. "A Review On Automatic Detection of Brain Tumor Using Computer Aided Diagnosis System Through MRI." *EAI Endorsed Transactions on Energy Web* 5.20 (2018).

Individual Contribution :-

Name	**Contribution**
Raghav Agrawal	Flask Application and Machine Learning models
Shrishail Terni	Data Preprocessing and Documentation work
Singamsetty Aditya Sai	Data Augmentation and Poster Making

Appendix :-

- **PPT Link**
- **Source Code :-**
  1. **Data Synthesis**
  2. **Main Source Code(Preprocessing + Model Generation)**
- **High Resolution Image**
- **Demo Video**
- **Poster**