BRAIN TUMOUR DETECTION SYSTEM

A PROJECT REPORT

Submitted in partial fulfillment of requirement
for the degree of

Bachelor of Engineering

in

Electronics & Telecommunication

Submitted by

ISHAN MODI (B150953039)

ISHAN SRIVASTAVA (B150953023)

JANHAVI PANAMBOR (B150953045)

Under the Guidance of

DR. VARSHA DEGAONKAR

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE(MH)-411057

SAVITRIBAI PHULE PUNE UNIVERSITY

A.Y. 2021-22

CERTIFICATE

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE-411057

This is to certify that

Ishan Modi (B150953039)

Ishan Srivastava (B150953023)

Janhavi Panambor (B150953045)

Class: BE(E&TC) have satisfactorily completed Project titled, '**Brain Tumour Detection System**' under my supervision as a part of Bachelor of Engineering in **Electronics and Telecommunication (A.Y. 2021-2022)** of Savitribai Phule Pune University.

Dr. Varsha Degaonkar
Project Guide

Dr. Risil Chhatrala
HOD(E&TC)

Principal

Place : Pune
Date :

External Examiner

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been 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 penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. We take sole responsibility for the work presented by us in this report. We also declare that we will submit our completed project along with all necessary hardware and software to the department at the end of the 2nd semester.

Ishan Modi

Ishan Srivastava

Janhavi Panambor

Place : Pune

Date :

Abstract

Brain tumours, in medical terms are the intentional or unintentional growth of mass cells which hamper the conventional functioning of the shape of brain. For correct diagnosis and efficient treatment planning, it is necessary to detect the brain tumour in the early stages. The tumour within the brain is one of the most dangerous diseases and might be diagnosed easily and reliably with the assistance of detection of the tumour using automated techniques on MRI Images. Positron Emission Tomography, Cerebral Arteriogram, spinal tap, Molecular testing are used for tumour detection. Digital image processing plays an important role in the analysis of medical images. Segmentation of tumours involves the separation of abnormal brain tissues from normal tissues of the brain. Over few past years, various researchers have proposed semi and fully automatic methods for the detection and segmentation of Brain tumours. The motivation behind this project is to detect neoplasm and supply better treatment for the suffering. The objectives for the project are to develop an end-product (Web Application) that can be installed at hospitals. To facilitate this a detection model is developed that may accurately predict if an uploaded MRI scan of brain shows it is affected by tumour or not. To implement the project a Convolutional Neural Network(CNN) was used to define the model. Transfer Learning is implemented in order to efficiently train the model. The data-set used is split into 3 sets which are train, test and validation, in the ratio 80:10:10. The model is meant to be trained for 12 epochs. Callbacks also have been given to automate the model save process. The test accuracy of 97% is achieved. This trained model will be connected with an online Application via API. Within the proposed Web App the user is having access to four routes which is welcome page and this contains information about the system, second route is information and awareness about the brain tumour in medical terms, third is detection page, where the trained model is deployed. The user is able to provide an input image,

MRI images in our case, and last route is the team information. Images which are fed to the model route will be processed by the developed convolutional neural network which is able to then confirm if a tumour is present or not and intimidate the user for the same through an output Display. The advantage of using this system is that it will automate the detection process, and ease the workload of the hospital staff. However for the advantage to become a reality, careful selection of accurate data is needed, else there is a chance of false results.

Keywords:

Brain Tumour Detection, Medical Image Processing, Brain tumour, MRI, CNN, Web Application, Case History

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Initiating the chapter, Artificial Intelligence and its allied areas are witnessing the rapid growth and advancements for removing the barrier between humans and machines for a better world and future ahead. This deals with the biomedical domain of technology very specifically with the brain tumours and their detection with traditional ways and a brief light to modern future approaches with the use of technology. The introduction starts with an overview of the problem statement which is an early-stage detection of brain tumours. Traditional approaches of detecting and factors associated with it. Image processing is the heart of the detection through Deep Learning and Machine learning models. The detection is done via deep learning methods comprising particular algorithms. Convolutional Neural Network is the heart of image analysis using machine learning or deep learning models which offers ease of analysis with good efficiency. Followed in the next section shows the significance of Computer Vision that can be helpful for image analysis.

1.1 Overview of the detection system

Computer vision and image processing are a few of the most significant domains in upcoming technical advancements of the world. The main objective of computer vision and Image processing is to enable machines and devices to view and capture the world in the way humans see it. It is the entire process involving object detection, classifi-

cation and analysing the results. Image processing has majorly two parts involved in the entire process. Pre-processing is the first step in image processing which consists of operations such as image enhancement, resizing, adjusting images. Post Processing comes as the second part in the process which has special modifications as per the needs - highlighting the segmented areas, removing noise, applying texts to the area needed. Thus, any image dataset which consists of similar fields can be processed and modified for particular needs to tackle the problems.

Artificial Intelligence and its effects have seen an exponential rise in advancements of the world through automation and smart technologies. Artificial Intelligence alongside machine learning and deep learning techniques have been embedded into almost every aspect of life. With the advent of Smart tools and systems using AI-ML-DL, both developers and consumers have been benefited in terms of better business deals and revenue generation. It is highly significant to today's highly sophisticated technical era of the world to make better decisions using such advanced systems and implement - integrate them into their real-life as per their needs.

In this project, a system for the early-stage detection of brain tumours is proposed. As the human brain is the most important body organ and its improper working or disease can lead to mortality death with tumours. Hence a system based on image segmentation and the deep learning algorithm is proposed for early-stage detection of these tumours.

Brain tumours or neoplasm are basically the growth or mass of abnormal cells which grow inside the Brain. There has been an unprecedented growth in the number of humans diagnosed with Brain Tumours. There are two prominent types of brain Tumours one is benign (non - cancerous) and another one is Malignant (cancerous). There are various methodologies to detect them and validate the same with tissue tests. Traditional approaches to detecting brain tumours and treating them cost a

lot of money and time, which is in most cases not suitable for middle-class men and dependents. All these factors contribute as disadvantages and thus it is beneficial to bring technology into consideration and use existing infrastructure as an algorithm so that we can detect a tumour at an early stage and save a lot of money and consume less time. Adding to this, the mortality rate can be reduced by early detection of tumours. Humans are living in an age where health becomes a very crucial factor in order to maintain and then survive thus, having such a biomedical system can be highly efficient in terms of money and time.

The solution is an easy and simple brain tumour detection system that uses CNN - convolutional neural network algorithm and feeds on image input by the user which then segments the image and applies image processing and highlights if it detects brain tumour with proper highlighting.

The motivation behind selecting the particular project is the statistics observed till now. Brain Tumour incidence has increased more than 10% over the past 20 years [5]. Such an increase in incidence puts higher pressure on the existing medical facilities. Thereby a simple solution is proposed that can help in solving the issue.

What our system does is it uses a end product (web app) which will have inputs as MRI scanned images of brain, it will perform digital image processing on the input image and will undergo CNN algorithm process. This involves validation and comparison with our trained model and based on its learning, it will classify whether it a tumour or not. If tumour, it will highlight the area affected inside the brain. Since brain tumour varies on the location it is affected, we can have a long run effect insight also from the same output. These insigts and reports can be stored on the same web application and cab be used further to compare the change in earlier records of the same patient also.

Overall, in a nutshell, early information on the brain tumour and how well it can be diagnosed can be obtained.

The report is divided into 6 chapters. Chapter 1 deals with the basic introduction of the project. The chapter also deals with what a brain tumour actually is. Furthermore, problem motivation for the project and an overview of the project system is provided. Chapter 2 deals with the Literature Survey conducted. The literature survey highlights papers that give an insight into different algorithms, techniques and processing that can be used to detect brain tumours. Chapter 3 deals with Problem statements, Outcomes, Requirement Analysis, Impact Analysis and Limitations. A brief has also been given on Professional Ethics to be followed during the implementation of the project. Chapter 4 deals with the actual project implementation. Here a summary as to what resources are required actually, the basic block diagram have been discussed. Furthermore, the Detection Model and Webapp have been discussed in brief with flowcharts and algorithms. A model summary used has also been provided. At the end images of the proposed layout have also been attached for ease of understanding. Chapter 5 deals with the results obtained from the detection model, as well as results from the data augmentation. Chapter 6 deals with the future scope and conclusion.

Chapter 2

Literature Survey

A system that detects brain tumour using Convolutional Neural Network (CNN) is proposed in [1]. The paper mentions the dataset used for training. The same data has been used for the development of the system proposed in this project. The authors mention various other steps implemented which include image augmentation, pre processing, segmentation of the skull, which can be implemented for better training of the model. The authors have used a custom architecture of CNN which provides an efficiency of 95.3% on test images. The model used by the authors is as follows :

Model: "BrainTumorDetectionModel"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 240, 240, 3)]	0
zero_padding2d (ZeroPadding2	(None, 244, 244, 3)	0
conv0 (Conv2D)	(None, 238, 238, 32)	4736
bn0 (BatchNormalization)	(None, 238, 238, 32)	128
relu0 (Activation)	(None, 238, 238, 32)	0
max_pool0 (MaxPooling2D)	(None, 59, 59, 32)	0
max_pool1 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
fc (Dense)	(None, 1)	6273

Total params: 11,137
Trainable params: 11,073
Non-trainable params: 64

Figure 2.1: Paper 1 Model Summary

Insights into various symptoms, causes and statistics of brain tumour is identified from [2] , [3] and [5]. This data will be used into the information module of the web application.

A survey on major deep learning systems that have been implemented to do various types of brain tumour analysis can be inferred from [4]. This survey gives an insight into the existing detection models. The paper further gives an insight into datasets, techniques into consideration as well as the evaluation of each particular paper considered into the survey.

A tumour detection system that used image segmentation as the base technique is described in breif in [5]. The authors use a clustering based approach using Self Organizing Map (SOM) algorithm. The detection works in two phases, the first being image acquisition and pre processing. The second phase is image segmentation which basically segments principle tissue structures in the input images. The authors propose a new MR image segmentation method based on fuzzy C-Means clustering algorithm for the Segmentation which is unsupervised.

Comparision between nonlinearity reduction and linear method is done in [6]. The authors experiment to check if the non linear method provide better results (unsupervised classification) of MRS brain tumour data. Data reduction is performed using Laplacian eigenmaps (LE) or independent component analysis(ICA). This was then followed by k-means clustering or agglomerative hierarchical clustering (AHC) for unsupervised learning to assess tumour grade and for tissue type segmentation of MRSI data. Based on the results obtained LE method is promising for unsupervised clustering to separate brain and tumour tissue with automated color-coding for visualization of 1 H MRSI data after cluster analysis.

A survey on different segmentation techniques is given [7] conduct a survey on dif-

ferent segmentation techniques. The paper provides different papers, their datasets, and various parameter information that gives a brief insight into different algorithms that can be implemented to detect brain tumour. The survey is divided into the following categories: Thresholding techniques, Region growing techniques, Edge based techniques, Clustering techniques, Watershed technique, and Deformable model-based techniques.

A image segmentation based approach for tumour detection is explained in brief in [8]. From the paper it can be inferred that Magnetic Resonance Imaging (MRI) scans are the best medical scan for the training and detection of brain tumour. A deep insight is given into the different parameters associated with image enhancement using different filters and techniques. Furthermore insights are also provided into segmentation techniques.

A custom model named as Deep Wavelet AutoEncoder Model (DWAE Model) is proposed in [9]. A high pass filter is used to show the heterogeneity of MRI images. After preprocessing is performed, the DWAE Model analyzes the pixel pattern of the input scan, and classifies the tumour accordingly. The test results show that the model achieves an accuracy of 99.3% with a validation loss of 0.1%.

Chapter 3

Proposed Methodology

3.1 Problem Statement

To detect brain tumour using Deep Learning algorithms and implement the same into a web application that will act as an end product. The system being developed aims to facilitate early detection of tumour so as to reduce mortality rates.

3.1.1 Objective

Objectives of the project are to reduce human intervention using automation. To implement the automation part a web application will be used. It will integrate a detection model which will automatically which type of tumour is present, based on the inference it obtains. Another objective is to reduce medical treatment costs, by early detection of tumour.

3.1.2 Outcome

Outcomes of this project are as follows: A model system, which analyses the presence of brain tumour, after scanning. It traces the brain tumour present in the MRI Scan, and identifies the presence of Brain Tumour. The Brain Tumour is identified through the automation, which is trained, to identify the presence or absence through simple algorithm. After implementing the specific methodology, the model system is able to

accurately detect brain tumour and then provide further insights for the same.

3.2 Basic Block Diagram

The web application will use the trained model for detection of brain tumour. This detection system works on pre trained models that are able to analyze the input images provided by the end users. Once the system has input images, it performs pre-processing operations on it. Pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. Some of the steps in the Pre-processing include converting the image to grayscale, detecting the skull and brain part in the MRI image and so on. Once the Pre-processing is done, the system determines whether the input MRI image contains any tumour or not. Based on the results obtained from the detection, the system performs post processing. Post processing includes operations such as enhancement of the image. This segment is the processed result image. As part of output result, the system will provide a message on the web application regarding the tumour inference, as well as suggestions for treatment.

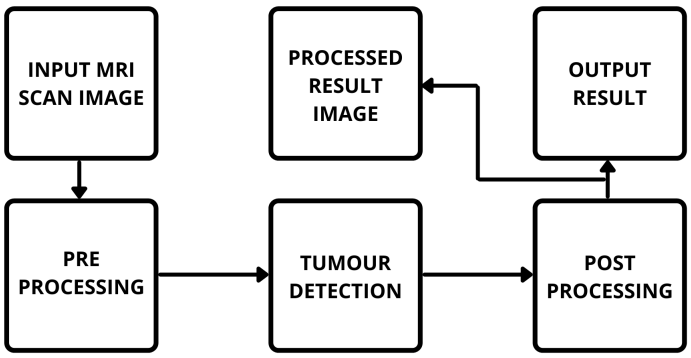


Figure 3.1: Block Diagram

3.3 Requirement analysis

3.3.1 Hardware Requirement

The only hardware requirement for the project which has been observed till now are as follows:

- An End device such as Laptop/PC(Both to train the model and for developing and accessing the web application)

3.3.2 Modern Engineering Tools and Software Requirement

a) Open Source Libraries / Softwares / Tools Requirement:

- A python development environment : Any python environment is needed to train the model which will inturn detect the tumour. For our particular project we have opted for Kaggle Cloud Editor. For the mentioned below reasons:
  - Higher computational power.
  - Ease of importing data from datasets.
  - Version control of code.

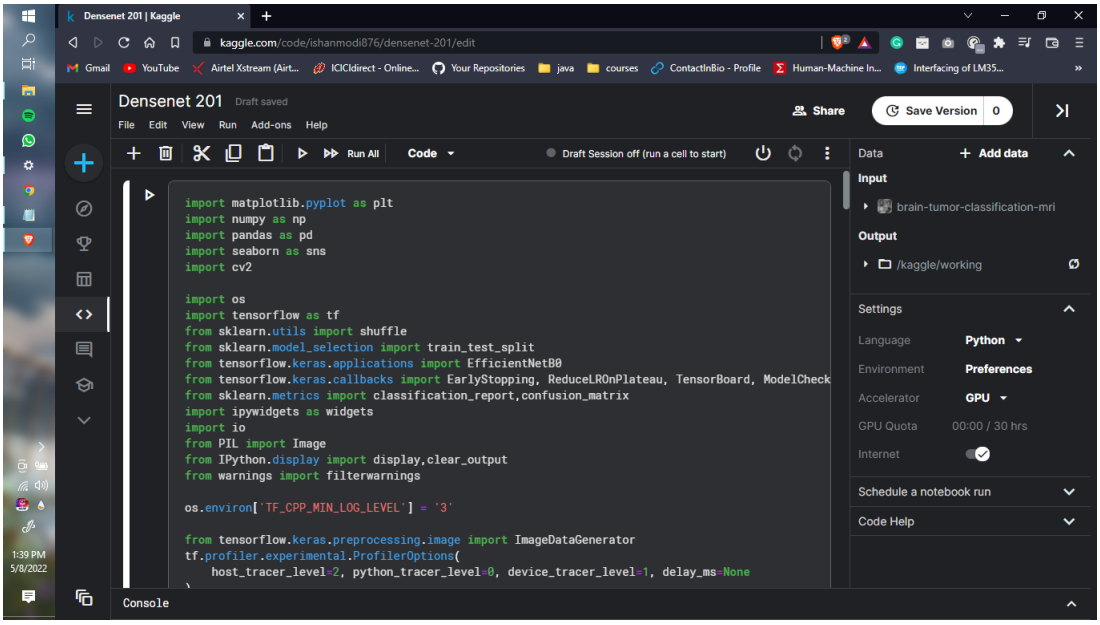


Figure 3.2: Kaggle Editor

- Python Libraries : To implement the project, various libraries are needed to perform the required functionality that we desire. We need Python libraries both for the training of the model as well as to develop the web application. The libraries which have been used until now are as follows:

Table 3.1: List of Python libraries used:

Library	Function	Version(if applicable)
OS	Creating and removing a directory(folder), fetching its contents, changing and identifying the current director.	N/A
Keras	Provide python interface for Neural Networks. Also acts as an interface for Tensorflow Library. Also used to define layers to the model.	2.6.0
Tensorflow	Training and inference of deep neural networks. Also used for establishing callbacks	2.6.1
Numpy	Library for working with multi-dimensional arrays.	1.21.4
Pandas	Data Manipulation and Analysis	1.3.5
Matplotlib	Data visualization and graphical plotting library	3.5.1
Sklearn	Machine Learning library of Python. Features various algorithms such as random forests, K-neighbours etc.	0.23.2
Time	Representing time in code	3.7.0
Imutils	Basic Image Processing	0.5.4
Cv2	Image Processing and computer vision	4.5.4.60
Shutil	Copy and Removal of files	3.10.1
Seaborn	Python Data Visualization library based on Matplotlib	0.11.2
Ipywidgets	Interactive HTML widgets for Jupyter and Ipython Console	8.00rc0

- LaTeX Editor : LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX being a language system needs a system software to support it's functionality. For the project Papeeria has been used to write the project report.

- Streamlit Framework : Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

b) Proprietary Softwares / Libraries / Cloud Requirement:

Proprietary Software include both OS and software applications contained within it. The Proprietary Softwares which have been used in the duration of Phase-1 of the project are as follows:

- Windows OS : Windows OS is the base Operating System, which has been used to develop the code base as well as to do the paper work.

- Microsoft Powerpoint : Microsoft Powerpoint is a proprietary software by Microsoft which has been used to make the presentations for the project as well as develop a proposed UI of the web application.

3.3.3 Resources Requirement

- Datasets : To train the model efficiently datasets are needed which contain the data(MRI scan images in our case). This data is fed to the model which then trains it to detect whether the MRI scan is tumorous or non-tumorous and further classify the tumorous images into 3 categories : Meningioma, Pituitary and Glioma. For the project one dataset has been utilized. The dataset contains 4 subfolders each containing images under four labels as follows : Pituitary, Meningioma, Glioma and Non-Tumorous . The Dataset is as follows:

  – Brain Tumor Classification (MRI) :

    Table (3.2) gives an insight into the contents within the dataset:

Table 3.2: Contents of Brain MRI Images for Brain Tumor Detection Dataset

	Data Label	No. Of Images
Test Set	Glioma	100
	Meningioma	115
	No Tumour	105
	Pituitary	74
Train Set	Glioma	826
	Meningioma	822
	No Tumour	395
	Pituitary	827

Figure (3.3) plots some sample images from the dataset.

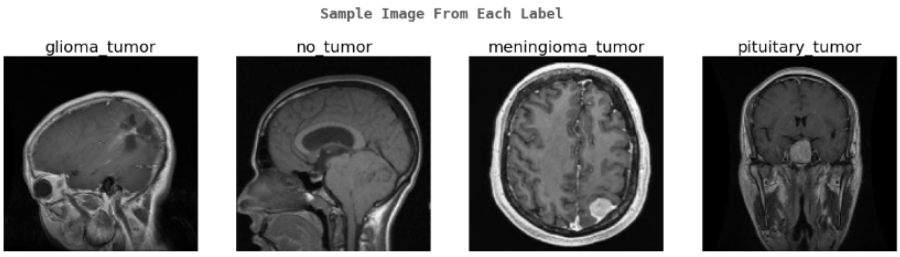


Figure 3.3: Sample Images from Dataset

- Graphical Images : Graphical Images are needed for the Web Application. This is purely a resource requirement for an end user product and is not needed for any core functionality of the system.

3.4 Impact analysis

3.4.1 Impact of project on society

Positive Impact of project on society:

- Early detection of tumour can reduce the cost required to treat the tumour.
- Early detection can overall improve the lifespan of humans.

Negative Impact of project on society:

- It has a risk of errors and wrong detection due to less training.
- It has a risk of not detecting Initial stages of tumour due to small size.

3.4.2 Impact of project on environment

Positive Impact of project on environment:

- Less Bio Medical waste is generated due to early detection and treatment.

Negative Impact of project on environment:

- Bio Medical waste generated during treatment after the detection of tumour is done through this system, if not treated properly can pose environmental risks.

3.5 Limitations

- Limited detection due to less datasets being used in training.
- Accuracy of Dataset is an important factor. Unverified datasets can lead to false results.
- The detection model in it's base form cannot provide accurate results on MRI captured using higher tesla MRI Machines.

3.6 Professional ethical practices to be followed

- Giving credits wherever due.
- Keeping technical guidelines in mind.
- Following the norms of engineering Practices.
- Liability for outcome caused by one's actions or decisions.

Chapter 4

Project Implementation

Initiating the chapter, here project implementation is discussed. The resources required, flowcharts, algorithms as well as proposed layouts have been highlighted here.

4.1 Requirement of Resources

- Software Requirement
    1. OS Requirement:
        - Windows 8/10(Can be used both for Training and at User End)
        - Linux(Only at User End)
    2. Training of the model:
        - Python ide(Anaconda Jupyter Notebook/Jupyter Lab)
        - Dataset for Training
    3. Web Application
- Hardware Requirement
    1. Basic Hardware for Training Purpose:
        - RAM : 8 GB

- ROM : 2 GB
- Processor : i5 Processor

4.2 Development of the Detection model

This section explains about the Detection model and it's training. The model developed is the backend of this project, as it handles the core function of determining whether the MRI image present has tumour or not.

4.2.1 Algorithm

Step 1 : Import the libraries.

Step 2 : Import the dataset.

Step 3 : Define Label Classes.

Step 4 : Append data into training set

Step 5 : Shuffle the training data.

Step 6 : Splitting data into train, test, and validation.

Step 7 : Import pre-trained model.

Step 8 : Add your training layers on top of imported model.

Step 9 : Compile the model.

Step 10 : Define Callbacks.

Step 11 : Train Model.

Step 12 : Defining Model Layers.

Step 13 : Plotting the accuracy and loss graphs.

Step 14 : Evaluating the Model using classification report, manual and automated testing.

4.2.2 Flowchart

The flowchart below describes the process as to how the detection model is developed.

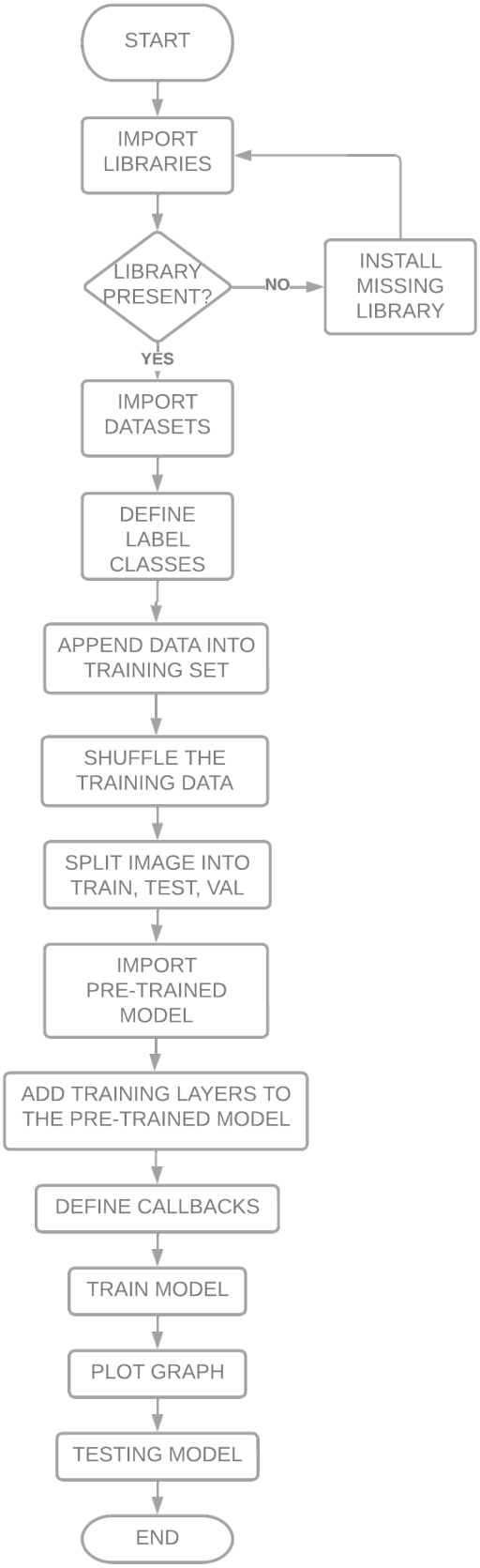


Figure 4.1: Model Training Flowchart

4.2.3 Transfer Learning

Transfer learning is a machine learning concept where a pre-trained model is reused or retrained as the base layer for a model on a new task. In simple words, an already trained model on one task or one problem statement is repurposed for a second, related task. This acts as an optimization that allows rapid progress when modelling the second task is taking place. As observed in figure 4.2 it can be seen that the training layers from Task 1 are transferred as base layers to the task 2, thereby reducing training time, increasing optimization time.

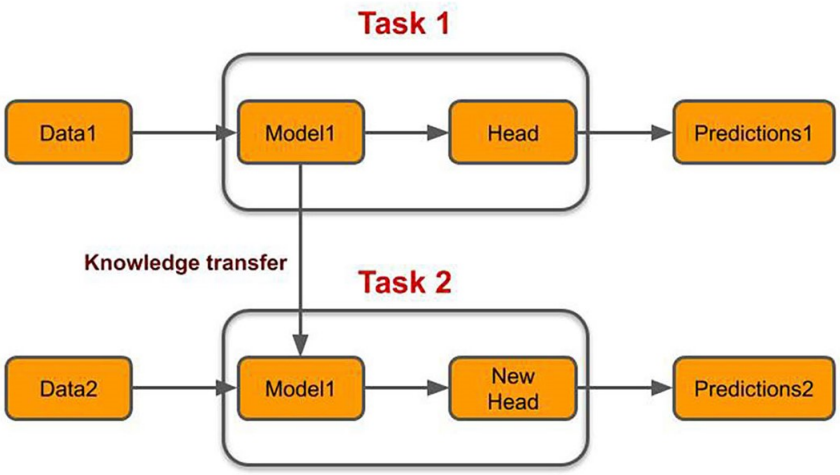


Figure 4.2: Transfer Learning

4.2.4 Defining the Model

A transfer learning based model was used for the project. It is one of the the easiest way to build a model for machine learning applications. As explained in section 4.2.3, already trained model on one task or one problem statement is repurposed for a second, related task. To define the model following libraries must be imported, these libraries are a sub class of "keras.layers" :

- Sequential : It is used to initialize the neural network.
- Convolution2D : This layer deals with creating a convolutional network that will deal with the MRI images.

- MaxPooling2D : This layer adds the pooling layers.
- Flatten : This layer converts the pooled feature map to a single column. This is passed to the fully connected layer.
- Dense : Dense layer will add this fully connected layer to the neural network.
- Dropout : Dropout is a technique used to prevent a model from overfitting.
- BatchNormalization : It is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch.

4.3 Proposed Web Application

This section explains about the Proposed Web Application. In this section detailed Algorithm is shown, as well as gives an insight on Local Server Deployement and Logs.

4.3.1 Welcome Page

Step 1 : Complete Web application runs on the local host web page in your browser.
Step 2 : Video division and basic information is displayed on this page.
Step 3 : Top left corner of the web application has option button and drop down menu access which consists of the different routes for the web application.
Step 4 : You can click on various routes to get the particular page displayed.

4.3.2 Information Page

Step 1 : Here you find all the basic and medical related info for the Brain Tumours.
Step 2 : Scroll down to get additional information for diagnosis, traditional methods and the web application method.
Step 3 : Interactivity buttons are added in between to display information about the various tumours and treatment methods.

4.3.3 Algorithm

Step 1 : Web-App is running on Streamlit framework.

Step 2 : User inputs the image. Step 3 : If any other file other than allowed image extension, system logs message as "Enter valid file type only".

Step 4 : If input file in accordance with allowed image extension, move to step 5.

Step 5 : Input image is passed onto the preexisting Knowledge Base (Detection Model).

Step 6 : If tumour is detected, the system further gives an inference between whether the tumour is of : meningioma, pituitary or glioma. Needed output will be printed along with the confidence score.

Step 7 : If tumour is not detected, the system returns output as "Tumour Not Detected".

4.3.4 Backend Results

Result : This is the result section where in it shows all the result in the form of confusion matrix, classification report which are respectively explained in the web application's particular route.

4.3.5 Team Information Page

Step 1 : This page consists of the information of our developing team.

Chapter 5

Results and Discussion

For the development phase of the project 1 Dataset has been used for training namely:

- Brain Tumor Classification (MRI)

The dataset is divided into four labels: Glioma, Meningioma, No Tumour and Pituitary tumour. This data is then shuffled using shuffle command of keras. All this data was then passed onto a final dataset, which was then separated into 3 segments: Train , Test and Validation in ratio of 80:10:10. Once the data has been split, a pre-trained model is imported and on top of that pre-trained layers, we add new training layers for the task being performed. This concept is known as transfer learning. Then these train images are passed on to the model. The model has been trained for 19 versions, where each version differs in the pre-trained model being imported. These results associated with each version can be understood in the section 5.4. Based on the results from each verion the highest accuracy achieved is 97% .

5.1 Dataset Information

Table 5.1 displays the number of imgaes in the dataset.

Table 5.1: Contents of Brain MRI Images for Brain Tumor Detection Dataset

	Data Label	No. Of Images
Test Set	Glioma	100
	Meningioma	115
	No Tumour	105
	Pituitary	74
Train Set	Glioma	826
	Meningioma	822
	No Tumour	395
	Pituitary	827

5.2 Sample Data from Dataset

Figure (5.1) plots some sample images from the dataset. These images are generated using a random function and the matplotlib library. One image from each label (Glioma, No Tumour, Meningioma, Pituitary) is taken from the dataset using the random function.

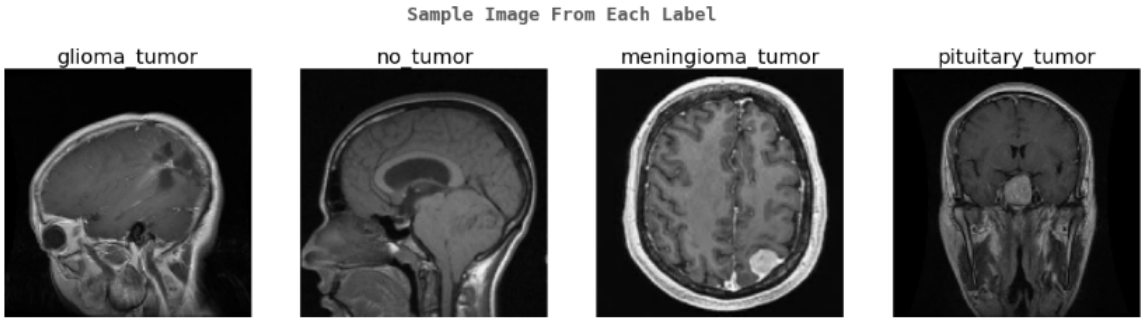


Figure 5.1: Sample Images from Dataset1

5.3 Training the Model

5.3.1 Preparing the Data:

X-train and Y-train data are prepared first so as to feed them to the model for training. X-train is an array which consists of the actual images which will be fed to the model. Y-train refers to the label about which type of tumour that particular image is. Figure 5.2 shows the output log of the number of images from each label being fed into the X train array. The log shows us the items/second the system was able to append to the array, based on the label; as well as the time the system takes to perform the required

action on each particular label.

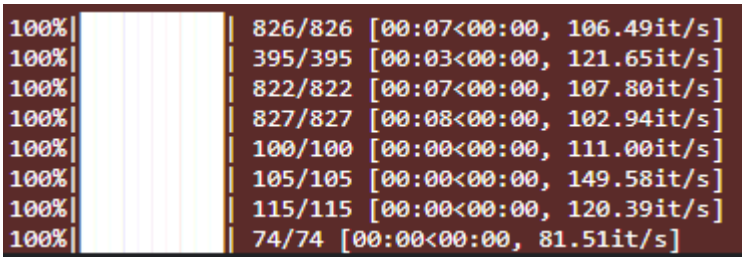


Figure 5.2: X train and Y train generation Log

5.3.2 Processing the Images:

Processing involves techniques such as cropping, RGB to Gray conversion, blurring the image and so on. This processing is performed within the web application. Figure 5.3 shows the processing results performed on meningioma tumor MRI scan.

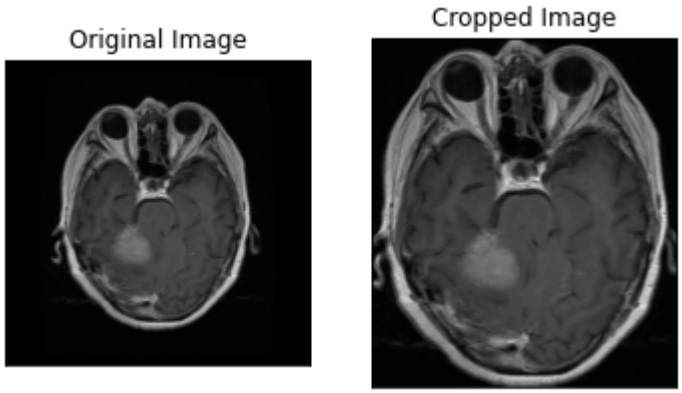


Figure 5.3: Meningioma Tumor Processing Results

Figure 5.4 shows the processing results performed on glioma tumor MRI scan.

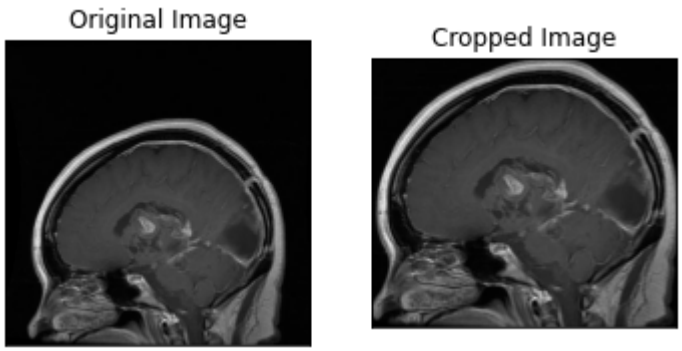


Figure 5.4: Glioma Tumor Processing Results

Figure 5.5 shows the processing results performed on pituitary tumor MRI scan.

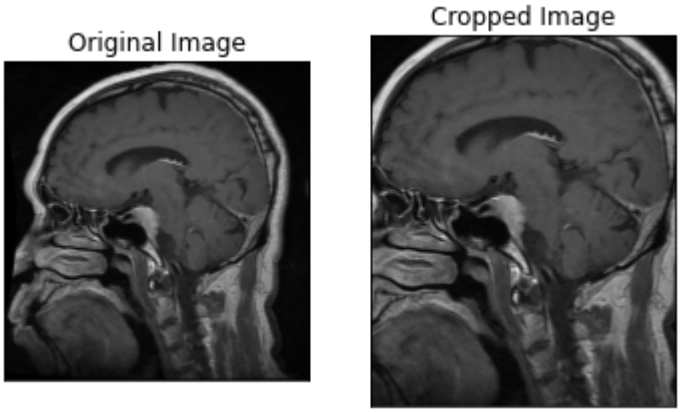


Figure 5.5: Pituitary Tumor Processing Results

Figure 5.6 shows the processing results performed on non tumor MRI scan.

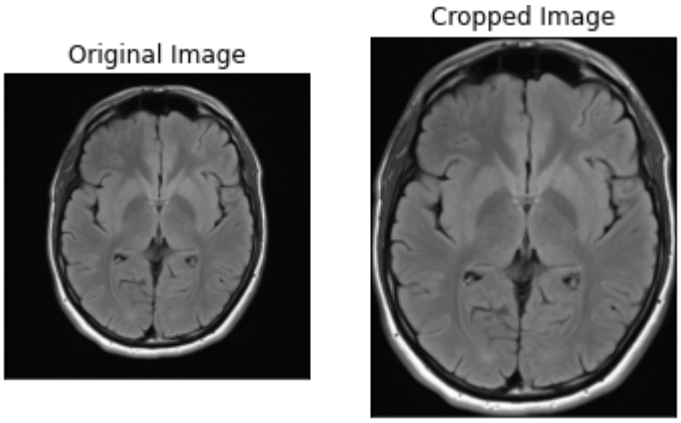


Figure 5.6: No Tumor Processing Results

5.3.3 Downloading the pre-trained model:

Once the data has been prepared, it has to be provided as input to the model. Since the project uses, transfer learning as the model defining principle, using "tf.keras.applications" we first download the model, and on top of that new training layers are added. Figure 5.7 shows the download log of the Densenet201 Model downloaded. The model downloaded is notop model, since on top of this model new layers have to be defined.

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet101v2_weights_tf_di
m_ordering_tf_kernels_notop.h5
171319296/171317808 [==============================] - 4s 0us/step
171327488/171317808 [==============================] - 4s 0us/step
```

Figure 5.7: Model Download Log

5.3.4 Training the Model:

Once the model has been defined, the X-Train and Y-Train data is split into a ratio of 80:10:10, where 80% of data is train set, and 10% data is each test and validation data set. Next, the model is compiled. It is done using model.compile function. Here the different metrics to be used/displayed are defined. This model.compile also deals with defining the optimizer to be used. Here "Adam" optimizer has been used. Once this has been done, callbacks are defined. Callbacks are the functions, which can interrupt the training of the model so as to prevent overfitting. Callbacks can also be used to keep track of the accuracy of every epoch or training cycle, and save the best model. Two callbacks have been used in the project; they are as follows

- ModelCheckpoint : ModelCheckpoint callback is used to save model at particular epoch/training based on the validation accuracy or any other metric the callback is asked to check for.

- ReduceLROnPlateau : ReduceLROnPlateau callback reduces the learning rate of the model when the metric stops improving.

Next, the actual parameters for training like how many iterations the model should train for, input data, callbacks and various other parameters are provided using the model.fit command. Here the model has been trained for 12 epochs. Figure 5.8 shows the log of the epoch training. The log shows the number of images used in each iteration, the time it took to perform the training, and the loss and accuracy metrics.

```
Epoch 00011: val_accuracy did not improve from 0.96939
Epoch 12/12
83/83 [==============================] - 14s 170ms/step - loss: 0.0074 - accuracy: 0.9989 - val_loss: 0.1271 - v
al_accuracy: 0.9592
```

Figure 5.8: Epoch Log

5.4 Training Results

As part of model training different pre-trained models were considered. These pre-trained models are open source and are inbuilt functions of Keras.application library.

Table 5.2 highlights the different results obtained with each version. Each version uses a different Pre-Trained model from the keras library. Due to different layers being imported in different version, the Test Accuracy and Test Loss for each version varies. Based on the results obtained, Densenet201 is the best version for the application of the project.

Table 5.2: Results obtained from each version

Version	Model	Test Accuracy	Test Loss
1	Densenet121	0.960244656	0.186115995
2	Densenet169	0.966360867	0.16515483
3	Densenet201	0.972477078	0.142509162
4	EfficientNetB0	0.788990855	0.551147401
5	EfficientNetB1	0.440366983	2.732621908
6	EfficientNetB2	0.773700297	0.561737001
7	EfficientNetB3	0.65749234	1.091159701
8	EfficientNetB4	0.76452601	0.656704843
9	EfficientNetB5	0.409785926	3.710002422
10	EfficientNetB6	0.865443408	0.362860888
11	EfficientNetB7	0.446483195	2.516769886
12	InceptionResNetV2	0.960244656	0.217766672
13	InceptionV3	0.963302732	0.139801413
14	MobileNet	0.966360867	0.169000089
15	MobileNetV2	0.926605523	0.51417464
16	MobileNetV3Large	0.840978622	0.996828198
17	MobileNetV3Small	0.85015291	0.87231195
18	ResNet101	0.948012233	0.204600975
19	ResNet101V2	0.929663599	0.251542419

5.5 Results of the Best Version

Based on the results obtained from table 5.2 Densenet201 proves to be the best model for the problem the project aims to solve. Figure 5.9 shows the accuracy and loss graph of the model trained using Densenet201 as base layers.

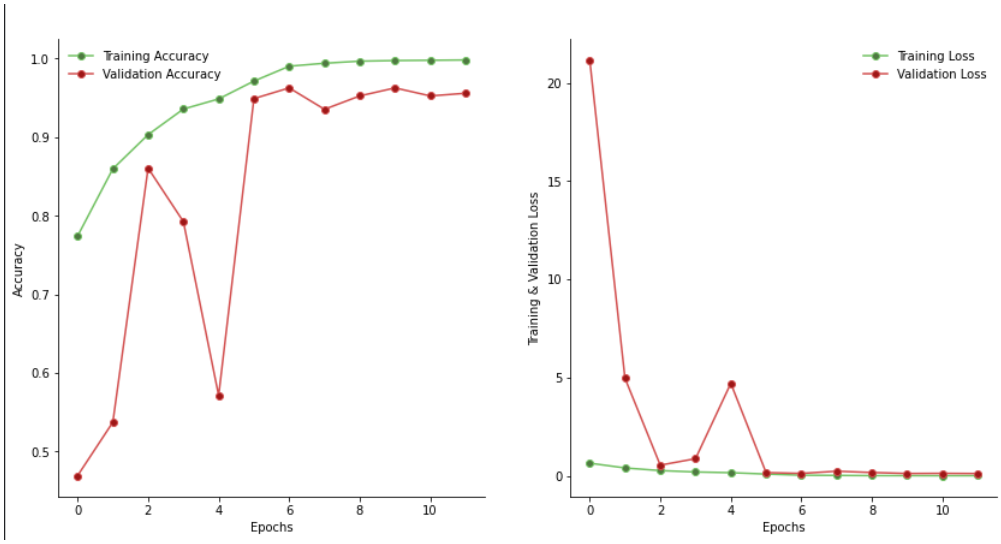


Figure 5.9: Accuracy Graph from the best version

Figure 5.10 shows the confusion matrix heatmap of the Densenet201 model. Confusion matrix is an evaluation parameter which gives an insight into the model accuracy by comparing the actual and predicted outputs with each other. As it can be observed, the model has a very high accuracy and is able to correctly classify images into their correct class.

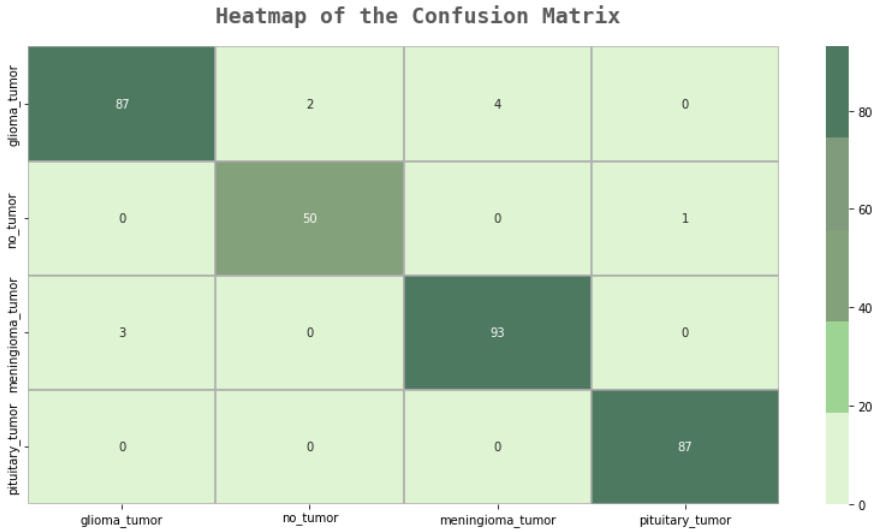


Figure 5.10: Confusion Matrix

Figure shows the classification report. A classification report, gives information about the quality of the predictions. There are factors like, precision, recall, f1-score and accuracy. The numbers 0,1,2, and 3 are the numbers of the labels, in order of glioma, no-tumor, meningioma and pituitary tumor. T

	precision	recall	f1-score	support
0	0.97	0.94	0.95	93
1	0.96	0.98	0.97	51
2	0.96	0.97	0.96	96
3	0.99	1.00	0.99	87
accuracy			0.97	327
macro avg	0.97	0.97	0.97	327
weighted avg	0.97	0.97	0.97	327

Figure 5.11: Classification Report

5.6 Accuracy and Loss graph of every version in table 5.2

Figure 5.12 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using Densenet121 as the base model.

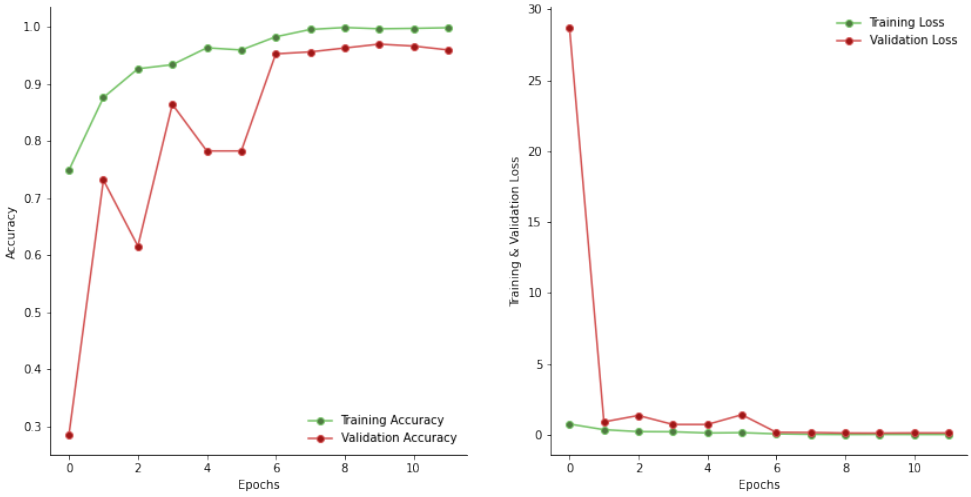


Figure 5.12: Version 1 : Densenet121

Figure 5.13 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using Densenet169 as the base model.

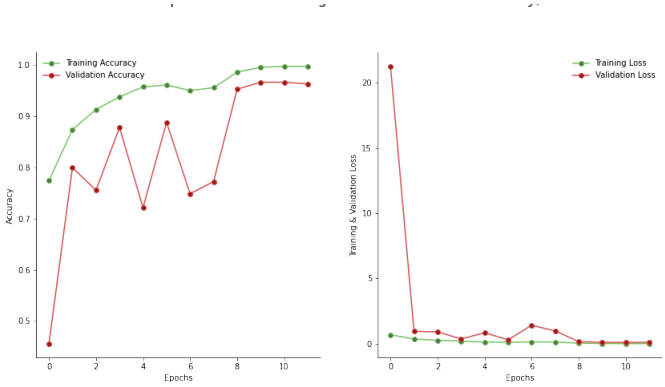


Figure 5.13: Version 2 : Densenet169

Figure 5.14 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using Densenet201 as the base model.

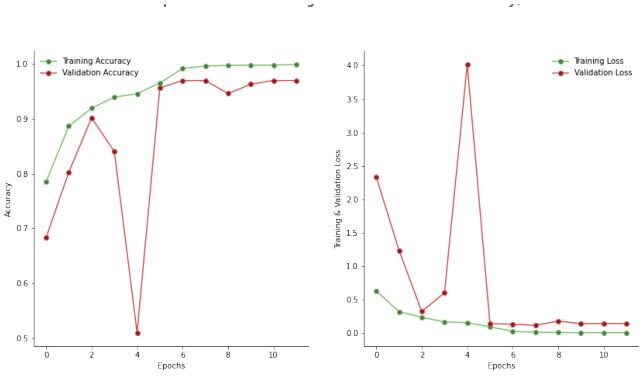


Figure 5.14: Version 3 : Densenet201

Figure 5.15 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB0 as the base model.

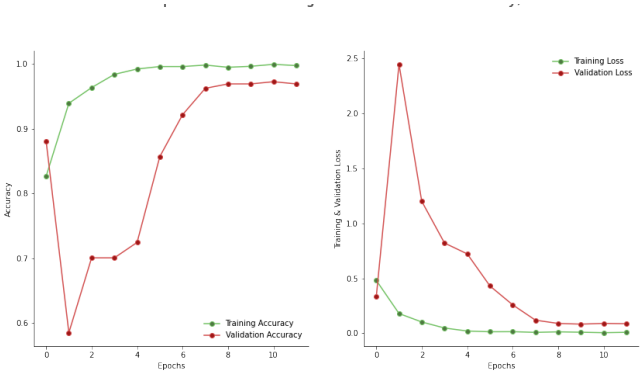


Figure 5.15: Version 4 : EfficientNetB0

Figure 5.16 shows the test loss vs epochs and test accuracy vs epoch graph of the

model generated using EfficientNetB1 as the base model.

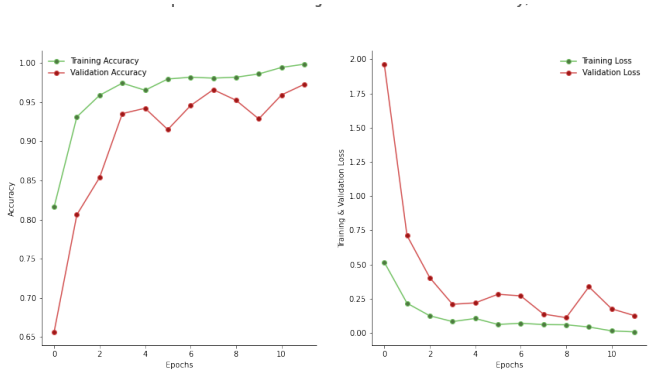


Figure 5.16: Version 5 : EfficientNetB1

Figure 5.17 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB2 as the base model.

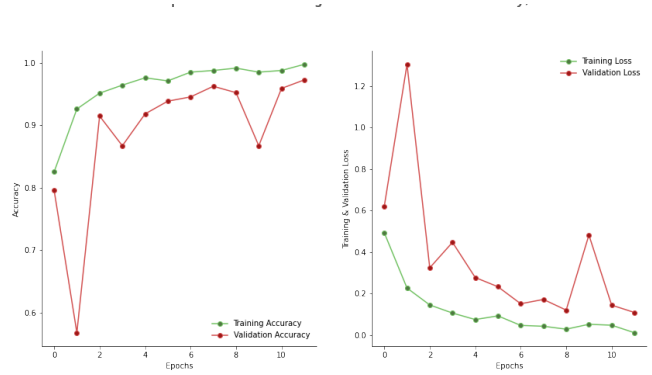


Figure 5.17: Version 6 : EfficientNetB2

Figure 5.18 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB3 as the base model.

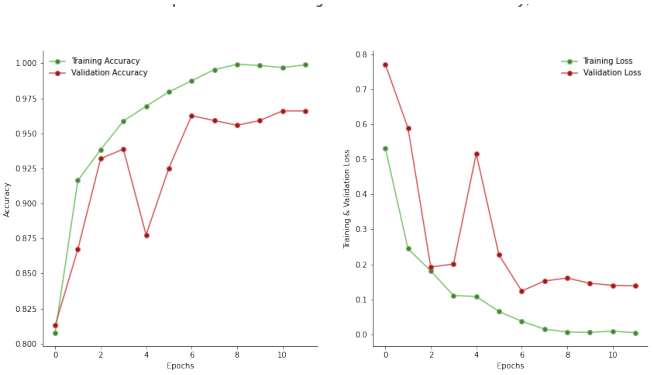


Figure 5.18: Version 7 : EfficientNetB3

Figure 5.19 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB4 as the base model.

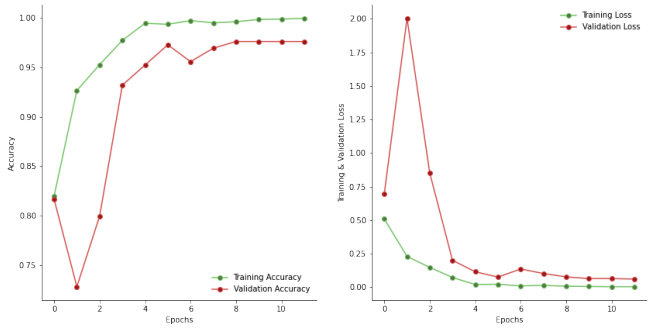


Figure 5.19: Version 8 : EfficientNetB4

Figure 5.20 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB5 as the base model.

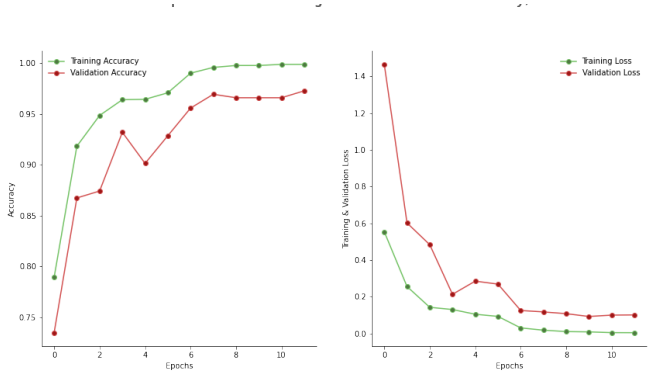


Figure 5.20: Version 9 : EfficientNetB5

Figure 5.21 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB6 as the base model.

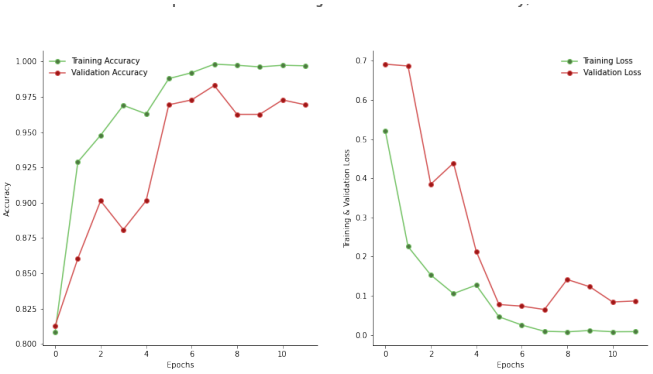


Figure 5.21: Version 10 : EfficientNetB6

Figure 5.22 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using EfficientNetB7 as the base model.

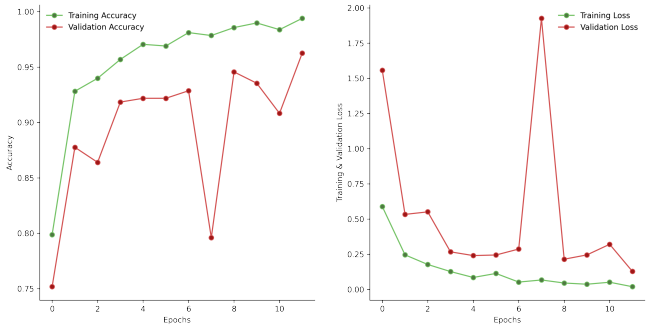


Figure 5.22: Version 11 : EfficientNetB7

Figure 5.23 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using InceptionResNetV2 as the base model.

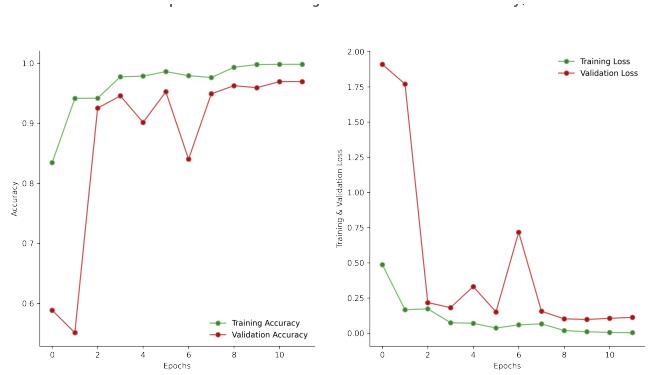


Figure 5.23: Version 12 : InceptionResNetV2

Figure 5.24 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using InceptionV3 as the base model.

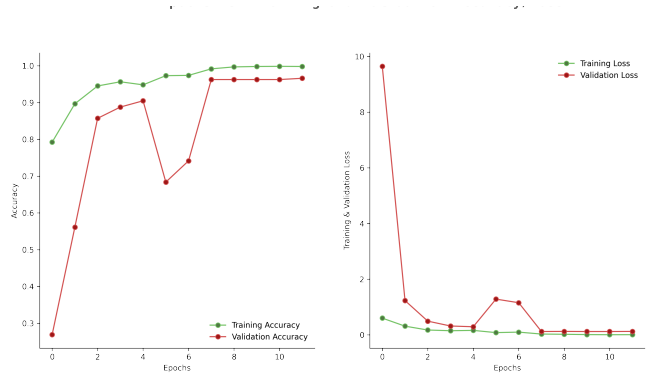


Figure 5.24: Version 13 : InceptionV3

Figure 5.25 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using MobileNet as the base model.

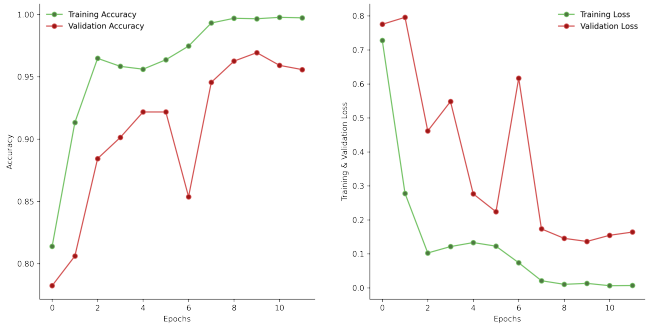


Figure 5.25: Version 14 : MobileNet

Figure 5.26 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using MobileNetV2 as the base model.

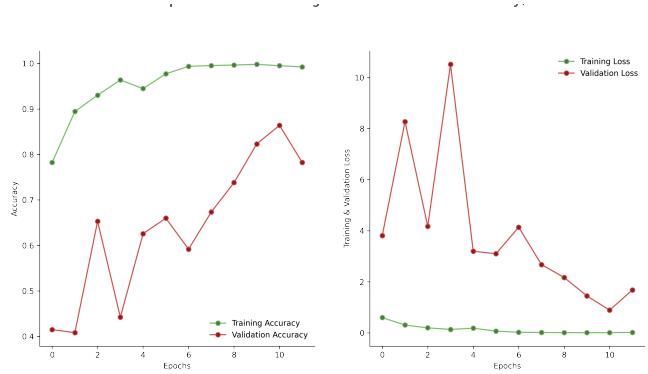


Figure 5.26: Version 15 : MobileNetV2

Figure 5.27 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using MobileNetV3Large as the base model.

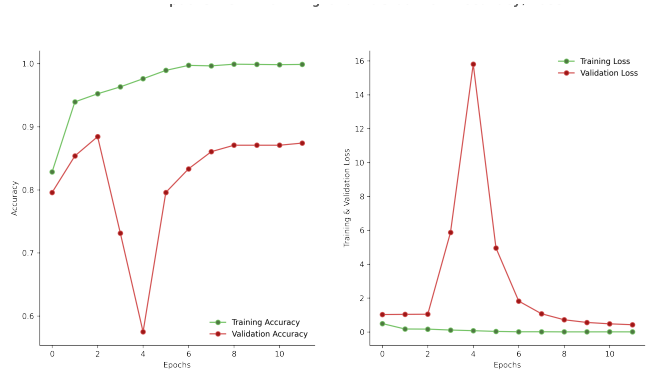


Figure 5.27: Version 16 : MobileNetV3Large

Figure 5.28 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using MobileNetV3Small as the base model.

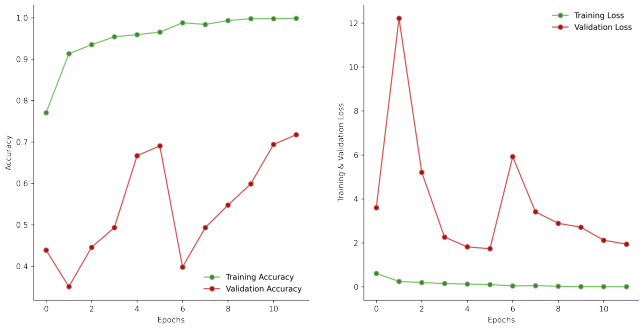


Figure 5.28: Version 17 : MobileNetV3Small

Figure 5.29 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using ResNet101 as the base model.

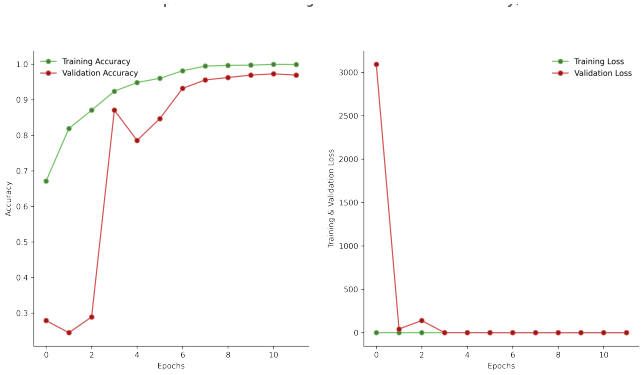


Figure 5.29: Version 18 : ResNet101

Figure 5.30 shows the test loss vs epochs and test accuracy vs epoch graph of the model generated using ResNet101V2 as the base model.

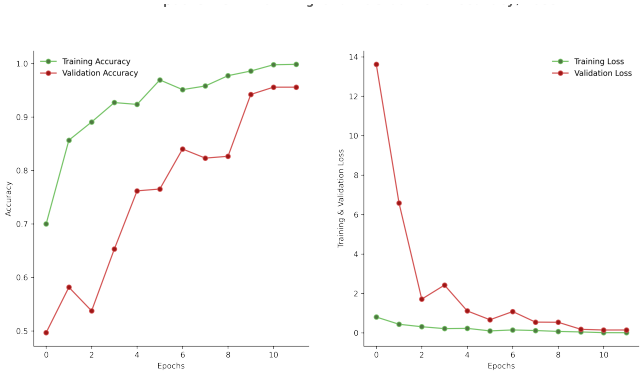


Figure 5.30: Version 19 : ResNet101V2

5.7 Manual Test Results

In Manual Testing images from an unbiased dataset is used to test the model on individual images. The images from the unbiased dataset have not been used to training of the model at any point; hence it acts as unbiased data. The results from the manual testing are as follows:

- Glioma Tumour Present in the Brain:

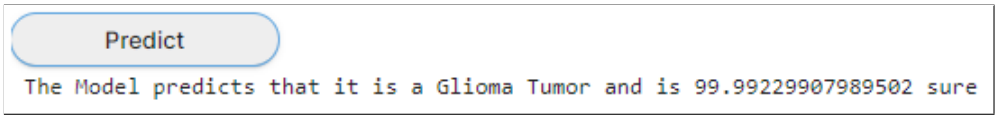


Figure 5.31: Glioma Tumour Manual Testing Results

- Meningioma Tumour Present in the Brain:

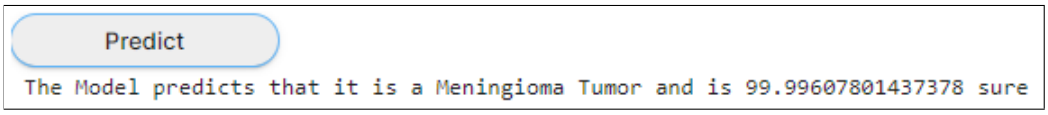


Figure 5.32: Meningioma Tumour Manual Testing Results

- Pituitary Tumour Present in the Brain:

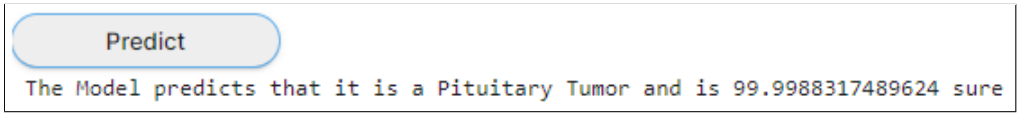


Figure 5.33: Pituitary Tumour Manual Testing Results

- Tumour Absent in the Brain:

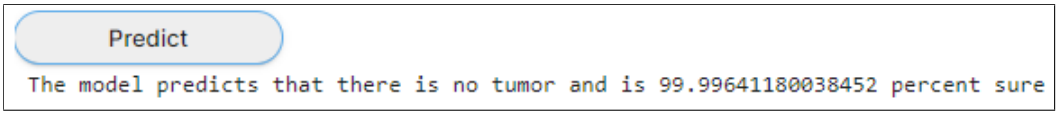


Figure 5.34: No Tumour Manual Testing Results

5.8 Terminal logs for starting the Web Application

The figure 5.35 displays the initial commit which is used for, starting the web application. We use the command "streamlit run filename.fileextention" to start the streamlit

server. The figure 5.36 displays the terminal logs post server deployment. Once the "streamlit run filename.fileextension" command has been executed, the server begins running and provides with a local server link for deploying it to the local browser. This local server link is as follows: "Local URL: http://localhost:8501" and "Network URL: http://172.27.112.252:8501". This log also will contain every backend process that occurs within the web application, like uploading an image, or processing some data.

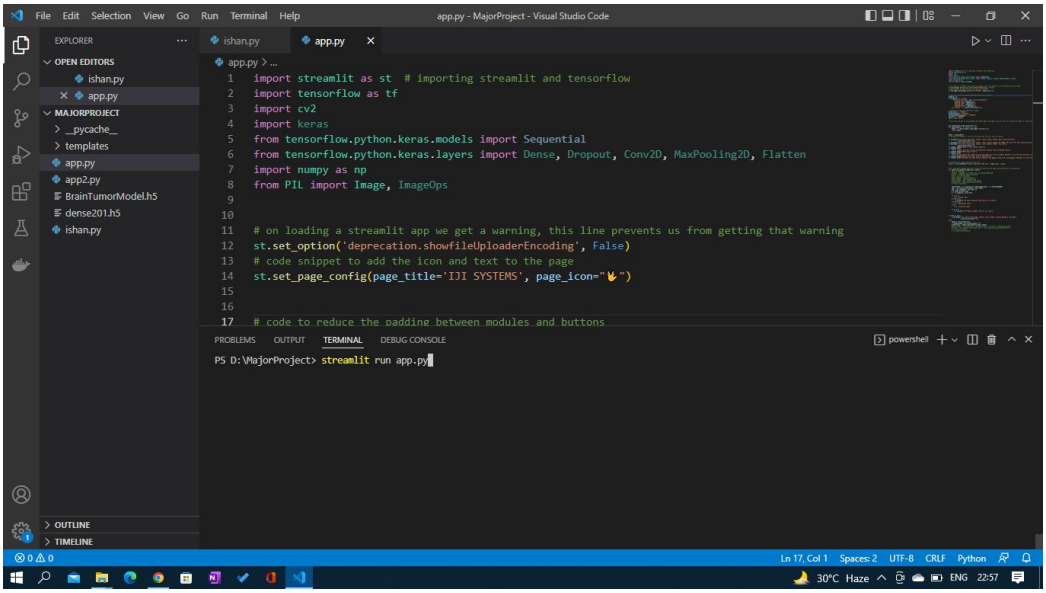


Figure 5.35: Initial Terminal Commit

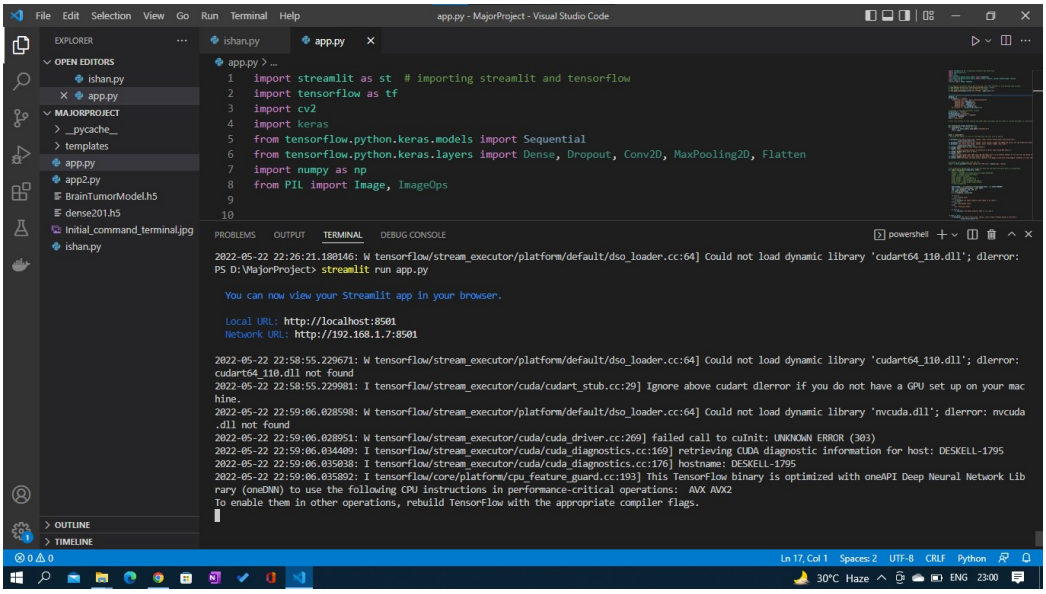


Figure 5.36: Terminal Logs when Application starts

5.9 Web Application User Interface

The web application has been developed using Streamlit framework. Figure 5.37 shows the comprehensive web application front end constituents where it has a sidebar on the left side and central area of the web app has the welcome page. This sidebar contains the navigation of the application, thereby enabling to change to different routes based on the user choice.

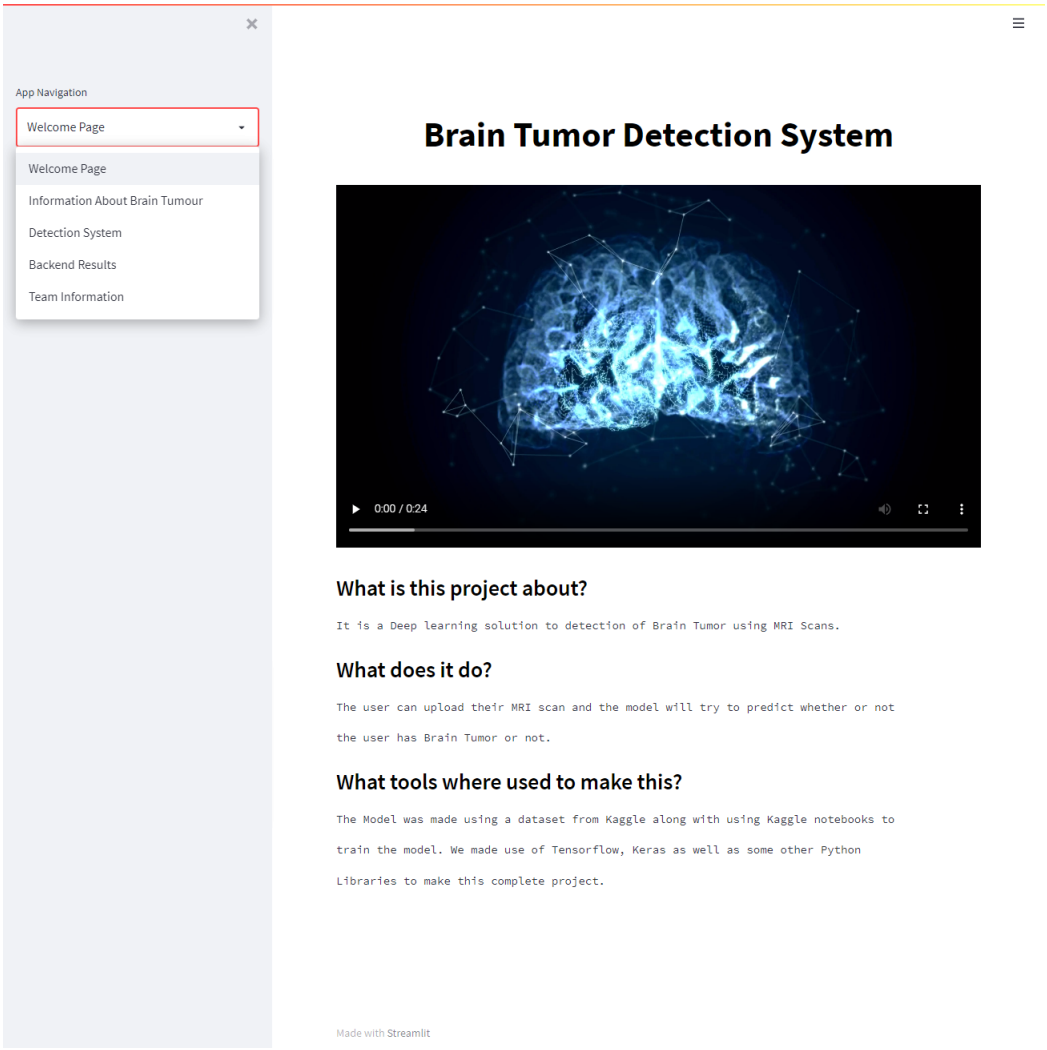


Figure 5.37: User Interface with sidebar

Figure 5.38 is the first page or route of the web application which is having the welcome page and information about how this model works with a video animation embedded on top. The route also highlights the tools used to make the application.

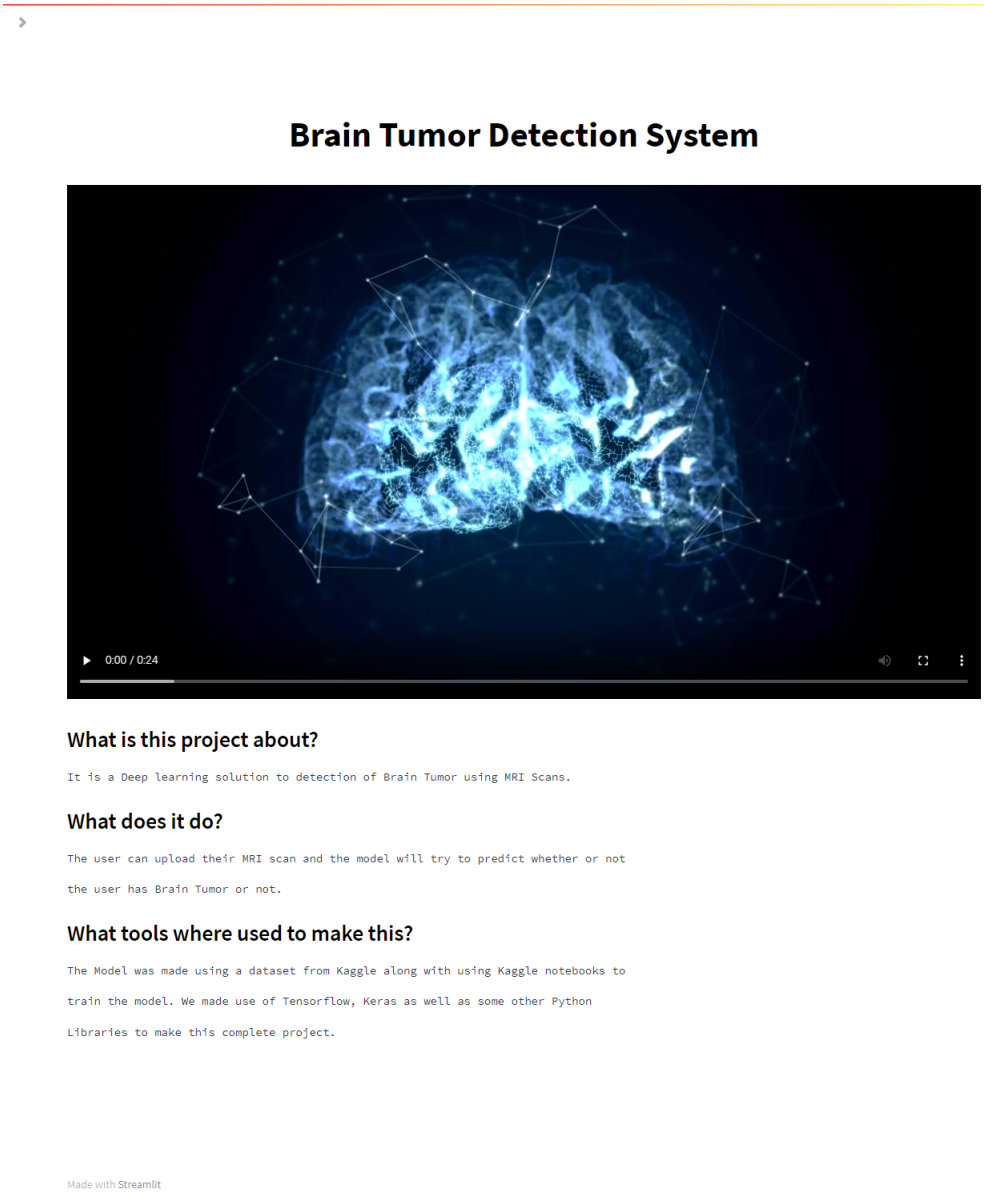


Figure 5.38: Welcome Page

Figure 5.39 is the information page route of the web application which has interactivity with the information mentioned about brain tumours, it's types, diagnosis, awareness and the treatment methods. It has the hyperlinks connected for displaying more information about the particular type of tumour and treatment methods. The page gives a brief insight into three types of tumors in particular : glioma tumor, pituitary tumor, and meningioma tumor.

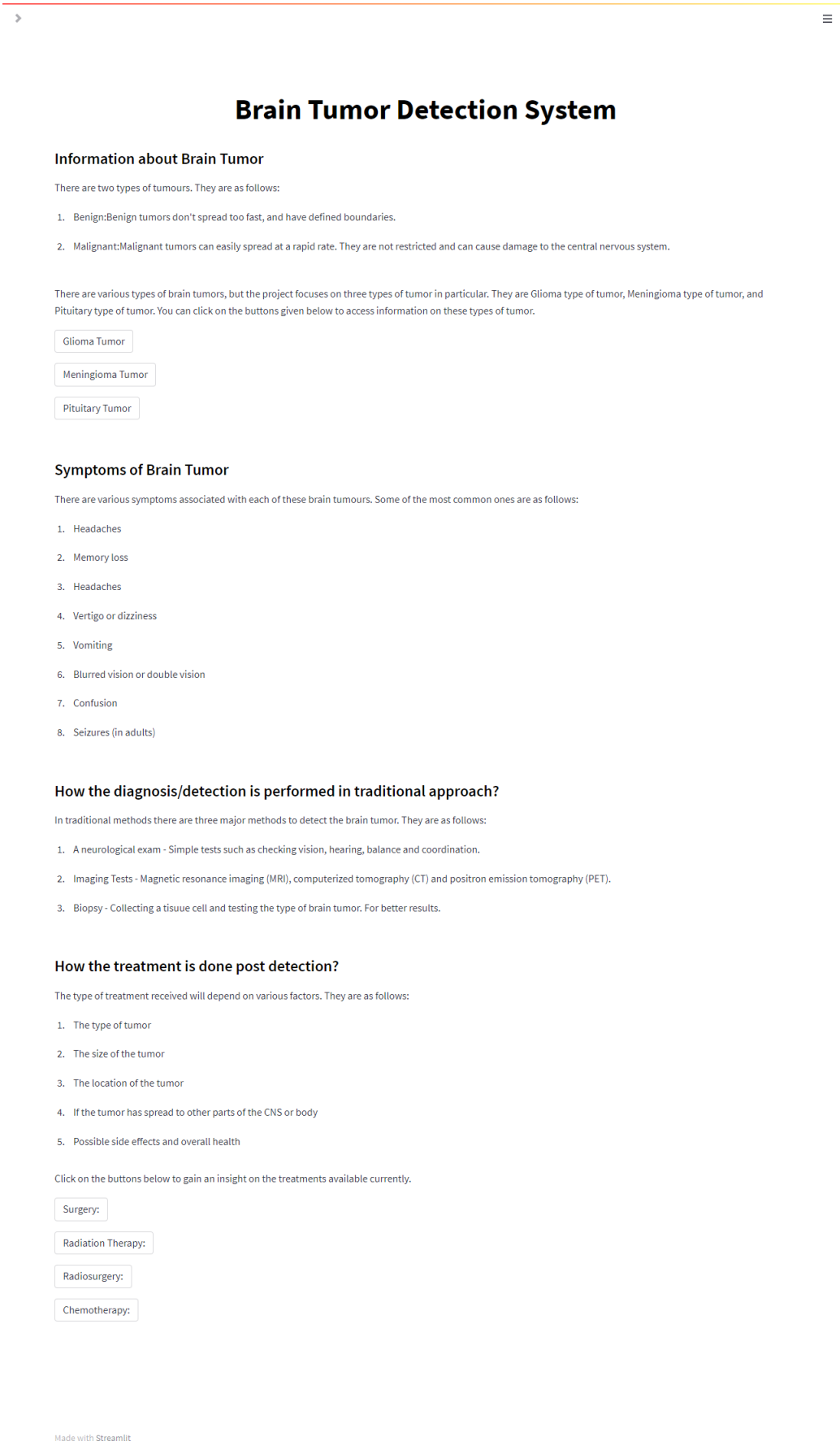


Figure 5.39: Brain Tumor Information Page

Figure 5.40 is the main detection model system which has the constituents of a input dialog box with specified memory size and format of the image which acts as input to the detection model. After model detects , it shows what type of tumour it is and a small description about the tumour found on the image. The web application also prints a confidence score as to how much the model has confidence that the particular uploaded image is of that type of label.

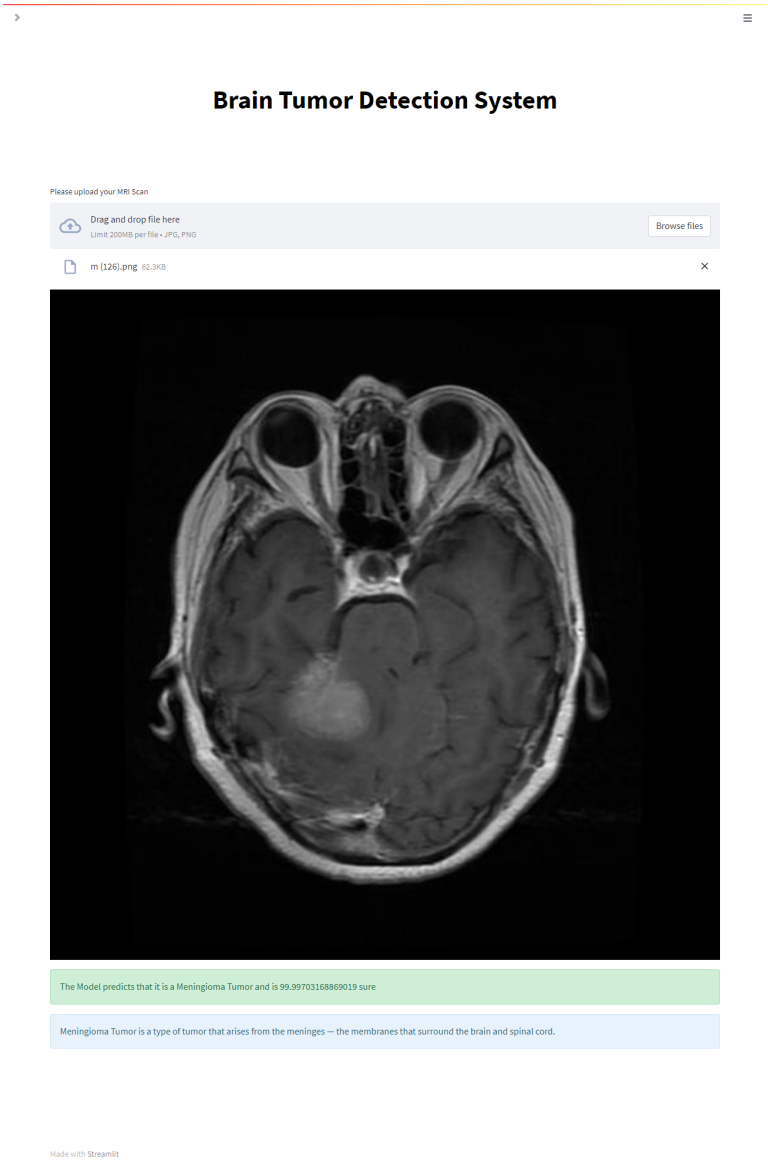


Figure 5.40: Detection Page

Figure 5.41 shows the general backend result images which includes classification report, confusion matrix and explanation.

Brain Tumor Detection System

Backend Results

1. Classification Report

	precision	recall	f1-score	support
0	0.97	0.94	0.95	93
1	0.96	0.98	0.97	51
2	0.96	0.97	0.96	96
3	0.99	1.00	0.99	87
accuracy			0.97	327
macro avg	0.97	0.97	0.97	327
weighted avg	0.97	0.97	0.97	327

A classification report, gives information about the quality of the predictions. There are factors like, precision, recall, f1-score and accuracy. The numbers 0,1,2, and 3 are the numbers of the labels, in order of glioma, no-tumor, meningioma and pituitary tumor. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. TN / True Negative: when a case was negative and predicted negative.
2. TP / True Positive: when a case was positive and predicted positive.
3. FN / False Negative: when a case was positive but predicted negative.
4. FP / False Positive: when a case was negative but predicted positive

• Precision is the accuracy of positive predictions.

Precision = TP/(TP + FP)

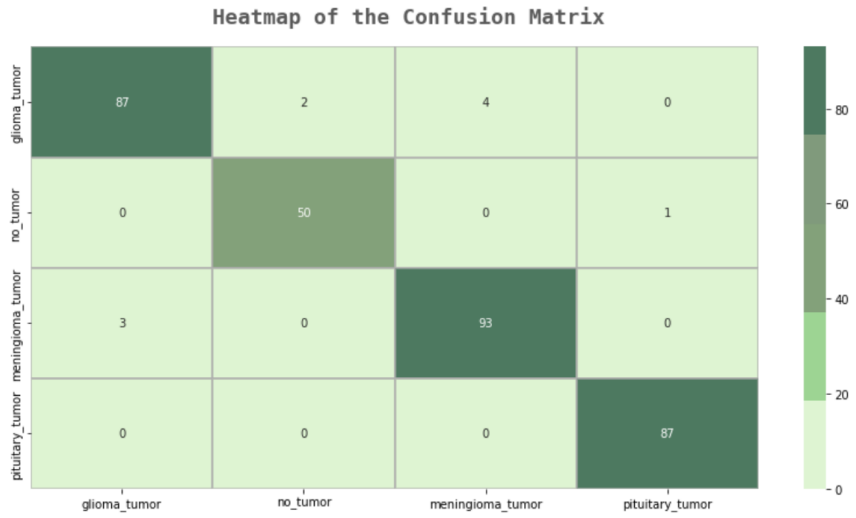
• Recall is Fraction of positives that were correctly identified.

Recall = TP/(TP+FN)

• F1 score is the percentage of positive predictions that were correct.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

2. Confusion matrix



Confusion matrix is an evaluation parameter which gives an insight into the model accuracy by comparing the actual and predicted outputs with each other. As it can be observed, the model has a very high accuracy and is able to correctly classify images into their correct class.

Made with Streamlit

Figure 5.41: Backend Results

Figure 5.42 is the route screenshot for the team information.

Figure 5.42: Team Information

5.10 Results from the Web Application

- Glioma Tumour Present in the Brain: The figure 5.43 displays the result on the Web Application when the User uploads a MRI Scan which is of glioma type.

Figure 5.43: Glioma Tumour Web Application Results

- Meningioma Tumour Present in the Brain: The figure 5.44 displays the result on the Web Application when the User uploads a MRI Scan which is of meningioma type.

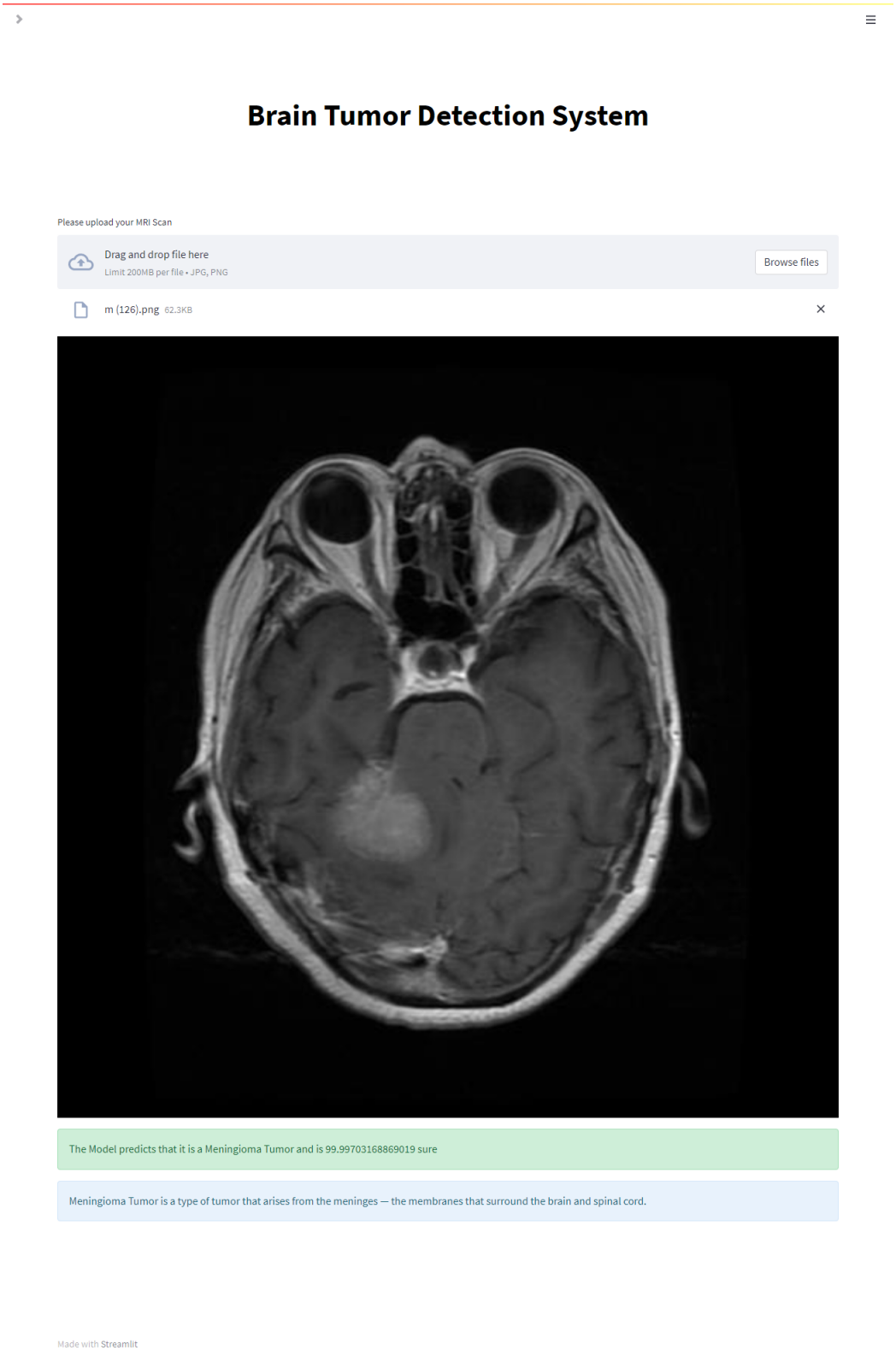


Figure 5.44: Meningioma Tumour Web Application Results

- Pituitary Tumour Present in the Brain: The figure 5.45 displays the result on the Web Application when the User uploads a MRI Scan which is of pituitary type.

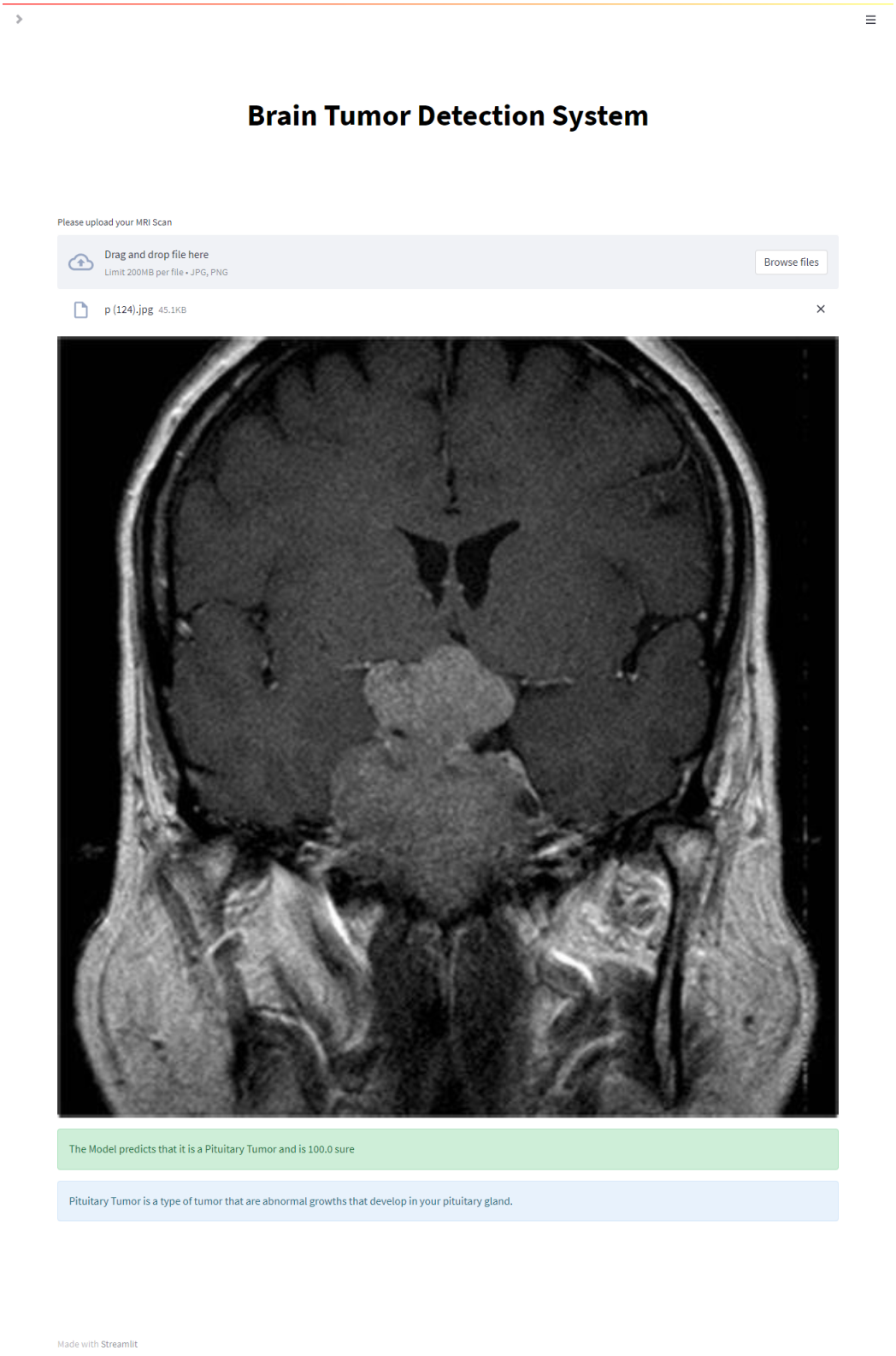


Figure 5.45: Pituitary Tumour Web Application Results

- Tumour Absent in the Brain: The figure 5.46 displays the result on the Web Application when the User uploads a MRI Scan which is of non-tumorous type.

Figure 5.46: No Tumour Web Application Results

Chapter 6

Conclusions and Future Scope

6.1 Conclusions

Implementation of a Deep Learning Model which detects whether there is brain tumour present or not using image segmentation and analysis. The model if it observes tumorous image is further able to classify the tumorous images into 3 categories : Meningioma, Pituitary and Glioma. A total of 19 versions based on CNN Transfer Learning are developed with various parameters into consideration for detection of the tumour images which are being fed to the model. Based on these results and further testing the model with the best accuracy gives an system accuracy of 97%. This result is verified using model evaluate function as well as the manual image set testing. Input image is passed to the trained model which then determines whether image contains tumour or not. As per the inference by the model the output will be displayed on the web application accordingly.

6.2 Future Scope

- The particular trained model can be further trained to detect tumour of initial stages.
- The system in it's base form can be trained further to detect the tumour even

more efficiently

- The entire Web Application can be further developed to understand more types of human body scans such as X-ray, CT - Scans etc. So as to become a full fledged health assessment portal.
- Such web application could be installed at hospitals which will then connect patients and the hospital staff virtually.
- The system can be upgraded to detect more types of tumours such as ovarian, breast, skin, lung tumours etc.

References

[1] Avigyan Sinha, Aneesh R P, Malavika Suresh, Nitha Mohan R, Abinaya D, Ashwin G Singerji "Brain Tumour Detection Using Deep Learning" *Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*, 2021

[2] Pär Salander, A Tommy Bergenheim, Katarina Hamberg, Roger Henriksson , "Pathways from symptoms to medical care: a descriptive study of symptom development and obstacles to early diagnosis in brain tumour patients" *Family Practice, Volume 16, Issue 2, April 1999, Pages 143–148*

[3] McKinney PA, "Brain tumours: incidence, survival, and aetiology," *Journal of Neurology, Neurosurgery & Psychiatry 2004*

[4] Muhammad Waqas Nadeem, Mohammed A. Al Ghamdi, Muzammil Hussain, Muhammad Adnan Khan, Khalid Masood Khan, Sultan H. Almotiri and Suhail Ashfaq Butt, "Brain Tumor Analysis Empowered with Deep Learning: A Review, Taxonomy, and Future Challenges" *Brain Sci.*2020

[5] Logeswari, T.; Karnan, M."An improved implementation of brain tumor detection using segmentation based on hierarchical self organizing map." *Int. J. Comput. Theory Eng. 2010, 2, 591.*

[6] Yang, G.; Raschke, F.; Barrick, T.R.; Howe, F.A."Manifold Learning in MR spectroscopy using nonlinear dimensionality reduction and unsupervised clustering." *Magn. Reson. Med. 2015, 74, 868–878.*

[7] Mansi Lathera, Dr. Parvinder Singh"Investigating Brain Tumor Segmentation and Detection Techniques " *International Conference on Computational Intelligence and Data Science (ICCIDS 2019)*

[8] Nandita Goyal and Dr. Bharti Sharma"Image Processing Techniques for Brain Tumor Identification" *IOP Conf. Ser.: Mater. Sci. Eng. 1022 012011*

[9] Isselmou Abd El Kader, Guizhi Xu, Zhang Shuai, Sani Saminu, Imran Javaid, Isah Salim Ahmad and Souha Kamhi"Brain Tumor Detection and Classification on MR Images by a Deep Wavelet Auto-Encoder Model" *Diagnostics 2021*

[10] Saurabh Kumar, Iram Abid, Shubhi Garg, Anand Kumar Singh and Vivek Jain"Brain Tumor Detection using Image Processing" *International Journal of Information Sciences and Application (IJISA). ISSN 0974-2255, Vol.11, No.1, 2019, (Special Issue)*

[11] Ramin Ranjbarzadeh, Abbas Bagherian Kasgari, Saeid JafarzadehGhoushchi, ShokofehAnari, Maryam Naseri & Malika Bendechache"Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images" *Sci Rep. 2021 May 25;11(1):10930. doi: 10.1038/s41598-021-90428-8. PMID: 34035406; PMCID: PMC8149837.*