BRAIN TUMOR DETECTION AND CLASSIFICATION USING CNN

Submitted In Partial Fulfillment of The Requirements for The Degree Of

Bachelor of Technology in **Electrical Engineering**

By

Siddhesh Jadhav - E2010028

Vaibhavi Kuche - E2010032

Nidhi Shah - E2010061

Guided by:

Dr. Anupa Sabnis

Associate Professor,
Electrical Engineering Department,
SPCE - Mumbai

Department of Electrical Engineering

SARDAR PATEL COLLEGE OF ENGINEERING

(An Autonomous Institute Affiliated to University of Mumbai)

MUNSHI NAGAR, ANDHERI (WEST) - 400058

(Academic Year 2023-24)

REPORT APPROVAL SHEET

This report entitled "Brain Tumor Detection and Classification Using CNN" by Siddhesh Jadhav, Vaibhavi Kuche and Nidhi Shah has been approved for the degree of Bachelor of Technology in Electrical Engineering.

EXAMINERS:

______________________	______________________
Dr. Sangeeta Daingade	Prof. Matilda Justin
Electrical Engg. Dept., Sardar Patel College of Engineering, Mumbai	Electrical Engg. Dept., Sardar Patel College of Engineering, Mumbai

SUPERVISOR:

____

Dr. Anupa Sabnis
Electrical Engineering Department,
Sardar Patel College of Engineering, Mumbai

HEAD OF DEPARTMENT:

____

Dr. Sangeeta Daingade
Electrical Engineering Department,
Sardar Patel College of Engineering, Mumbai

PRINCIPAL:

____

Dr. M.M. Murudi
Sardar Patel College of Engineering, Mumbai.

DATE: ________________
PLACE: SPCE, Mumbai

BONAFIDE CERTIFICATE

This is to certify that the project on “Brain Tumor Detection and Classification Using CNN” is a bonafide work presented by Siddhesh Jadhav, Vaibhavi Kuche and Nidhi Shah as a partial fulfillment for award of **DEGREE IN BACHELOR OF TECHNOLOGY** in **ELECTRICAL ENGINEERING** as laid down by S.P.C.E. during the academic year 2023-24.

Supervisor:

Dr. Anupa Sabnis,
Associate Professor,
Electrical Engineering Department,
Sardar Patel College of Engineering, Mumbai

DECLARATION

We declare that this written submission represents our ideas in our own words and in places where other's ideas or words are included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penalty action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____

Siddhesh Jadhav

(E2010028)

____

Vaibhavi Kuche

(E2010032)

____

Nidhi Shah

(E2010061)

Date: __________

PLACE: SPCE, Mumbai

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. Anupa Sabnis, our project guide, Dean Academics for her continuous support, guidance, advice and motivation throughout the project.

We express our gratitude towards The Principal Dr. M.M. Murudi, Head of the department Dr. Sangeeta Daingade and all the faculty members of the Electrical Engineering Department, SPCE Mumbai.

We would also like to express our gratitude towards Dr. Sangeeta Daingade and Prof. Matilda J., who were part of our Project evaluation panel for their insightful comments and encouragement for the project.

We express sincere appreciation to all contributors, whose collective efforts greatly enriched this report. Your commitment and expertise were pivotal in achieving comprehensive insights. Special thanks for your valuable feedback and support throughout the process.

ABSTRACT

A brain tumor is an abnormal collection of cells in or around your brain wherein cells grow and multiply uncontrollably. Any such growth in a constrained area like your skull can be potentially dangerous. It is a major health risk in adults and can even lead to death. When it comes to detection and identification of brain tumors, Magnetic Resonance Imaging (MRI) plays a very important role. Going through the MRIs manually to detect a tumor is a challenging and exhausting practice. With the advancements in technology, we propose a system that utilizes machine learning to detect a brain tumor from MRI scans and identify it with maximum accuracy to solve the existing issue. A Convolutional Neural Network (CNN) is deployed as the base of our model. A VGG-16 CNN architecture is used to provide better results without compromising accuracy. To improve the training of our proposed model, various data augmentation methods have been used on an accessible brain tumor dataset. The results verify the efficiency of our suggested model, which achieves higher accuracy than the previously existing methods. As a result, this scheme has a lot of potential as a useful decision-making tool for healthcare professionals in the field of diagnosing brain tumors.

Keywords: Brain Tumor, Convolutional Neural Network (CNN), Magnetic Resonance Imaging (MRI), Brain tumor detection, Machine learning, Deep learning

INDEX

Table of Contents

Acronym

CNN	Convolutional Neural Networks
MRI	Magnetic Resonance Imaging
CNS	Central Nervous System
ICMR	Indian Council of Medical Research
IACR	International Association of Cancer Registries
GUI	Graphical User Interface
VGG 16	Visual Geometry Group 16
FCNN	Fully Connected Neural Network
CT	Computed Tomography
ReLU	Rectified Linear Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
ResNets	Residual Networks
CMS	Content Management System
UI	User Interface
UX	User Experience
JWT	JSON Web Token
HMR	Hot Module Replacement
AWS	Amazon Web Services
IAM	Identity and Access Management

List of Figures

List of Tables

CHAPTER-1

INTRODUCTION

1.1 Brain Tumor

A human brain is an organ that forms the core of our nervous system and is the largest cluster of neurons in the body. It is located in the head, usually near organs for special senses such as vision, hearing and olfaction. It is the most energy-consuming organ of the body, and the most specialized one, responsible for performing various crucial tasks. The structure of a human brain is extremely complex. Illnesses like dementia, brain tumors, stroke, migraine, infections, multiple sclerosis and many more are examples where the diagnosis and treatment are very challenging.

A brain tumor is a collection of cells that grow or multiply uncontrollably in an abnormal fashion. These cells can be cancerous (malignant) or noncancerous (benign). Any such abnormal growth in a very restricted space in the skull, which encloses the brain, can cause damage to the brain and can be very dangerous as it increases the pressure inside the skull.

There are many types of tumors depending on the size, texture and location. There are more than 100 types of tumors that can affect the brain directly or indirectly. These are broadly classified as:

1) Primary brain tumors: These are the tumors that originate in the brain. This type of brain tumor can be malignant or benign.
2) Secondary brain tumors: Tumors that begin in one area of the body and progress to the brain. These are cancerous and fatal. Some examples are breast cancer, kidney cancer, or skin cancer.

A study indicated that approximately 85–90 percent of notable tumors in the central nervous system (CNS) are attributed to brain tumors. Early detection plays a crucial role in substantially reducing the mortality rate associated with these tumors. Healthcare professionals have extensively employed medical imaging as a means of identifying tumors, with magnetic resonance imaging (MRI) emerging as one of the widely preferred techniques for the early diagnosis of brain tumors.

1.1.1 Signs and symptoms:

The signs and symptoms of a brain tumor vary based on its size and location within the brain. Additionally, these symptoms can be influenced by the tumor's growth rate, also referred to as its grade. Some common signs and symptoms of brain tumors may include:

- Headache or pressure in the head (usually worse in the morning)

- Nausea or vomiting
- Eye problems (such as blurry vision, seeing double)
- Losing feeling or movement in an arm or a leg
- Trouble balancing
- Memory problems

1.1.2 Global trends of brain tumor:

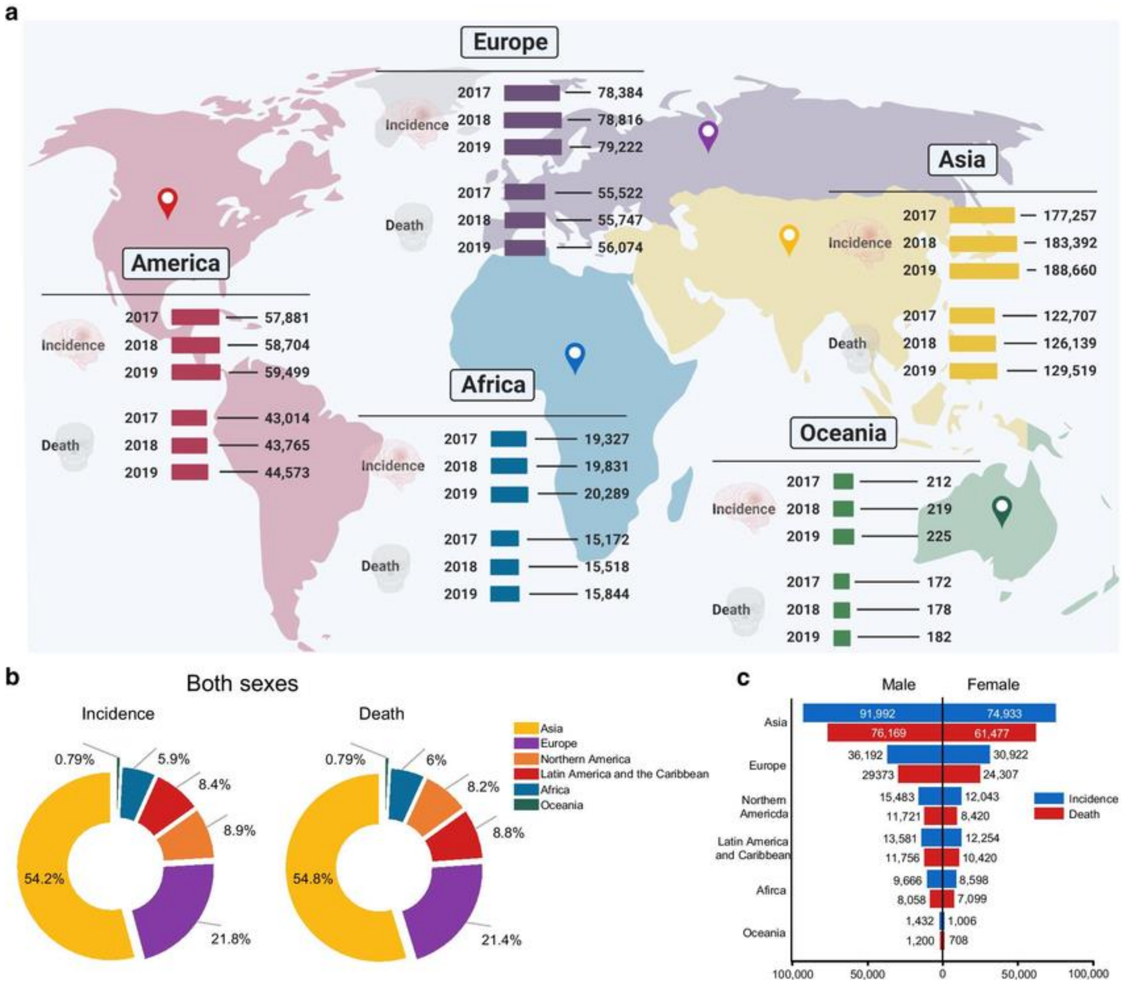


Figure 1.1

(a) A global map illustrating the incidence and mortality rates associated with brain tumors worldwide from 2017 to 2019. (b) Pie charts display the distribution of brain tumor incidence and mortality across major world regions in 2020, encompassing all sexes and age groups. (c) A bar graph illustrates the incidence of brain tumors and the associated mortality rates in 2020 for both males and females. This data is sourced from global health data exchange and GLOBALCAN 2020 at the Global Cancer Observatory (GCO).

1.2 Problem Statement

Traditional methods for brain tumor detection suffer from inefficiencies, leading to delayed diagnoses and potential inaccuracies. There is a need to address issues related to time-consuming manual processes, limited precision in current methods, and the critical need for prompt diagnosis in brain tumor detection and classification. This project aims to address the need for a swift and accurate solution by leveraging Convolutional Neural Networks (CNNs). The project will work towards developing an automated system that enhances the speed and accuracy of brain tumor detection and classification, ultimately contributing to improved patient outcomes.

1.3 Objectives and Scope

1.3.1 Objectives and goals

- To develop a Convolutional Neural Network (CNN) model for the detection of brain tumors in medical images.
- To implement a classification system to categorize tumors into different types.
- To achieve a high level of accuracy in tumor detection and classification.
- To develop a web application that is easy to use and accessible to a wide range of users.

1.3.2 Scope of the project

- Prepare and refine the data for analysis.
- Investigate data patterns and relationships.
- Enhance the dataset with additional examples.
- Create and design the model(s).
- Train and evaluate model performance.
- Develop and deploy the website online.

1.4 Methodology

The entire project can be viewed as a two-step process -

1. Development of CNN Model: The CNN (Convolutional Neural Network) model will serve as the backend, responsible for image classification and analysis.
2. Development of Web Application: Web application will provide a user-friendly interface for interacting with the model and visualizing its results.

This integrated approach will require the seamless interaction of the CNN model and web application, ensuring real-time processing and an engaging user experience.

The methodology is divided into a few important stages. First, we collect our data from an available online resource (kaggle.com), then this dataset is pre-processed. Then various machine learning models are compared to give us an effective and accurate model in order to train our data. Our dataset was split into groups for training and testing purposes. Then, in order to validate our findings, we will consider several types of metrics including the accuracy and loss. Once the model is ready, website designing will be done using Figma and then the web development will take place where the model will be integrated with the backend.

Model Building Pipeline

1) Data analysis

2) Data augmentation (Image Rotation, Brightness adjustments)

3) Data Generator (Given a list of paths to images, and the labels, this function augments the images, normalizes them, encodes the label, and then returns the batch on which the model can train on.)

4) Model Development (CNN)

5) Train the model

6) Evaluate

Website development Pipeline

1) Design on Figma

2) Make the backend on Django

3) Develop the server + backend database on MongoDB atlas

4) Frontend development using Vite.js + tailwind CSS

5) Integrate the frontend with the backend

6) Deploy on AWS

1.5 Project Workflow

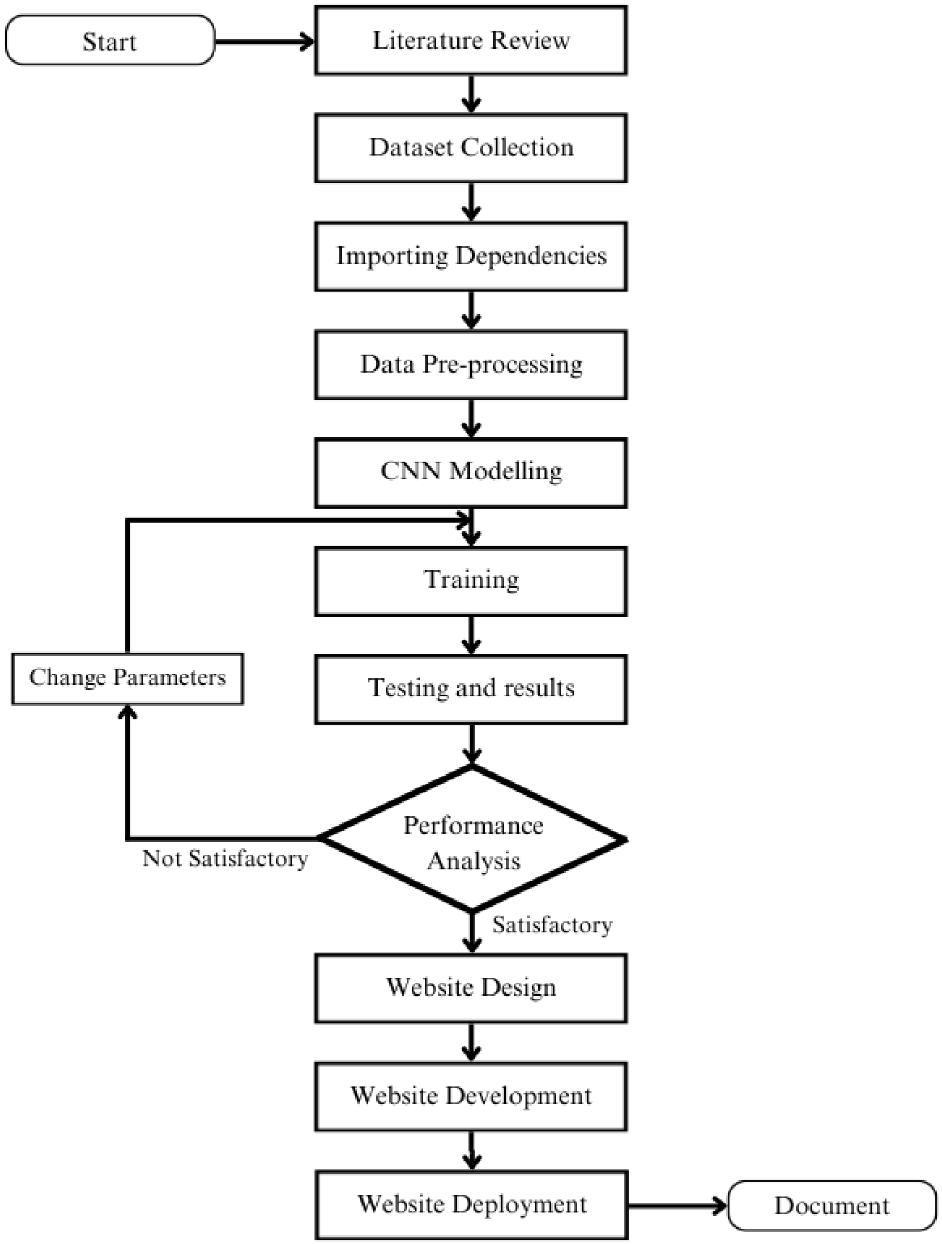


Figure 1.2 Project Workflow

CHAPTER-2

LITERATURE REVIEW

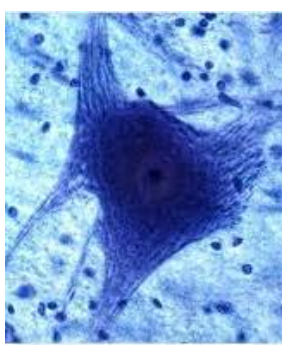
In this section, the existing literature related to our topic has been reviewed. This helps to understand what has already been studied and discovered by other researchers. This section explores various approaches, methodologies, and findings from previous studies to establish a foundation for understanding the current state of research and identifying gaps that the current study aims to address. Given below is a description of the study of various tumor types, models, training and testing techniques, evaluation criteria, and platforms used for website design.

2.1 Types of Brain Tumor

There exist various types of brain tumors. The type of brain tumor depends upon the kind of cells that constitute the tumor. These tumor cells under special lab testing can give information about the cells. This information can then be used by the healthcare professionals to identify the type of brain tumor.

2.1.1 Gliomas and related brain tumors:

Gliomas are cell growths in the brain that resemble glial cells (in the central nervous system). The glial cells connect and support nerve cells in the brain tissue. Types of gliomas and related brain tumors include astrocytoma, glioblastoma, oligodendroglioma and ependymoma. While gliomas can be benign, the majority are malignant. Gliomas represent about 33% of all brain cancers, with glioblastoma being the most common type of malignant brain tumor.

(a) 

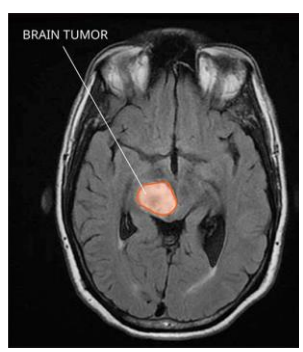
(b) 

Figure 2.1 (a) Glial Cell (b) Glioma tumor in the brain

2.1.2 Meningiomas:

Brain tumors that originate in meninges, the outer three layers of tissue between the skull and the brain that cover and protect the brain just under the skull are known as meningiomas (shown in fig. 2.2(a)). Meningiomas are generally benign, but sometimes they can be malignant. These are the most common types of benign brain tumors.

2.1.3 Pituitary:

Tumors that originate in and around the pituitary gland are referred to as pituitary tumors. This small gland is situated near the base of the brain (shown in fig. 2.2(b)). Most tumors occurring in and around the pituitary gland are benign. These tumors develop within the pituitary gland itself. Craniopharyngioma is a type of brain tumor that occurs near the pituitary gland.

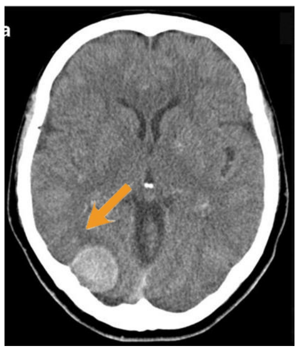
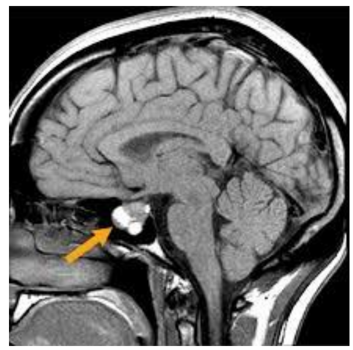
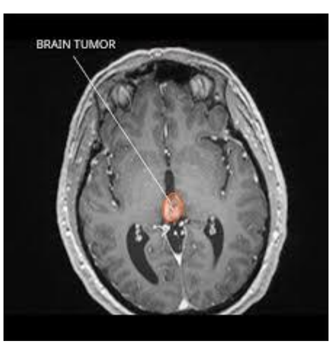
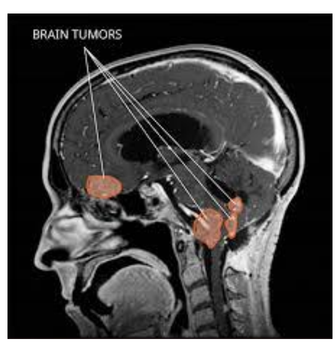
(a)  (b) 

Figure 2.2 (a)Meningioma (b) Pituitary

2.1.4 Other brain tumors:

Various other rare tumors can develop in and around the brain. These tumors can start building up in the muscles, blood vessels, and connective tissues surrounding the brain, as well as in the bones of the skull. These can be pineal tumors which originate in and around the brain's pineal gland, choroid plexus tumors that grows in cells that make the fluid that surrounds the brain and spinal cord and germ cell tumors that start in reproductive cells, called germ cells, that go on to become the sperm and egg cells. Germ cells are mostly in the ovaries and testicles, but sometimes they're in other parts of the body, including the brain.

(a) 

(b) 

(c) 

Figure 2.3 (a) Pineal tumor (b) Choroid plexus tumor (c) Germ cell tumor

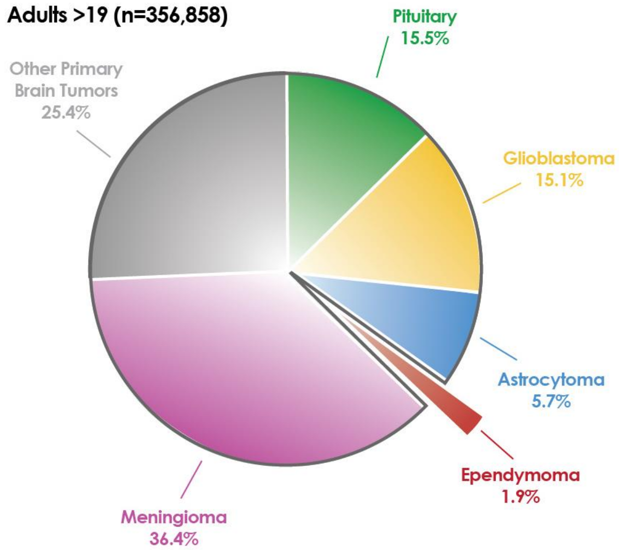


Figure 2.4 Distribution of Adult primary brain tumors, sourced from CBTRUS Statistical Report from 2008-2012

2.2 Imaging Tests

2.2.1 Magnetic Resonance Imaging (MRI):

Magnetic Resonance Imaging (MRI) is a medical imaging technique that uses a strong magnetic field and radio waves (computer generated) to produce detailed images of organs and tissues within the body. It can differentiate between white and gray matter in the brain and can be used to diagnose aneurysms and tumors. As MRI does not use x-rays or other radiation, it is the preferred imaging technique when frequent imaging is required for diagnosis or therapy, especially in the brain.

Thus, the MRI technique is becoming more and more popular as a solution to address the limitations of human diagnosis. Their high-resolution and detailed images allow for accurate detection and characterization of tumors.

2.2.2 CT (Computed Tomography) Scan:

Computed tomography, or CT, is a computerized x-ray imaging technique where a narrow beam of x-rays rotates around the body. Signals from these x-rays are processed by a computer to create cross-sectional images, known as tomographic slices. These slices provide clinicians with more detailed information compared to traditional x-rays. By digitally combining (stacking) successive slices, a three-dimensional (3D) image of the patient is formed. This 3D image facilitates easier identification of anatomical structures and potential tumors or abnormalities.

2.3 Convolutional Neural Network (CNN):

In the past few decades, deep learning has established itself as a very powerful tool as it possesses the ability to handle large amounts of data. A Convolutional Neural Network is a type of deep learning algorithm specifically created for image recognition and processing tasks. It is made up of different layers like the convolutional layer, pooling layer and many more. These layers are where the filters are applied to the input image to extract its features, spatial dimensions are reduced and the important information is retained to make a prediction or identify the image. Training of CNNs involves a large dataset of labeled images being fed to the network where it learns to recognize patterns and features associated with specific classes. This concept is applied to a vast range of applications like self-driving cars, medical imaging and security systems. The initial application of CNN in the healthcare industry dates back to the 1980s. Different CNN models include ResNet, AlexNet, VGG, etc.

2.3.1 Layers of CNN:

The convolutional neural network is a multi-layered feed-forward neural network, made by assembling many layers on top of each other in a particular order. These layers are:

- **Input Layer:** This is where we provide input to the model. In CNNs, the input is typically an image or a series of images. This layer contains the raw input data of the image, which typically has a width of 32, a height of 32, and a depth of 3.
- **Convolutional Layers:** These layers function by applying convolutional operations to input images by using filters to capture features like edges, textures, and other complex patterns. These operations help preserve the spatial relationships between the pixels. The filters are smaller matrices, typically with dimensions of 2×2, 3×3, or 5×5. They move across the input image data and compute the dot product between the filter weight and the corresponding patch of input image.
- **Pooling Layers:** These layers are used to decrease the spatial dimensions of the input, thereby lowering the computational load and the number of parameters in the network. This makes the computation fast, reduces memory and also avoids overfitting. Two common types of pooling layers are max pooling and average pooling. Max pooling, a popular pooling method, selects the maximum value from a set of neighboring pixels.

- **Activation Functions:** Non-linear activation functions, like the Rectified Linear Unit (ReLU), add non-linearity to the model by applying element-wise activation functions, enabling it to capture more complex patterns in the data.
- **Fully Connected Layers:** These layers make predictions using the high-level features extracted by the preceding layers. They connect each neuron in one layer to every neuron in the subsequent layer.
- **Output Layer:** The output from the fully connected layers is passed into a logistic function, such as sigmoid or softmax, for classification purposes. This function converts the outputs into probability scores for each class.

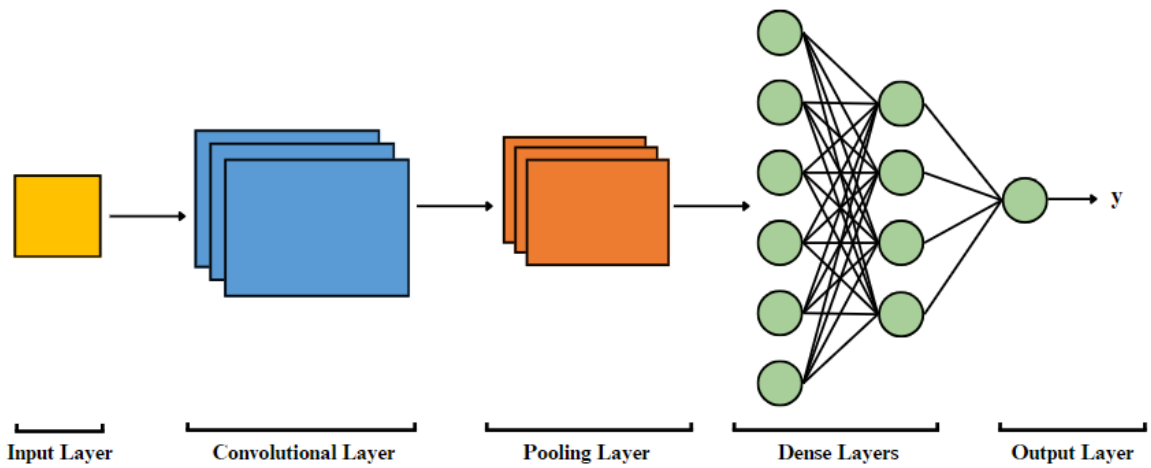
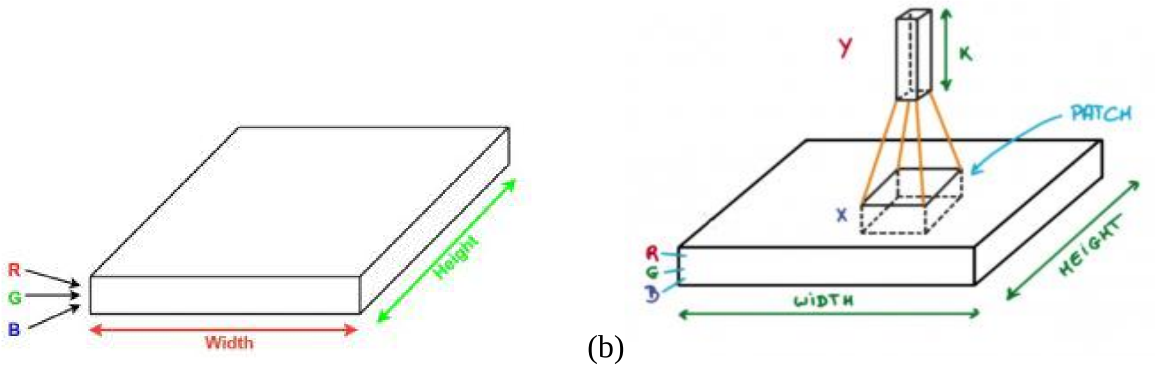


Figure 2.5 Different Layers of CNN

2.3.2 Working of convolutional layers:

Convolutional Neural Networks (CNNs or covnets) are neural networks where parameters are shared. Imagine an image as a cuboid with dimensions representing its length, width (dimensions of the image), and height (typically for red, green, and blue channels). Then consider a small patch of this image, processed by a small neural network called a filter, which outputs K vertical representations. By sliding this neural network across the entire image, a new image is generated with altered width, height, and depth dimensions. This process is known as convolution, which increases the number of channels beyond just R, G, and B, but with reduced width and height. If the patch size matches the image size, it functions like a regular neural network but with fewer weights due to the smaller patch size.

(a) 

(b)

Figure 2.6 (a) Image representation (b) Patch being processed by a filter

2.4 CNN Models

2.4.1 LeNet:

LeNet is an early convolutional neural network (CNN) architecture created by Yann LeCun and his team in the late 1990s. It was designed specifically for recognizing handwritten digits and became one of the first successful CNNs for image recognition. The LeNet architecture comprises two sets of convolutional and pooling layers, each followed by a subsampling layer, and three fully connected layers. Trained on the MNIST dataset, which includes 70,000 images of handwritten digits, LeNet achieved high recognition accuracy. Despite being simpler than modern architectures, LeNet laid the groundwork for many subsequent CNNs and is considered a classic model for image recognition tasks.

2.4.2 AlexNet:

AlexNet is a convolutional neural network (CNN) architecture created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012. It was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a prestigious image recognition competition, which solidified CNNs as a powerful tool for image recognition. The AlexNet architecture consists of five convolutional layers, three pooling layers, and three fully connected layers. It was trained on the ImageNet dataset, containing 1.2 million images across 1000 classes, and achieved high recognition accuracy. AlexNet demonstrated that CNNs could significantly outperform traditional machine learning methods in image recognition tasks, paving the way for deeper architectures like VGGNet, GoogleNet, and ResNet.

2.4.3 ResNet:

ResNets (Residual Networks) are a type of deep learning algorithm especially effective for image recognition and processing tasks. They are known for their capability to train very deep networks without overfitting. ResNets have attained state-of-the-art results on various key point detection benchmarks, including the COCO Key point Detection Challenge and the MPII Human Pose Estimation Dataset.

2.4.4 GoogleNet:

GoogleNet, also known as InceptionNet, is renowned for achieving high accuracy in image classification tasks while using fewer parameters and computational resources compared to other leading CNNs. It employs global average pooling to reduce the size of feature maps before passing them to the fully connected layers, enhancing performance on image classification tasks. Additionally, GoogleNet uses factorized convolutions to lower the number of parameters and computational resources needed for training. This makes GoogleNet a powerful tool for image classification, applicable in a wide range of tasks, such as categorizing images of cats and dogs, cars and trucks, and various flowers and animals.

2.4.5 MobileNet:

MobileNets are a type of CNN specifically designed for image recognition and processing tasks on mobile and embedded devices. They are notable for achieving high accuracy in image classification tasks while utilizing fewer parameters and computational resources compared to other advanced CNNs. MobileNets are also used for key point detection tasks and have achieved state-of-the-art results on various key point detection benchmarks.

2.4.6 VGG-16:

VGG-16 is a type of convolutional neural network (CNN) known for its simplicity and effectiveness. It typically consists of a series of convolutional and pooling layers, followed by several fully connected layers. VGG networks are used in self-driving cars to detect and classify objects on the road, such as other vehicles, pedestrians, and traffic signs, aiding in safe navigation. VGGs are a strong and reliable tool for image processing and recognition tasks.

2.5 Different ways to train and test the CNN model

2.5.1 Train-Test Split:

It divides the dataset into two sets, one for training and one for testing.

Advantages: Simple and quick to implement.

Disadvantages: Performance evaluation might be biased if the split is not representative of the entire dataset.

2.5.2 K-Fold Cross-Validation:

This method splits the dataset into k subsets (folds). Train the model k times, each time using a different fold as the test set and the remaining folds as the training set.

Advantages: Provides a better estimate of model performance.

Disadvantages: Computationally expensive.

2.5.3 Stratified K-Fold Cross-Validation:

It is similar to K-Fold Cross-Validation but ensures each fold has the same proportion of classes as the entire dataset.

Advantages: Maintains class distribution, especially useful for imbalanced datasets.

Disadvantages: Computationally intensive, similar to K-Fold Cross-Validation.

2.6 Criteria for evaluation

After completing its training phase, a CNN is assessed using a separate test set composed of images it hasn't encountered before. How effectively the CNN performs on this test set serves as a reliable indicator of its performance on real-world data. The performance of a CNN in tasks such as image categorization can be evaluated using several metrics:

- Accuracy: The percentage of test images correctly classified by CNN.
- Precision: The percentage of test images predicted by the CNN as belonging to a specific class that actually belong to that class.
- Recall: The percentage of test images of a particular class that the CNN correctly identifies as belonging to that class.
- F1 Score: The harmonic mean of precision and recall, which is particularly useful for assessing CNN performance on classes that are imbalanced in the dataset.

2.7 Website design platforms

2.7.1 Figma

This is a powerful web-based design tool ideal for professional designers and teams. Figma excels in collaborative design, prototyping for user testing, and creating comprehensive design systems. However, it requires some design knowledge and might be overkill for simple websites.

2.7.2 WordPress

This is a popular content management system (CMS) that allows you to build and manage websites without extensive coding knowledge. WordPress offers a wide range of themes and plugins to customize your website's design and functionality. While user-friendly, complex design elements might require additional coding or theme customization.

2.7.3 Canva

This is a user-friendly graphic design platform perfect for creating basic website elements like banners, social media graphics, and presentations. Canva offers pre-designed templates and drag-and-drop functionality, making it easy for beginners to create visually appealing content. However, for complex website layouts and interactivity, Canva's capabilities might be limited.

2.7.4 Adobe Express

Similar to Canva, Adobe Express is a web-based design tool offering templates and drag-and-drop functionality for creating basic website graphics and social media content. It integrates well with other Adobe products and offers a freemium model. However, like Canva, extensive website design and interactivity might be challenging with Adobe Express.

CHAPTER - 3

CNN MODEL DEVELOPMENT

Developing an efficient and reliable CNN model for brain tumor detection is very crucial as it enhances early detection capabilities, improves diagnostic accuracy, optimizes healthcare resources and facilitates personalized patient care. CNNs have emerged as a powerful tool in medical image analysis, leveraging their ability to learn intricate patterns and features from images. This section explores various aspects of CNN model development, including dataset considerations, model architecture, training methodologies and evaluation metrics.

3.1 Dataset

The development of the Convolutional Neural Network (CNNs) model relies heavily on the availability and quality of datasets. These datasets are crucial for training and evaluating CNN models, as they provide the diverse examples needed for the network to learn and generalize effectively. Understanding these datasets is very essential in order to appreciate the capabilities and limitations of CNN models, highlighting their characteristics and importance in advancing the field of image recognition and processing.

With the intention of creating a robust model to detect the brain tumor from MRIs, there was a requirement of vast sets of images (MRI) to train the model successfully. The dataset has been sourced from available online platforms like kaggle.com. Multiple datasets were collected and studied. The final dataset is a combination of 3 different datasets- figshare, SARTAJ and Br35H integrated to enhance data completeness and diversity. The dataset roughly contains 7023 images which are classified into 4 categories namely Glioma, Meningioma, Pituitary and No Tumor, where the first three are different types of tumors. With this, a reasonably balanced dataset is obtained (as shown in figure 3.1).

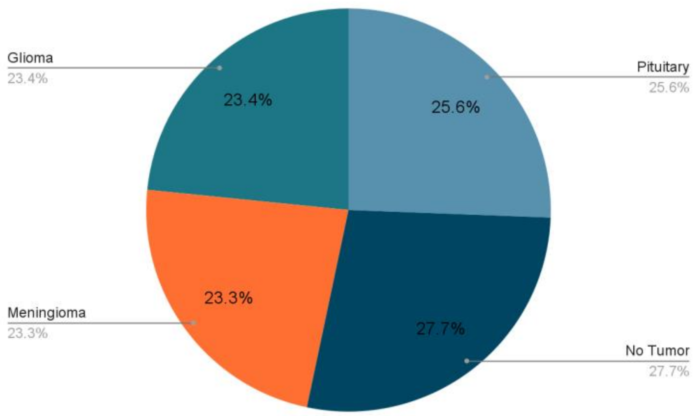


Figure 3.1 Classification of dataset

3.2 Importing the dependencies

Dependencies refer to the external libraries or frameworks that a project relies on to function properly. These dependencies often include specialized tools that provide essential functionalities for building and training neural networks. Dependencies are necessary to ensure that software or systems function correctly and reliably, they ensure that the different components of a software system work together seamlessly, their software remains secure against potential threats and standard development practices and methodologies are used across the entire project.

The following table (Table 3.1) shows the different dependencies deployed for different purposes in this model development.

Application	**Dependencies used**
ML Models	Tensorflow, keras
Data Manipulation	Numpy, pandas, sklearn
Data Visualization	Matplotlib, seaborn

Table 3.1 List of dependencies used

3.3 Data Augmentation

Data augmentation is a pre-processing method used on images before inputting them into a network. It involves creating additional data through random generation to enhance the robustness of a model. The primary goal of data augmentation is to expand the dataset, addressing challenges like over fitting and imbalanced classification. This technique employs diverse transformations like rotation, mirroring, zoom, etc. Over fitting arises during training, causing the model's outcome to be overly reliant on the input data. This is also done as the CNN is not capable of handling rotated images. Data augmentation is done in Keras, an open-source library in python.

Advantages of data augmentation:

- To prevent models from over fitting.
- The initial training set is too small.
- To improve the model accuracy.
- To lower the operational cost of labeling and cleaning the raw dataset.

Data augmentation techniques include:

- Rotation: Rotating the image by a certain angle.
- Flip: Flipping the image horizontally or vertically.
- Translation: Shifting the image horizontally or vertically.
- Scaling: Zooming in or out of the image.
- Brightness and Contrast Adjustment: Changing the brightness and contrast levels of the image.
- Noise Injection: Adding random noise to the image.

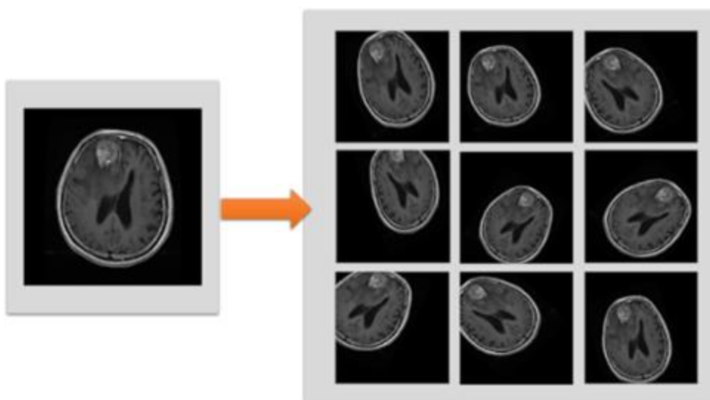


Figure 3.2 Data augmentation

3.4 CNN Model

We have discussed the various CNN models available in section 2.4. A brief comparison is done to select the suitable model that fits best for our purpose.

Model	Year	Architecture	Parameter	Applications	Key Features
LeNet	1998	Convolutional layers followed by fully connected layers	~60K	Handwritten digit recognition	Simple architecture
AlexNet	2012	Alternating	~60M	ImageNet	ReLU

		convolutional and pooling layers		Classification	activations, dropout, local response normalization
ResNet	2015	Residual Blocks	~25M	Image classification, object detection	Residual connections, skip connections
GoogleNet	2014	Inception modules	~7M	Deep feature extraction	Inception modules, global average pooling
MobileNet	2018	Depth wise separable convolutions	~3.4M	Mobile and embedded applications	Lightweight, efficient on mobile devices
VGG-16	2014	Very deep network with small 3x3 convolutional filters	~138M	ImageNet classification, object recognition	Uniform architecture, deep layers

Table 3.2 Comparison between different CNN Models

Here, the ‘Architecture’ gives a brief description of the model's layer structure and the ‘Parameters’ denotes the approximate number of parameters in millions (M) or thousands (K).

3.4.1 Finalized Architecture

Out of the six different models, three models were shortlisted. These are AlexNet, MobileNet and the VGG-16. Among these three VGG-16 was selected over the other two because of the following reasons:

- MobileNet is specifically crafted for mobile and edge devices, prioritizing lightweight design. However, it compromises accuracy for enhanced efficiency. Thus, in our scenario where accurate predictions of medical conditions are paramount, it is not suitable. The VGG-16 model places a higher emphasis on accuracy.
- AlexNet comprises 8 layers, whereas VGG16 consists of 16 layers and thus, is more effective and accurate.

3.5 Architecture of VGG-16:

VGG-16 (Visual Geometry Group 16) was suggested by the Visual Geometry Group at the University of Oxford. It is known for its simplicity and uniformity in architecture, making it easy to understand and implement.

Given below is a breakdown of the layer structure in VGG-16:

- Input layer
- Convolutional layers (13 layers)
- Max pooling layers (5 layers)
- Fully connected layers (3 layers)

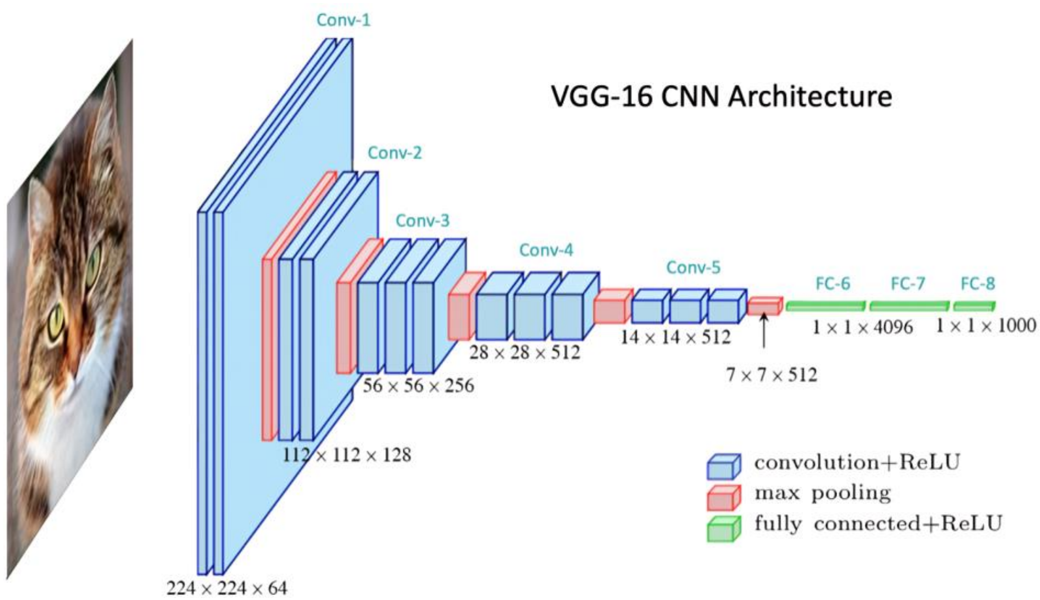


Figure 3.3 VGG-16 CNN Architecture

VGG-16 has three types of layers:

1) **Convolution Layer-** It is the core building block of CNN which carries the main portion of the network's computational load. This layer performs a dot product between two matrices.
2) **Pooling Layer-** This helps in lowering the spatial size of the representation, which reduces the required amount of computation and weights.
3) **Dense Layer/ Fully Connected (FC) Layer-** Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. The FC layer is used to map the representation between the input and the output.

3.6 Model Structure

The model structure illustrated in Figure 3.4(a) showcases a sophisticated Convolutional Neural Network (CNN) architecture. At its core is the VGG16 model, which acts as a pre-trained convolutional base. This layer processes input images, extracting rich feature representations, and outputs a shape of (4, 4, 512) with 14,714,688 parameters. Following this, a flattened layer reshapes the 4x4x512 feature maps into a single vector of 8192 elements, facilitating the transition from convolutional to fully connected layers without adding parameters. To prevent overfitting, a dropout layer is applied, maintaining the output shape of (8192) with no additional parameters.

Next, a dense layer with 128 neurons reduces the input vector's dimensionality, encapsulating essential features with 1,048,704 parameters. Another dropout layer is included for further regularization, keeping the output shape at (128) with no additional parameters. Finally, the model concludes with a dense layer of 4 neurons, which corresponds to the output classes of the classification task, adding 516 parameters.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14714688
flatten_2 (Flatten)	(None, 8192)	0
dropout_4 (Dropout)	(None, 8192)	0
dense_4 (Dense)	(None, 128)	1048704
dropout_5 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 4)	516

Total params: 15,763,908
Trainable params: 8,128,644
Non-trainable params: 7,635,264

Fig 3.4 (a) Model Structure

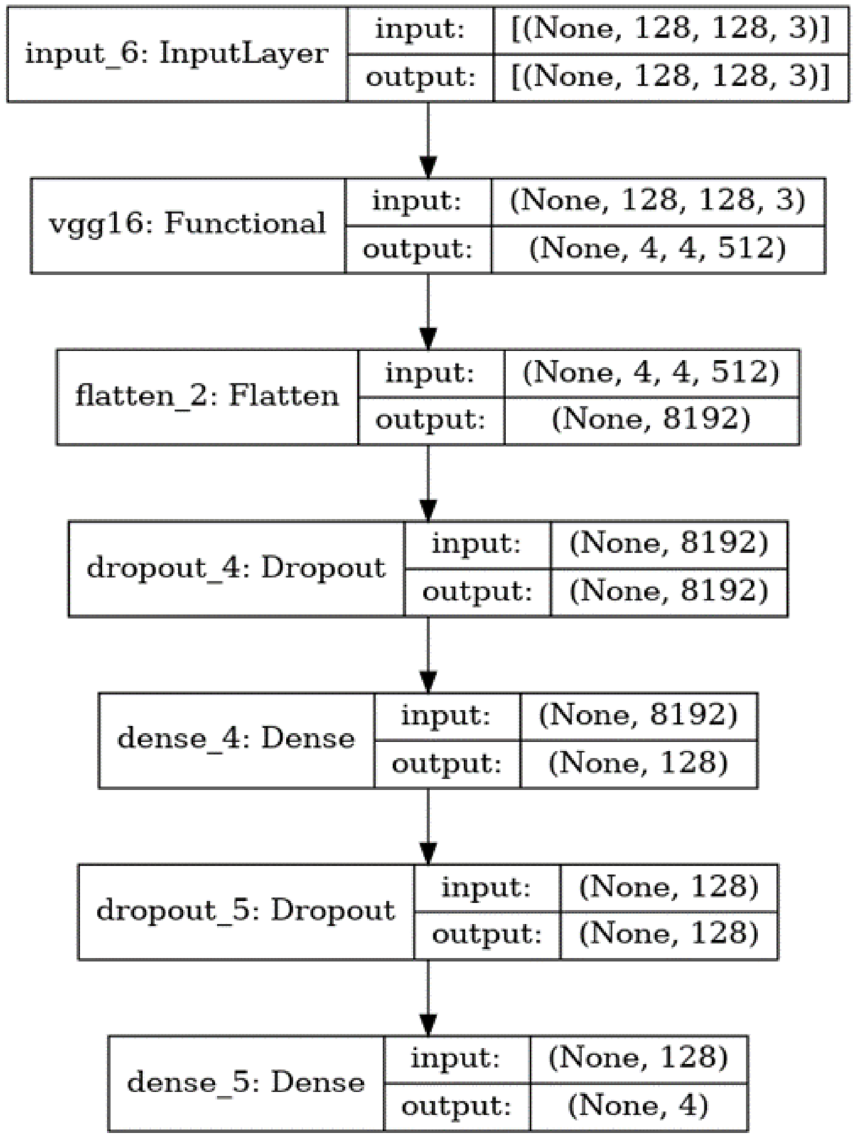


Figure 3.4 (b) Model Structure

3.7 Model Training

Model training in CNN model development is a pivotal process wherein the network learns to recognize patterns and features in the input data. This phase involves feeding the CNN with datasets and evaluating its performance to achieve desired accuracy levels. Effective training not only enhances the CNN's ability to generalize to new data but also ensures robustness and efficiency in real-world applications, in this case, brain tumor detection.

The train-test split method was used to train and test our model. Once the model was ready, the dataset was split into two parts for training and testing purposes. A standard splitting rule of 80-20 split was followed. We have split the dataset into 81.3-18.7 (81.3% for training and 18.7% for testing) as we obtained more accurate results for this particular split percentage.

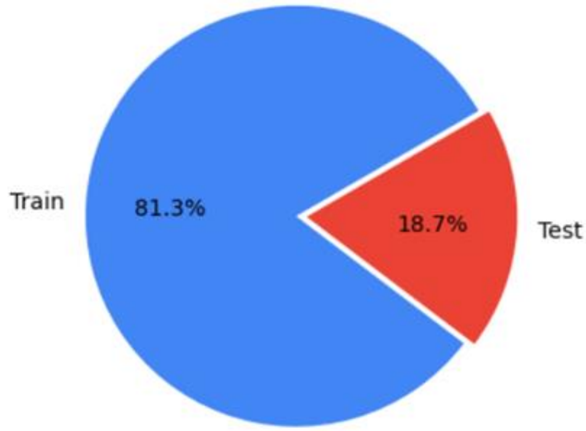


Figure 3.5 Splitting of dataset

An epoch refers to one complete pass through the entire training dataset during the training of a model. During each epoch, the model is presented with each training example once, and the learning algorithm updates the model's parameters to improve its performance on the task at hand. The number of epochs is a hyper parameter that determines how many times the learning algorithm will work through the entire training dataset.

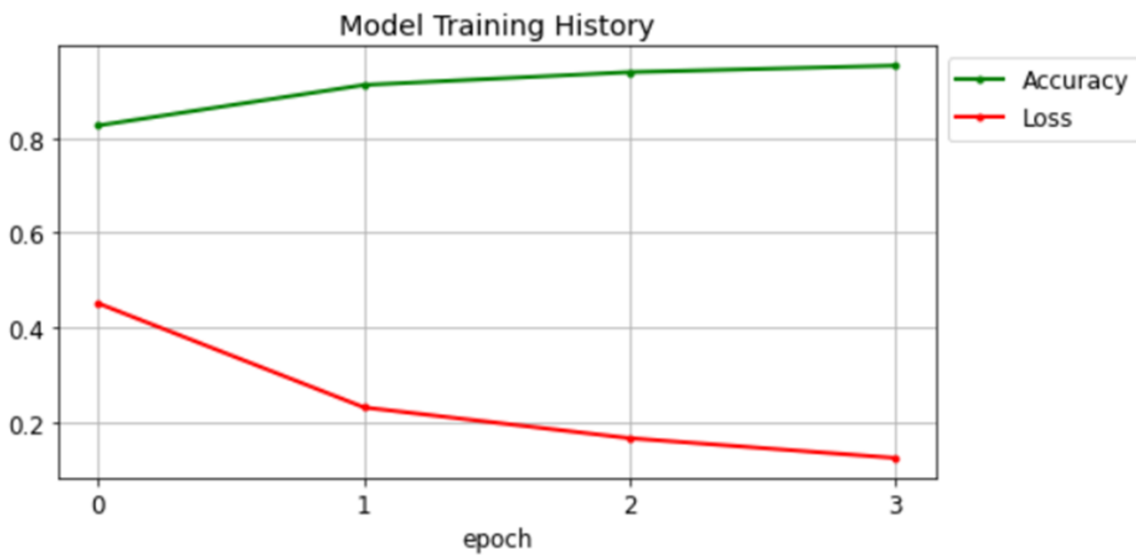


Figure 3.6 Model Training Graph

Here, no. of epochs was fixed to 4 for optimal accuracy and low training loss while keeping computational cost minimum.

3.8 Classification Matrix

	precision	recall	f1-score	support
glioma	0.90	0.92	0.91	300
meningioma	0.91	0.93	0.92	306
notumor	0.98	0.99	0.99	405
pituitary	1.00	0.95	0.97	300
accuracy			0.95	1311
macro avg	0.95	0.95	0.95	1311
weighted avg	0.95	0.95	0.95	1311

Table 3.3 Accuracy Report

As per the evaluation criteria explained in section 2.6, the F1 score tells how accurate our model was in detecting the type of tumor correctly. According to the classification report, we got an accuracy of 95% (weighted avg of f1-score). Generally, the accuracy must lie between 92-97%. Anything more/less than that means the model is overfitted or the training data was not sufficient.

CHAPTER - 4

WEBSITE DEVELOPMENT

Website development plays an important role in today's digital landscape, serving as a cornerstone for businesses, organizations, and individuals to establish their online presence and reach a global audience. In this digital age, a well-designed and functional website is not just a digital storefront but also a powerful tool for communication, interaction, and engagement with users. Whether it's showcasing products and services, disseminating information, or facilitating transactions, a carefully crafted website can significantly impact brand visibility, customer satisfaction, and overall business success.

We have utilized a modern development pipeline for the development process of our website and to make it reach to the ones who need it. The application utilizes various technologies for the backend, frontend, and deployment, providing a versatile and scalable solution.

Website Developments Process involves the following steps:

1) Design on Figma
2) Making the frontend
3) Making the backend
4) Integrate
5) Deploy

4.1 Website Design

A website design serves as a blueprint to guide the actual development process. While the final website may not precisely match the proposed design, it adheres to the design principles to create the basic interfaces.

4.1.1 Key design elements or user interface (UI) considerations:

UI (User Interface) is an important part of a website as it directly influences user experience by enhancing usability and navigation. A well-designed UI ensures intuitive interaction, clarity in information presentation, and seamless functionality, thereby improving user satisfaction, engagement, and retention on the website. To ensure a smooth UI interaction, the key design elements that we formulated for our website are:

- **User-friendly Interface:** For a clean, intuitive design which is easy to navigate. Users should be able to find the information they need quickly and efficiently.

- **Informational Content:** Provide comprehensive and reliable information about the medical condition, including risk factors, diagnosis, treatment options, and prevention strategies.
- **Search Functionality:** Implement a search bar that allows users to quickly search for any required information within the website.
- **Personalized Recommendations:** Offer personalized recommendations based on the user's symptoms, medical history, age, gender, and other relevant factors.
- **Doctor Directory:** Provide a directory of healthcare professionals, including doctors, specialists, clinics, and hospitals. Users should be able to search for healthcare providers based on location, specialty, and other criteria.
- **Privacy and Security:** Prioritize user privacy and security by implementing measures to protect sensitive information, such as encryption protocols.

4.1.2 The Designing Platform:

As discussed in section 2.7, there are various designing platforms available. Out of all the different platforms, Figma was chosen for the purpose as it offers collaborative features, real-time editing, and cloud-based platform. It is ideal for creating prototypes and iterating designs. Figma is a powerful web-based design tool that offers several advantages for designing our brain tumor detection website as mentioned below:

- **Collaborative Design:** Figma allows real-time collaboration, enabling our team to work on the design simultaneously and iterate quickly. This is crucial for incorporating feedback and ensuring a cohesive user experience (UX).
- **Prototyping:** Figma's built-in prototyping features allow us to create interactive prototypes that simulate the website's functionality. This helps us test user flows, identify potential issues, and refine the design before development begins.
- **Cloud-Based Access:** Figma is accessible from any device with an internet connection. This allows us to work on the design from anywhere and ensures version control for easy tracking of changes.
- **Extensive Design Library:** Figma offers a vast library of pre-built design components and icons. This saves time and ensures consistency throughout the website.
- **Easy Handoff to Development:** Figma allows developers to easily access design specifications, including measurements, colors, and fonts. This makes transition from design to development easier.

- **User interface (UI) and user experience (UX) design:** Create mockups, wireframes, and prototypes for websites and apps.
- **Vector graphics editing:** Create and edit vector graphics for your designs.

4.1.3 Design Process:

Our design process on Figma involved the following stages:

- **Wireframing:** We created low-fidelity wireframes to visualize the website's layout, functionality, and user flow. Wireframes focus on structure and functionality, not aesthetics.
- **Prototyping:** Once the wireframes are finalized, we created interactive prototypes in Figma. Prototypes allowed us to test user interactions and refine the user flow before investing time in visual design.

The figures below are from a prototype of a website made on Figma.

1. **Home Page**: The landing page provides an overview of the website's purpose and features. It has a 'Learn more' button in order to know more about the website.

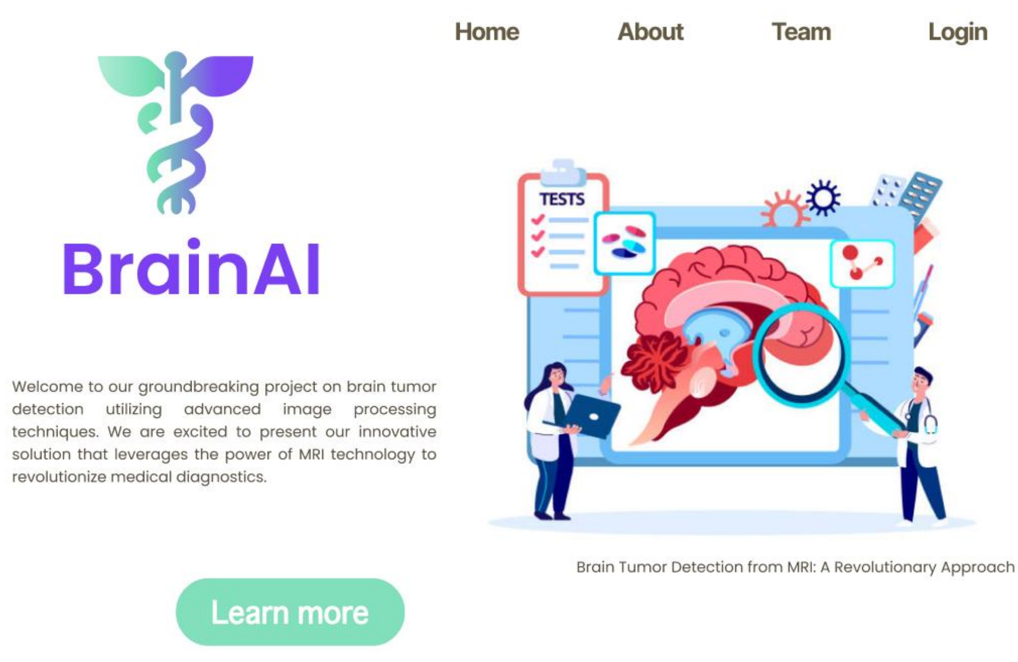


Figure 4.1 (a) Home Page

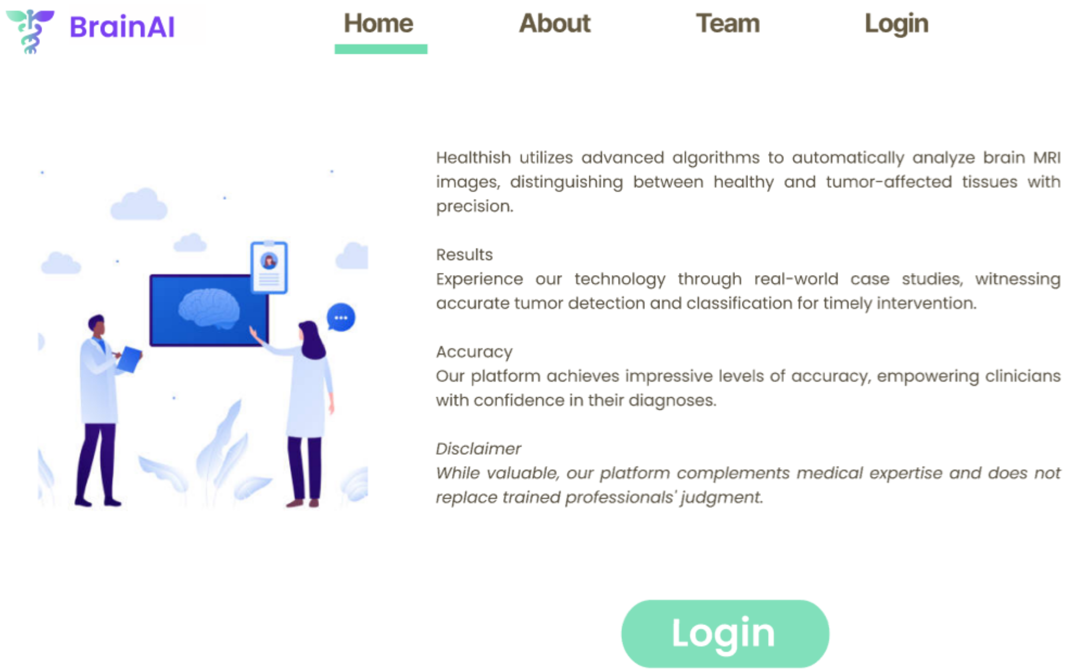


Figure 4.1 (b) Learn more section of Home Page

2. **About Us:** This page describes the website, its goals along with some of its key features. There is a button provided to directly redirect to the dashboard if needed.

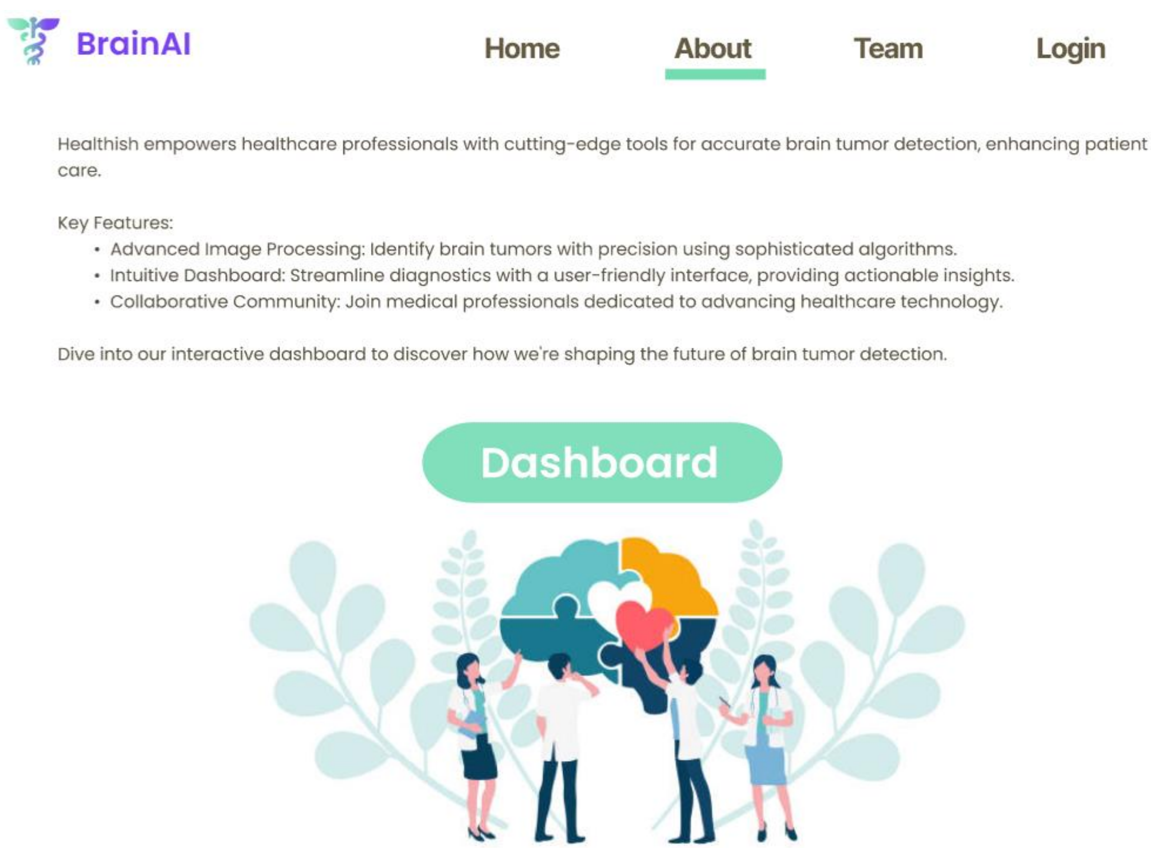


Figure 4.2 About Us

3. **Team and Contact Us:** This page provides information about the creators/ developers with a form for users to reach out to the website administrators.

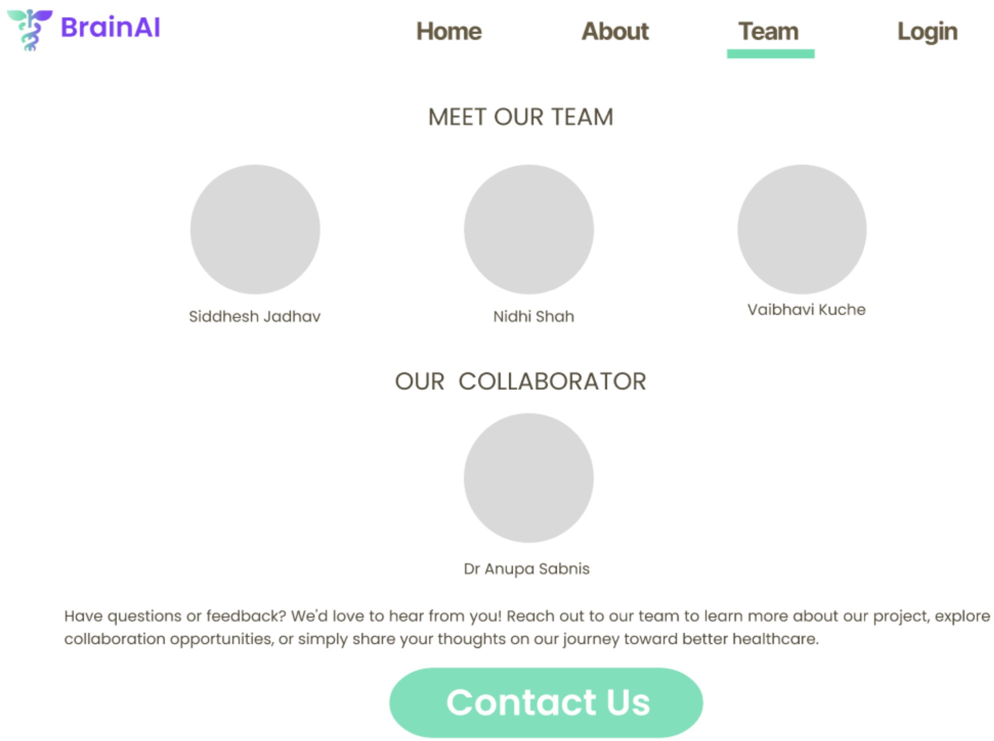


Figure 4.3 Team and contact page

4. **Login Page:**

A clean and user-friendly login page with name, phone number, username/email and password fields. It implements secure protocols (HTTPS) and considers two-factor authentication for added security. There is a provision for sign-up as well as sign-in for first time users.

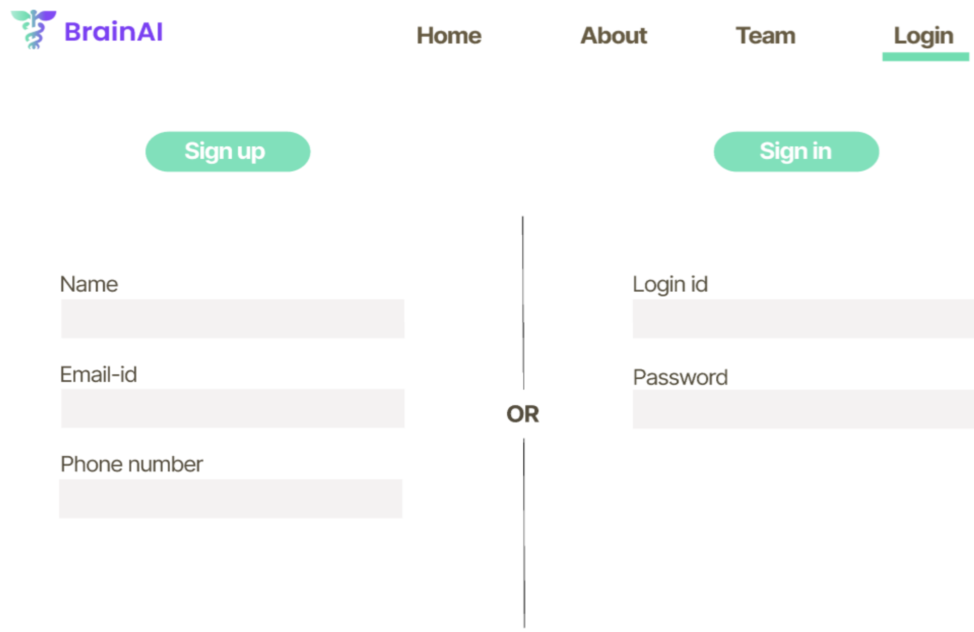


Figure 4.4 Login Page

5. Dashboard:

The dashboard serves as the central hub for logged-in users. It provides an overview of relevant information and functionalities. It has the following elements:

- Personal Details: Write age, address and consulting doctors' details.
- Scan Upload: An option to upload MRI scans for analysis by the AI model.
- Results: A list of previously uploaded scans and their corresponding analysis results along with the summary of analysis findings (e.g., tumor probability, location).
- Account Settings: A section for managing user information, such as password updates or contact details.
- Disclaimer: A disclaimer is provided so as to avoid misinterpreting the AI prediction for Doctor's report.

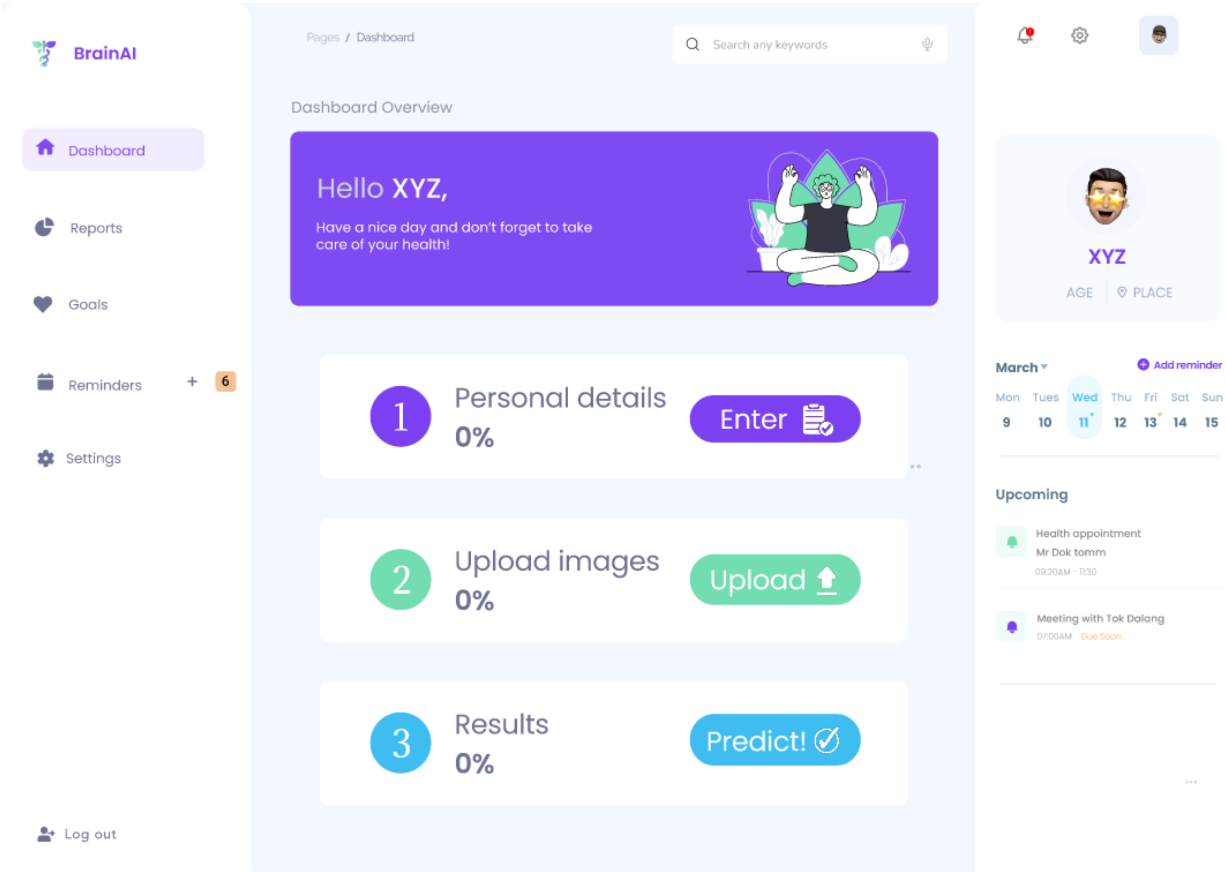


Figure 4.5 (a) Dashboard

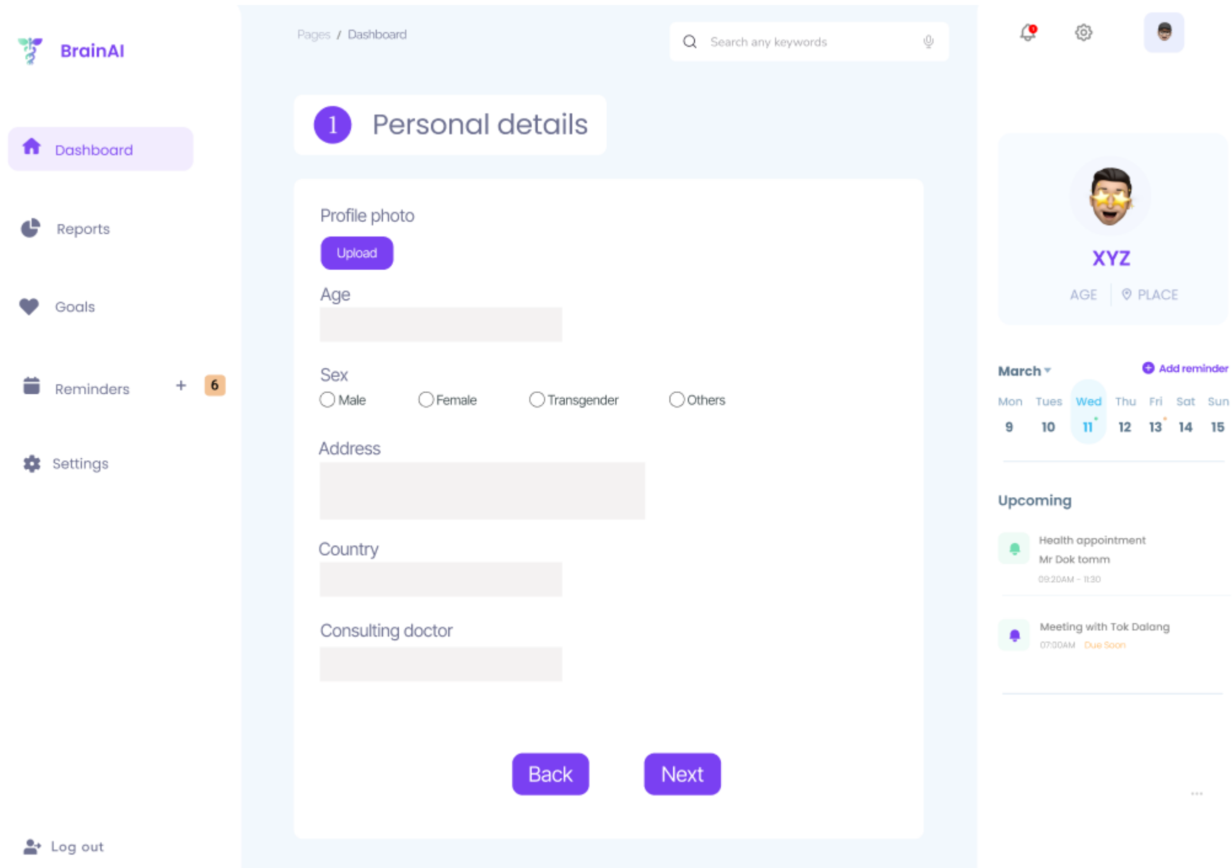


Figure 4.5 (b) Personal Details section of Dashboard

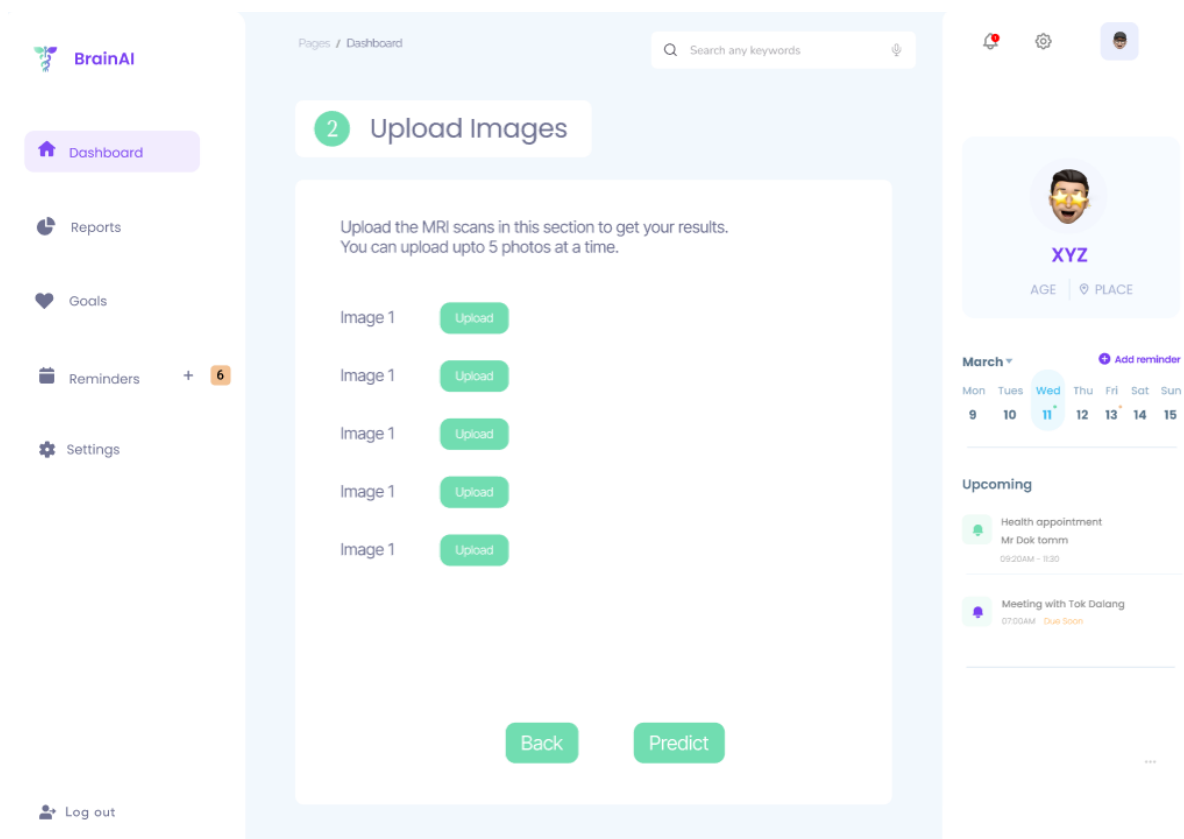


Figure 4.5 (c) Image Upload section of Dashboard

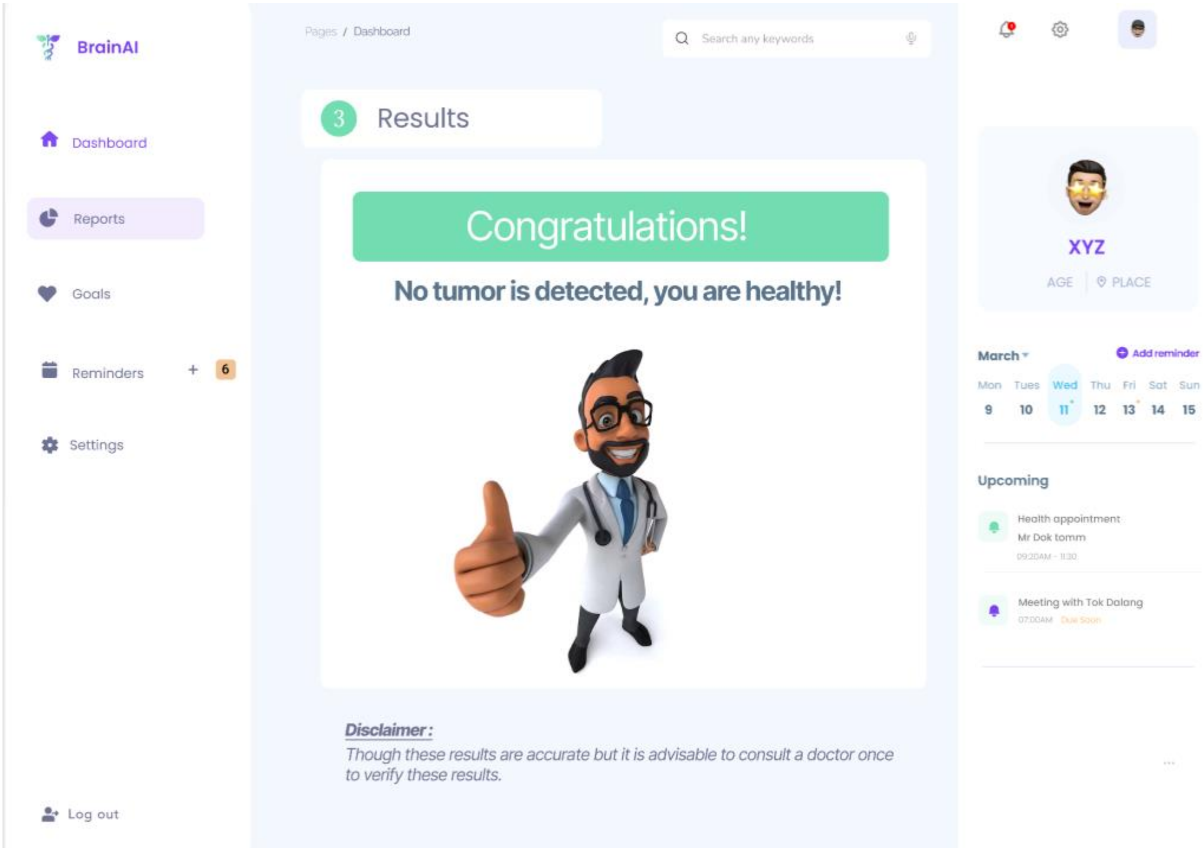


Figure 4.5 (d) Results section of Dashboard that will be displayed in case of no tumor detected

If a tumor is detected, the results will display information about the type of tumor present, the treatment that can be taken up, a few videos to increase knowledge about the tumor and a few specialist names with location for consulting them if required.

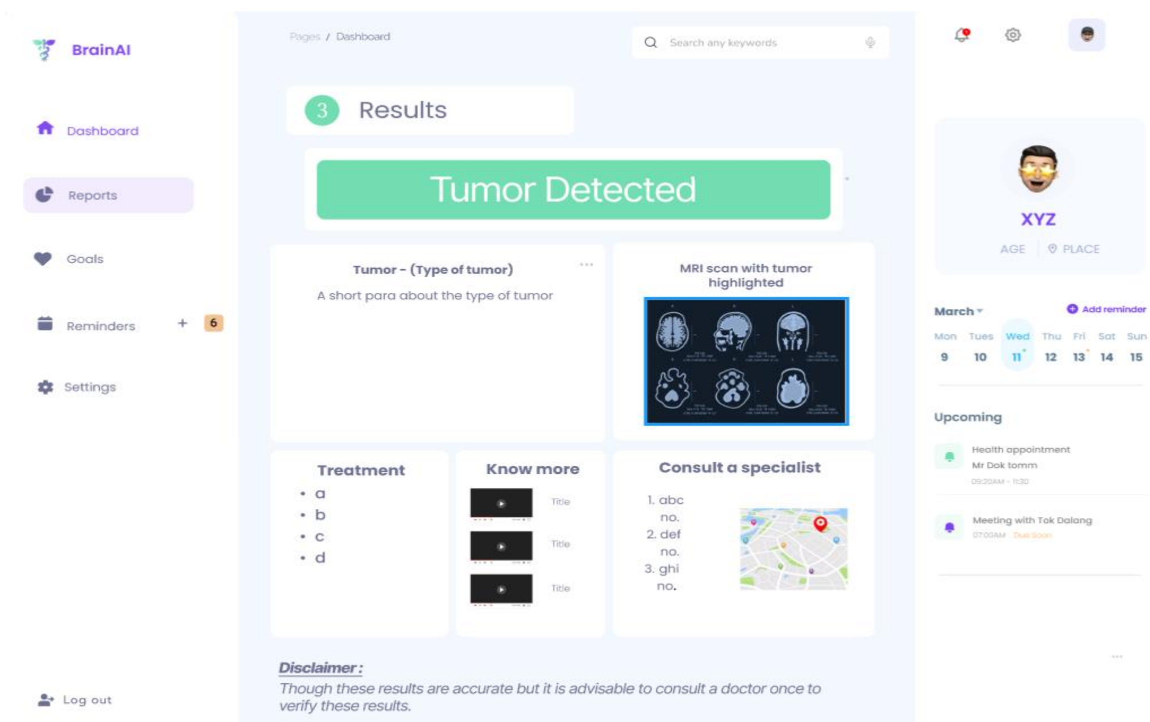


Figure 4.5 (e) Results section of Dashboard that will be displayed in case of tumor detected

4.1.4 Some Design Features:

- **Easy to Find:** Make contact information readily available on the website, ideally in the footer or a dedicated "Contact Us" page.
- **Social Media Links:** Consider including links to relevant social media profiles for updates and community engagement.
- **Clear Layout:** Organize the dashboard with a clear and intuitive layout for easy navigation.
- **Scan Upload:** Dedicated section for uploading MRI scans.
- **Drag-and-Drop:** Implement drag-and-drop functionality for easier file uploads.
- **Progress Indicator:** Include a progress bar or notification system to inform users about upload status.
- **Clear Presentation:** Present the analysis results in a clear and concise manner.
- **Terminology:** Use clear and understandable language, avoiding overly technical jargon.
- **Disclaimer:** Emphasize that the website's results are for informational purposes only and should not be used for self-diagnosis.
- **General Information:** Provide basic information about the predicted tumor type, avoiding overwhelming users with complex medical details.
- **Resources:** Include links to reliable resources for further information about specific tumor types.
- **Location Services :** With user consent, leverage location services to suggest nearby doctors or hospitals specializing in brain tumor treatment or directories for finding qualified oncologists or neurosurgeons.

4.2 Frontend development

Frontend development is essential for creating user-facing aspects of websites and applications. It focuses on building responsive, accessible, and visually appealing interfaces using HTML, CSS, and JavaScript. It integrates design elements, optimizes performance, and enhances interactivity across various devices and browsers. Our frontend was developed using Vite.js and tailwind CSS.

4.2.1 Login Page with JWT Authentication:

The login page uses JSON Web Token (JWT) authentication to securely manage user sessions. Users input their credentials, which are sent to the server for verification. Upon successful authentication, a JWT is returned and stored on the client-side to authenticate subsequent requests, ensuring secure and stateless user sessions.

4.2.2 Pages:

Home: A clear and concise explanation of the website's purpose – brain tumor detection using AI and MRI scans.
About: Outline the website's mission and core values, emphasizing patient care or advancements in medical technology.
Contact: Introduce the team behind the website, their expertise, and their dedication to brain tumor detection.
Include various ways for users to contact the website administrators, such as email address, contact form, or social media links.
Predict: A page where users can input data to get predictions from the website's service.
Results: A page that displays the outcome of predictions or any other relevant data.
Login: A page for user authentication using JWT.

4.2.3 Built with Vite.js and Tailwind CSS:

- Vite.js: A modern build tool that offers a fast and efficient development environment. Vite.js improves the development experience with its quick start-up times and hot module replacement (HMR), allowing for rapid development and debugging.
- Tailwind CSS: A utility-first CSS framework that provides low-level utility classes for rapid and responsive UI development. Tailwind CSS enables developers to design custom and scalable user interfaces quickly without writing much custom CSS.

4.3 Backend Development

Backend development forms the core of web applications, handling server-side logic, databases, and ensuring functionality. It powers the interactive elements users interact with, manages data storage, and supports seamless integration with frontend interfaces. The backend of the project is developed using Node.js and Express.js on Django, providing a robust and scalable environment to handle server-side logic and API requests. The backend serves as a REST API, managing various endpoints to interact with the frontend and perform CRUD operations on the database.

4.3.1 API Features:

- Node.js and Express.js: The core of the backend, Node.js offers an asynchronous, event-driven architecture, while Express.js simplifies the creation of robust API routes.
- REST API: The API follows REST principles, ensuring stateless, client-server communication. Endpoints are designed to handle requests for user authentication, data retrieval, predictions, and other functionalities.
- Database: MongoDB Atlas, a cloud-based NoSQL database, stores user data, prediction results, and other necessary information. MongoDB's flexible schema design allows easy handling of diverse data types.

4.3.2 Machine Learning Integration:

- Flask API for ML Model: To handle machine learning tasks, a separate Flask API is created. Flask, a lightweight WSGI web application framework in Python, is used to serve the machine learning model.
- Flask API: Flask is chosen for its simplicity and ease of integration with Python-based machine learning models. It provides a straightforward way to create APIs that can handle requests, perform computations, and return results.
- Loading the Model: The machine learning model is pre-trained and loaded into the Flask API when the server starts. This allows the Flask API to quickly process prediction requests by running the input data through the model and returning the results to the Node.js backend, which then forwards them to the frontend for display.

4.4 Integrate the frontend with the backend

The integration of frontend and backend in a MERN stack application involves seamlessly connecting MongoDB, Express.js, React.js, and Node.js to create a cohesive and efficient web application. The frontend, developed using React.js, communicates with the backend via HTTP requests to RESTful APIs built with Express.js and Node.js. These APIs handle data operations and business logic, interfacing with MongoDB for data storage and retrieval. This architecture ensures a smooth flow of data between the client and server, providing a dynamic and responsive user experience. The MERN stack's unified use of JavaScript across all layers simplifies development and maintenance, enhancing overall application performance and scalability.

4.5 Deploy on AWS (Amazon Web Services)

First, we set up an AWS account and then configured IAM (Identity and Access Management) to create a user with administrative permissions. Next, we installed the AWS CLI and configured it with our access keys and preferred settings. For a static website, we created an S3 bucket, enabled static website hosting, and uploaded our website files. If needed, we registered a domain and configured DNS settings using Route 53. For a dynamic website, we launched an EC2 instance, connected to it via SSH, and installed web server software such as Apache or Nginx. We then deployed the website files to the web server. To ensure accessibility, we updated DNS settings for the EC2 instance. If a database was required, we created an RDS instance and connected it to our web application. We configured security groups to allow necessary traffic and performed regular patching. To handle traffic spikes, we set up auto-scaling. Finally, we monitored the website's performance using CloudWatch.

CHAPTER - 5

RESULTS

The website for brain tumor detection and classification using CNNs provides an intuitive platform for uploading MRI scans, receiving instant, accurate diagnoses, and accessing detailed reports. This enhances diagnostic efficiency, supports healthcare professionals with real-time data, and improves patient outcomes through timely and precise analysis. The final website is shown below:

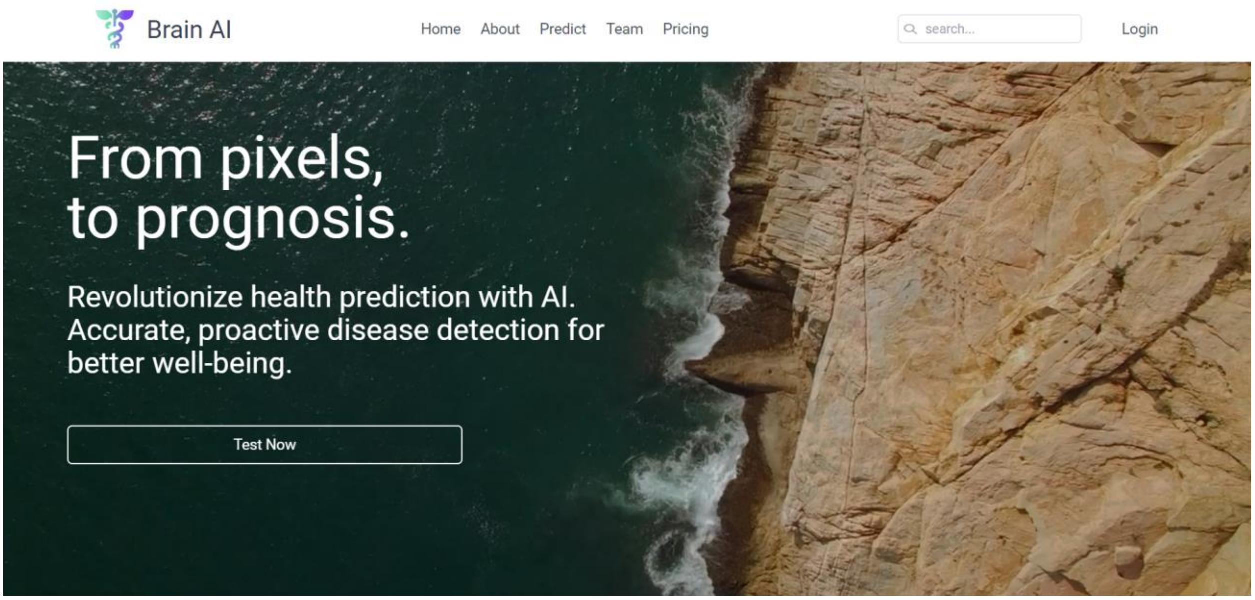


Figure 5.1 Home Page

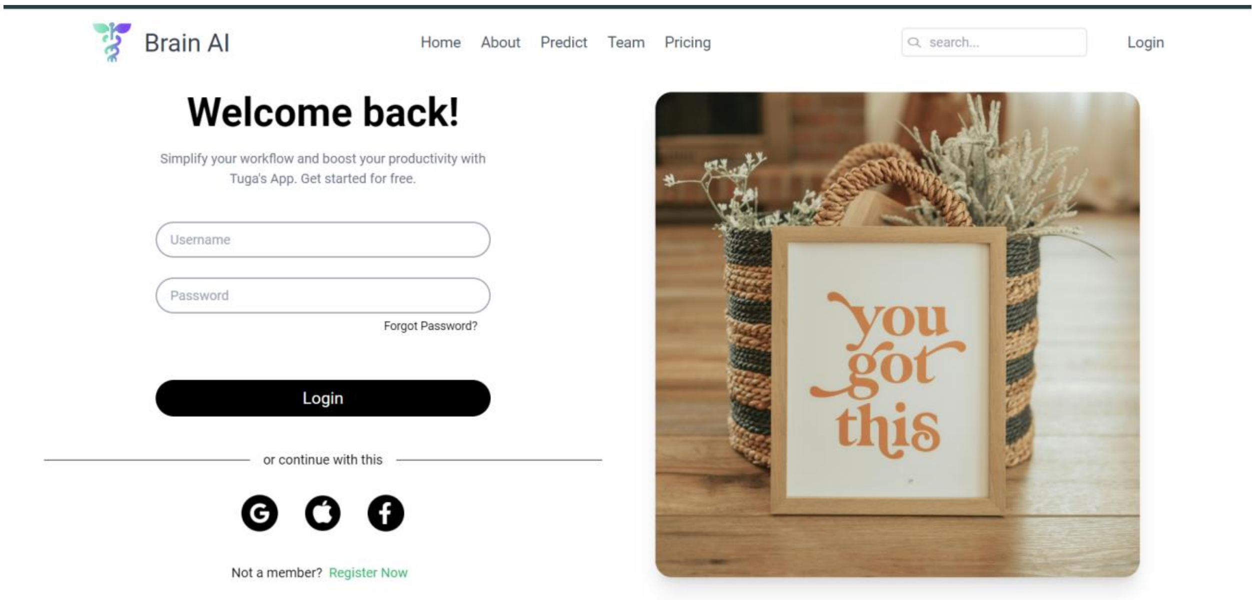


Figure 5.2 Login Page

Model Accuracy

Accuracy upto 96%

Category	Precision	Recall	F1-Score	Support
Glioma	0.9	0.92	0.91	300
Meningioma	0.91	0.93	0.92	306
Notumor	0.98	0.99	0.99	405
Pituitary	1	0.95	0.97	300
Accuracy	0.96	0.96	0.96	1311

Figure 5.3 About Page

Brain AI   Home   About   Predict   Team   Pricing   search...   Login

Upload your MRI Scan to get Results

Upload

Figure 5.4 Page to Upload MRI Scans

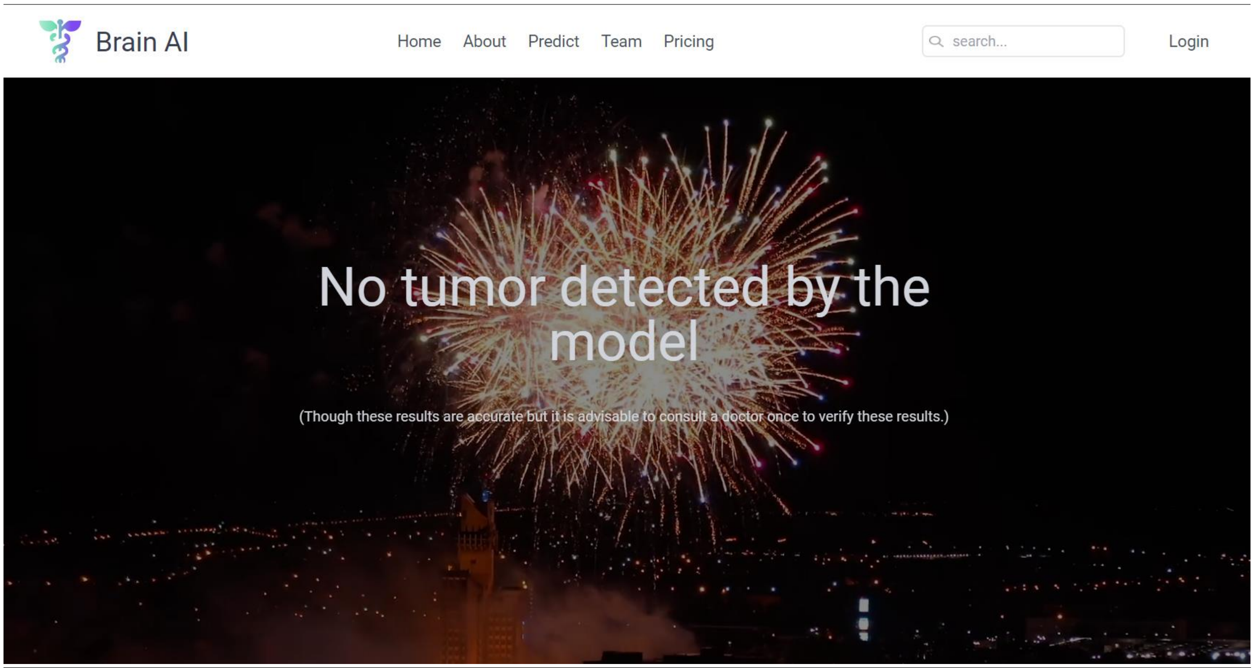


Figure 5.5 Page to be displayed if no tumor is detected

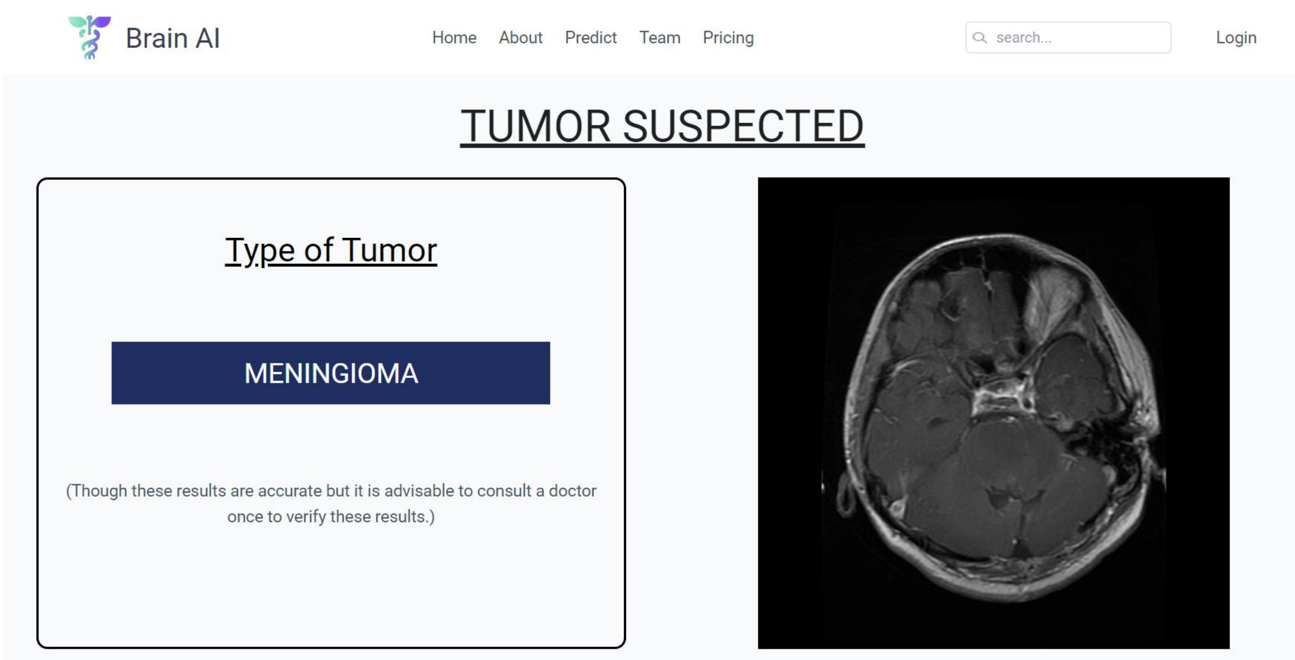


Figure 5.6 Page to be displayed if tumor is detected

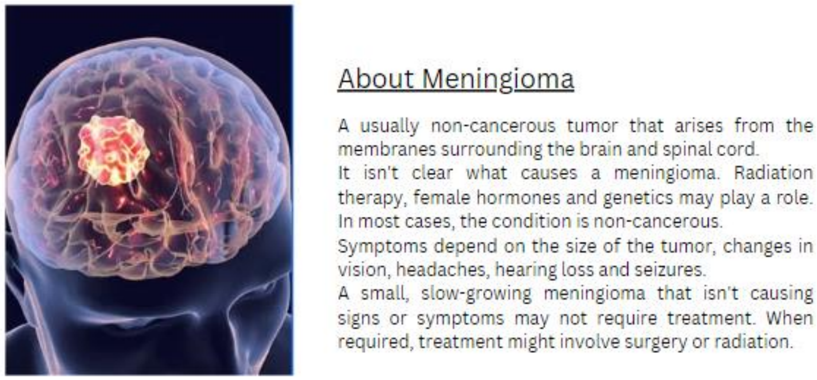


Figure 5.7 Page displaying details of the tumor if tumor is detected

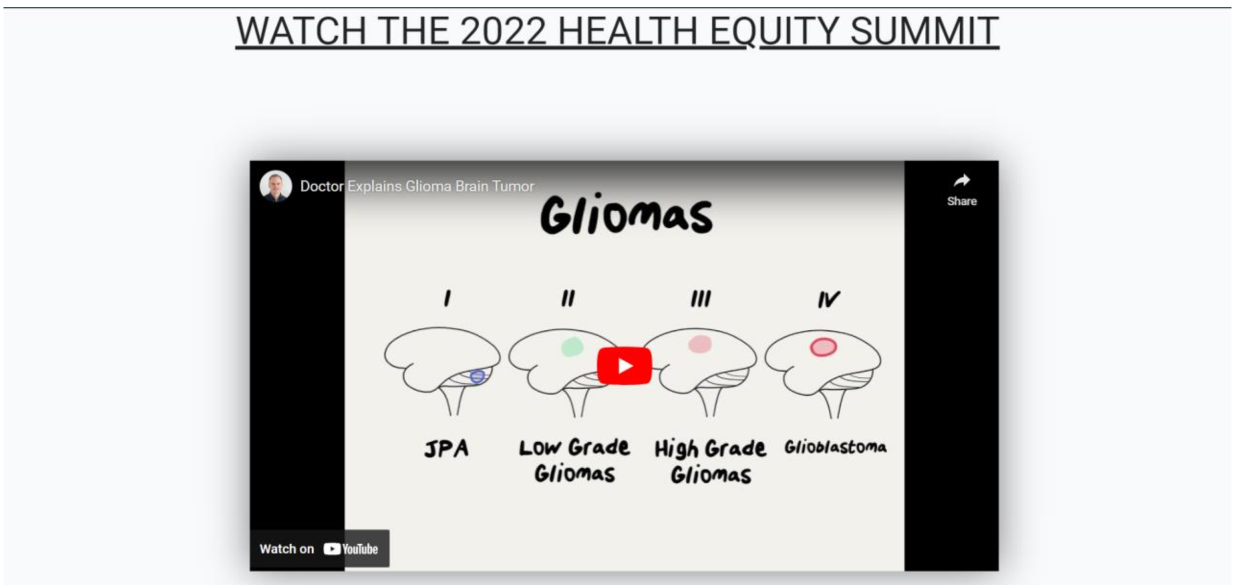


Figure 5.8 Page to suggest a few video links related to the type of tumor detected

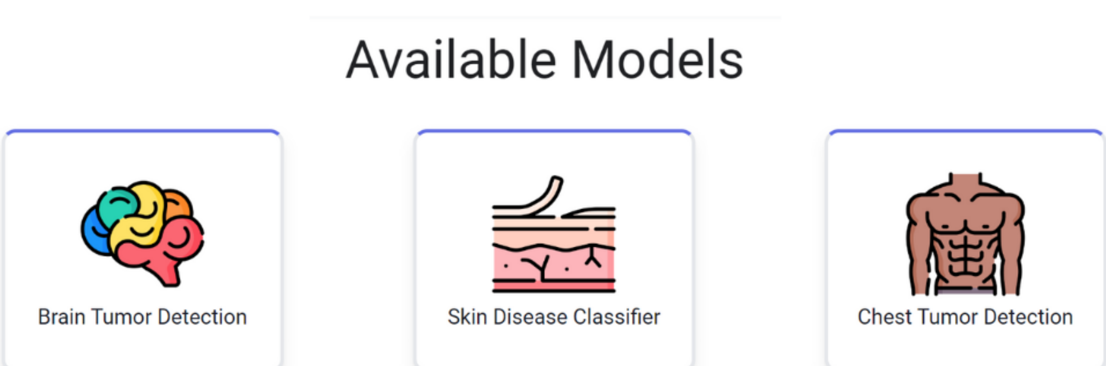


Figure 5.9 Page depicting future scope of the website

CHAPTER - 6

CONCLUSION

The brain tumor detection and classification project utilizing a Convolutional Neural Network (CNN), demonstrates the potential of deep learning in medical imaging. This study covers model development, website design, development and deployment. The VGG-16 model was considered which has achieved significant success in training and testing with a weighted average F1 score of 0.95, implying an accuracy of 95%. The model efficiently classifies brain tumors into four distinct types, showcasing the model's capability to learn intricate patterns and features from MRI scans and its robust performance in medical image analysis. This automated approach not only enhances diagnostic precision but also reduces the burden on healthcare professionals, enabling faster and more reliable tumor detection. The design on Figma gave an overview of the features that the website will showcase. With the effective use of Vite.js and tailwind CSS, the frontend came to life and the backend was made possible using Node.js and Express.js on Django. After their successful integration, the deployment was carried out using Amazon Web Services (AWS) and the website's performance was monitored using CloudWatch. Overall, this project underscores the transformative impact of CNNs in medical diagnostics, offering a promising platform for early detection and personalized treatment suggestions of brain tumors.

CHAPTER - 7

FUTURE SCOPE

Moving forward, there are several exciting future objectives to consider. Firstly, refining and expanding the existing model could be explored to enhance its capabilities further. Incorporating larger and more diverse datasets may improve the model's generalization across different patient demographics, ensuring its effectiveness in real-world scenarios.

Additionally, extending the application of deep learning models to other medical domains is a promising avenue. For instance, developing a similar model for the detection and classification of skin diseases using dermatological images could be a valuable contribution to the field of healthcare. Skin disease diagnosis is another area where automated systems can assist healthcare professionals, potentially improving diagnostic accuracy and efficiency.

Future research could also focus on the interpretability of the models, making them more transparent and understandable for healthcare practitioners. This could involve developing visualization techniques to explain the decision-making process of the model, promoting trust and confidence in its recommendations.

Furthermore, developing a web app or a mobile app will make advanced diagnostic tools accessible, enabling early and accurate detection. Healthcare providers can carry a powerful diagnostic tool in their pocket, enabling them to make informed decisions anywhere, anytime which is useful especially for fieldwork or in-home patient visits. Apps can be easily updated with the latest advancements in AI and medical research, ensuring that users always have access to the most current diagnostic capabilities.

References:

[1] "Deep Neural Network Correlation Learning Mechanism for CT Brain Tumor Detection" (2021) - Woźniak M., Siłka J. and Wieczorek M.

[2] "Computer Vision and Pattern Recognition (cs.CV)" (2014) - Karen Simonyan and Andrew Zisserman

[3] "Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging" (2023) - Akmalbek Bobomirzaevich Abdusalomov, Mukhriddin Mukhiddinov and Taeg Keun Whangbo

[4] "Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images" (2019) : A study by Qing Guan, Yunjun Wang, Bo Ping, Duanshu Li, Jiajun Du, Yu Qin, Hongtao Lu, Xiaochun Wan, and Jun Xiang

[5] "Brain Tumor Detection and Classification Using Intelligence Techniques" : An overview by Shubhangi Solanki, Uday Pratap Singh, Siddharth Singh Chouhan and Sanjeev Jain

Figures -

Fig 1.1 - https://images.app.goo.gl/H5S1QKCRYkhwBbmU6

Fig 2.1 (a) - https://images.app.goo.gl/p73YttLD6KHxgpwp6

Fig 2.1 (b) - https://images.app.goo.gl/pxh1acS1xWcxJjKU9

Fig 2.2 (a) - https://images.app.goo.gl/TmPDERtBUNZcRFe56

Fig 2.2 (b) - https://images.app.goo.gl/pADdhC54moyR46jG7

Fig 2.3 (a) - https://images.app.goo.gl/NykETBEMXk6PY2SB7

Fig 2.3 (b) - https://images.app.goo.gl/D9Xaz92S7NNUhrHH6

Fig 2.3 (c) - https://images.app.goo.gl/ngjC14mFc8HcMKSGA

Fig 2.4 - https://images.app.goo.gl/w1A542ypd3CyCLJt6

Fig 2.5 - https://images.app.goo.gl/a1sUqbxtgLv9FwHFA

Fig 2.6 (a) - https://images.app.goo.gl/AHSwVkRnwcWafJCGA

Fig 2.6 (b) - https://images.app.goo.gl/5LEspzqEcAdEW5h96

Fig 3.2 - https://images.app.goo.gl/VYe2zsFuP9Xun12G7

Fig 3.3 - https://images.app.goo.gl/FXG66Ay58xwgjyRf9