Brain Tumor Detection Using A.I.

Submitted by

Ayush Paul (Roll No : 213223-21-0012 ; Reg. No : 223-1111-0220-21)

In fulfillment for the award of the degree of

Bachelor of Science (Honours)

in

COMPUTER SCIENCE

Under the supervision of

Ms. Nivedita Saha

Department of Computer Science

Scottish Church College

Kolkata, India

CERTIFICATE OF COMPLIANCE

This is to certify that the research project entitled

Brain Tumor Detection Using A.I.

Is a Bonafide record of work done by Ayush Paul Roll No : 213223-21-0012 ; Reg. No : 223-1111-0220-21 of under our supervision, in fulfillment of the requirements for the award of Degree of Bachelor of Science (Honours) with specialization in Computer Science from Scottish Church College, University of Calcutta in the year 2024, in the examination under CBCS Semester System.

Ms. Nivedita Saha	**Ms. Moumita Banerjee**
Project Guide	**Head of Department**
Department of Computer Science	**Department of Computer Science**
(Scottish Church College, C.U.)	**(Scottish Church College, C.U.)**

DATE

ACKNOWLEDGEMENT

We take this opportunity to express a profound sense of gratitude and respect to all those who helped me throughout the duration of this project.

First and foremost, we would like to thank our supervisor of this project, Ms. Nivedita Saha , for her valuable time and guidance. Her willingness to motivate us has contributed tremendously to our project.

We would also like to thank Ms. Moumita Banerjee, Head of Department of Computer Science, for providing a good environment and facilities to work on this project.

We regret any inadvertent omissions.

Ayush Paul: 213223-21-0012

Semester VI (Honours)

Scottish Church College,

University Of Calcutta,

Kolkata: 700006

TABLE OF CONTENTS

ABSTRACT

Health is very important for human life. In particular, the health of the brain, which is the executive of vital resources, is very important. Diagnosis for human health is provided by **magnetic resonance imaging (MRI)** devices, which help health decision makers in critical organs such as brain health. Images from these devices are a source of big data for artificial intelligence. This big data enables high performance in image processing classification problems, which is a subfield of artificial intelligence.

In this study, we aim to detect brain tumors from brain MRI images. 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. We can use a Deep Learning architecture **CNN (Convolution Neural Network)** generally known as **NN (Neural Network)** for detecting the brain tumor. CNN architecture is very important for human health in early diagnosis and rapid treatment of such diseases. The performance of the model is to predict from image if tumor is present or not in image. If the tumor is present it returns **yes** otherwise returns **no**.

INTRODUCTION

1.1 Brain Tumor Detection System

The healthcare industry has been rapidly transformed by technological advances in recent years, and an important component of this transformation is **Artificial intelligence (AI)** technology. AI is a computer system that simulates human-like intelligence and has many applications in medicine. One such area is the fight against brain tumors. Brain tumors are a major public health problem in the healthcare sector, and accurate diagnosis, treatment, and follow-up processes are critical. AI has become an important tool for improving these processes and has great potential for early diagnosis and treatment of brain tumors.

Brain tumors affect human health due to their location. AI is designed to help diagnose and treat complex diseases such as brain tumors by combining technologies such as **Big data analytics**, **Machine learning**, and **Deep learning**. AI has the ability to detect and classify tumors by analyzing brain imaging techniques, such as **Magnetic Resonance Imaging (MRI)**. AI algorithms can help determine the size, location, class, and aggressiveness of tumors. This helps physicians make a more accurate diagnosis and treatment plan, and helps patients better understand their health.

AI can also be used to track a patient's progress through treatment. AI-based analytics can be used to assess treatment response and predict potential tumor recurrence. In this way, patients' treatment plans can be more effectively organized and individualized treatment approaches can be developed.

- ## Overview of Brain and Brain Tumor

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

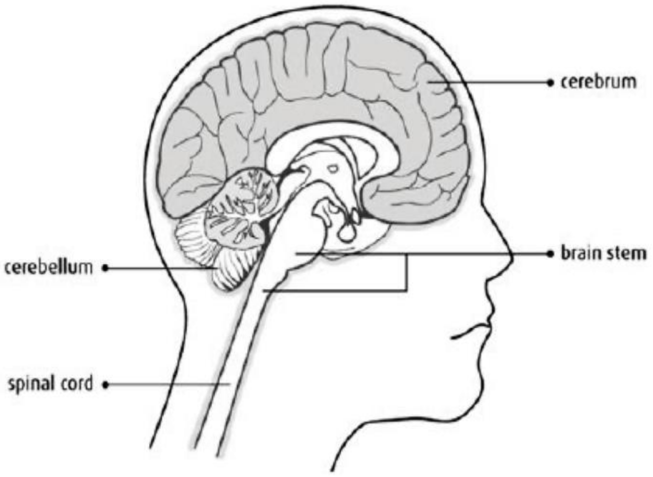


Figure 1.1: Basic Structure of Human Brain

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 that 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, the brain stops working properly. The secondary tumors are more aggressive and more quick to spread into other tissue.

Secondary brain tumors originate through other parts of the body. These types of tumors have a cancer cell in the body that is **metastatic** which spreads into different areas of the body like brain, lungs etc. Secondary brain tumor is very malignant. The reason for secondary brain tumors is mainly due to lung cancer, kidney cancer, bladder cancer etc.

• Magnetic Resonance Imaging (MRI)

Raymond Damadian invented the first **magnetic image** in **1969**. In 1977 the first MRI images were invented for the human body in the most perfect technique. Because of MRI we are able to visualize the details of the internal structure of the brain and from that we can observe the different types of tissues of the human body. MR images have a better quality as compared to other medical imaging techniques like X-ray and computer tomography. MRI is a good technique for knowing the brain tumor in the human body. There are different images of MRI for mapping tumor induced Change including T1 weighted, T2 weighted and **FLAIR** (Fluid attenuated inversion recovery) shown in **Fig 1.2**.

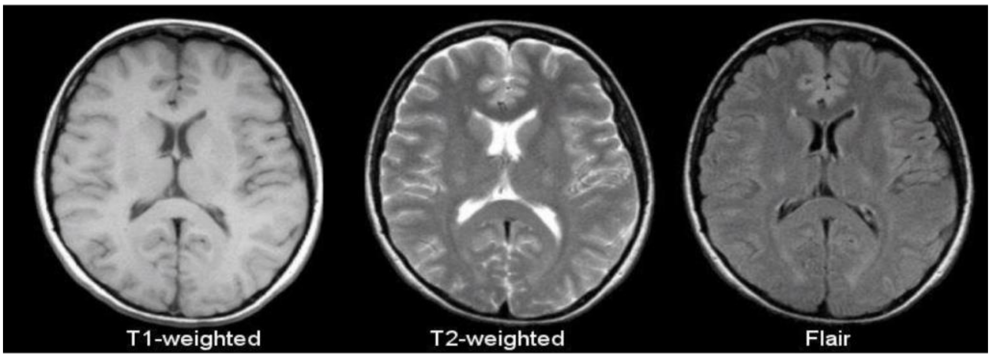


Figure 1.2: T1, T2 and Flair Image

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 and 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 1.3**.

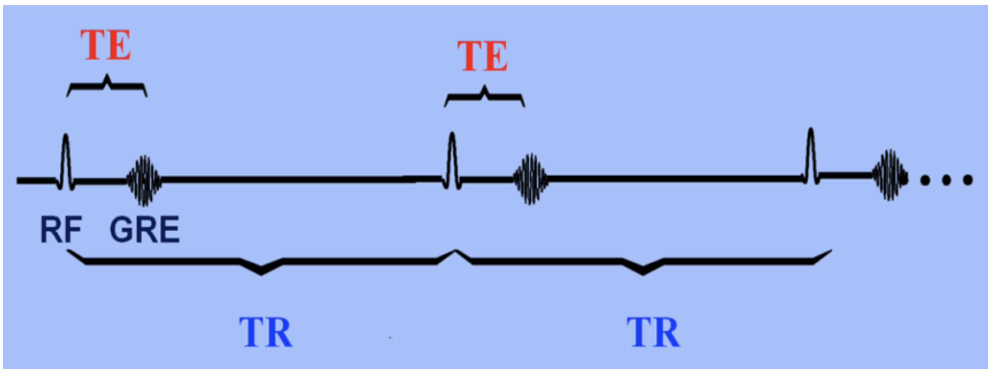
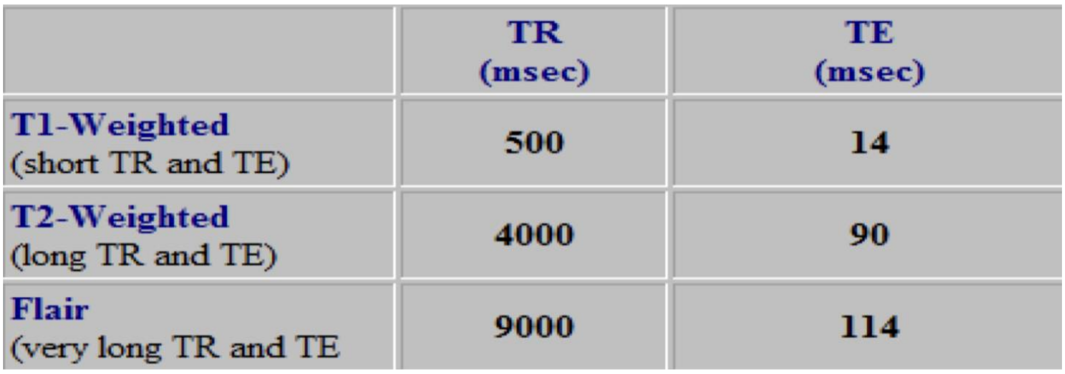


Figure 1.3: Graph of TE and TR

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

Figure 1.4: Table of TR and TE time

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 patients.
- The manual identification is not so fast, more accurate and efficient for the user. To overcome those problems this application is designed.
- It is a user-friendly application.

1.3 Objective

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

1.4 Motivation

The primary point here is to carry out a method of programmed detection of neoplasm utilizing **Convolutional Neural Networks (CNN)** utilizing MRIs as tests. CNN is utilized to make the model and train it utilizing past tumor patients' records and utilizing that model to anticipate a substitution picture if it's infected or not.

METHODOLOGY

2.1 Problem Statement

Brain Tumor detection is one of the principal vital and challenging errands inside the landscape of the clinical picture; preparing as a human-helping manual characterization may bring about inaccurate forecasts and analysis. Moreover, it's an exasperating errand when there's a larger-than-usual measure of information present to be helped. Since brain tumors have a wide variety for all intents and purposes, and there is a similitude between tumors, and typical tissues, a productive and powerful framework is required for extracting tumor districts from pictures.

2.2 Domain Description

In this study, difference detection was performed on brain images. Detection was performed with **CNN**, a type of **Deep Learning** algorithm.

The contribution of the study is as follows.

- We investigate the transfer learning method with the highest performance in the detection process of transfer learning methods on brain images.
- We investigate the performance of CNN and transfer learning on brain images using **CNN as a multi-layer without using transfer learning**.
- We investigate whether it is possible to achieve good results with a skewed and poor quality dataset.

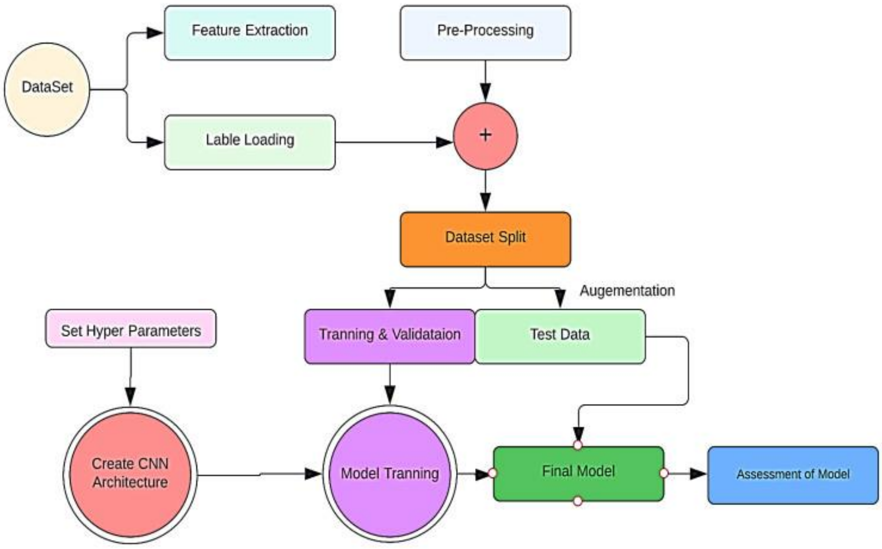


Figure 2.1: Flow diagram of proposed methodology for tumor detection.

2.2.1 Deep Learning

Deep learning is a subset of machine learning that focuses on training artificial neural networks to perform complex tasks by learning patterns and representations directly from data. Unlike traditional machine learning approaches that require manual feature engineering, deep learning algorithms autonomously extract hierarchical features from data, leading to the creation of powerful and highly accurate models. In this study, a CNN architecture is employed.

2.2.2 Convolutional Neural Network

Convolutional neural networks represent a major breakthrough in deep learning and computer vision. These architectures are specifically designed to extract meaningful features from complex visual data, such as images and video. The inherent structure of the CNN, consisting of convolutional layers, pooling layers, and fully connected layers, mimics the ability of the human visual system to recognize patterns and hierarchical features. Convolutional layers use convolutional operations to detect local features, which are then progressively abstracted by pooling layers that condense the information. The resulting hierarchical representations are then fed into fully connected layers for classification or regression tasks. CNN have redefined the landscape of image recognition, achieving remarkable success in diverse domains ranging from image classification and object detection to face recognition and medical image analysis.

2.3 Proposed Neural Network Architecture

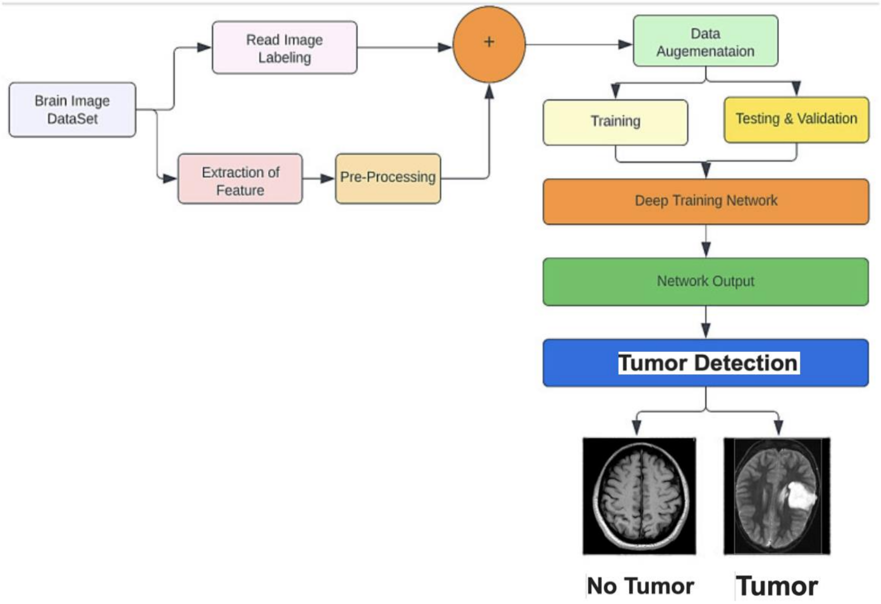


Figure 2.2: Proposed model progressing in its level

Using the convolutional layer in conjunction with the **Fledgling Layer**, an information state of the MRI images of 64 × 64 × 3 is created, converting each snapshot into a homogenous measurement. After assembling all images at the same angle, we used 32 convolutional channels, each measuring 3 × 3, and three channel tensors to produce a convolutional component entangled with the information layer. **ReLU** does not need to prove the yield since it is utilized as an enactment task. Reduce the piece of bounds and computational season of the organization by logically abbreviating the spatial size of the representation in this **CNN Model**. The pollution of overfitting may also be costly to the Brain MR image, and our Max Pooling layer flawlessly works for this distinction. We employ **MaxPooling2D** as the model for spatial information that proves our information picture. This convolutional layer uses measures of 32 x 32 x 64. The pool size is (2, 2) due to the separation of the information images in both spatial dimensions, which recommends a tuple of two integers to downscale in an upward direction and to a level plane. A pooled highlight map is acquired after the pooling layer. After pooling, one of the most critical layers is leveling. It is essential for planning since we must improve the framework by addressing the information photos in a single vector section. The Neural Network was then tasked with processing.

We used two layers that were entirely intertwined. **Dense-1** and **Dense-2** addressed the thick layer. In **Keras**, the thick capacity is used to handle the Neural Network and the gotten vector functions as a contribution to this layer. Inside the hidden layer, there are 128 hubs. We kept it as low as possible since the number of measurements or hubs we need is proportional to the processing assets we need to match our model, and 128 hubs produce the most liberal outcome. Because the beginning job requires improved intermingling execution, **ReLU** is used. Following the first thick layer, the model's final layer was created using the second entirely associated layer.

The proposed system has mainly five modules. Data Source, Preprocessing, Data Split the data, Build CNN model train Deep Neural network for epochs, and detection. In dataset we can take multiple MRI images and take one as input image. In preprocessing image to encoded the label and resize the image. In split the data we set the image as 62.5% Training Data, 12.5% Testing Data and 25% Validation Data. Then build CNN model and train deep neural network for epochs. Then classify the image as yes or no if tumor is positive then it returns yes and the tumor is negative the it returns no.

2.3.1 Data Source

The dataset consists of a total of 3060 human brain MRI images systematically classified into two different categories: **'NO' (no tumor)** and **'YES' (tumor)** .

- 'NO' and 'YES' each contain 1500 images.
- Availability link of used dataset, "Br35H :: Brain Tumor Detection 2020": https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection

2.3.2 Data Preprocessing

The following image preprocessing techniques are used to enhance the image quality:

1) **Rescaling:** Rescaling the image to a desired size is often one of the first steps in preprocessing. This step is essential to ensure that the image has a consistent size, which makes it easier to process.

2) **Denoising:** Removing noise from the image can improve its visual quality and make it easier for the algorithm to process. There are various techniques for denoising an image, including **Gaussian** and **Median Filtering**.

3) **Color Correction:** Adjusting an image's brightness, contrast, and saturation can improve its visual quality and make it easier for the algorithm to identify patterns and features.

4) **Rotations and translations:** Images may need to be rotated or translated to align with a reference image. It is crucial in applications such as medical imaging, where the images must be registered accurately to obtain meaningful results.

5) **Image segmentation:** Image segmentation is the process of dividing an image into multiple regions, or segments, based on the similarity of pixel values. It is an essential step in object recognition and classification tasks.

The following preprocessing procedures were conducted for each image:

Remove the area of the image that solely depicts the brain (which is the image's most fundamental piece). Adjust the image's size such that (64,64, 3) = (picture width, picture height, and the number of channels) is the state: since the dataset's photos arrive in different sizes. As a result, all images should have an indistinguishable form to ensure that they are taken care of as a contribution to the neural network. Use standards to scale pixel values from 0 to 1.

2.3.3 Data Split

The information was divided into the following categories:
Information for training = **62.5%**
Information for validation = **25%**
Information for testing = **12.5%**.

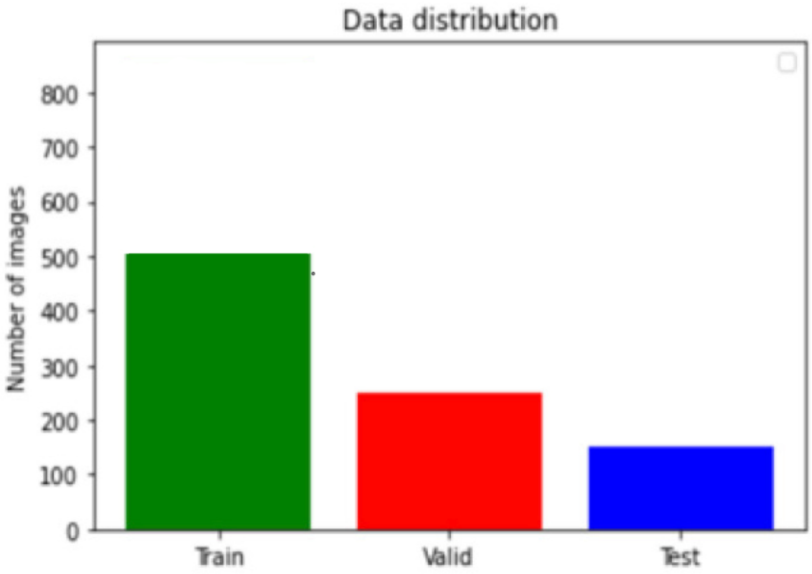


Figure 2.3: Dataset Distribution

CNNS contains different layers. Each layer has a different function.

- **Convolution Layer:** This is where the learning process occurs; it computes the complications between the neurons and the different patches in the input. Image data is stored in a **4D Tensor** (A tensor is a multidimensional array of components that describe functions relevant to the coordinates of a space), usually processed by **2D Convolutional Layers**.

- **Pooling Layer:** This layer decreases the network's count of parameters (weights). A model that fits the training data too well is said to overfit. The model becomes so familiar with the details of the data and the noise in the training data that it performs poorly on the new data. This layer also increases the network's dependability. The pooling layer conserves the essential characteristics while reducing the size of the image and is mainly placed between two convolution layers.

- **Flattening Layer:** Neural networks can learn only 1D data; this layer converts the **2D data** (tensor/array) into **1D data**.

- **Fully Connected Layer:** Here, the input image from the previous layer is fed to the FC layer. This is the last layer placed before the output layer, which comprises the weights along the network's neurons. This layer provides help in connecting the neurons between two different layers. A nonlinear combination of these parameters can be learned by attaching a fully connected layer that is feasible and cost-effective.

- **Activation Function:** The activation functions are significant parameters of the convolutional neural network model. There are several activation functions, such as SoftMax, ReLU, Sigmoid functions, etc.

2.3.4 Working of CNN Model

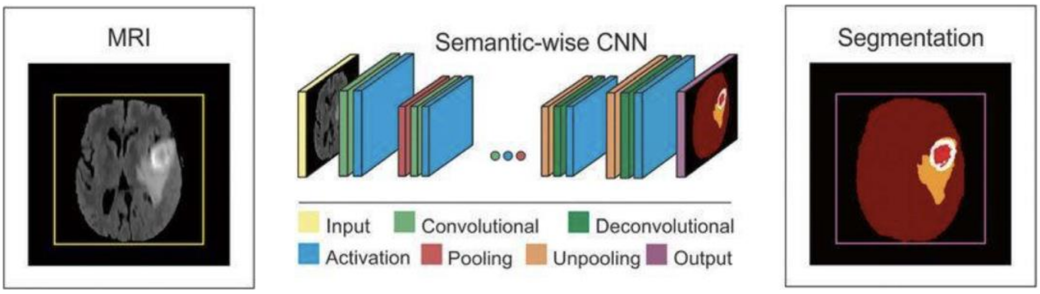


Figure 2.4: Working of CNN for brain tumor detection

Layers of CNN model:

- ★ Convolution 2D
- ★ MAX Pooling2D
- ★ Dropout
- ★ Flatten
- ★ Dense
- ★ Activation

➢ **Convolution 2D:** In the Convolution 2D extract the featured from input image. It gives the output in matrix form.

➢ **MAX Pooling 2D:** In the MAX pooling 2D it takes the largest element from rectified feature map.

➢ **Dropout:** Dropout is randomly selected neurons that are ignored during training.

➢ **Flatten:** Flatten feeds 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 is followed by a nonlinear activation function.

➢ **Activation:** It uses **ReLU** function and predict the probability 0 and 1.

- ➔ In the compile model we used binary cross entropy because we have two layers and 1.
- ➔ We used Adam optimizer in compile model.

➢ **Adam (Adaptive moment estimation):** It is used for non convex optimization problem like straight forward to implement.

- ➔ Computationally efficient.
- ➔ Little memory requirement.

2.3.4.1 ReLU (Rectified Linear Unit) correction layer

Applying any activation function to the output of the preceding layer is the duty of the ReLU Correction Layer. It adds nonlinearity to the network. These activation functions are used to learn and find the accuracy of the continuous and complex relationships between network variables.

Usually defined as ReLU(i) = max (0, i). Visually represented in **Figure 2.5**.

$$\mathbf{f(i) = max(0, j)}$$

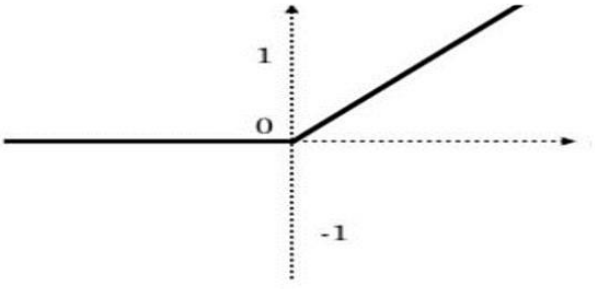


Figure 2.5: ReLU Function

2.3.4.2 SoftMax Layer

SoftMax extends ideas into a multiclass world related to cross-entropy functions. This layer intends to test the model's efficiency by employing a loss function and the cross-entropy (used to measure the randomness in the information being processed) function to maximize the network's performance.

$$s_i = \frac{exp(a_i)}{\sum_{b=1}^{b=n} exp(b_i)}$$

For i = 1, 2, 3…….., n.

Any neural network's final activation function, the SoftMax function, is often employed to normalize the output of a network.

IMPLEMENTATION AND RESULT

3.1 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 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 a straight answer to all questions.

  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.

- **Noise Removal and Sharpening:** Unwanted data of element are removed using filter and image can be sharpened and black and white gray scale image is used as an input.

- **Threshold:** Thresholding is a process of image segmentation. It converts the grayscale image into binary image.

- **React:** React is the library for web and native user interfaces. Build user interfaces out of individual pieces called components written in JavaScript. It was used to build the UI for this project. This was a straightforward call for many reasons.

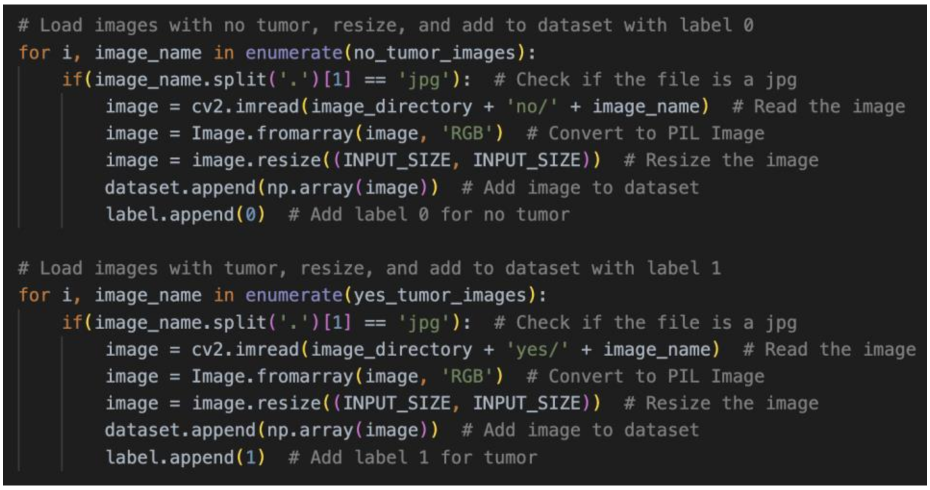
  1. React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

  2. Declarative views make your code more predictable and easier to debug.

3. React can also render on the server using Node and power mobile apps using React Native

3.2 Results

● Give the label of the image



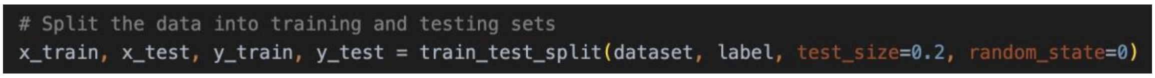
```
# Load images with no tumor, resize, and add to dataset with label 0
for i, image_name in enumerate(no_tumor_images):
    if(image_name.split('.')[1] == 'jpg'):  # Check if the file is a jpg
        image = cv2.imread(image_directory + 'no/' + image_name)  # Read the image
        image = Image.fromarray(image, 'RGB')  # Convert to PIL Image
        image = image.resize((INPUT_SIZE, INPUT_SIZE))  # Resize the image
        dataset.append(np.array(image))  # Add image to dataset
        label.append(0)  # Add label 0 for no tumor

# Load images with tumor, resize, and add to dataset with label 1
for i, image_name in enumerate(yes_tumor_images):
    if(image_name.split('.')[1] == 'jpg'):  # Check if the file is a jpg
        image = cv2.imread(image_directory + 'yes/' + image_name)  # Read the image
        image = Image.fromarray(image, 'RGB')  # Convert to PIL Image
        image = image.resize((INPUT_SIZE, INPUT_SIZE))  # Resize the image
        dataset.append(np.array(image))  # Add image to dataset
        label.append(1)  # Add label 1 for tumor
```

Figure 3.1: Label of the image

These outputs in images are resized and label names are given to all images.

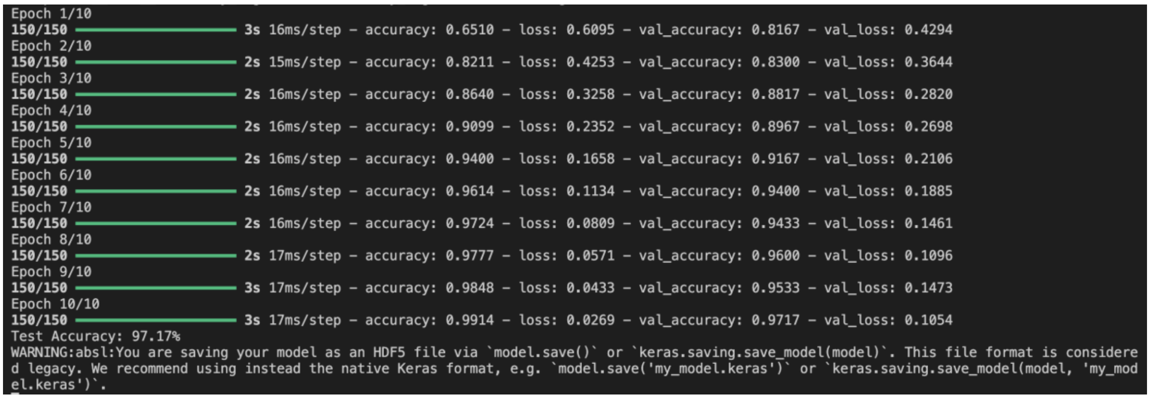
● Split Data



```
# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(dataset, label, test_size=0.2, random_state=0)
```

Figure 3.2: Split the Image Data

● Train Data

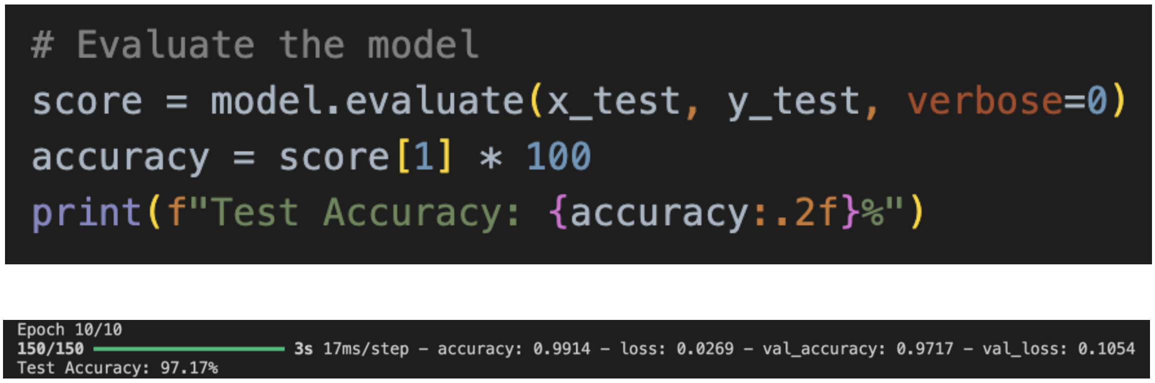


```
Epoch 1/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 3s 16ms/step - accuracy: 0.6510 - loss: 0.6095 - val_accuracy: 0.8167 - val_loss: 0.4294
Epoch 2/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.8211 - loss: 0.4253 - val_accuracy: 0.8300 - val_loss: 0.3644
Epoch 3/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - accuracy: 0.8640 - loss: 0.3258 - val_accuracy: 0.8817 - val_loss: 0.2820
Epoch 4/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - accuracy: 0.9099 - loss: 0.2352 - val_accuracy: 0.8967 - val_loss: 0.2698
Epoch 5/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - accuracy: 0.9400 - loss: 0.1658 - val_accuracy: 0.9167 - val_loss: 0.2106
Epoch 6/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - accuracy: 0.9614 - loss: 0.1134 - val_accuracy: 0.9400 - val_loss: 0.1885
Epoch 7/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - accuracy: 0.9724 - loss: 0.0809 - val_accuracy: 0.9433 - val_loss: 0.1461
Epoch 8/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 2s 17ms/step - accuracy: 0.9777 - loss: 0.0571 - val_accuracy: 0.9600 - val_loss: 0.1096
Epoch 9/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9848 - loss: 0.0433 - val_accuracy: 0.9533 - val_loss: 0.1473
Epoch 10/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9914 - loss: 0.0269 - val_accuracy: 0.9717 - val_loss: 0.1054
Test Accuracy: 97.17%
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considere
d legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_mod
el.keras')`.
```

Figure 3.3: Train CNN Image Data

Figure 3.3 consists output of trained convolutional neural network

- **Test Data**



```
# Evaluate the model
score = model.evaluate(x_test, y_test, verbose=0)
accuracy = score[1] * 100
print(f"Test Accuracy: {accuracy:.2f}%")
```

```
Epoch 10/10
150/150 ━━━━━━━━━━━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9914 - loss: 0.0269 - val_accuracy: 0.9717 - val_loss: 0.1054
Test Accuracy: 97.17%
```

Figure 3.4: Test CNN Image Data

Figure 3.4 consists output of convolutional neural network testing accuracy score **97.17%**

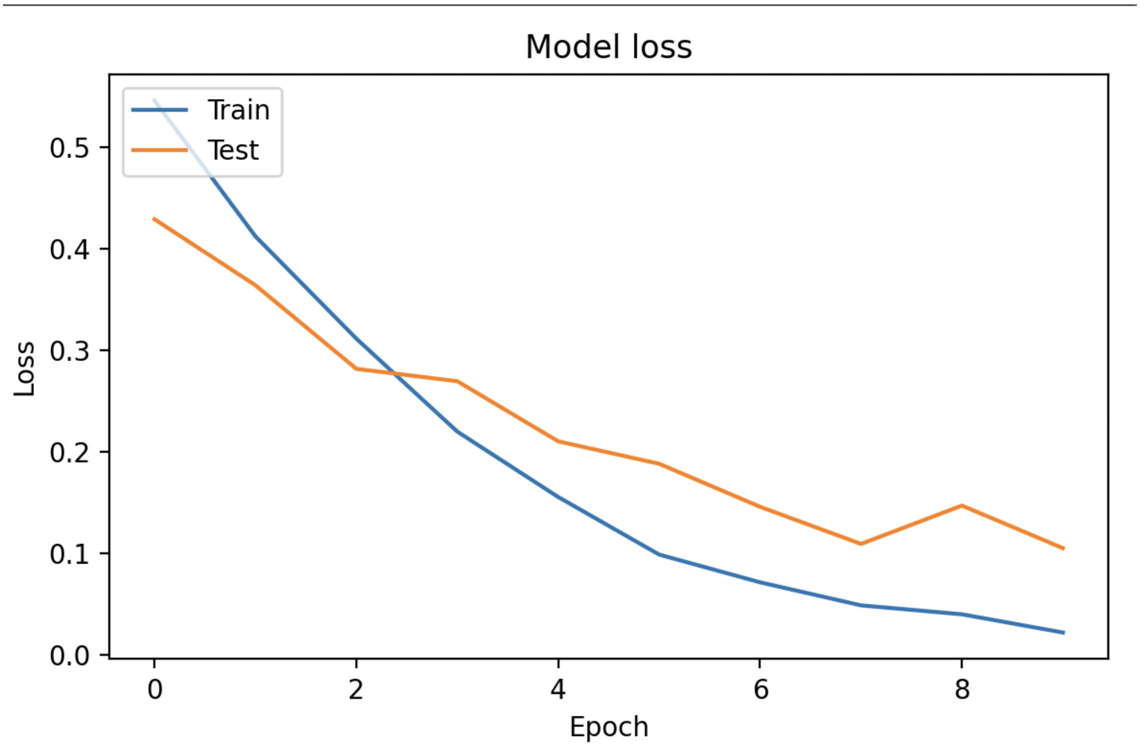


Figure 3.5: Training and Testing performance of loss.

Figure 3.5 shows the comparative analysis of loss during training and validation of the proposed model

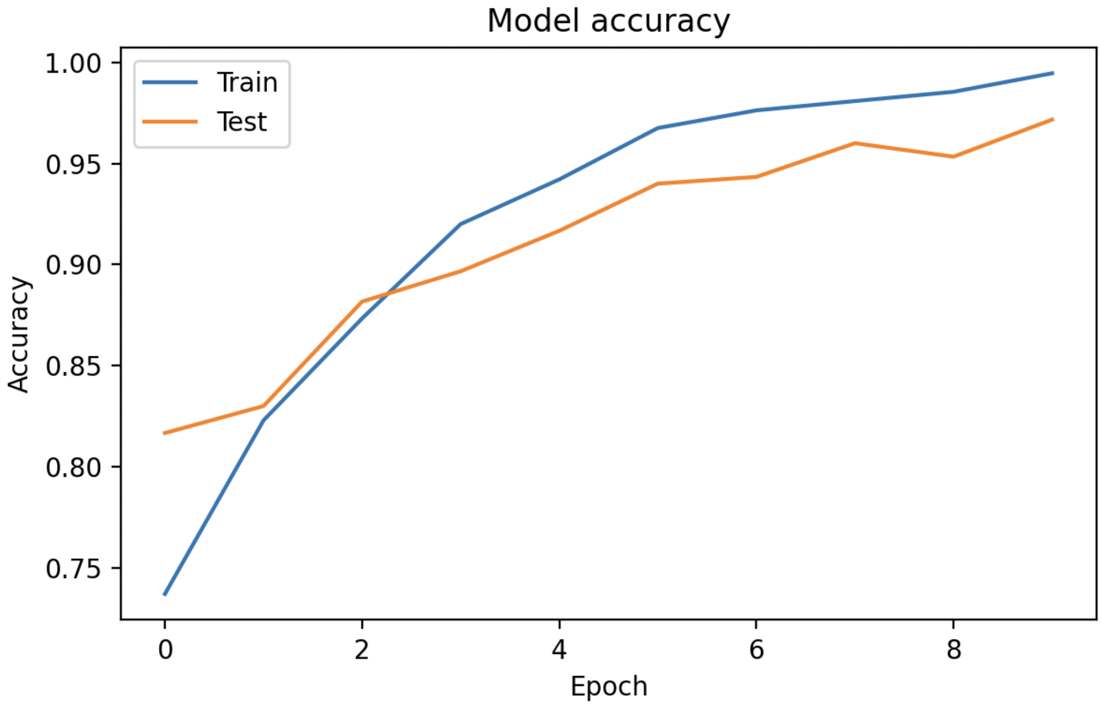


Figure 3.6: Training and Testing performance of accuracy.

Figure 3.6 shows the comparative analysis of accuracy during training and validation of the proposed model.

- **UI Design**

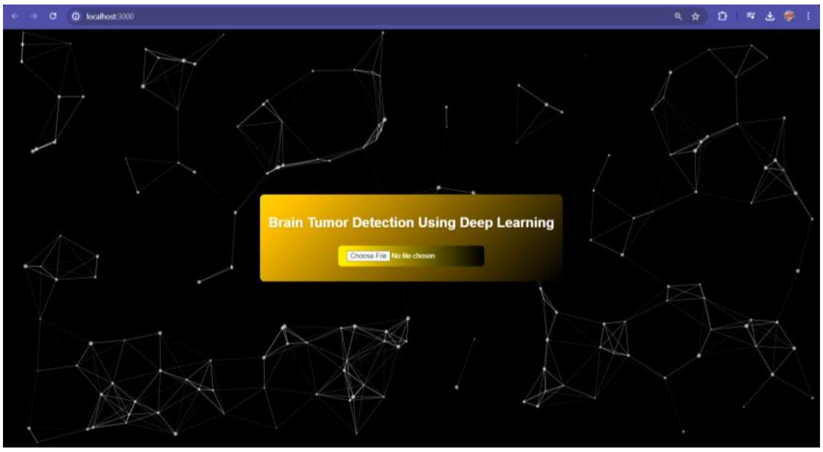


Figure 3.7: Landing Page of Our Website.

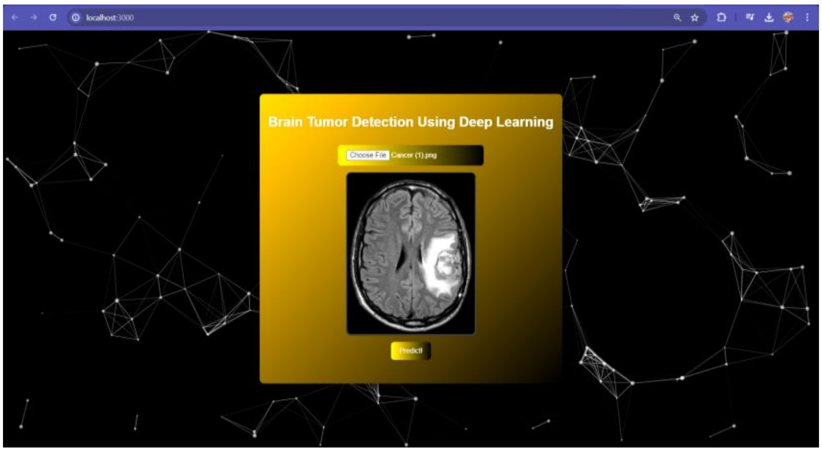
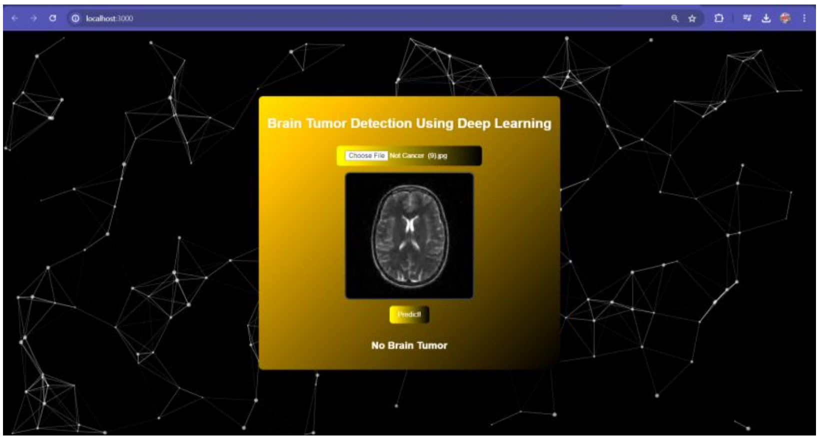
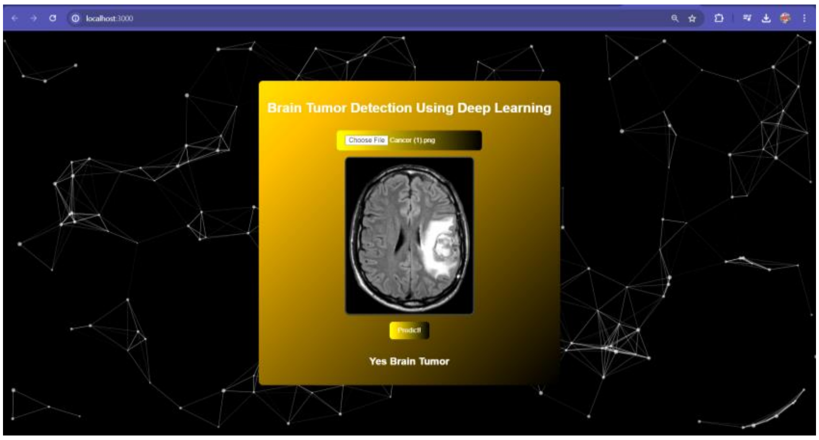


Figure 3.8: After Uploading an Axial MRI Scan.



No Brain Tumor

Figure 3.9: Prediction of an Axial MRI not having Tumor.

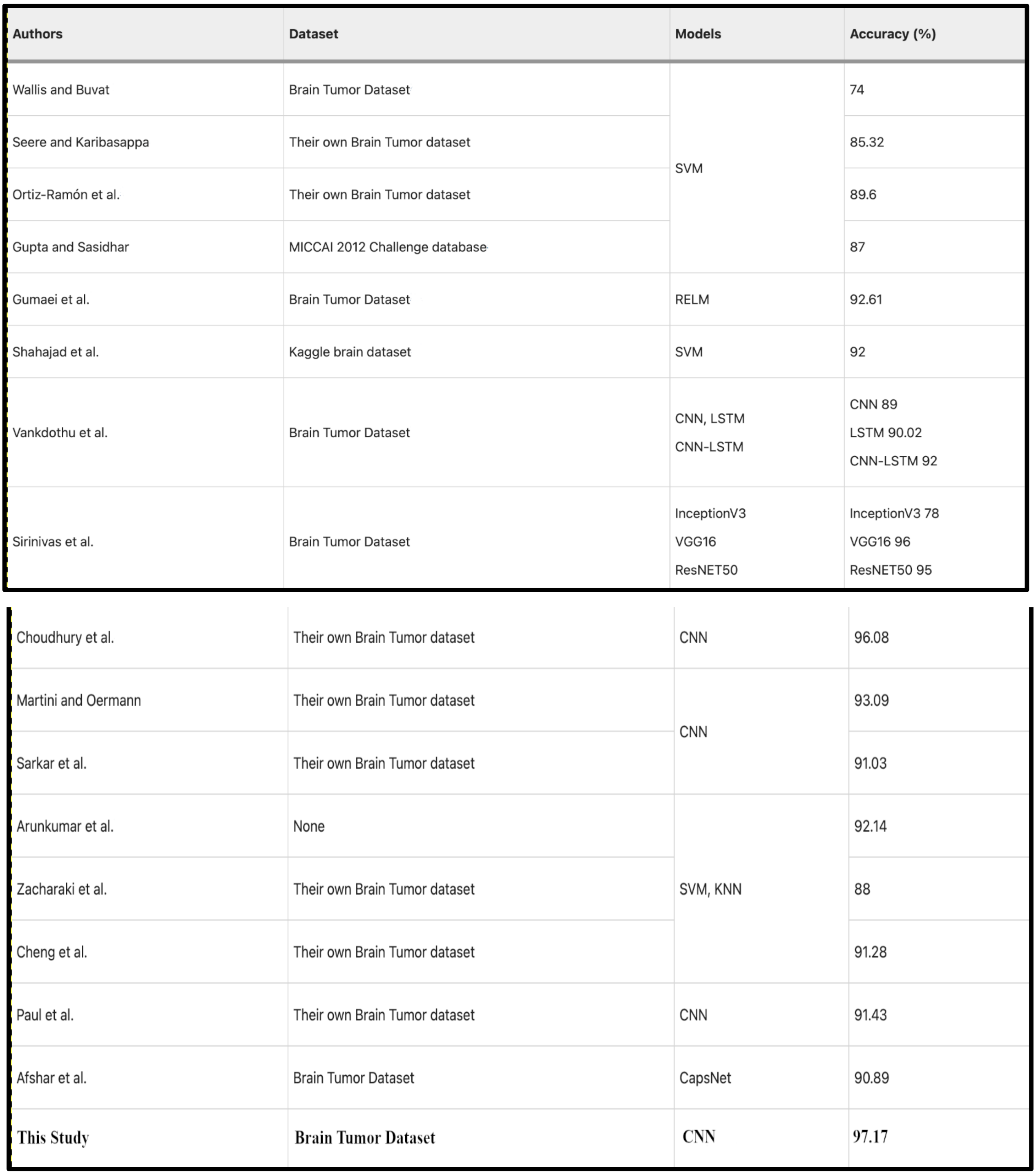


Yes Brain Tumor

Figure 3.10: Prediction of an Axial MRI having Tumor.

DISCUSSION

Figure 4.0 shows that our proposed work outperformed in comparison to the existing methods and achieved more accuracy.



Authors	Dataset	Models	Accuracy (%)
Wallis and Buvat	Brain Tumor Dataset	SVM	74
Seere and Karibasappa	Their own Brain Tumor dataset		85.32
Ortiz-Ramón et al.	Their own Brain Tumor dataset		89.6
Gupta and Sasidhar	MICCAI 2012 Challenge database		87
Gumaei et al.	Brain Tumor Dataset	RELM	92.61
Shahajad et al.	Kaggle brain dataset	SVM	92
Vankdothu et al.	Brain Tumor Dataset	CNN, LSTM CNN-LSTM	CNN 89 LSTM 90.02 CNN-LSTM 92
Sirinivas et al.	Brain Tumor Dataset	InceptionV3 VGG16 ResNET50	InceptionV3 78 VGG16 96 ResNET50 95
Choudhury et al.	Their own Brain Tumor dataset	CNN	96.08
Martini and Oermann	Their own Brain Tumor dataset	CNN	93.09
Sarkar et al.	Their own Brain Tumor dataset		91.03
Arunkumar et al.	None	SVM, KNN	92.14
Zacharaki et al.	Their own Brain Tumor dataset		88
Cheng et al.	Their own Brain Tumor dataset		91.28
Paul et al.	Their own Brain Tumor dataset	CNN	91.43
Afshar et al.	Brain Tumor Dataset	CapsNet	90.89
This Study	**Brain Tumor Dataset**	**CNN**	**97.17**

Figure 4.0: Comparisons with previous studies on brain tumor

LIMITATION

With our motivation to investigate how it will work in single **CNN** and **Multilayer CNN** based transfer learning models, we subjected the dataset to classification as it is without rotation and cropping operations, which is the most important limitation of our study.

Also the CNN model provided for this study performs and results in the most accurate manner when the frontal brain MR image is presented. Otherwise in case of brain MR images of different angles, the CNN model doesn’t always detect the presence or absence of brain tumor in the most accurate way.

FUTURE SCOPE

AI in healthcare plays an important role in the management of **complex diseases** such as **brain tumors**. AI enables faster, more accurate, and more effective diagnosis and treatment processes. However, AI technology is not intended to completely replace doctors, but to support and enhance their work. To realize the full potential of AI, it is important to consider issues such as ethics, security and privacy. In the future, AI-based solutions will continue to contribute to better management of brain tumors and other health problems, and improve the quality of life for patients. As seen in this study, AI-based studies will increase their importance to human health, from early diagnosis to positive progress in the treatment process.

Based on the results of this study, **CNN** methods should be preferred especially in image processing-based applications to support health decision makers. The data obtained from **MRI** or **CT** can be used as an early warning system to help health decision makers make quick and accurate decisions. Therefore, in addition to empirical analysis, AI-based applications should take a more active role as soon as possible. To this end, the diagnosis of diseases from instant CT or MR images will be investigated in the coming years.

CONCLUSION

A large number of researchers have examined a wide variety of algorithms with the goal of accurately detecting and classifying brain cancers in a quick and efficient manner. **Deep learning (DL)** makes it possible to use a **Convolutional Neural Network (CNN)** model that has already been pre-trained for the analysis of medical images, particularly for the categorization of brain tumors. The fundamental objective of this research is to develop an improved model to make brain tumor diagnosis more accurate.

In this paper, a **Convolutional Neural Network (CNN)** based on a **Brain Tumor Detection Model** was proposed. Cancers such as carcinoma, glioma, meningioma, and pituitary tumors are becoming increasingly common worldwide, and our proposed framework can aid in the early detection of these dangerous infections in other clinical areas related to medical imaging. Carcinomas, which originate in the epithelial cells, are one of the most common types of cancer and include subtypes such as adenocarcinoma and squamous cell carcinoma. Gliomas are a type of tumor that occurs in the brain and spinal cord, arising from glial cells. Meningiomas develop in the meninges, the protective membranes covering the brain and spinal cord. Pituitary tumors, although often benign, can impact hormone levels and overall health.

We will apply this methodology to other logical domains in addition to the current debate over the availability of massive data, or we can use alternative transfer learning strategies with the same proposed approach. This methodology is not limited to brain tumors but can also be extended to the detection and classification of other cancers such as leukemia, lymphoma, melanoma, sarcoma, and breast cancer. Leukemia is a type of cancer that affects blood-forming tissues, including the bone marrow and lymphatic system. Lymphoma originates in the lymphatic system and includes Hodgkin and non-Hodgkin lymphoma. Melanoma is the most serious type of skin cancer, developing in the melanocytes. Sarcomas are a diverse group of cancers arising in the bones and soft tissues. Breast cancer originates in the cells of the breast and is one of the most common cancers among women.

However, not every task is perfect in this development field, and further improvement may be possible in this application. Continuous advancements in AI and DL can lead to the refinement of these models, increasing their accuracy and reliability. Incorporating multi-modal data, such as combining imaging data with genetic information, can also enhance diagnostic capabilities.

We have learned many things and gained a lot of knowledge about the development field through this research. The integration of AI in medical imaging is a promising area that holds the potential to revolutionize early cancer detection and improve patient outcomes.

REFERENCES

[1] https://www.nature.com/articles/s41598-024-52823-9#Fig1

[2] https://aimspress.com/article/doi/10.3934/era.2023146?viewType=HTML

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9854739/

[4] https://bvmengineering.ac.in/NAAC/Criteria1/1.3/1.3.4/18CP814_Thesis.pdf

[5] https://legacy.reactjs.org/

APPENDIX

1	AI	Artificial Intelligence
2	DL	Deep Learning
3	NN	Neural Network
4	CNN	Convolutional Neural Network
5	MRI	Magnetic Resonance Imaging
6	TR	Time Repetition
7	TE	Pulse Sequence Parameter
8	FLAIR	Fluid-Attenuated Inversion Recovery
9	FC	Fully Connected Layer
10	Adam	Adaptive Moment Estimation
11	ReLU	Rectified Linear Unit
12	UI	User Interface
13	CT	Computed Tomography
14	SVM	Support Vector Machine
15	KNN	K-Nearest Neighbor
16	VGG16	Visual Geometry Group
17	LSTM	Long Short-Term Memory