

# ENHANCED APPROACH FOR BRAIN TUMOR DETECTION

**M A R KUMAR**, *Associate Professor, Sreyas Institute Of Engineering And Technology,  
Hyderabad*

**VINUTA PATIL**, *Sreyas Institute Of Engineering And Technology, Hyderabad*  
**SUSHITHA RACHAMALLA**, *Sreyas Institute Of Engineering And Technology , Hyderabad*  
**HEPSEEBA GAJULAVARTHI**, *Sreyas Institute Of Engineering And Technology, Hyderabad*  
**MARTHA BHAVANA**, *Sreyas Institute Of Engineering And Technology, Hyderabad*

## ABSTRACT

Automated defect detection in medical imaging has become the emergent field in several medical diagnostic applications. Automated detection of tumor in MRI is very crucial as it provides information about abnormal tissues which is necessary for planning treatment. The conventional method for defect detection in magnetic resonance brain images is human inspection. This method is impractical due to large amount of data. Hence, trusted and automatic classification schemes are essential to prevent the death rate of human. So, automated tumor detection methods are developed as it would save radiologist time and obtain a tested accuracy. The MRI brain tumor detection is complicated task due to complexity and variance of tumors. In this project, we propose the machine learning algorithms to overcome the drawbacks of traditional classifiers where tumor is detected in brain MRI using machine learning algorithms. 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.

## INTROD CTION

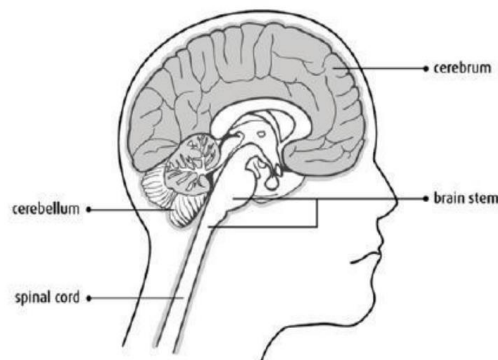
### BRAIN TUMOR DETECTION SYSTEM

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 MRI 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.



**Basic Structure of Human Brain**

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. 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.

## **PROBLEM STATEMENT**

To detect whether a person is affected by a brain tumor or not by checking his MRI scan. If affected he can take the necessary precautions and medical treatments. Our study deals with automated brain tumor detection and classification. Normally the anatomy of the brain is analyzed by MRI scans or CT scans. The aim of the paper is tumor identification in brain MR images. The main reason for detection of brain tumors is to provide aid to clinical diagnosis. The aim is to provide an algorithm that guarantees the presence of a tumor by combining several procedures to provide a foolproof method of tumor detection in MR brain images. The method involved is called Convolution Neural Network(CNN) and the 4 main processes involved are Image Pre-Processing, Image segmentation, Feature Extraction, and Classification.

The focus of this project is MR brain images tumor extraction and its representation in simpler form such that it is understandable by everyone. The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in extracted image of the tumor from the MR brain image. The resultant image will be able to provide about its boundary provides us with information related to the tumor that can prove useful for various cases, which will provide a better base for the staff to decide the curing procedure. Finally, we detect whether the given MR brain image has tumor or not using Convolution Neural Network.

## **PROJECT OBJECTIVE**

- To capture MRI images to analyze the presence of tumor using machine learning techniques.
- To perform segmentation process on MRI images to separate the tumor from normal brain tissues.
- To extract features from captured images to find the tumor.
- To detect the presence of tumor
- 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

## **SCOPE**

This project can create a user interface in hospitals which allows doctors to easily determine the impact of tumor and suggest treatment accordingly. Since performance and complexity depend on the input data representation we can try to predict the location as well as stage of the tumor from Volume based 3D images. By creating three dimensional (3D) anatomical models from individual patients, training, planning and computer guidance during surgery is improved.

## 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.

## LITERATURE SURVEY

1. Machine Learning Assisted Methodology for Multiclass Classification of Malignant Brain Tumor By Ankit Vidyarthi, Ruchi Agarwal, Deepak Gupta, Rahul Sharma.
2. Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging by Andres Anaya-Isaza and Leonel Mera-Jimenez.
3. A Robust Approach for Brain Tumor Detection in Magnetic resonance Images Using Finetuned EfficientNet by Hasnanin Ali Shah, Faisal Saeed, Sangseok Yun and Jae-Mo kang.
4. Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network by Ming Li, Lishan Kuang and Zhanguo Sha.
5. Deep Learning by Ian GoodFellow , Yoshua Bengio and Aaron Courville.(Textbook)
6. Brain Tumor Detection Analysis using CNN by Sunil Kumar, Renu Dhir, Nisha Chaurasia.
7. A Review Paper on Brain Tumor Segmentation and Detection by Priya Patil, Seema Pawar, Sunayna Patil, Arjun Nichal

## SYSTEM ANALYSIS

### EXISTING SYSTEM

#### BRAIN TUMOR DETECTION BASED ON THRESHOLD

Threshold based approach uses image segmentation methods. Since there are various segmentation mechanisms like(ANN, clustering, histogram based methods edge-based methods) , it is a challenging task to choose appropriate technique for a particular kind of image.

Segmentation require separating an image into regions corresponding to objects. Threshold is the simplest method of segmenting image. Threshold can be used to convert Gray-Scale image to binary images. The simplest property that pixels in a pixels can share is intensity. So, a natural way to segment such pixels is through threshold, the separation of light and dark pixels. Threshold convert the gray images to binary image, its value to convert the zero's and one's.

Threshold have a many types. One of the approach is OTSU threshold. Otsu's threshold technique includes iterating via all the feasible threshold values and calculating a measure of unfold for the pixel ranges every aspect of the threshold, i.e. the pixels that both fall in foreground or background. The intention is to discover the threshold value of foreground and background of the binary images.

Image segmentation is a technique of segregating the image into many parts. The basic aim of this segregation is to make the images easy to analyze and interpret with preserving the quality. This technique is also used to trace the objects' borders within the images. This technique labels the pixels according to their intensity and characteristics. Those parts represent the entire original image and acquire its characteristics such as intensity and similarity. The image segmentation technique is used to create contours of the body for clinical purposes. Segmentation is used in machine perception, malignant disease analysis, tissue volumes, anatomical and functional analyses, virtual reality visualization, and anomaly analysis, and object definition and detection. Segmentation methods has ability to detect or identify the abnormal portion from the image which is useful for analyzing the size, volume, location, texture and shape of the extracted image. MR image segmentation with the aid of preserving the threshold information, which is convenient to identify the broken regions extra precisely. It was a trendy surmise that the objects that are placed in close propinquity might be sharing similar houses and characteristics.

### • THRESHOLDING:

Thresholding is the simplest method of image segmentation. It is a non-linear operation that converts a grey-scale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value. In Open CV, we use **cv2.threshold()** function:

```
cv2.threshold(src, thresh, maxval, type[dst])
```

This function applies fixed-level thresholding to a single-channel array. The function is typically used to get a bi-level (binary) image out of a grayscale image for removing a noise, that is, filtering out pixels with too small or too large values. "maxval" is the set threshold value which compares with input values, when the input is greater than the set threshold value it gives output is set maxval value and it is shown with white color in gray images. when the input pixel intensity values are less than the set threshold, its output is black color. There are several types of thresholding supported by the function.

The function returns the computed threshold value and thresholder image.

**1. src** - input array (single-channel, 8-bit or 32-bit floating point). This is the source image, which should be a grayscale image.

**2. thresh** - threshold value, and it is used to classify the pixel values.

**3. maxval** - maximum value to use with the THRESH\_BINARY and THRESH\_BINARY\_INV thresholding types. It represents the value to be given if pixel value is more than (sometimes less than) the threshold value.

**4. type** - thresholding type

- cv2.THRESH\_BINARY
- cv2.THRESH\_BINARY\_INV

- **MORPHOLOGICAL OPERATIONS:**

Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

The Morphological techniques are also used with segmentation techniques. The morphological action is normally performed on binary images. It processes the operations based on shape and it has a wide set of the image processing operation. Erosion and Dilation are two methods of morphological operations which used in this proposed work. We perform both Erosion and dilation operations used together.

Two main steps of the erosion and dilation morphological operation are opening and closing. The first step is the opening of the MRI binary image. The main work of opening operation is open up a gap which is present in between object and connect that to a small collection of pixels. After setting of the bridge, the erosion again restored with their actual size using dilation. If the binary image has been opened then the subsequent opened same structured elements have not affected on that image. After completing the opening operations next step is the closing operation. Based on the closing operation while keeping the original region sizes, the erosion and dilation can handle different hole in the image region. Dilation and Erosion are the basic morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

Watershed Method: considers the gradient magnitude of an image as a topographic surface where high gradient denotes peaks, while low gradient denotes valleys. Start by filling every isolated valley with different coloured water. As the water rises, water from different valleys will start to merge. To avoid that, barriers are built in the locations where water merges. Continue the work of filling water and building barriers until all the peaks are under water. Then the created barriers give the segmentation result.

## **PROPOSED METHODS**

### **BRAIN TUMOR DETECTION USING CONVOLUTIONAL NEURAL NETWORK**

Classification is the best approaches for identification of images like any kind of medical imaging. All classification algorithms are based on the prediction of image, where one or more features and that each of these features belongs to one of several classes.

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 details. 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 preprocessing 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.

➤ **SEQUENTIAL:**

- To initialize the neural network, we create an object of the Sequential class.
- classifier = Sequential ()

➤ **CONVOLUTION:**

- To add the convolution layer, we call the add function with the classifier object and pass in Convolution2D with parameters. The first argument feature\_detectors which is the number of feature detectors that we want to create. The second and third parameters are dimensions of the feature detector matrix.
- We used 256 feature detectors for CNNs. The next parameter is input shape which is the shape of the input image. The images will be converted into this shape during pre-processing. If the image is black and white it will be converted into a 2D array and if the image is coloured it will be converted into a 3D array.
- In this case, we'll assume that we are working with coloured images. Input\_shape is passed in a tuple with the number of channels, which is 3 for a coloured image, and the dimensions of the 2D array in each channel. If you are not using a GPU it's advisable to use lower dimensions to reduce the computation time. The final parameter is the activation function. Classifying images is a nonlinear problem. So, we use the rectifier function to ensure that we don't have negative pixel values during computation. That's how we achieve non-linearity.

### ➤ **POOLING:**

- The Pooling layer is responsible for reducing the spatial size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.
- There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Generally, we use max pooling. • In this step we reduce the size of the feature map. Generally, we create a pool size of 2x2 for max pooling. This enables us to reduce the size of the feature map while not losing important image information.

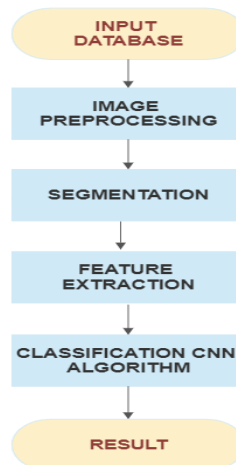
### ➤ **FLATTENING:**

- In this step, all the pooled feature maps are taken and put into a single vector for inputting it to the next layer.
- The Flatten function flattens all the feature maps into a single long column.

### ➤ **FULLY CONNECTION:**

- The next step is to use the vector we obtained above as the input for the neural network by using the Dense function in Keras. The first parameter is output which is the number of nodes in the hidden layer. You can determine the most appropriate number through experimentation. The higher the number of dimensions the more computing resources you will need to fit the model. A common practice is to pick the number of nodes in powers of two.
- The next layer we have to add is the output layer. In this case, we'll use the sigmoid activation function since we expect a binary outcome. If we expected more than two outcomes, we would use the SoftMax function.
- The output here is 1 since we just expect the predicted probabilities of the classes.



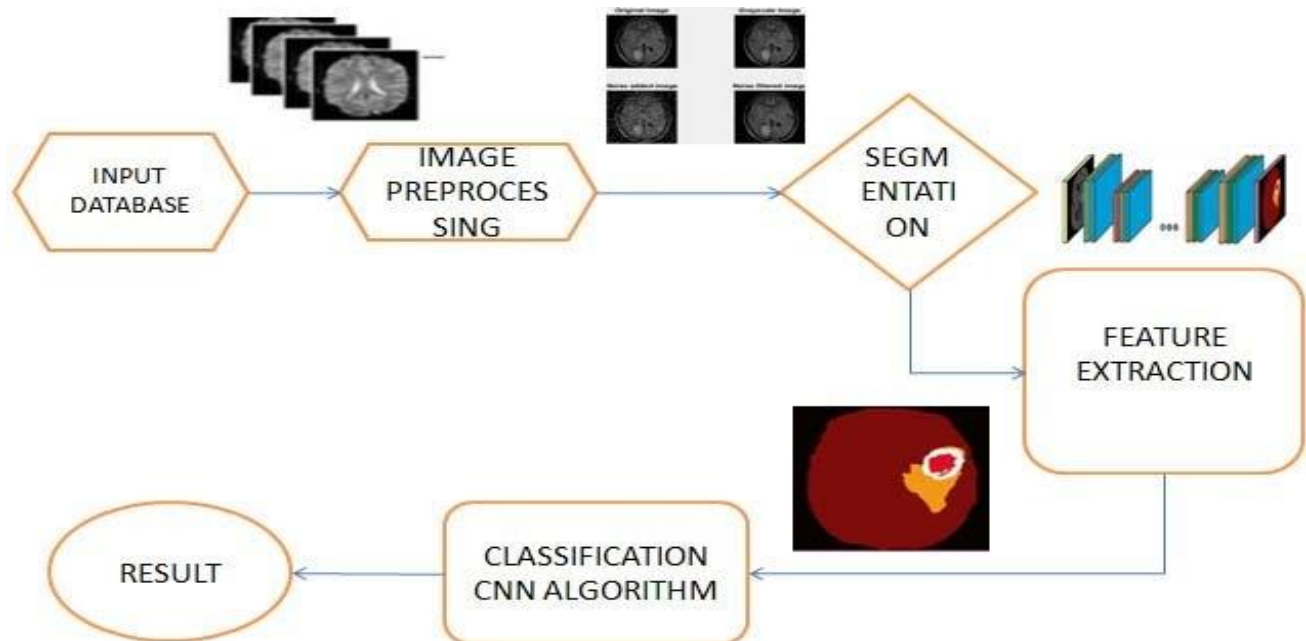


### Work Flow of Brain Tumor Detection

- **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. As explained in system flow we are using high pass filter for noise removal and preprocessing.
- **Segmentation:** Region growing is the simple region-based image segmentation technique. It is also classified as a pixel based image segmentation technique since it is involve the selection of initial seed points.
- **Feature Extraction:** The feature extraction is used for edge detection of the images. It is the process of collecting higher level information of image such as shape, texture, color, and contrast.

## SYSTEM DESIGN

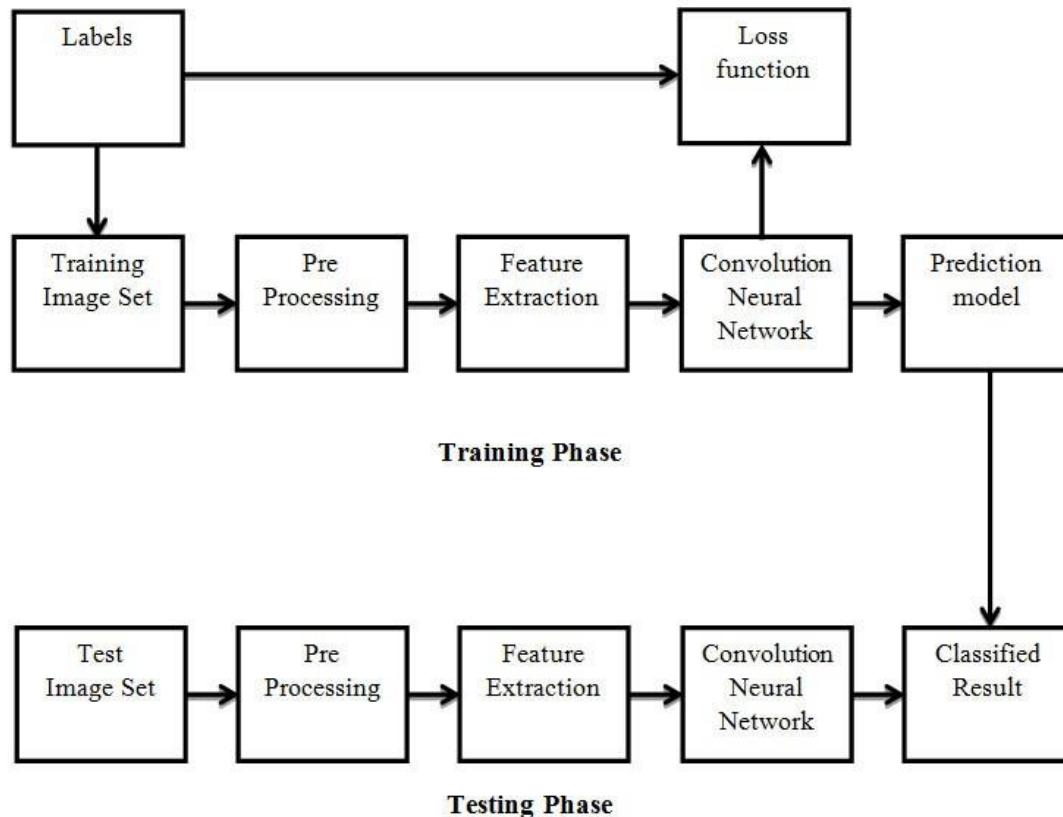
The system architecture diagram is an abstract depiction of the system's component architecture. It provides a succinct description of the system's component architecture in order to assist in component-component relationships and system functioning. The system architecture diagram is a visual representation of the system architecture. It shows the connections between the various components of the system and indicates what functions each component performs. The general system representation shows the major functions of the system and the relationships between the various system components.



### System Architecture

#### DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.



### Dataflow Diagram

## IMPLEMENTATION

### ➤ Modules

**Os:** It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system. Consider **Current Working Directory(CWD)** as a folder, where the Python is operating. Whenever the files are called only by their name, Python assumes that it starts in the CWD which means that name-only reference will be successful only if the file is in the Python's CWD.

**Cv2:** OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a

human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

**Tensorflow:** TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow was designed for use both in research and development and in production systems.

**Keras:** Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras is relatively easy to learn and work with because it provides a python frontend with a high level of abstraction while having the option of multiple back-ends for computation purposes. This makes Keras slower than other deep learning frameworks, but extremely beginner-friendly. Keras allows you to switch between different back ends. Keras has features such as it runs smoothly on both CPU and GPU ,it supports almost all neural network models ,it is modular in nature, which makes it expressive, flexible, and apt for innovative research.

**Numpy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

**PIL:** The Python Pillow Library is a fork of an older library called PIL. PIL stands for Python Imaging Library, and it's the original library that enabled Python to deal with images. PIL was discontinued in 2011 and only supports Python 2. To use its developers' own description, Pillow is the friendly PIL fork that kept the library alive and includes support for Python 3. Pillow is often the preferred option for high-level image processing tasks that don't require more advanced image processing expertise. It's also often used for exploratory work when dealing with images.

**Normalize:** The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model. Four common normalization techniques may be useful such as scaling to a range, clipping, log scaling, z-score.

**Sequential:** The Sequential Model is just as the name implies. It consists of a sequence of layers, one after the other. A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. A CNN can be instantiated as a Sequential

model because each layer has exactly one input and output and is stacked together to form the entire network.

**Conv2D:** 2D convolution layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. If `use_bias` is `True`, a bias vector is created and added to the outputs. Finally, if `activation` is not `None`, it is applied to the outputs as well. A 2-D convolutional layer applies sliding convolutional filters to 2-D input. The layer convolves the input by moving the filters along the input vertically and horizontally and computing the dot product of the weights and the input, and then adding a bias term.

**MaxPooling2D:** Keras `MaxPooling2D` is a pooling or max pooling operation which calculates the largest or maximum value in every patch and the feature map. The results will be down sampled, or it will pool features map which was highlighting the most present feature into the patch which contains the average feature presence from the average pooling. The max pooling is found to work well in average pooling for vision tasks.

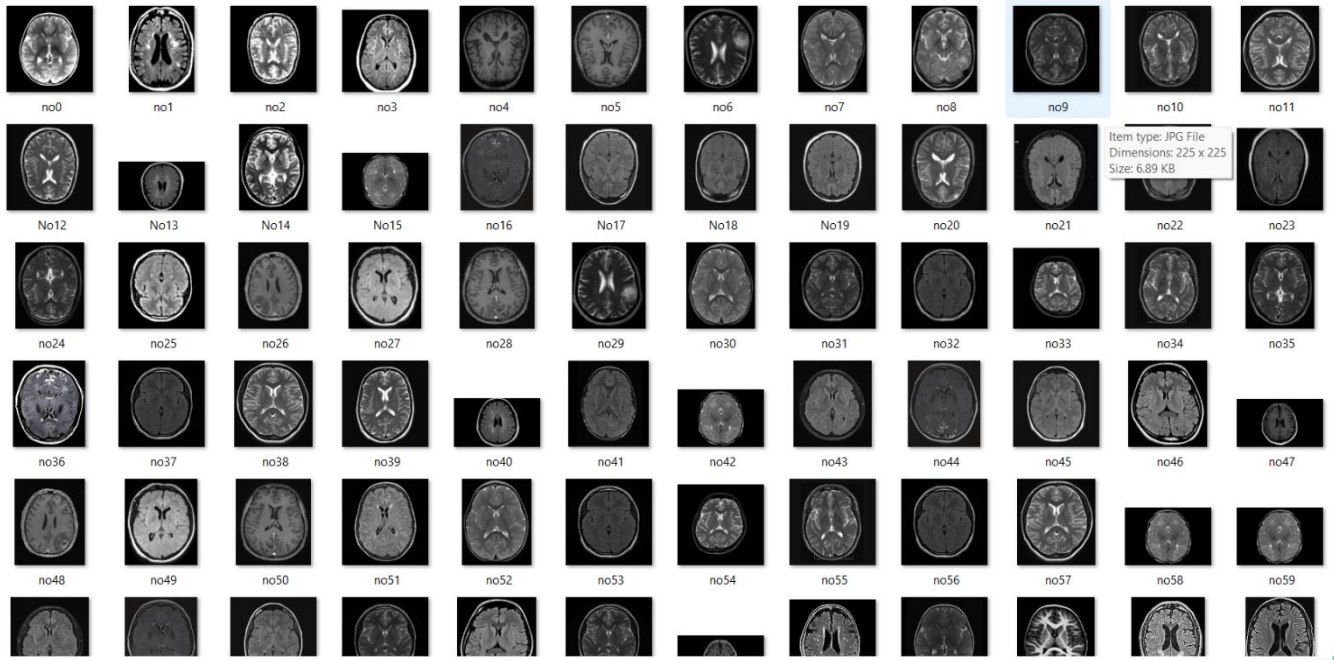
**to\_categorical:** The function of `to_categorical` is used to convert the class vector to the matrix of the binary class. We can apply limitations in cases where we are involving the network communications which was creating additional sources of randomness while non-deterministic apps are involved. The `to_categorical` function is useful while converting the binary class matrix.

**Train\_test\_split:** The Sklearn `train_test_split` function helps us create our training data and test data. This is because typically, the training data and test data come from the same original dataset. To get the data to build a model, we start with a single dataset, and then we split it into two datasets: train and test. So the scikit learn train test split function enables us to split a dataset into training data and test data. 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. The `train_test_split` function returns a list that contains the train-test splits of the inputs.

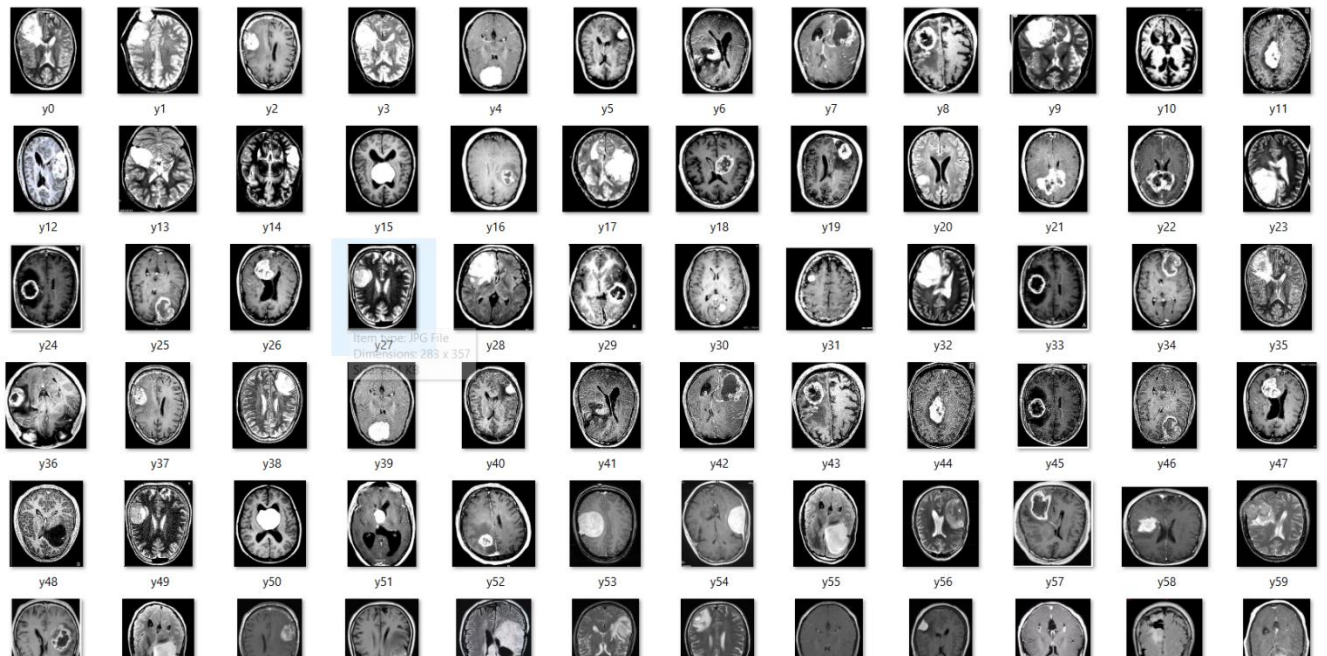
## Datasets

- **Training Dataset**

The training dataset contains two types of images which are no and yes folders. These folders are used for training the model. Each folder contains about 1500 images.



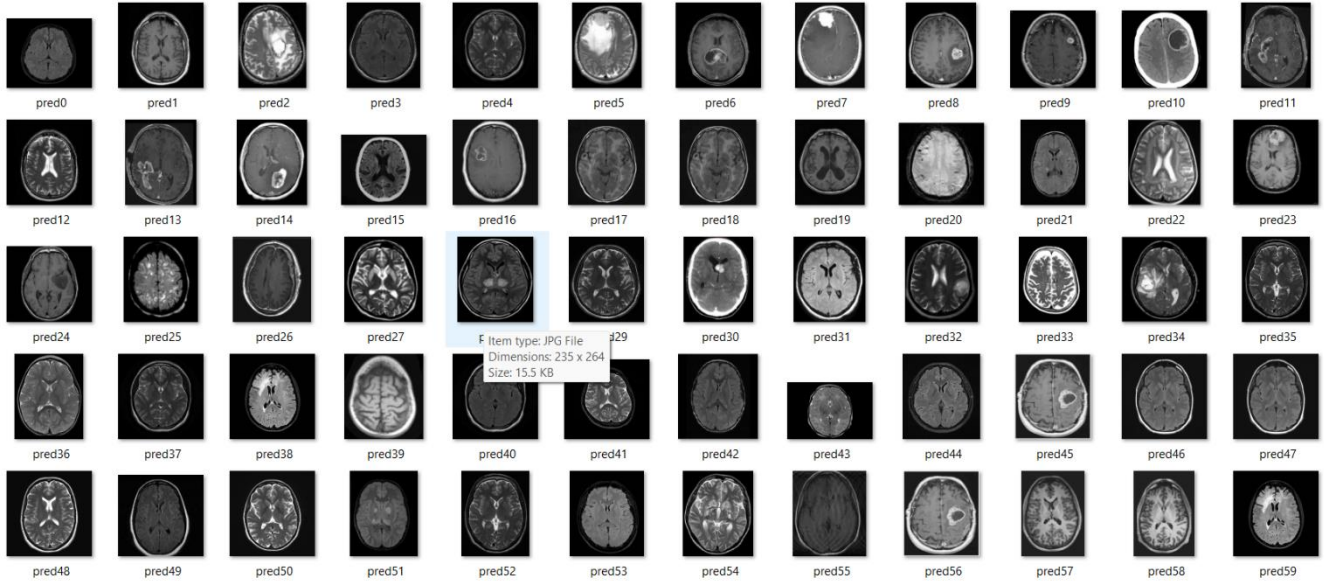
**Training Dataset 1**



**Training Dataset 2**

- **Testing Dataset**

This folder contains brain MRI images for testing the model after it is trained. These images are used to check if the model is giving the expected results.



**Testing Dataset**

**RESULTS**

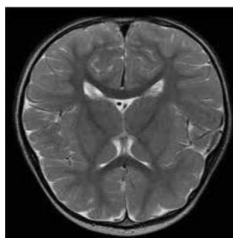
**Result on Webpage**

The following images show the result after clicking the predict button on the webpage.

- **Negative Result**



**Brain Tumor Classification Using CNN**

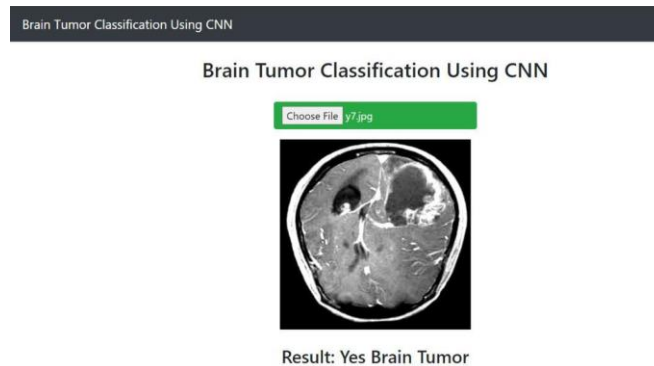


Result: No Brain Tumor

**Negative Result**

The above image displays the result on the webpage which is no brain tumor.

- **Positive Result**



### **Positive Result**

The above image displays the result on the webpage which is presence on brain tumor

## **CONCLUSION AND FUTURE SCOPE**

### **CONCLUSION**

In brain tumor detection we have studied about feature based existing work. In feature based we have study about image processing techniques like image pre-processing, image segmentation, features extraction, classification. In existing system the accuracy of the model is not more than 70%. But, in the proposed system the accuracy is increased to 90%. So, we can clearly understand that the efficiency of the model is very high when compared to other models because of the efficiency and accuracy of this model. So, if any extension of this model is required then we can furtherly use some hybrid combinations of CNN algorithm.

### **FUTURE SCOPE**

The goal is to try detect the grade and even the stage of tumor, try to predict the location of tumor from 3D images. For more future work it can also be tested for tumor detection of other parts of the body. Future research on the project will lead towards improving the computational speed of the proposed approach as well as minimizing the amount of time taken for the process. The proposed framework can be used as a quick guidance tool for the radiologists. The future work will consist of more precise extraction of tumor based on current results. Further development in the field can cause a great impact on medical industry



## REFERENCES

- “Machine Learning Assisted Methodology for Multiclass Classification of Malignant Brain Tumor” By Ankit Vidyarthi, Ruchi Agarwal, Deepak Gupta, Rahul Sharma mentioned in volume 10 and page numbers 50624 to 50640 Of IEEE .Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9772610>
- Data Augmentation and Transfer Learning for Brain Tumor Detection in Magnetic Resonance Imaging by Andres Anaya-Isaza and Leonel Mera-Jimenez.
- “A Robust Approach for Brain Tumor Detection in Magnetic resonance Images Using Finetuned EfficientNet” by Hasnanin Ali Shah, Faisal Saeed, Sangseok Yun and Jae-Mo kang mentioned in volume 10 and page numbers 65426 to 65432 of IEEE.  
Available : <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9798829>
- Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network by Ming Li, Lishan Kuang and Zhanguo Sha.
- Deep Learning by Ian GoodFellow , Yoshua Bengio and Aaron Courville.
- Brain Tumor Detection Analysis using CNN by Sunil Kumar, Renu Dhir, Nisha Chaurasia.
- A Review Paper on Brain Tumor Segmentation and Detection by Priya Patil, Seema Pawar, Sunayna Patil, Arjun Nichal.
- T. Hossain, F. S. Shishir, M. Ashraf, M. A. Al Nasim and F. Muhammad Shah, "Brain Tumor Detection Using Convolutional Neural Network," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934561.