Detection of Brain Tumor using MRI images

A Project report

Submitted By

Damarasing Abhishek	**318126511013**
Kapuganti Sowmya	**318126511026**
Pentakota Chaitanya	**318126511040**
Peyyala Mounika	**318126511041**

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN
INFORMATION TECHNOLOGY

PRAGNANAM BRAHMA
ANITS

Under the Esteemed Guidance of

Mr Pandit Samuel G
(M.Tech)
(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)

(Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist.(A.P)

2021-22

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)
(Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist.(A.P)

CERTIFICATE

This is to certify that the project report entitled **"Detection of brain tumor using MRI images"** submitted by **D. Abhishek (318126511013), K. Sowmya (318126511026), P. Chaitanya (318126511040), P. Mounika (318126511040),** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** of Anil Neerukonda Institute of technology and sciences, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide	**Head of the Department**
Pandit Samuel	**Dr. Poosapati Padmaja**
Assistant Professor	Professor
Department of IT	Department of IT
ANITS	ANITS

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Pandit Samuel** Assistant Professor, Department of Information Technology, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. Poosapati Padmaja**, Head of the Department, of Information Technology, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of the Department of IT, whose suggestions during reviews helped us in the accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of IT, ANITS for providing great assistance in the accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. Last but not that, we thank everyone for supporting us directly or indirectly in completing this project successfully.

Damarasing Abhishek	**318126511013**
Kapuganti Sowmya	**318126511026**
Pentakota Chaitanya	**318126511040**
Peyyala Mounika	**318126511041**

DECLARATION

We hereby declare that the project work entitled **“Detection of brain tumor using MRI images”** submitted to the Anil Neerukonda Institute of Technology and Sciences is a record of an original work done by **D. Abhishek (318126511013), K. Sowmya (318126511026), P. Chaitanya (318126511040), P. Mounika (318126511040),** under the esteemed guidance of **G Pandit samuel,** Assistant professor of Information Technology, Anil Neerukonda Institute of Technology and Sciences, and this project work is submitted in the partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in **Information Technology.** This entire project is done with the best of our knowledge and have not been submitted for the award of any other degree in any other universities.

Damarasing Abhishek	**318126511013**
Kapuganti Sowmya	**318126511026**
Pentakota Chaitanya	**318126511040**
Peyyala Mounika	**318126511041**

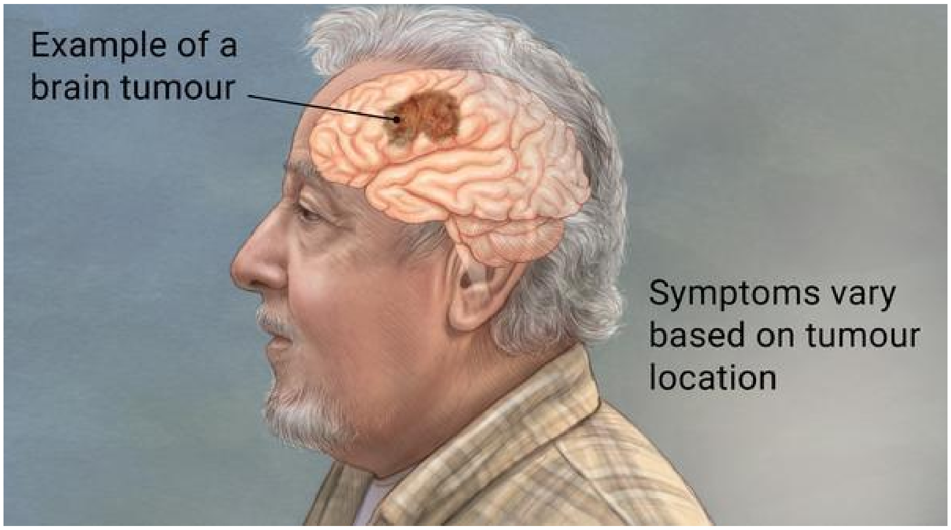
Place: Visakhapatnam
Date:

INDEX

ABSTRACT

A Brain Tumor or intracranial neoplasm happens when strange cells structure inside the mind. The side effects might incorporate cerebral pain, regurgitating, issues with vision and mental changes. More unambiguous issues might remember trouble for strolling, talking and sensation. As the sickness advances obviousness might happen. The reason for cerebrum growth isn't known. An illustration of cerebrum growth is displayed in the figure underneath.

We proposed an automated strategy for distinguishing proof of cerebrum growth utilizing the Brain Organization. The proposed model had acquired an exactness of 95% above and yields promising outcomes with practically no mistakes and significantly less computational time.



LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

S.no	Figure Name and Number	Page No
1.	MRI- MAGNETIC RESONANCE IMAGING	10
2.	GNN- GRAPH NEURAL NETWORKS	10
3.	CPU-CENTRAL PROCESSING UNIT GPU- GRAPHICS PROCESSING UNIT	15

CHAPTER-1

INTRODUCTION

Clinical pictures are one among the first significant assets used by specialists to analyze cerebrum cancers. An instrument with high precision to robotize this interaction might be very important. In any case, because of issues related to legitimate liabilities, such an instrument can't supplant the well-qualified assessments of prepared doctors. the fluctuated types of clinical imaging advances upheld painless methodologies like; MRI, CT check, Ultrasound, SPECT, PET, and X-beam. when contrasted with other clinical imaging procedures, reverberation Imaging (MRI) is significantly utilized and gives more noteworthy difference pictures to dangerous tissues. Hence, cerebrum cancer recognizable proof is much of the time done through MRI pictures. We present an AI way to deal with recognizing regardless of whether a MRI picture of a mind contains cancer. the resulting article discusses the indistinguishable.

1.1 Problem Statement and Motivation

A tumor could be an illness whose curability is basically snared into its initial recognition. the sooner the growth is recognized the adequate treatment might be selected and furthermore the more noteworthy the chance of endurance. during a comparative case, after the treatment, this can be generally a possibility for the growth to grow once more, and regardless early location is a vital stage to handle the matter. Specialists should address numerous patients and patients are expected to plan meetings with a specialist, now and again this is frequently not extremely effective. A quicker more precise choice to survey the MRI is expected to manage this kind of issue.

1.2 Research Objectives

We propose to make a web-based application for the discovery of growths inside the mind by dissecting MRI outputs of patients. the response is significantly pointed toward resolving the issues referenced in Section 1.1. Making the strategy more computerized and quicker. The proposed arrangement utilizes GNN calculation to distinguish highlights in MRI outputs of the patient. The model is prepared with more than 3000 MRI outputs of patients both with and without growths.

1.3 Project Scope and Direction

These days the necessity for cutting-edge clinical guides was expanded. Individuals have become more dependent on Doctors and clinical gear was expanding step by step. it's turned into a huge issue nowadays to require care surprisingly with restricted assets, as a number of specialists are extremely less.

1.4 Impact, Significance, and Contribution

The accessibility of specialists to a standard man makes the biggest difference here. Fewer specialists and emergency clinics are accessible for Neurology. neoplasm is one everyone of the premier and most hazardous infections its treatment costs are extremely high. Our Systems are helpful for individuals to distinguish the growth at its beginning phase previously or after the medical procedure. Likewise, it's essentially useful for clinical understudies to audit growth results without speaking with specialists.

1.5 Background Information

1.5.1 Project Field

This Project is typically dependent on the clinical field. X-ray is that the essential perspective for quite some time resulting for closing Brain and a lot of different sicknesses. In clinical terms, MRI might be a clinical imaging method that utilizes a field of power and PC-produced radio waves to frame point-by-point pictures of the organs and tissues in your body. Most MRI machines are enormous, tube-formed magnets.

1.5.2 Historical development prior to the project

There is now a created on mind growth location named cancer discovery using MRI pictures in that they used portrayal estimation. we are including GNN modules in our system which is more exact than this structure.

CHAPTER-2

LITERATURE SURVEY

2.1 Convolutional neural networks for brain tumor segmentation

The presentation of quantitative picture investigation has brought about fields like radionics which are acclimated to anticipate clinical sequelae. One developing area of interest for investigation is cerebrum growths, explicitly glioblastoma multiforme (GBM). Cancer division is a crucial stage in the pipeline of the investigation of this pathology. The manual division is normally conflicting on the grounds that it changes between eyewitnesses. The mechanized division has been proposed to battle this issue. Procedures like convolutional brain organizations (CNNs) which are AI pipelines demonstrated on the natural course of neurons (called hubs) and significant neurotransmitters (associations) are of interest in writing. We explore the job of CNNs to fragment mind cancers by first and foremost taking a scholarly analyze CNNs and playing out a writing search to see a model pipeline for division. We then research the more extended thermalization of CNNs by investigating a remarkable field of radionics. This inspects quantitative elements of cerebrum growths like shape, surface, and significant force to anticipate clinical results like endurance and reaction to treatment. With the introduction of methods to quantitatively analyse gliomas with computational methods comes a new frontier for radiology. It is important for radiologists to be abreast of advances in machine learning. This has been recognised by the recent changes in the Royal Australian and New Zealand College of Radiologists (RANZCR) curriculum that incorporates machine learning into the part I applied imaging technology examinations. Methods that incorporate quantitative analyses will add to the traditional visual analysis of images. An important step in the image analysis pipeline is the anatomical segmentation of regions of interest (ROI), for example, defining a volume of abnormal tissue from a background of normal tissue. This will allow for statistical analysis of features that is not visible by human perception. For example, the field of radiomics is fast developing as a method of predicting survival times from imaging features such as shape of a volume of interest and texture and intensity of the voxel habitat. With the development of these methods comes a greater need for automated segmentation. inconsistencies in blinded manual segmentation of brain tumours by the first and second authors. A measure of the consistency of image segmentation can be performed by the Sørensen–Dice coefficient, and this was calculated with the StudierFenster calculator (available at: http://studierfenster.tugraz.at/). This ranges from 0 to 1 with 1 having 100% consistency. The value obtained from the segmentation by the first and secauthorsthor was 0.91 which demonstrates the discrepancy in manual segmentation. As an example of machine learning, this educational paper will examine the use of convolutional neural networks for low-grade diffuse astrocytoma (World Health

Organization grade 2) and high-grade (World Health Organization grade 4) glioblastoma also known as glioblastoma multiforme (GBM) segmentation. Convolutional neural networks (CNNs) are a unique machine learning structure originally modelled on the human visual cortex. The brain was studied due to the abundance of segmentation methods that are already available and well established in the literature. Machine learning is fast developing and is exponentially being represented at international conferences. An educational perspective is needed for radiologists. This paper provides a novel balance between education and a state-of-the-art review on convolutional neural networks in glioblastoma.

2.2 MRI-Based Brain Tumor Classification Using Ensemble of Deep Features and Machine Learning Classifiers

There are elective ways of treating mind cancers relies upon the growth area, size, and type. As of now, the principal normal treatment for mind growth is a medical procedure since it significantly affects the cerebrum. varying sorts of clinical imaging advancements like modernized tomography (CT), positron discharge tomography (PET), and reverberation imaging (MRI) are accessible that are acclimated to notice the inward pieces of the structural conditions. Among these imaging modalities, MRI is considered the most ideal since it is the just painless and non-ionizing methodology that gives significant data in 2D and 3D organizations about cancer type, size, shape, and position. Notwithstanding, physically investigating these pictures is tedious, feverish, and, surprisingly, obligated to mistakes on account of the deluge of patients. to manage this issue, the occasion of a programmed PC supported conclusion (CAD) framework is expected to ease the responsibility of the order and determination of cerebrum MRI and go about as an instrument for aiding radiologists and specialists. Several efforts have been made to develop a highly accurate and robust solution for the automatic classification of brain tumors. However, due to high inter and intra shape, texture, and contrast variations, it remains a challenging problem. The traditional machine learning (ML) techniques rely on handcrafted features, which restrains the robustness of the solution. Whereas the deep learning-based techniques automatically extract meaningful features which offer significantly better performance. However, deep learning-based techniques require a large amount of annotated data for training, and acquiring such data is a challenging task. To overcome these issues, in this study, we proposed a hybrid solution that exploits (1) various pre-trained deep convolutional neural networks (CNNs) as feature extractors to extract powerful and discriminative deep features from brain magnetic resonance (MR) images, and (2) various ML classifiers to identify the normal and abnormal brain MR images. Also, to investigate the benefits of combining features from different pre-trained CNN models, we designed the novel feature ensemble method for the MRI-based brain tumor classification task. We proposed the novel feature evaluation and selection mechanism where the deep features from 13 different pre-trained CNNs are evaluated using 9 different ML classifiers and selected based on our proposed feature selection criteria. In our proposed framework, we concatenated the selected top three deep features from three different CNNs to form a synthetic feature. The concatenation process integrates the information from different CNNs to create a more discriminative feature representation than using the feature extracted from a single CNN model since different CNN architectures can capture diverse information in brain MR images. An ensemble of deep features is then fed into several ML classifiers to predict the final output, whereas most of the previous works have employed traditional feature extraction techniques. In our experiment, we provided an extensive evaluation using 13 different pre-trained deep convolutional neural networks and 9 different ML classifiers on three different datasets: (1) BT-small-2c, the small dataset with 2 classes (normal/tumor), (2) BT-large-2c, the large dataset with 2 classes (normal/tumor), and (3) the large dataset with 4 classes (normal, glioma tumor, meningioma tumor, and pituitary tumor) for brain tumor classification. The traditional ML methods are comprised of several steps: pre-processing, feature extraction, feature reduction, and classification. In traditional ML methods, feature extraction is a core step as the classification accuracy relies on extracted features. There are two main types of feature extraction. The first type of feature extraction is low-level (global) features, for instance, texture features and intensity, first-order statistics (e.g., mean, standard deviation, and skewness), and second-order statistics such as gray-level co-occurrence matrix (GLCM), wavelet transform (WT), Gabor feature, and shape.

2.3 Brain Tumor Classification Using Convolutional Neural Networks

Brain tumors, are the chief normal and forceful illness, bringing about an outrageously short life expectancy in their most noteworthy grade. Hence, treatment arranging could be a critical stage to support the norm of a lifetime of patients. By and large, different picture procedures like X-raying (CT), reverberation Imaging (MRI), and ultrasound pictures are wont to assess the growth in a very mind, lung, liver, bosom, prostate… and so on. Particularly, during this wor,k MRI pictures are wont to analyze cancer inside the cerebrum. Anyway, the monstrous measure of data created by the MRI filter defeats the manual grouping of cancer versus non-growth during a specific time. Yethasaving some limit (i.e) precise quantitative estimations are accommodated set number of pictures. Thus trusted and programmed order plots are fundamental to hinder the passing pace of a human. the robotized cerebrum cancer arrangement is an amazingly difficult assignment in enormous spatial and underlying inconstancy of encompassing locale of growth. during this work, programmed cerebrum growth identification is proposed by utilizing Convolutional Neural Networks (CNN) arrangement. The more deeply engineering configuration is performed by utilizing little pieces. the weight of the neuron is given as little. Exploratory outcomes show that the CNN exactness with low intricacy and contrasted and any remaining condition of expressions strategies.

CHAPTER-3

SYSTEM ANALYSIS

3.1 Software Requirement Specifications

Flask :

Flask could be a miniature web structure written in Python. it's named a microframework on the grounds that it doesn't need specific devices or libraries. it's no data set deliberation layer, structure approval, or different parts where previous outsider libraries give normal capacities. In any case, Flask upholds augmentations which will add application highlights as though they were executed in Flask itself. Expansions exist for object-social mappers, structure approval, transfer taking care of, different open validation advancements, and a few other normal system-related apparatuses.

PyTorch using Deep Graph Library:

Deep Graph Library (DGL) could be a Python bundle that worked for straightforward execution of diagram brain network model family, on top of existing DL systems (as of now supporting PyTorch, MXNet, and TensorFlow). It offers an adaptable control of message passing, speed enhancement through auto-clustering and exceptionally tuned meager network bits, and multi-GPU/CPU preparing to scale to charts of numerous army hubs and edges. Diagram level portrayals are basic in a very kind of true applications like foreseeing the properties of atoms and local area examination in informal organizations. Conventional diagram piece-based techniques are straightforward, yet viable for getting fixed-length portrayals for charts yet they experience the ill effects of unfortunate speculation due to hand-made plans. There are a few ongoing techniques that upheld language models (for example graph2vec) yet they tend to just think about specific foundations (for example subtrees) as chart delegates.

Programming Language:

Here we are utilizing Python 3.9.0 which can be supporting every one of the libraries we use i.e PyTorch, Flask, Deep diagram. A few highlights that assistance in Python 3.9 are:

PEP 573, Module State Access from C Extension Methods
PEP 584, Union Operators in dict
PEP 585, Type Hinting Generics In Standard Collections
PEP 593, Flexible capacity and variable comments
PEP 602, Python embraces a steady yearly delivery rhythm.
PEP 614, Relaxing Grammar Restrictions On Decorators
PEP 615, Support for the IANA zone Database inside the Standard Library
PEP 616, String strategies to dispose of prefixes and additions
PEP 617, New PEG parser for CPython.
BPO 38379, garbage pickup doesn't impede on revived objects.
BPO 38692, os.pidfd_open added that empowers cycle the board without races and signals;
BPO 39926, Unicode support refreshed to form 13.0.0;
BPO 1635741, when Python is introduced on numerous occasions inside a similar cycle, it doesn't spill memory any longer.

Various Python builtins (range, tuple, set, frozenset, list, dict) are presently accelerated utilizing PEP 590 vectorcall.

3.2 System Requirements

	Windows requirements	Mac requirements	Linux requirements
Operating system	Windows 8 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Screen resolution	1280 * 1024 or larger		
Application window size	1024 * 680 or larger		
Internet connection	Required		

CHAPTER-4

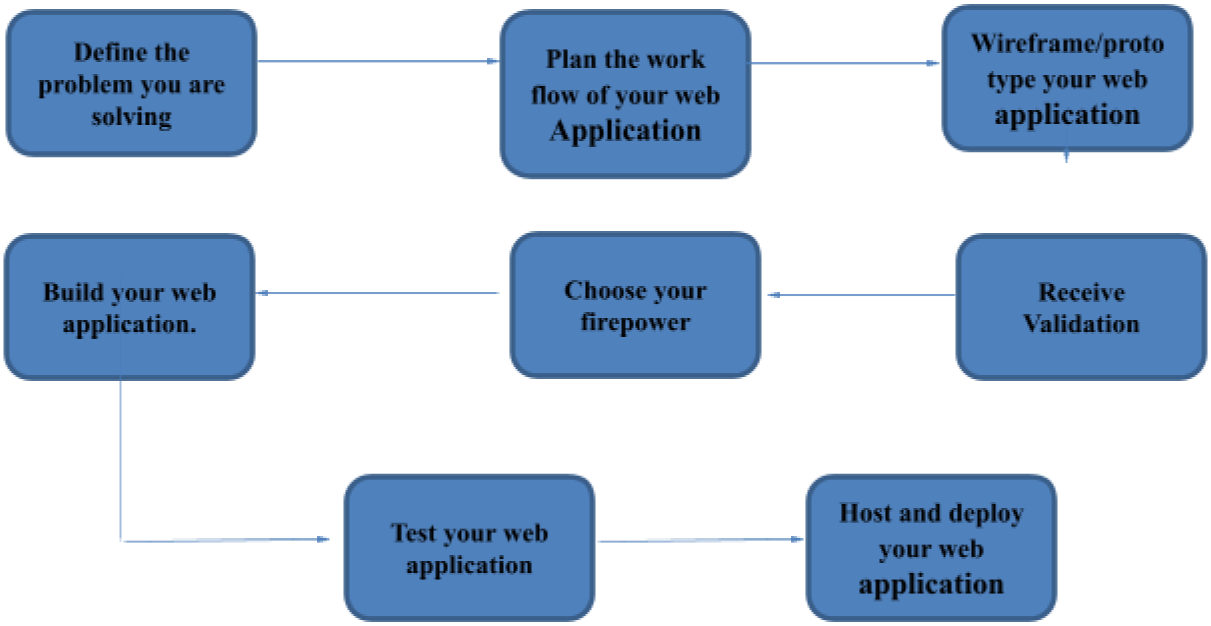
Software Development

There are two significant framework streams inside the product improvement area as displayed underneath:

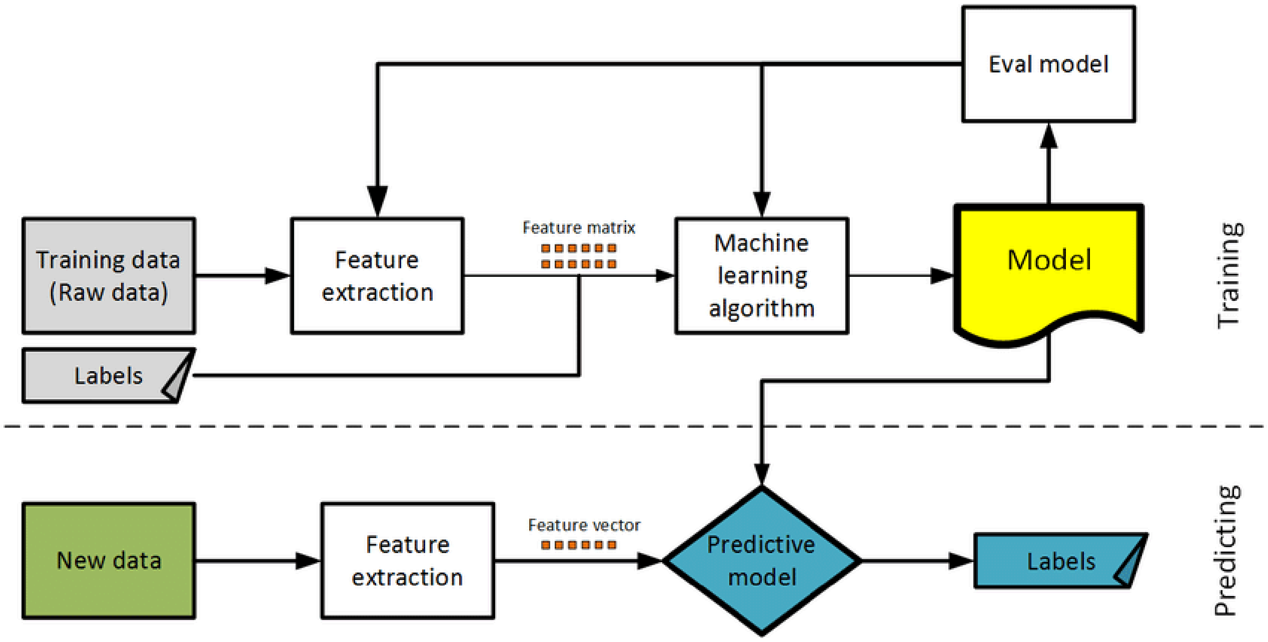
- The formation of the web-based Application.
- The most common way of making a Trained Model.

The two cycles referenced above are fundamental since they made up the foundation of the MRI cerebrum cancer Detection. during this part, the strategy for the two streams is momentarily depicted. In the interim, their full usefulness, explicit prerequisites, and furthermore the techniques/way to deal with achieving such targets are talked about in the forthcoming part.

4.1 The creation of the Web Application: fig- 4.1



4.2 Creating Model: fig 4.2



CHAPTER-5

Methodology

In our project, we use Graph Neural Networks. A Graph Neural Network (GNN) might be a Deep Learning calculation that could absorb an information picture, dole out significance (learnable loads and predispositions) to various perspectives/objects inside the picture, and have the option to separate one from the inverse.

A diagram brain organization (GNN) could be a class of brain networks for handling information addressed by chart information structures. They were promoted by their utilization in administered learning on properties of changed particles.

Since their commencement, a few variations of the simple message passing brain organization (MPNN) system are proposed. These models improve GNNs to be utilized on bigger charts and apply them to spaces like interpersonal organizations, reference organizations, and online networks.

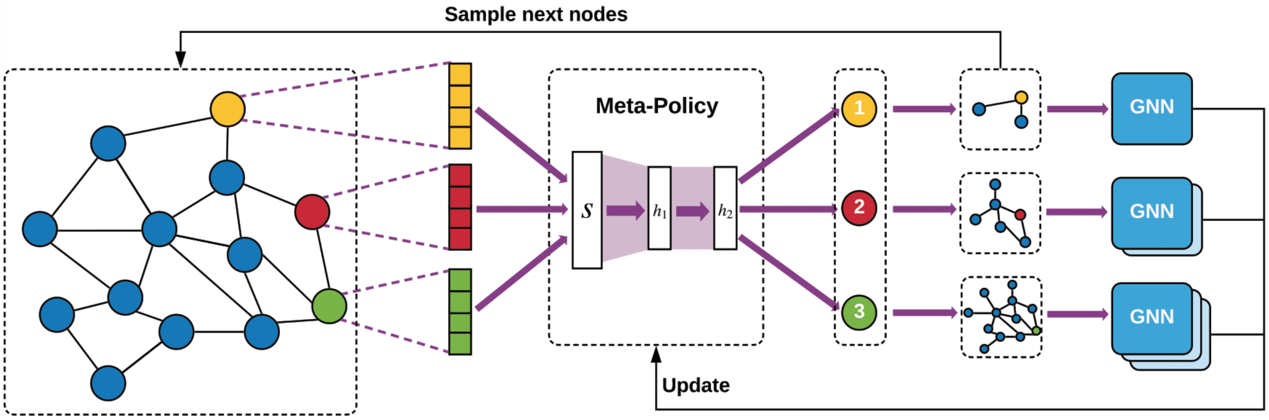


Fig- 5.1

In our framework, we train the model with and without cancer MRI checks, all together that our model will recognize regardless of whether our MRI filter duplicate we transfer has growth. you'll have the option to see well beneath graphs.

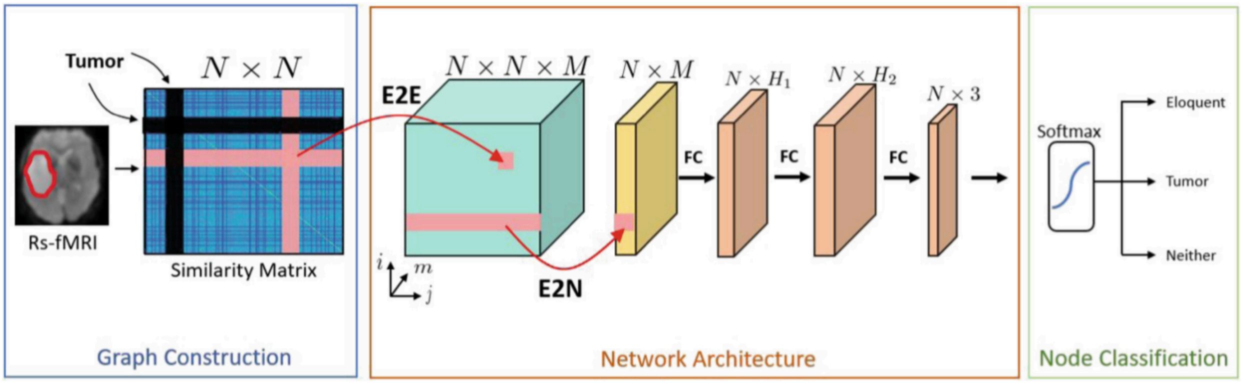


Fig- 5.2

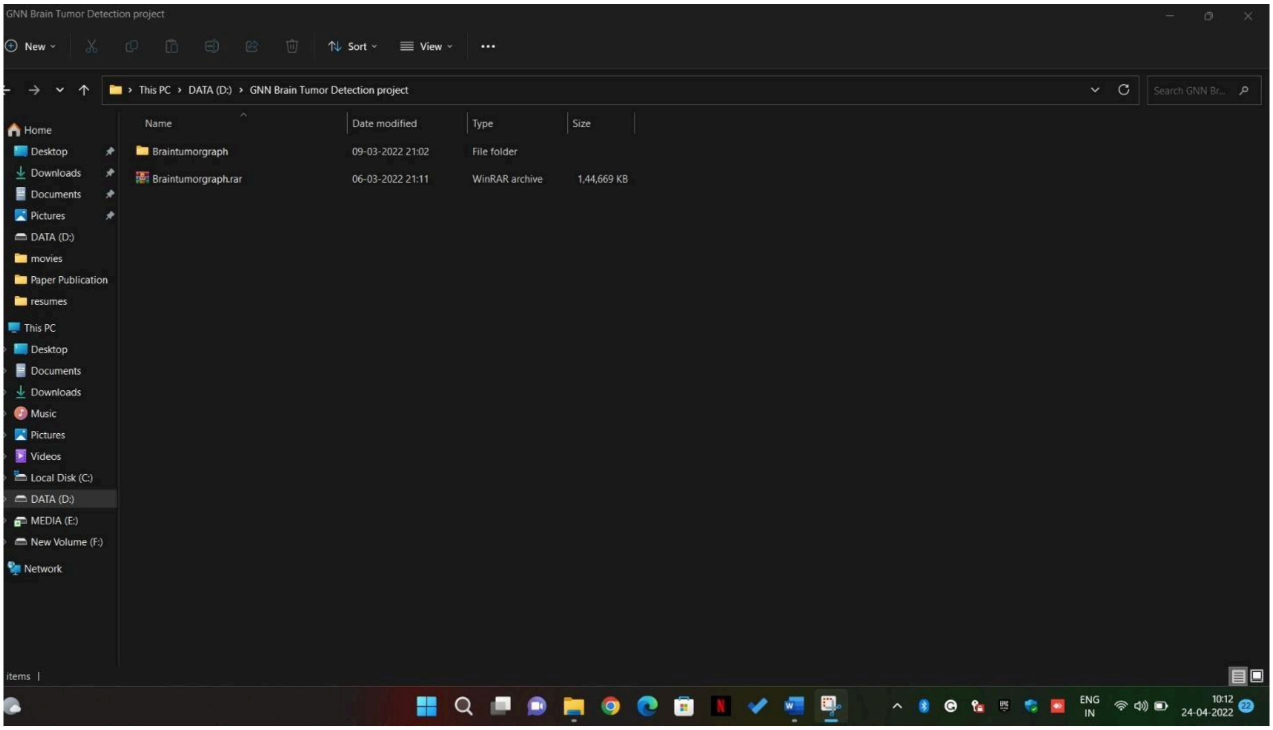
By the above picture, we will perceive the way the model will works, first the picture that is transferred will be seen in a N x N Similarity Matrix and it'll be set apart in the dark part. Furthermore, it'll be additionally examined profoundly utilizing a N x N x M framework utilizing E2E and E2N (Three aspects) and it'll be additionally dissected utilizing various Matrix. What's more, the last result will become about.

Before preparing, every patient example is changed over to a chart. We split the dataset into preparing (60%), approval (20%), and test sets (20%). The contribution to the GNN is characterized officially as a chart of the shape, and an element grid $H \in R$ n×f, where n is the number of hubs and f is the number of highlights per hub. $f = 20$ for all examinations. The result is of size n × c, where for each diagram hub, the model returns the likelihood of that hub having a place with everything about four classes (c) characterized to see the most straightforward hyperparameters for everything about GNN variations, we play out an irregular hyperparameter search on the approval set. We enkindle routinely dispersed time periods rate from 0.00001 to 0.001, including dropout somewhere in the range of 0 and 0.5, model profundity from 3 to 9, and secret layer size somewhere in the range of 64 and 256. For GAT models, we moreover analyze consideration dropout somewhere in the range of 0 and 0.5 and a spotlight heads somewhere in the range of 3 and 10 for each layer. Each model is prepared to constrict hub-wise multi-name cross-entropy misfortune on the approval set utilizing the Adam enhancer. the classification loads are acclimated to be contrarily corresponding to their pervasiveness inside the test set to manage the classification awkwardness.

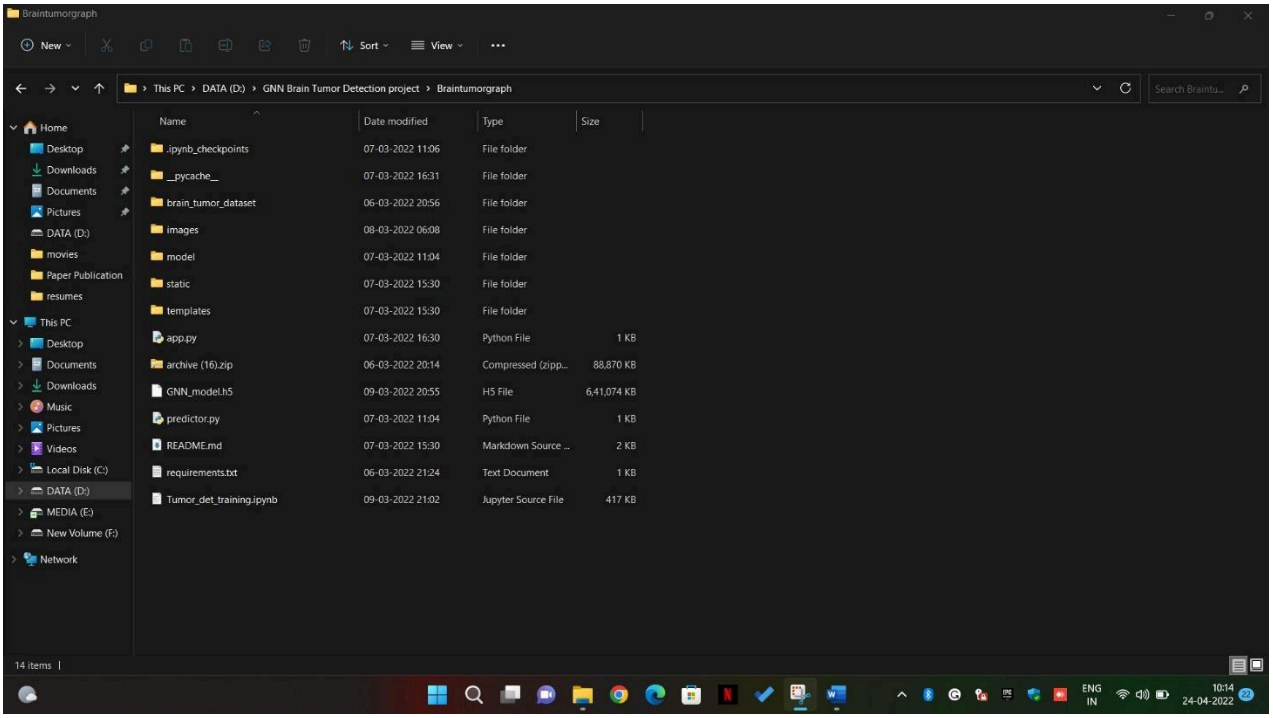
CHAPTER-6

Coding

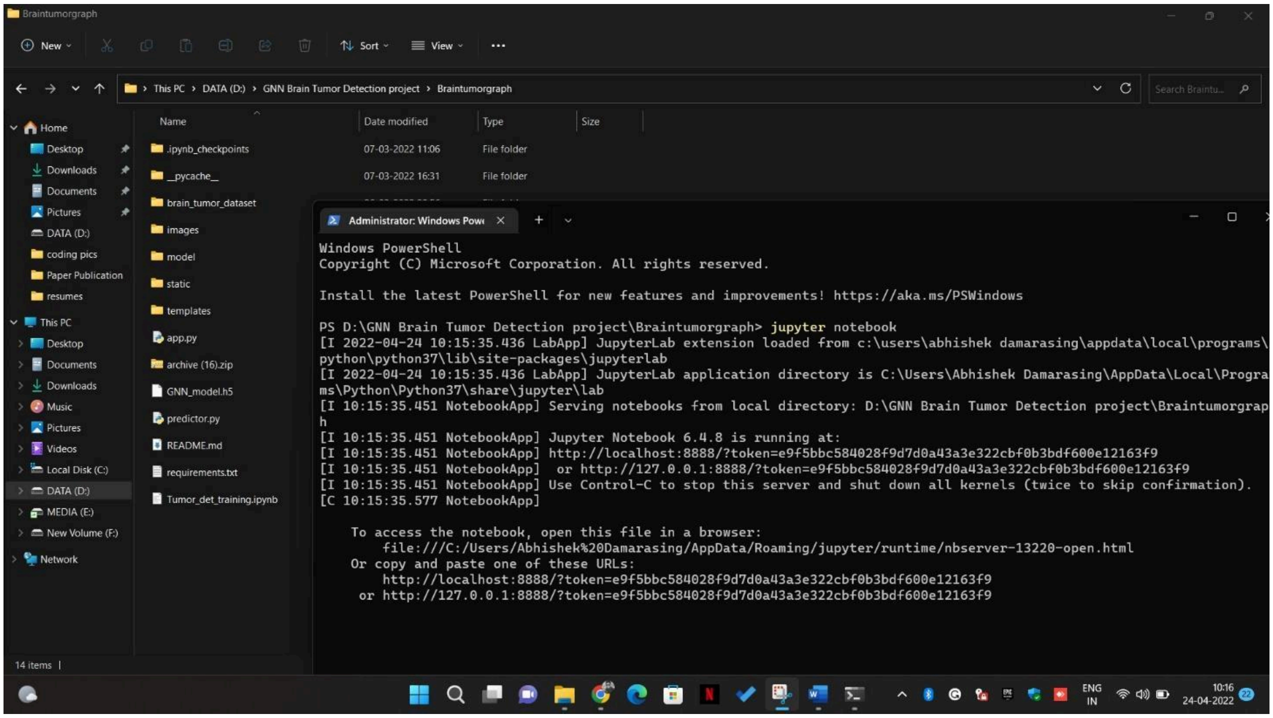
6.1 Tumor_det_training.ipynb:



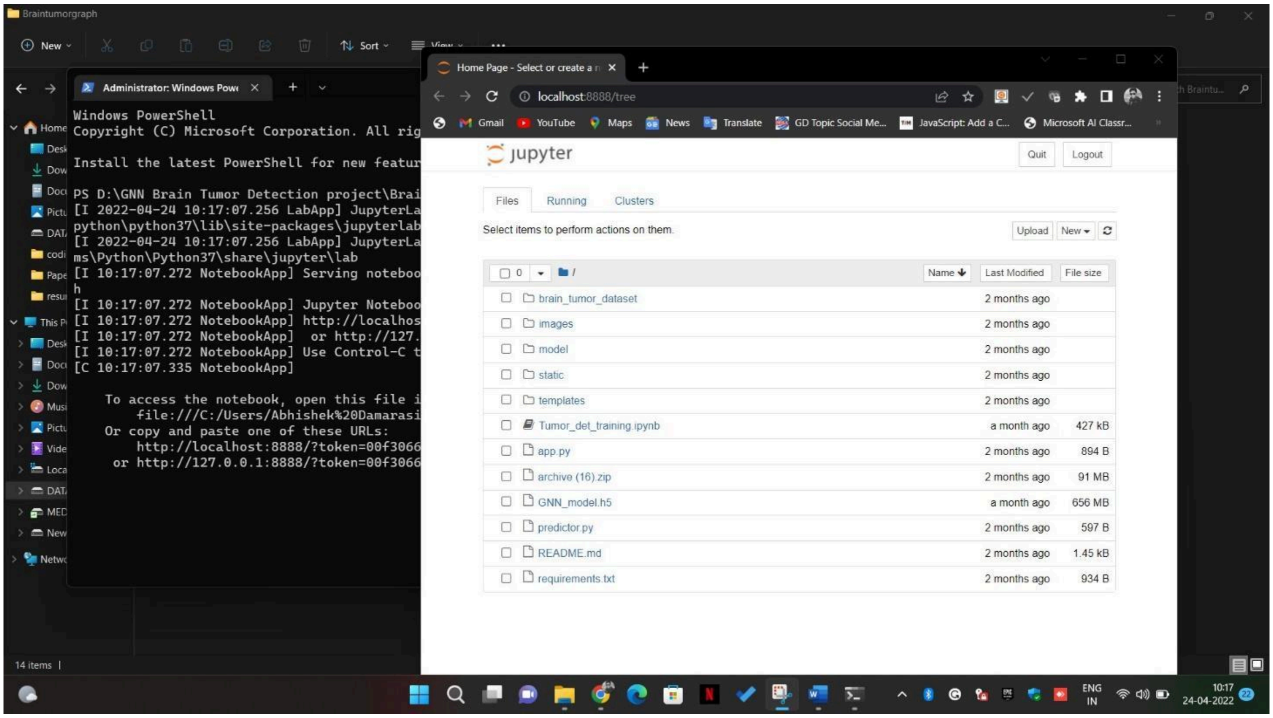
Name	Date modified	Type	Size
Braintumorgraph	09-03-2022 21:02	File folder	
Braintumorgraph.rar	06-03-2022 21:11	WinRAR archive	1,44,669 KB



Name	Date modified	Type	Size
.ipynb_checkpoints	07-03-2022 11:06	File folder	
__pycache__	07-03-2022 16:31	File folder	
brain_tumor_dataset	06-03-2022 20:56	File folder	
images	08-03-2022 06:08	File folder	
model	07-03-2022 11:04	File folder	
static	07-03-2022 15:30	File folder	
templates	07-03-2022 15:30	File folder	
app.py	07-03-2022 16:30	Python File	1 KB
archive (16).zip	06-03-2022 20:14	Compressed (zipp...	88,870 KB
GNN_model.h5	09-03-2022 20:55	H5 File	6,41,074 KB
predictor.py	07-03-2022 11:04	Python File	1 KB
README.md	07-03-2022 15:30	Markdown Source ...	2 KB
requirements.txt	06-03-2022 21:24	Text Document	1 KB
Tumor_det_training.ipynb	09-03-2022 21:02	Jupyter Source File	417 KB



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\GNN Brain Tumor Detection project\Braintumorgraph> jupyter notebook
[I 2022-04-24 10:15:35.436 LabApp] JupyterLab extension loaded from c:\users\abhishek damarasing\appdata\local\programs\
python\python37\lib\site-packages\jupyterlab
[I 2022-04-24 10:15:35.436 LabApp] JupyterLab application directory is C:\Users\Abhishek Damarasing\AppData\Local\Progra
ms\Python\Python37\share\jupyter\lab
[I 10:15:35.451 NotebookApp] Serving notebooks from local directory: D:\GNN Brain Tumor Detection project\Braintumorgrap
h
[I 10:15:35.451 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 10:15:35.451 NotebookApp] http://localhost:8888/?token=e9f5bbc584028f9d7d0a43a3e322cbf0b3bdf600e12163f9
[I 10:15:35.451 NotebookApp]  or http://127.0.0.1:8888/?token=e9f5bbc584028f9d7d0a43a3e322cbf0b3bdf600e12163f9
[I 10:15:35.451 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 10:15:35.577 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Abhishek%20Damarasing/AppData/Roaming/jupyter/runtime/nbserver-13220-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=e9f5bbc584028f9d7d0a43a3e322cbf0b3bdf600e12163f9
     or http://127.0.0.1:8888/?token=e9f5bbc584028f9d7d0a43a3e322cbf0b3bdf600e12163f9
```

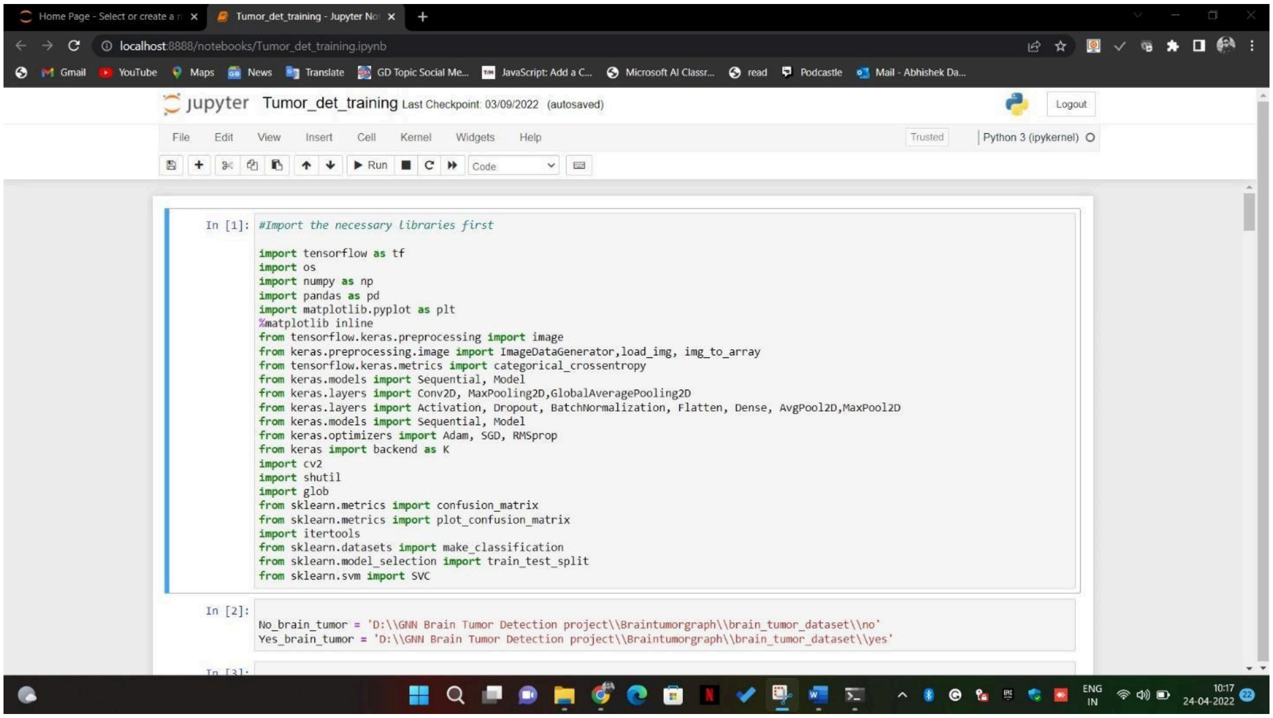


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rig

Install the latest PowerShell for new featur

PS D:\GNN Brain Tumor Detection project\Brai
[I 2022-04-24 10:17:07.256 LabApp] JupyterLa
python\python37\lib\site-packages\jupyterlab
[I 2022-04-24 10:17:07.256 LabApp] JupyterLa
ms\Python\Python37\share\jupyter\lab
[I 10:17:07.272 NotebookApp] Serving noteboo
h
[I 10:17:07.272 NotebookApp] Jupyter Noteboo
[I 10:17:07.272 NotebookApp] http://localhos
[I 10:17:07.272 NotebookApp]  or http://127.
[I 10:17:07.272 NotebookApp] Use Control-C t
[C 10:17:07.335 NotebookApp]

    To access the notebook, open this file i
        file:///C:/Users/Abhishek%20Damarasi
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=00f3066
     or http://127.0.0.1:8888/?token=00f3066
```

localhost:8888/tree

Select items to perform actions on them.

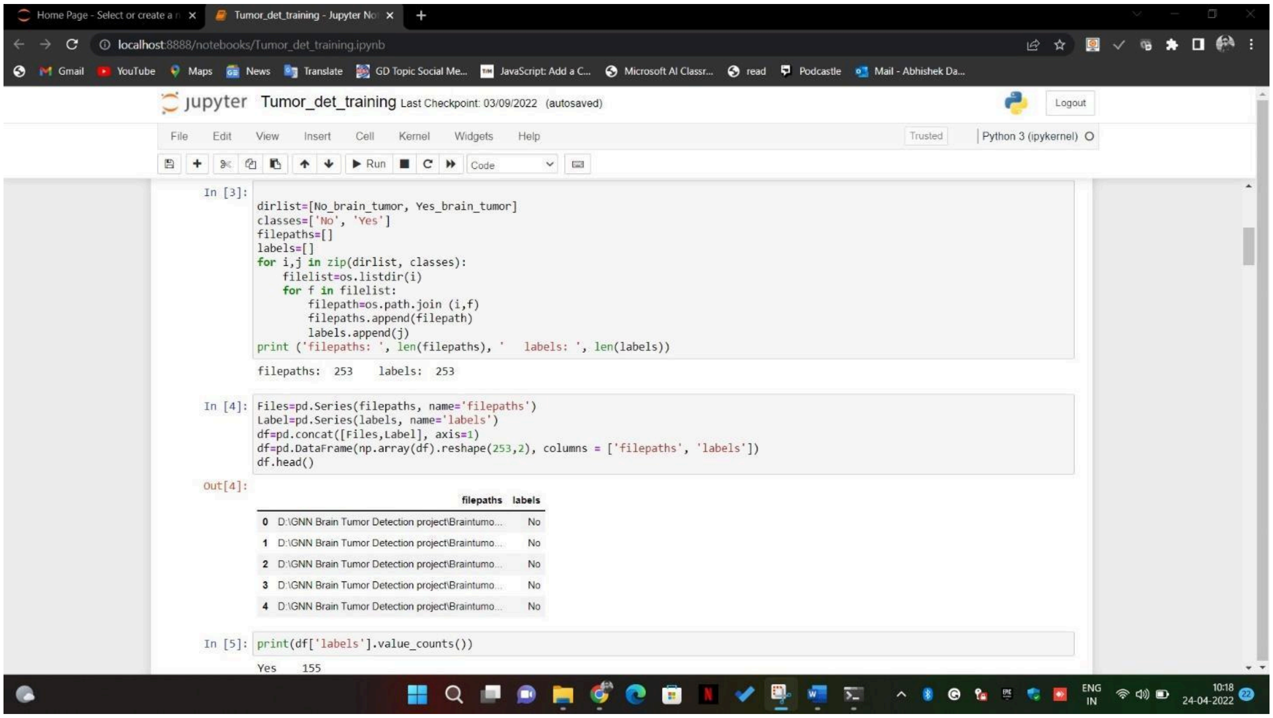
Name	Last Modified	File size
brain_tumor_dataset	2 months ago	
images	2 months ago	
model	2 months ago	
static	2 months ago	
templates	2 months ago	
Tumor_det_training.ipynb	a month ago	427 kB
app.py	2 months ago	894 B
archive (16).zip	2 months ago	91 MB
GNN_model.h5	a month ago	656 MB
predictor.py	2 months ago	597 B
README.md	2 months ago	1.45 kB
requirements.txt	2 months ago	934 B



localhost:8888/notebooks/Tumor_det_training.ipynb

Tumor_det_training Last Checkpoint: 03/09/2022 (autosaved)

```
In [1]: #Import the necessary libraries first

        import tensorflow as tf
        import os
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        from tensorflow.keras.preprocessing import image
        from keras.preprocessing.image import ImageDataGenerator,load_img, img_to_array
        from tensorflow.keras.metrics import categorical_crossentropy
        from keras.models import Sequential, Model
        from keras.layers import Conv2D, MaxPooling2D,GlobalAveragePooling2D
        from keras.layers import Activation, Dropout, BatchNormalization, Flatten, Dense, AvgPool2D,MaxPool2D
        from keras.models import Sequential, Model
        from keras.optimizers import Adam, SGD, RMSprop
        from keras import backend as K
        import cv2
        import shutil
        import glob
        from sklearn.metrics import confusion_matrix
        from sklearn.metrics import plot_confusion_matrix
        import itertools
        from sklearn.datasets import make_classification
        from sklearn.model_selection import train_test_split
        from sklearn.svm import SVC

In [2]:
        No_brain_tumor = 'D:\\GNN Brain Tumor Detection project\\Braintumorgraph\\brain_tumor_dataset\\no'
        Yes_brain_tumor = 'D:\\GNN Brain Tumor Detection project\\Braintumorgraph\\brain_tumor_dataset\\yes'
```



```
In [3]:
dirlist=[No_brain_tumor, Yes_brain_tumor]
classes=['No', 'Yes']
filepaths=[]
labels=[]
for i,j in zip(dirlist, classes):
    filelist=os.listdir(i)
    for f in filelist:
        filepath=os.path.join (i,f)
        filepaths.append(filepath)
        labels.append(j)
print ('filepaths: ', len(filepaths), '   labels: ', len(labels))
```

```
filepaths:  253    labels:  253
```

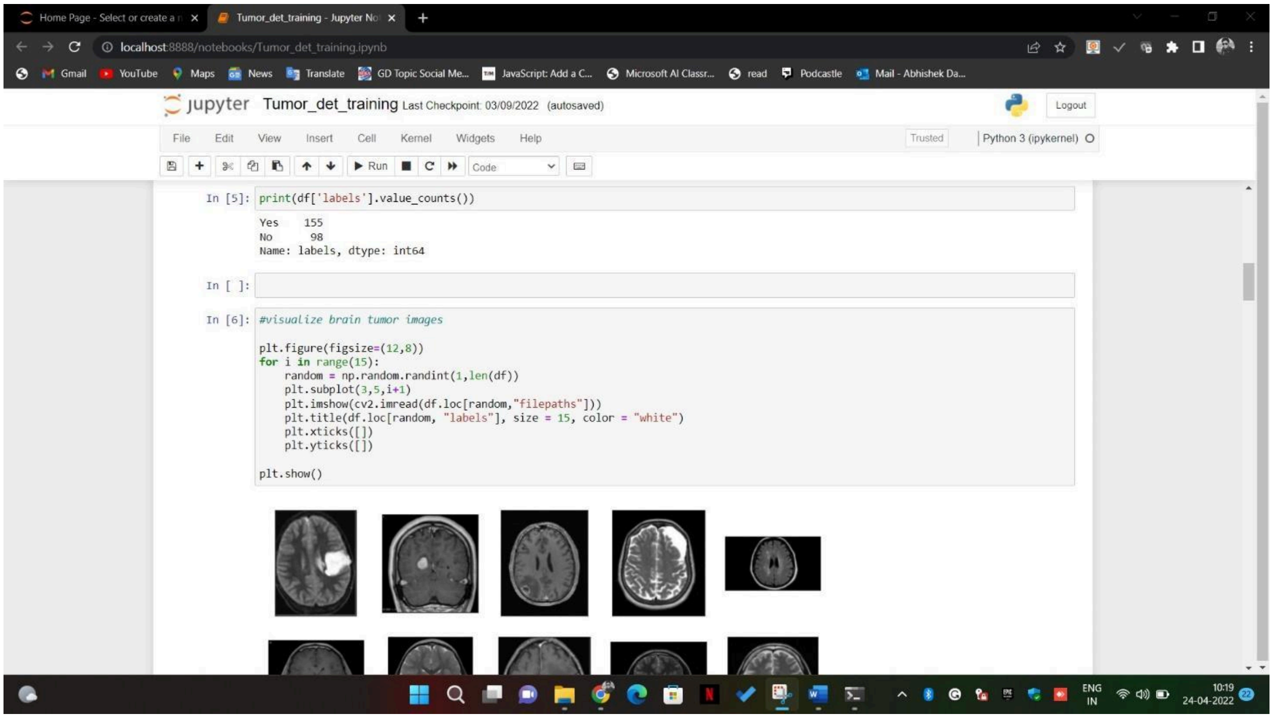
```
In [4]: Files=pd.Series(filepaths, name='filepaths')
Label=pd.Series(labels, name='labels')
df=pd.concat([Files,Label], axis=1)
df=pd.DataFrame(np.array(df).reshape(253,2), columns = ['filepaths', 'labels'])
df.head()
```

Out[4]:

	filepaths	labels
0	D:\GNN Brain Tumor Detection project\Braintumo...	No
1	D:\GNN Brain Tumor Detection project\Braintumo...	No
2	D:\GNN Brain Tumor Detection project\Braintumo...	No
3	D:\GNN Brain Tumor Detection project\Braintumo...	No
4	D:\GNN Brain Tumor Detection project\Braintumo...	No

```
In [5]: print(df['labels'].value_counts())
```

```
Yes     155
```

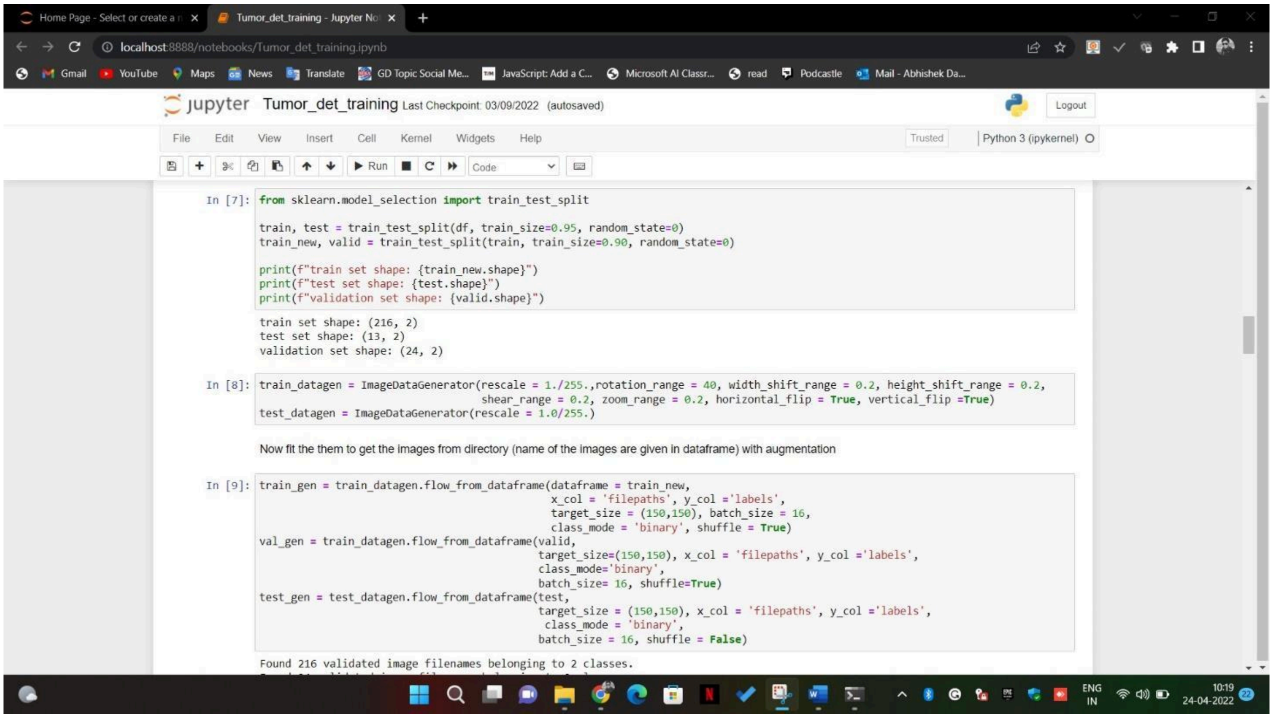


```
In [5]: print(df['labels'].value_counts())
```

```
Yes     155
No       98
Name: labels, dtype: int64
```

```
In [ ]:
```

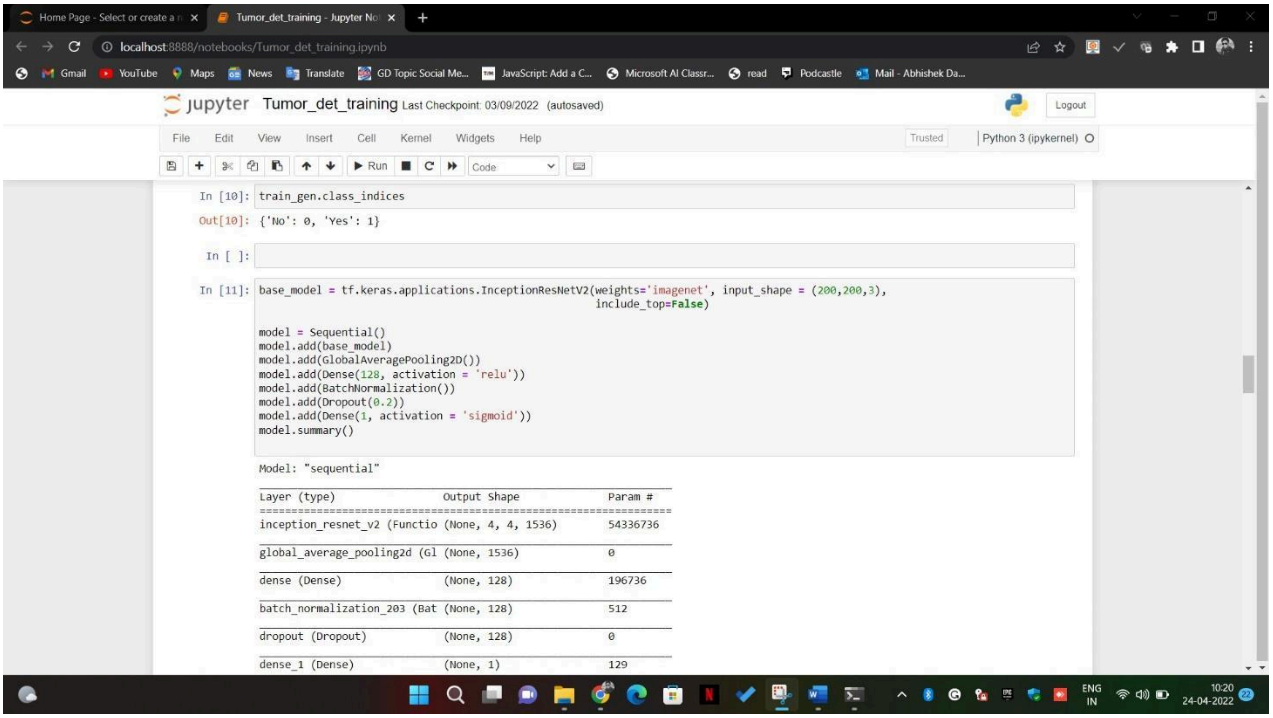
```
In [6]: #visualize brain tumor images

plt.figure(figsize=(12,8))
for i in range(15):
    random = np.random.randint(1,len(df))
    plt.subplot(3,5,i+1)
    plt.imshow(cv2.imread(df.loc[random,"filepaths"]))
    plt.title(df.loc[random, "labels"], size = 15, color = "white")
    plt.xticks([])
    plt.yticks([])

plt.show()
```



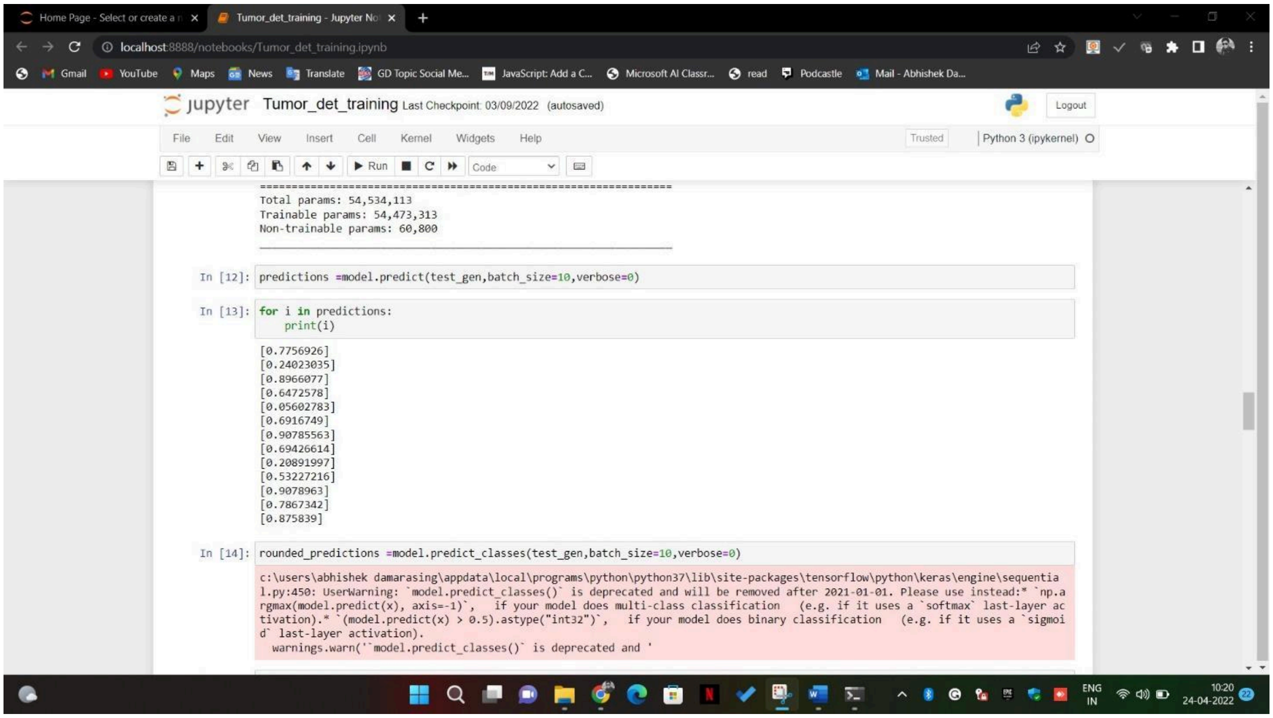
```
In [7]: from sklearn.model_selection import train_test_split

        train, test = train_test_split(df, train_size=0.95, random_state=0)
        train_new, valid = train_test_split(train, train_size=0.90, random_state=0)

        print(f"train set shape: {train_new.shape}")
        print(f"test set shape: {test.shape}")
        print(f"validation set shape: {valid.shape}")

        train set shape: (216, 2)
        test set shape: (13, 2)
        validation set shape: (24, 2)

In [8]: train_datagen = ImageDataGenerator(rescale = 1./255.,rotation_range = 40, width_shift_range = 0.2, height_shift_range = 0.2,
                                           shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True, vertical_flip =True)
        test_datagen = ImageDataGenerator(rescale = 1.0/255.)
```

Now fit the them to get the images from directory (name of the images are given in dataframe) with augmentation

```
In [9]: train_gen = train_datagen.flow_from_dataframe(dataframe = train_new,
                                                       x_col = 'filepaths', y_col ='labels',
                                                       target_size = (150,150), batch_size = 16,
                                                       class_mode = 'binary', shuffle = True)
        val_gen = train_datagen.flow_from_dataframe(valid,
                                                     target_size=(150,150), x_col = 'filepaths', y_col ='labels',
                                                     class_mode='binary',
                                                     batch_size= 16, shuffle=True)
        test_gen = test_datagen.flow_from_dataframe(test,
                                                     target_size = (150,150), x_col = 'filepaths', y_col ='labels',
                                                      class_mode = 'binary',
                                                     batch_size = 16, shuffle = False)

        Found 216 validated image filenames belonging to 2 classes.
```



```
In [10]: train_gen.class_indices

Out[10]: {'No': 0, 'Yes': 1}

In [ ]:

In [11]: base_model = tf.keras.applications.InceptionResNetV2(weights='imagenet', input_shape = (200,200,3),
                                                               include_top=False)

         model = Sequential()
         model.add(base_model)
         model.add(GlobalAveragePooling2D())
         model.add(Dense(128, activation = 'relu'))
         model.add(BatchNormalization())
         model.add(Dropout(0.2))
         model.add(Dense(1, activation = 'sigmoid'))
         model.summary()

         Model: "sequential"
         _________________________________________________________________
         Layer (type)                Output Shape              Param #
         =================================================================
         inception_resnet_v2 (Functio (None, 4, 4, 1536)       54336736
         _________________________________________________________________
         global_average_pooling2d (Gl (None, 1536)             0
         _________________________________________________________________
         dense (Dense)               (None, 128)               196736
         _________________________________________________________________
         batch_normalization_203 (Bat (None, 128)              512
         _________________________________________________________________
         dropout (Dropout)           (None, 128)               0
         _________________________________________________________________
         dense_1 (Dense)             (None, 1)                 129
```

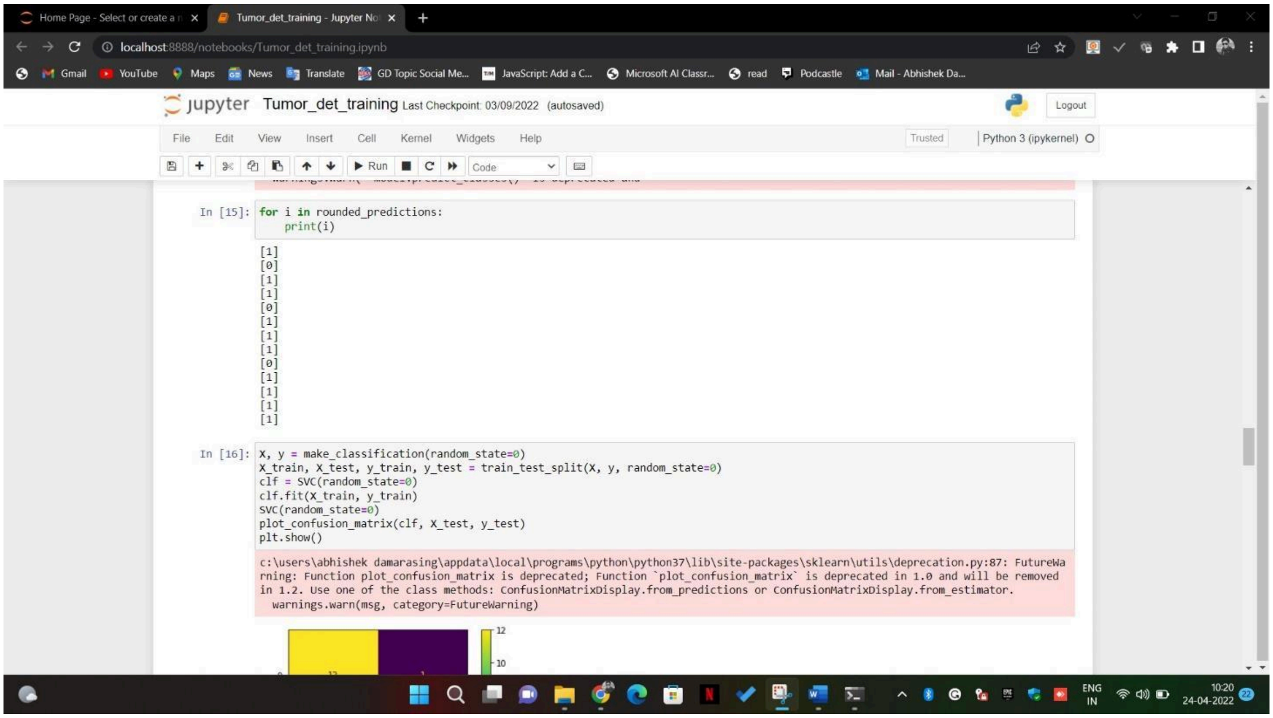


```
==================================================================
Total params: 54,534,113
Trainable params: 54,473,313
Non-trainable params: 60,800
__________________________________________________________________
```

```
In [12]: predictions =model.predict(test_gen,batch_size=10,verbose=0)
```

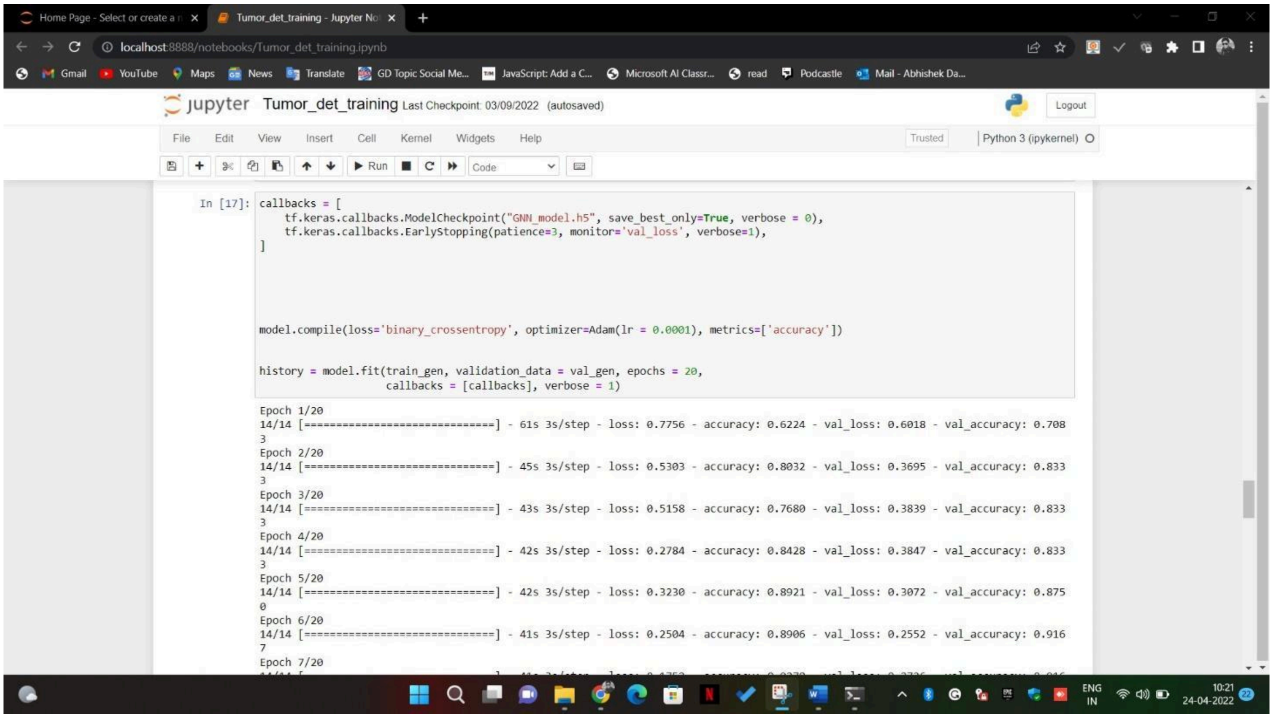
```
In [13]: for i in predictions:
             print(i)

[0.7756926]
[0.24023035]
[0.8966077]
[0.6472578]
[0.05602783]
[0.6916749]
[0.90785563]
[0.69426614]
[0.20891997]
[0.53227216]
[0.9078963]
[0.7867342]
[0.875839]
```

```
In [14]: rounded_predictions =model.predict_classes(test_gen,batch_size=10,verbose=0)

c:\users\abhishek damarasing\appdata\local\programs\python\python37\lib\site-packages\tensorflow\python\keras\engine\sequentia
l.py:450: UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.a
rgmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer ac
tivation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoi
d` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
```



```
In [15]: for i in rounded_predictions:
             print(i)

[1]
[0]
[1]
[1]
[0]
[1]
[1]
[1]
[0]
[1]
[1]
[1]
[1]
```

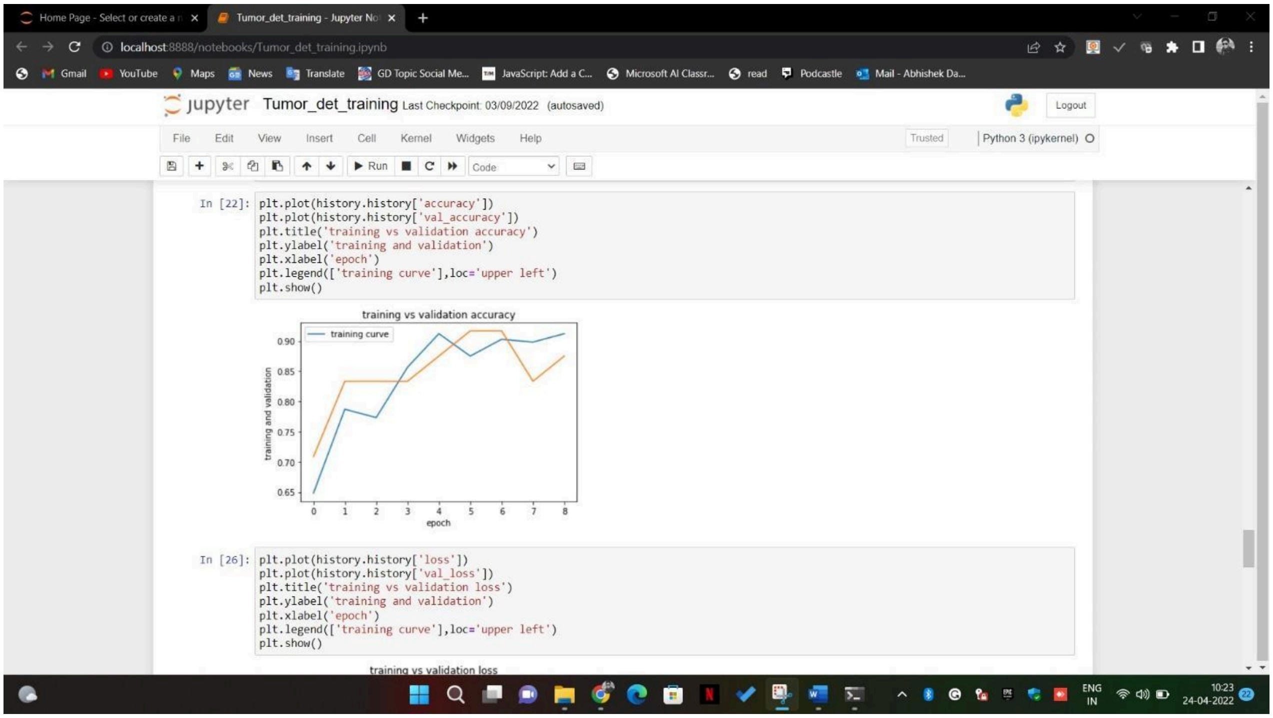
```
In [16]: X, y = make_classification(random_state=0)
         X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
         clf = SVC(random_state=0)
         clf.fit(X_train, y_train)
         SVC(random_state=0)
         plot_confusion_matrix(clf, X_test, y_test)
         plt.show()

c:\users\abhishek damarasing\appdata\local\programs\python\python37\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWa
rning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed
in 1.2. Use one of the class methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```



In [17]:

```
callbacks = [
    tf.keras.callbacks.ModelCheckpoint("GNN_model.h5", save_best_only=True, verbose = 0),
    tf.keras.callbacks.EarlyStopping(patience=3, monitor='val_loss', verbose=1),
]




model.compile(loss='binary_crossentropy', optimizer=Adam(lr = 0.0001), metrics=['accuracy'])


history = model.fit(train_gen, validation_data = val_gen, epochs = 20,
                    callbacks = [callbacks], verbose = 1)
```

```
Epoch 1/20
14/14 [==============================] - 61s 3s/step - loss: 0.7756 - accuracy: 0.6224 - val_loss: 0.6018 - val_accuracy: 0.708
3
Epoch 2/20
14/14 [==============================] - 45s 3s/step - loss: 0.5303 - accuracy: 0.8032 - val_loss: 0.3695 - val_accuracy: 0.833
3
Epoch 3/20
14/14 [==============================] - 43s 3s/step - loss: 0.5158 - accuracy: 0.7680 - val_loss: 0.3839 - val_accuracy: 0.833
3
Epoch 4/20
14/14 [==============================] - 42s 3s/step - loss: 0.2784 - accuracy: 0.8428 - val_loss: 0.3847 - val_accuracy: 0.833
3
Epoch 5/20
14/14 [==============================] - 42s 3s/step - loss: 0.3230 - accuracy: 0.8921 - val_loss: 0.3072 - val_accuracy: 0.875
0
Epoch 6/20
14/14 [==============================] - 41s 3s/step - loss: 0.2504 - accuracy: 0.8906 - val_loss: 0.2552 - val_accuracy: 0.916
7
Epoch 7/20
```

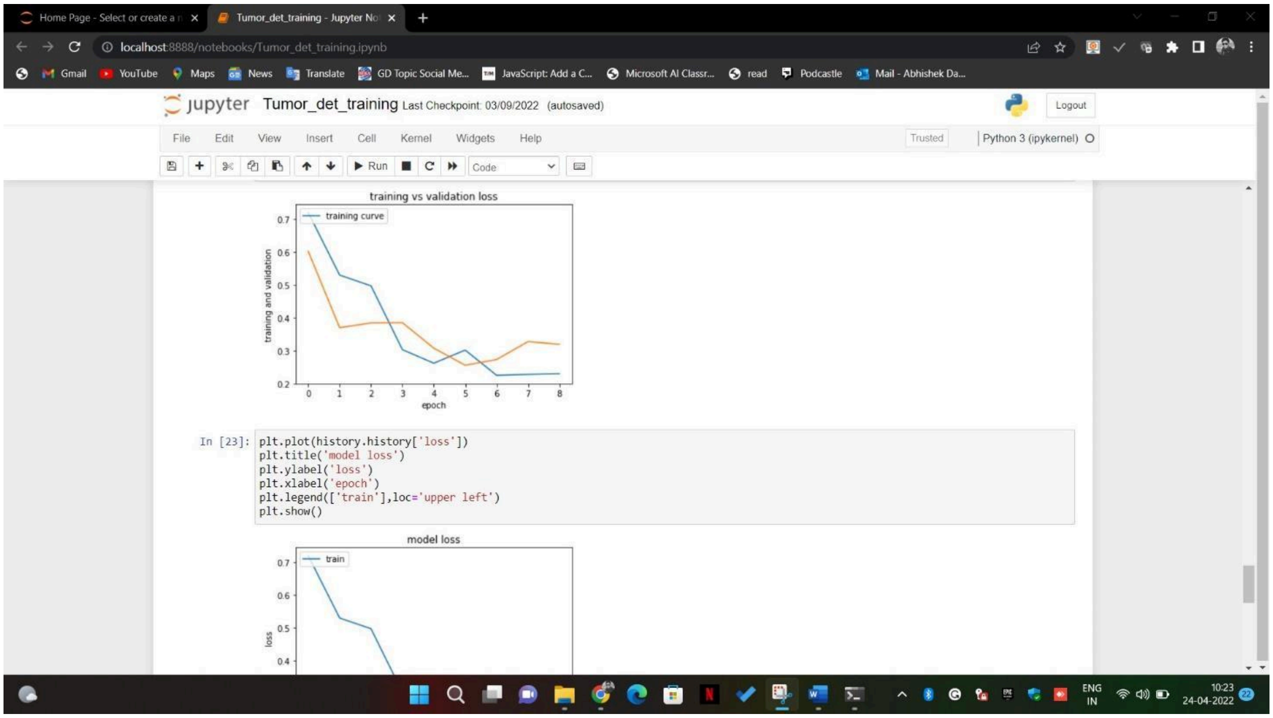


In [22]:

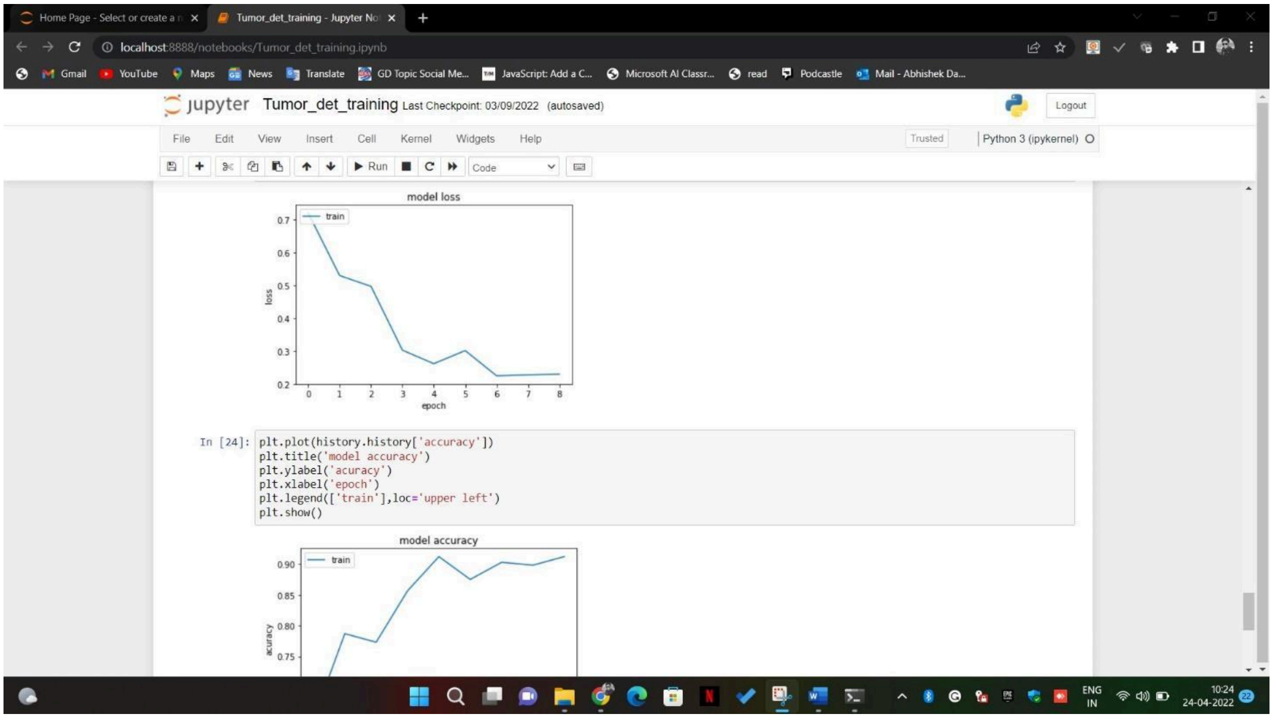
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('training vs validation accuracy')
plt.ylabel('training and validation')
plt.xlabel('epoch')
plt.legend(['training curve'],loc='upper left')
plt.show()
```

In [26]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('training vs validation loss')
plt.ylabel('training and validation')
plt.xlabel('epoch')
plt.legend(['training curve'],loc='upper left')
plt.show()
```

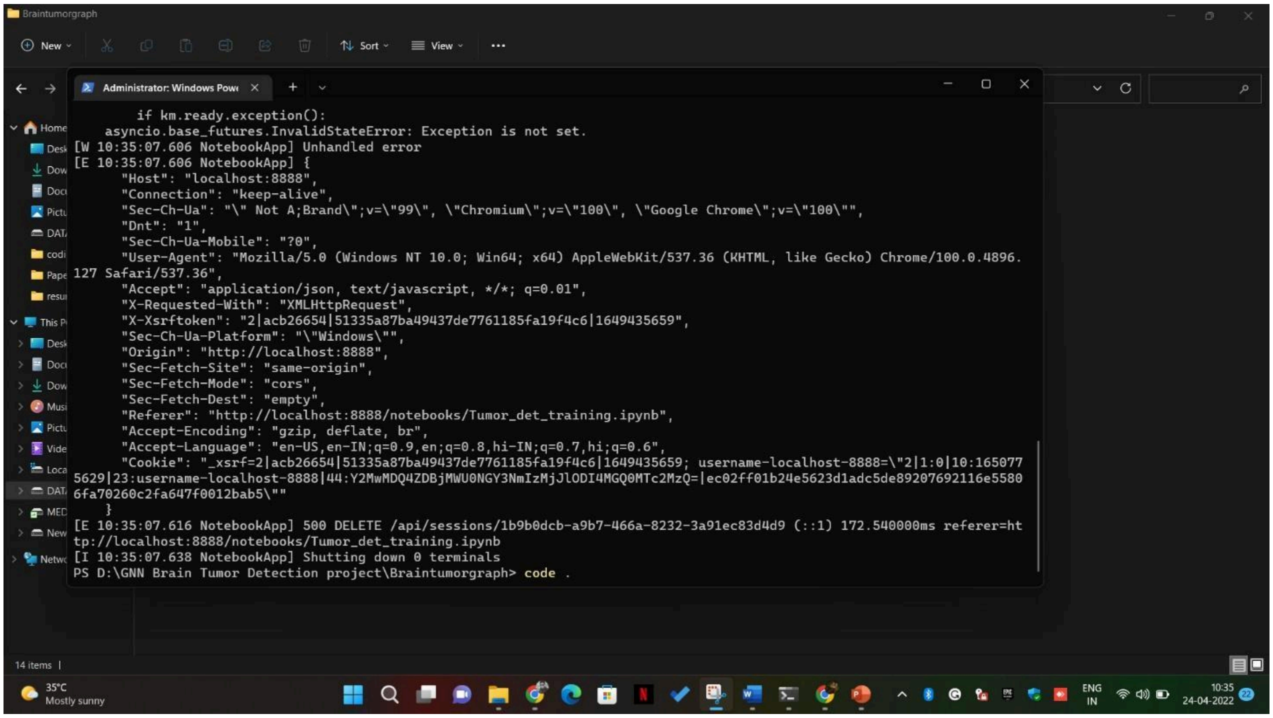


```
In [23]: plt.plot(history.history['loss'])
         plt.title('model loss')
         plt.ylabel('loss')
         plt.xlabel('epoch')
         plt.legend(['train'],loc='upper left')
         plt.show()
```

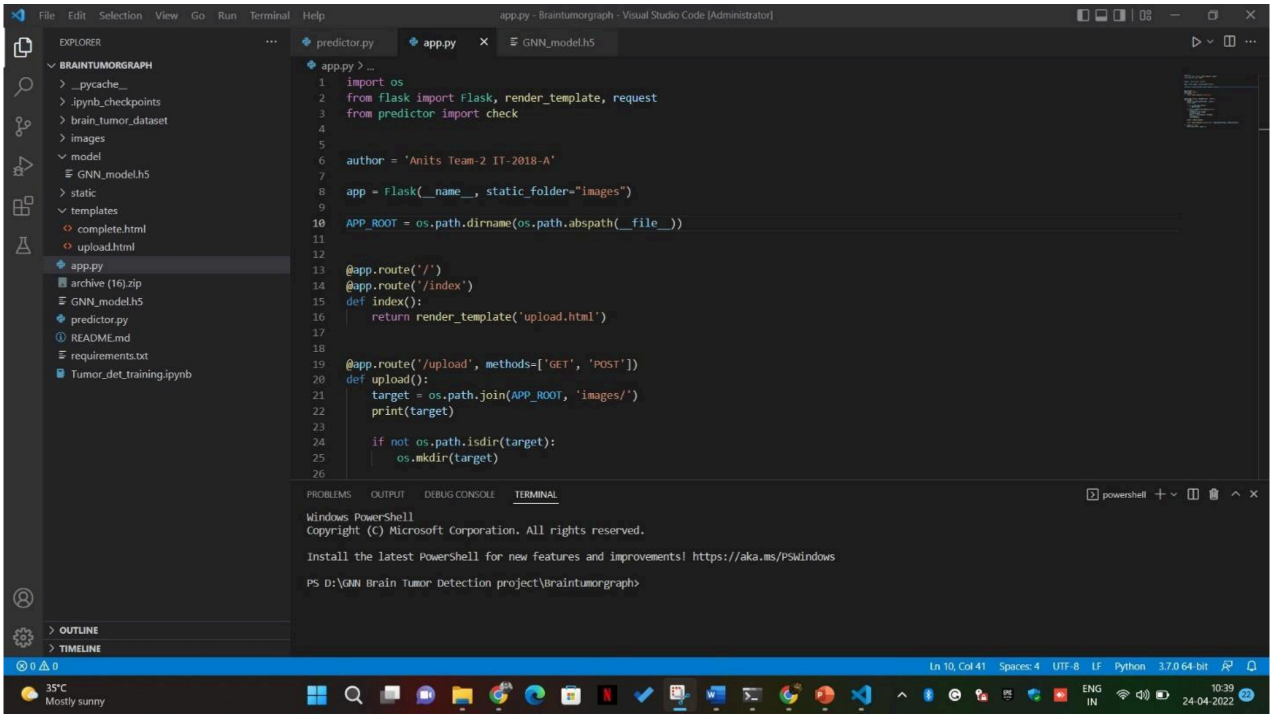


```
In [24]: plt.plot(history.history['accuracy'])
         plt.title('model accuracy')
         plt.ylabel('acuracy')
         plt.xlabel('epoch')
         plt.legend(['train'],loc='upper left')
         plt.show()
```

6.2 App.py:

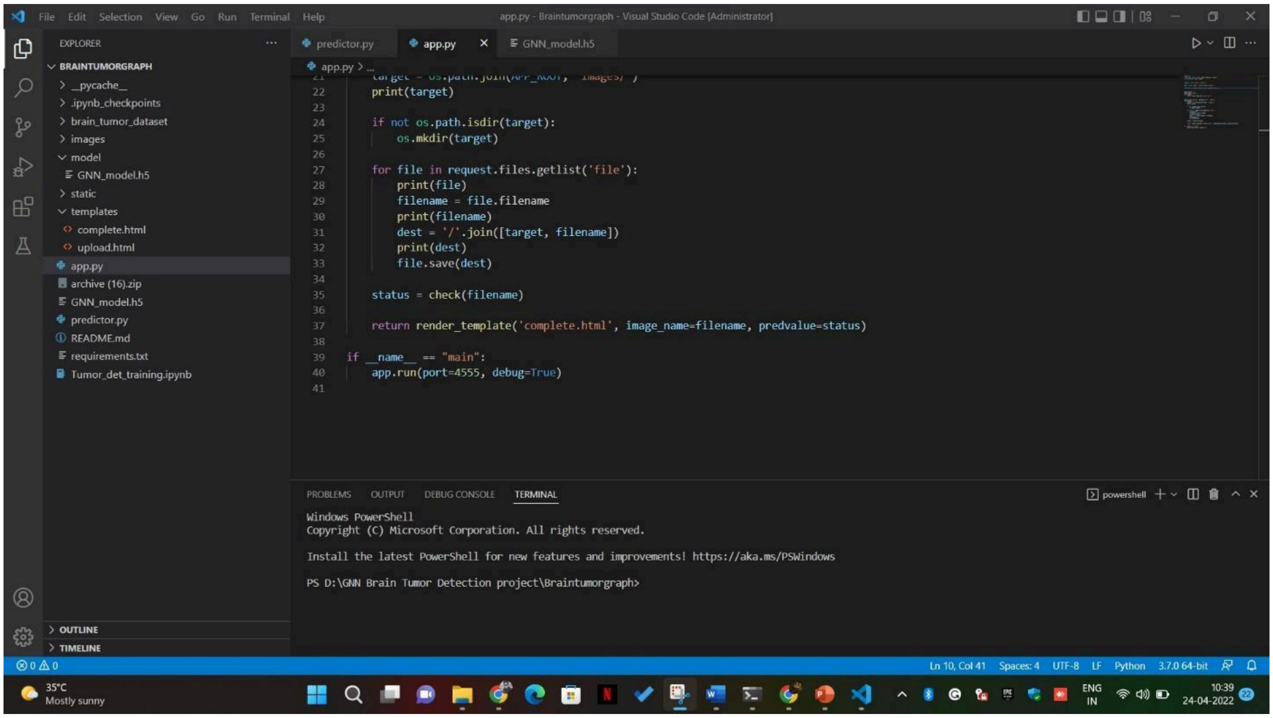


```
        if km.ready.exception():
    asyncio.base_futures.InvalidStateError: Exception is not set.
[W 10:35:07.606 NotebookApp] Unhandled error
[E 10:35:07.606 NotebookApp] {
      "Host": "localhost:8888",
      "Connection": "keep-alive",
      "Sec-Ch-Ua": "\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"100\", \"Google Chrome\";v=\"100\"",
      "Dnt": "1",
      "Sec-Ch-Ua-Mobile": "?0",
      "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.
127 Safari/537.36",
      "Accept": "application/json, text/javascript, */*; q=0.01",
      "X-Requested-With": "XMLHttpRequest",
      "X-Xsrftoken": "2|acb26654|51335a87ba49437de7761185fa19f4c6|1649435659",
      "Sec-Ch-Ua-Platform": "\"Windows\"",
      "Origin": "http://localhost:8888",
      "Sec-Fetch-Site": "same-origin",
      "Sec-Fetch-Mode": "cors",
      "Sec-Fetch-Dest": "empty",
      "Referer": "http://localhost:8888/notebooks/Tumor_det_training.ipynb",
      "Accept-Encoding": "gzip, deflate, br",
      "Accept-Language": "en-US,en-IN;q=0.9,en;q=0.8,hi-IN;q=0.7,hi;q=0.6",
      "Cookie": "_xsrf=2|acb26654|51335a87ba49437de7761185fa19f4c6|1649435659; username-localhost-8888=\"2|1:0|10:165077
5629|23:username-localhost-8888|44:Y2MwMDQ4ZDBjMWU0NGY3NmIzMjJlODI4MGQ0MTc2MzQ=|ec02ff01b24e5623d1adc5de89207692116e5580
6fa70260c2fa647f0012bab5\""
    }
[E 10:35:07.616 NotebookApp] 500 DELETE /api/sessions/1b9b0dcb-a9b7-466a-8232-3a91ec83d4d9 (::1) 172.540000ms referer=ht
tp://localhost:8888/notebooks/Tumor_det_training.ipynb
[I 10:35:07.638 NotebookApp] Shutting down 0 terminals
PS D:\GNN Brain Tumor Detection project\Braintumorgraph> code .
```



```python
import os
from flask import Flask, render_template, request
from predictor import check


author = 'Anits Team-2 IT-2018-A'

app = Flask(__name__, static_folder="images")

APP_ROOT = os.path.dirname(os.path.abspath(__file__))


@app.route('/')
@app.route('/index')
def index():
    return render_template('upload.html')


@app.route('/upload', methods=['GET', 'POST'])
def upload():
    target = os.path.join(APP_ROOT, 'images/')
    print(target)

    if not os.path.isdir(target):
        os.mkdir(target)

```

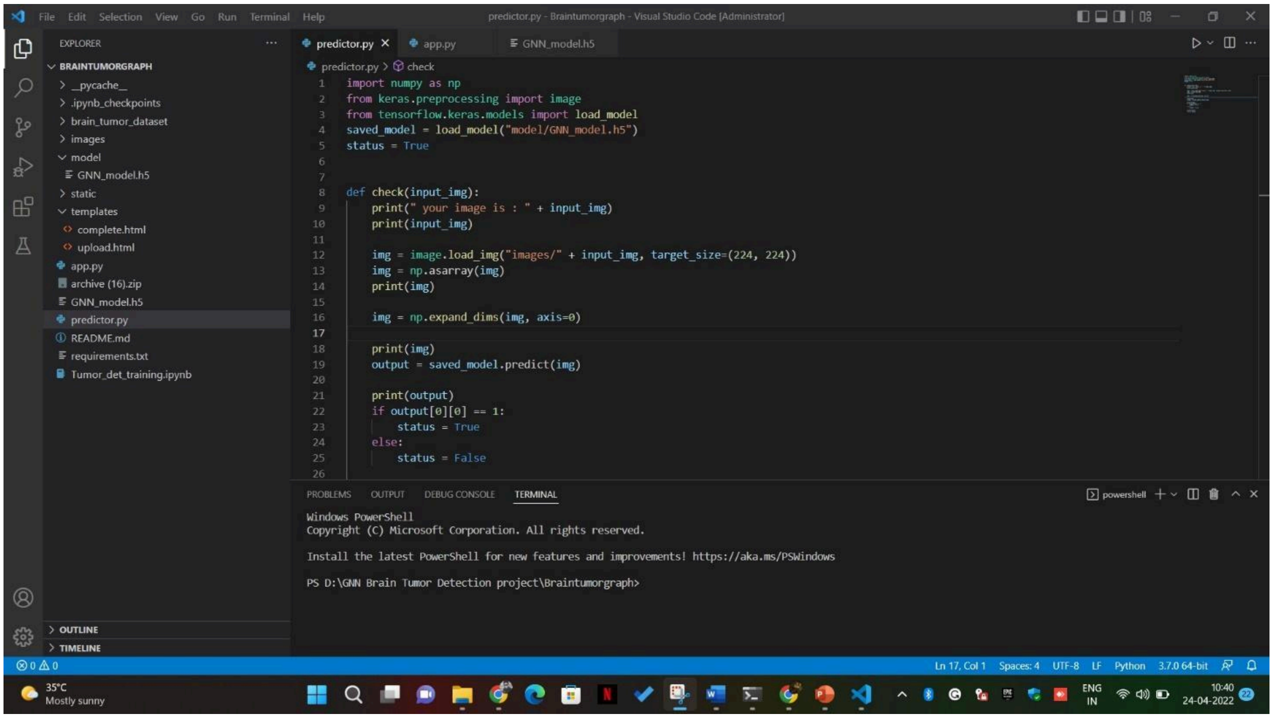
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```



```python
    target = os.path.join(APP_ROOT, 'images/')
    print(target)

    if not os.path.isdir(target):
        os.mkdir(target)

    for file in request.files.getlist('file'):
        print(file)
        filename = file.filename
        print(filename)
        dest = '/'.join([target, filename])
        print(dest)
        file.save(dest)

    status = check(filename)

    return render_template('complete.html', image_name=filename, predvalue=status)

if __name__ == "main":
    app.run(port=4555, debug=True)

```

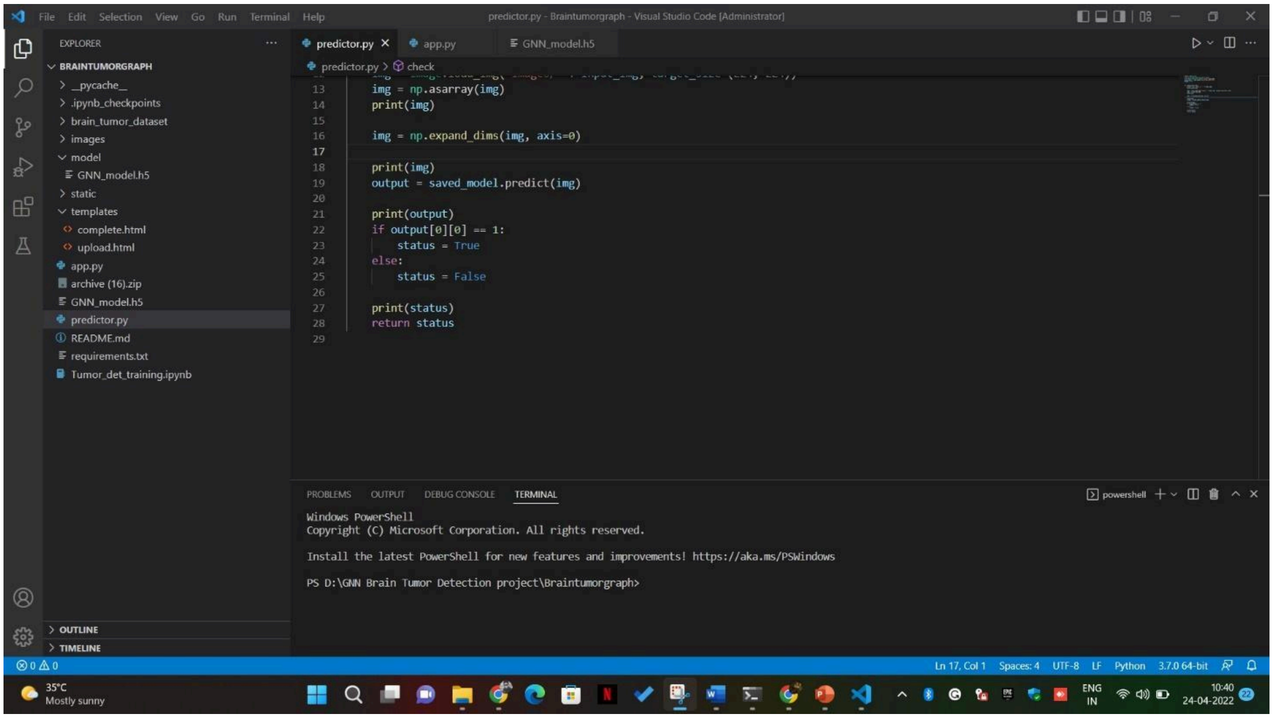
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```

6.3 Predictor.py:



```python
import numpy as np
from keras.preprocessing import image
from tensorflow.keras.models import load_model
saved_model = load_model("model/GNN_model.h5")
status = True


def check(input_img):
    print(" your image is : " + input_img)
    print(input_img)

    img = image.load_img("images/" + input_img, target_size=(224, 224))
    img = np.asarray(img)
    print(img)

    img = np.expand_dims(img, axis=0)

    print(img)
    output = saved_model.predict(img)

    print(output)
    if output[0][0] == 1:
        status = True
    else:
        status = False

```

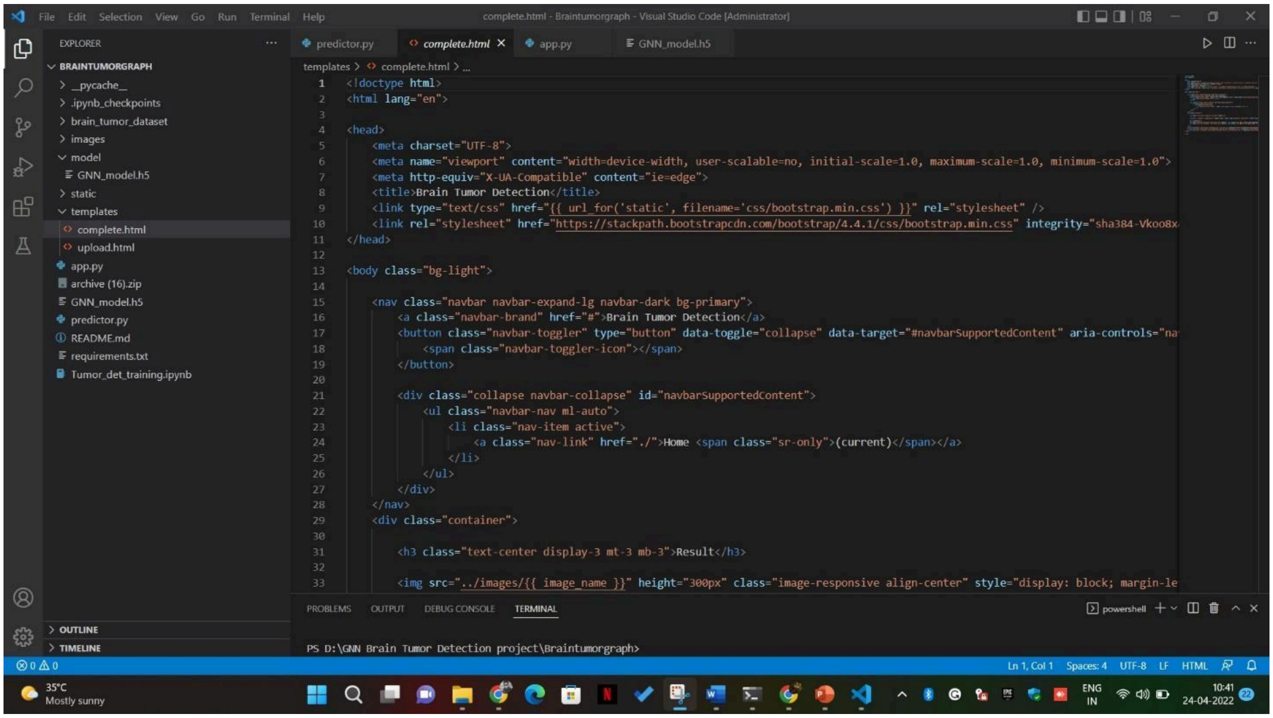
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```



```python
    img = np.asarray(img)
    print(img)

    img = np.expand_dims(img, axis=0)

    print(img)
    output = saved_model.predict(img)

    print(output)
    if output[0][0] == 1:
        status = True
    else:
        status = False

    print(status)
    return status

```

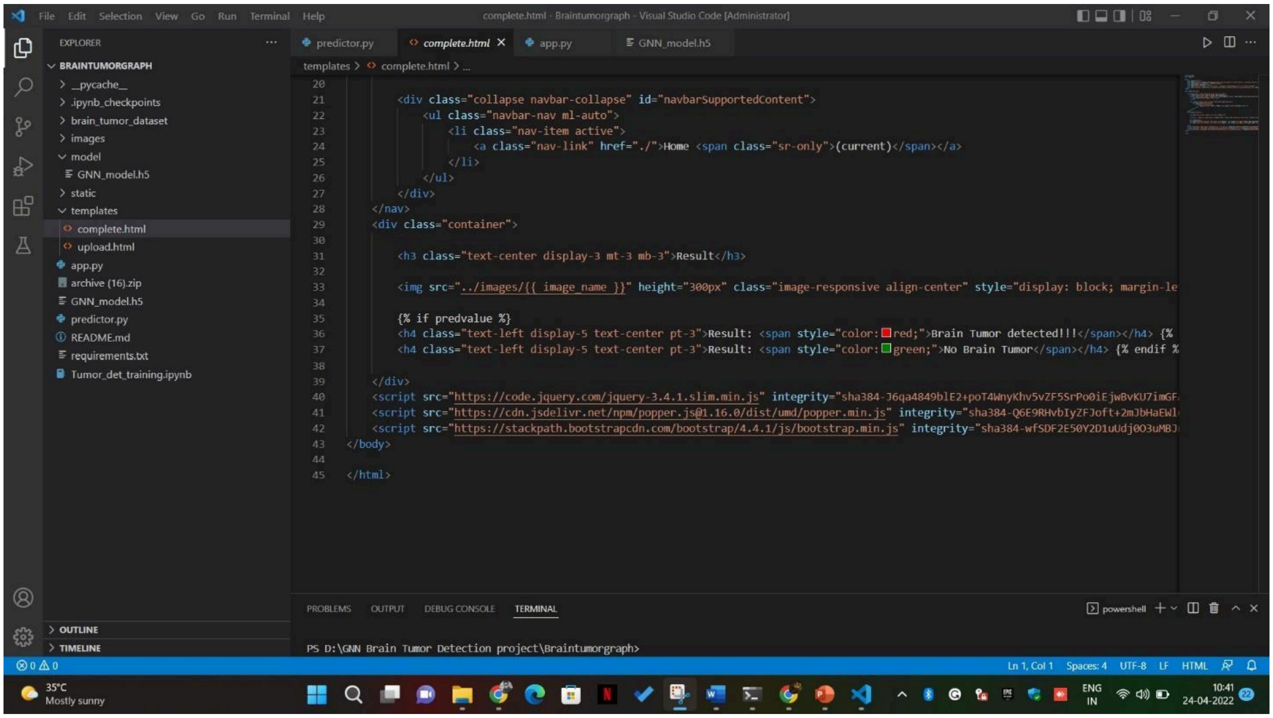
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```

6.4 Complete.html:



```html
<!doctype html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Brain Tumor Detection</title>
    <link type="text/css" href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet" />
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x
</head>

<body class="bg-light">

    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <a class="navbar-brand" href="#">Brain Tumor Detection</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="na
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="./">Home <span class="sr-only">(current)</span></a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container">

        <h3 class="text-center display-3 mt-3 mb-3">Result</h3>

        <img src="../images/{{ image_name }}" height="300px" class="image-responsive align-center" style="display: block; margin-le
```

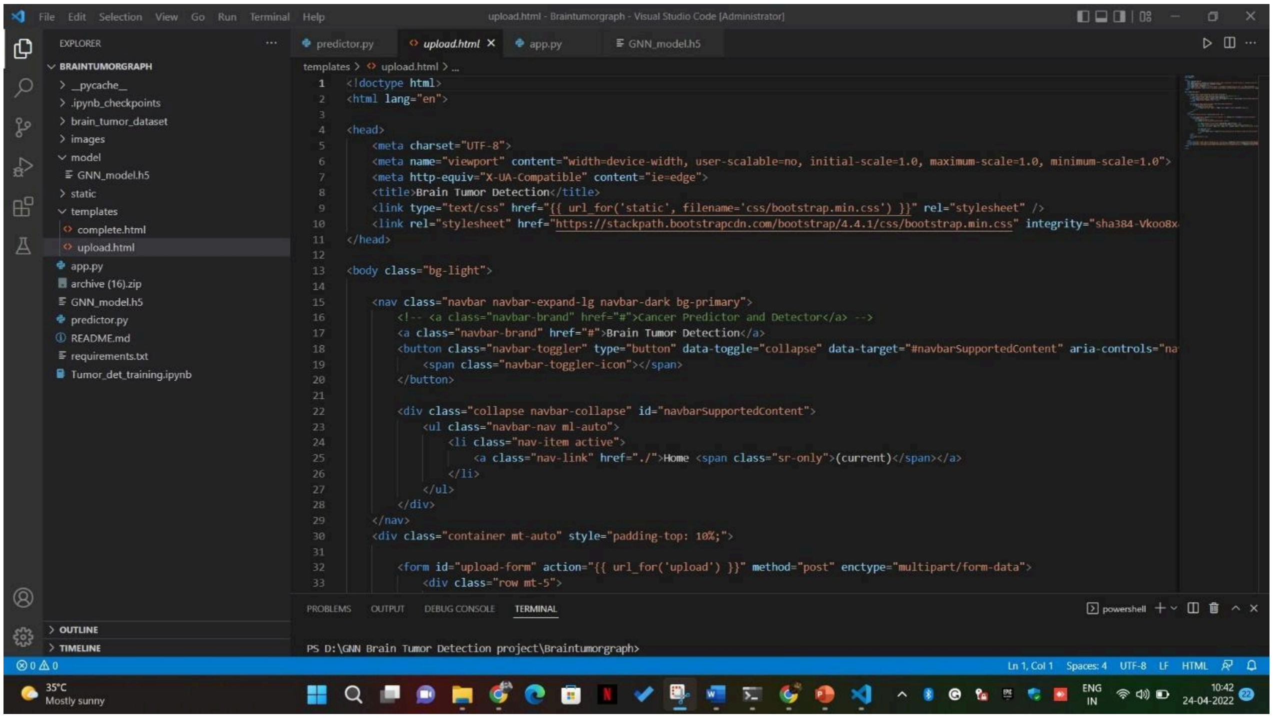
```
PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```



```html

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="./">Home <span class="sr-only">(current)</span></a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container">

        <h3 class="text-center display-3 mt-3 mb-3">Result</h3>

        <img src="../images/{{ image_name }}" height="300px" class="image-responsive align-center" style="display: block; margin-le

        {% if predvalue %}
        <h4 class="text-left display-5 text-center pt-3">Result: <span style="color:🟥red;">Brain Tumor detected!!!</span></h4> {%
        <h4 class="text-left display-5 text-center pt-3">Result: <span style="color:🟩green;">No Brain Tumor</span></h4> {% endif %

    </div>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGF
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWl
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJ
</body>

</html>
```

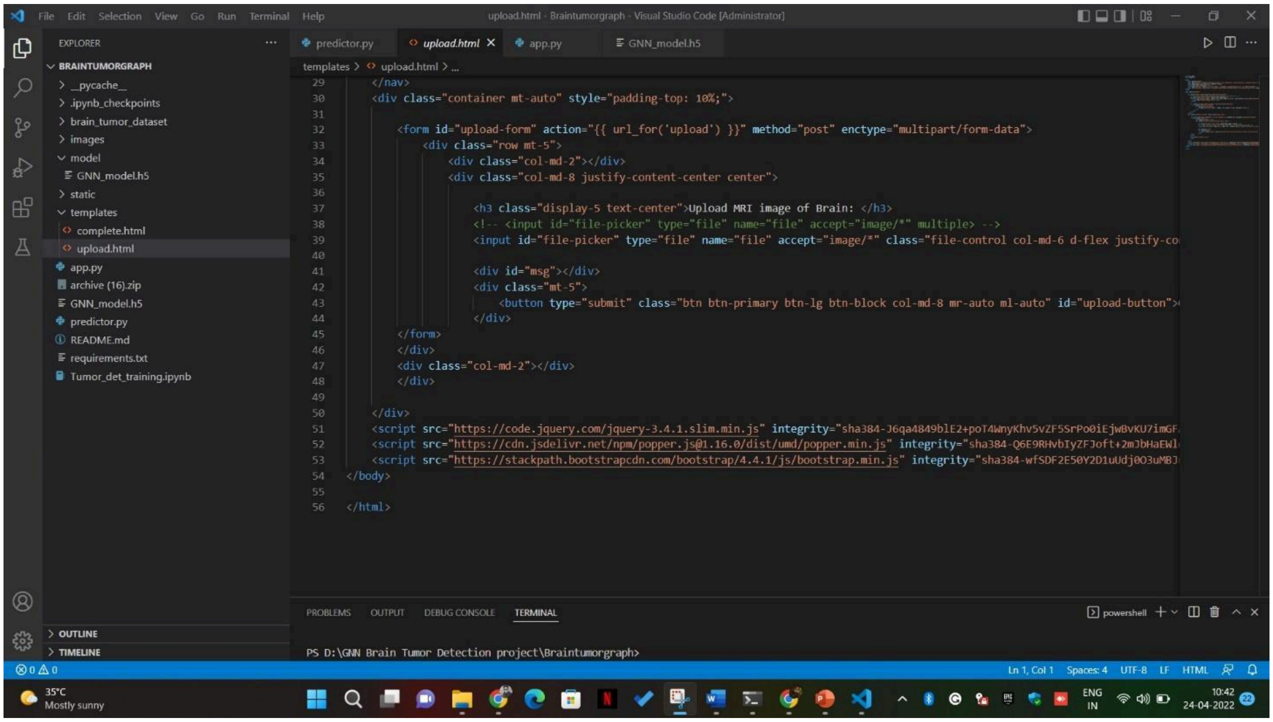
```
PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```

6.5 Upload.html:



```html
<!doctype html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Brain Tumor Detection</title>
    <link type="text/css" href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet" />
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x
</head>

<body class="bg-light">

    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <!-- <a class="navbar-brand" href="#">Cancer Predictor and Detector</a> -->
        <a class="navbar-brand" href="#">Brain Tumor Detection</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="na
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="./">Home <span class="sr-only">(current)</span></a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container mt-auto" style="padding-top: 10%;">

        <form id="upload-form" action="{{ url_for('upload') }}" method="post" enctype="multipart/form-data">
            <div class="row mt-5">
```

```
PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```



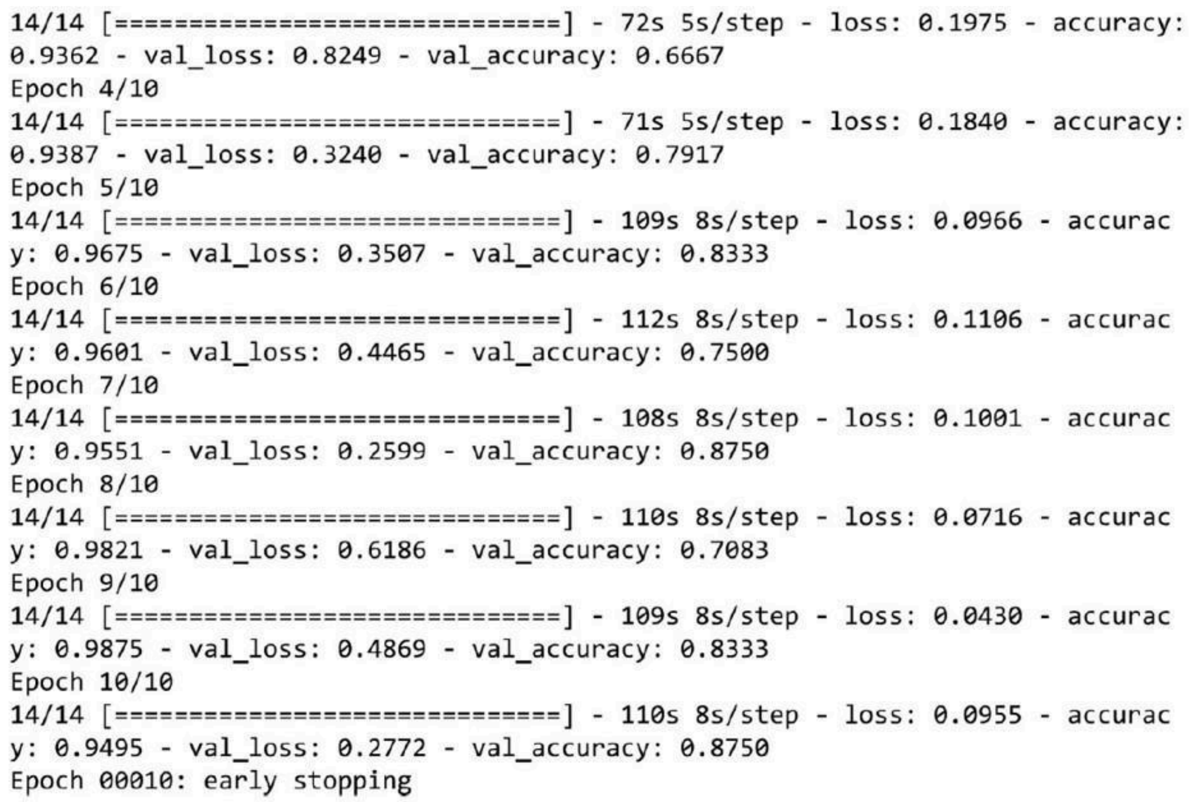
```html
    </nav>
    <div class="container mt-auto" style="padding-top: 10%;">

        <form id="upload-form" action="{{ url_for('upload') }}" method="post" enctype="multipart/form-data">
            <div class="row mt-5">
                <div class="col-md-2"></div>
                <div class="col-md-8 justify-content-center center">

                    <h3 class="display-5 text-center">Upload MRI image of Brain: </h3>
                    <!-- <input id="file-picker" type="file" name="file" accept="image/*" multiple> -->
                    <input id="file-picker" type="file" name="file" accept="image/*" class="file-control col-md-6 d-flex justify-co

                    <div id="msg"></div>
                    <div class="mt-5">
                        <button type="submit" class="btn btn-primary btn-lg btn-block col-md-8 mr-auto ml-auto" id="upload-button">
                    </div>
        </form>
        </div>
        <div class="col-md-2"></div>
        </div>

    </div>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGF
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWl
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJ
</body>

</html>
```

```
PS D:\GNN Brain Tumor Detection project\Braintumorgraph>
```

CHAPTER-7

Testing

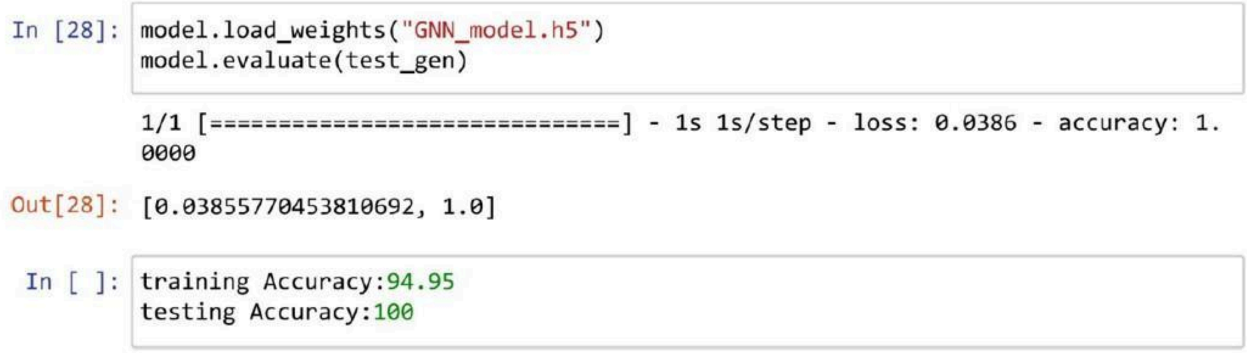
7.1 Accuracy we achieved using our model



```
14/14 [==============================] - 72s 5s/step - loss: 0.1975 - accuracy:
0.9362 - val_loss: 0.8249 - val_accuracy: 0.6667
Epoch 4/10
14/14 [==============================] - 71s 5s/step - loss: 0.1840 - accuracy:
0.9387 - val_loss: 0.3240 - val_accuracy: 0.7917
Epoch 5/10
14/14 [==============================] - 109s 8s/step - loss: 0.0966 - accurac
y: 0.9675 - val_loss: 0.3507 - val_accuracy: 0.8333
Epoch 6/10
14/14 [==============================] - 112s 8s/step - loss: 0.1106 - accurac
y: 0.9601 - val_loss: 0.4465 - val_accuracy: 0.7500
Epoch 7/10
14/14 [==============================] - 108s 8s/step - loss: 0.1001 - accurac
y: 0.9551 - val_loss: 0.2599 - val_accuracy: 0.8750
Epoch 8/10
14/14 [==============================] - 110s 8s/step - loss: 0.0716 - accurac
y: 0.9821 - val_loss: 0.6186 - val_accuracy: 0.7083
Epoch 9/10
14/14 [==============================] - 109s 8s/step - loss: 0.0430 - accurac
y: 0.9875 - val_loss: 0.4869 - val_accuracy: 0.8333
Epoch 10/10
14/14 [==============================] - 110s 8s/step - loss: 0.0955 - accurac
y: 0.9495 - val_loss: 0.2772 - val_accuracy: 0.8750
Epoch 00010: early stopping
```

fig- 7.1

7.2 Accuracy ranges from 95 to 100%



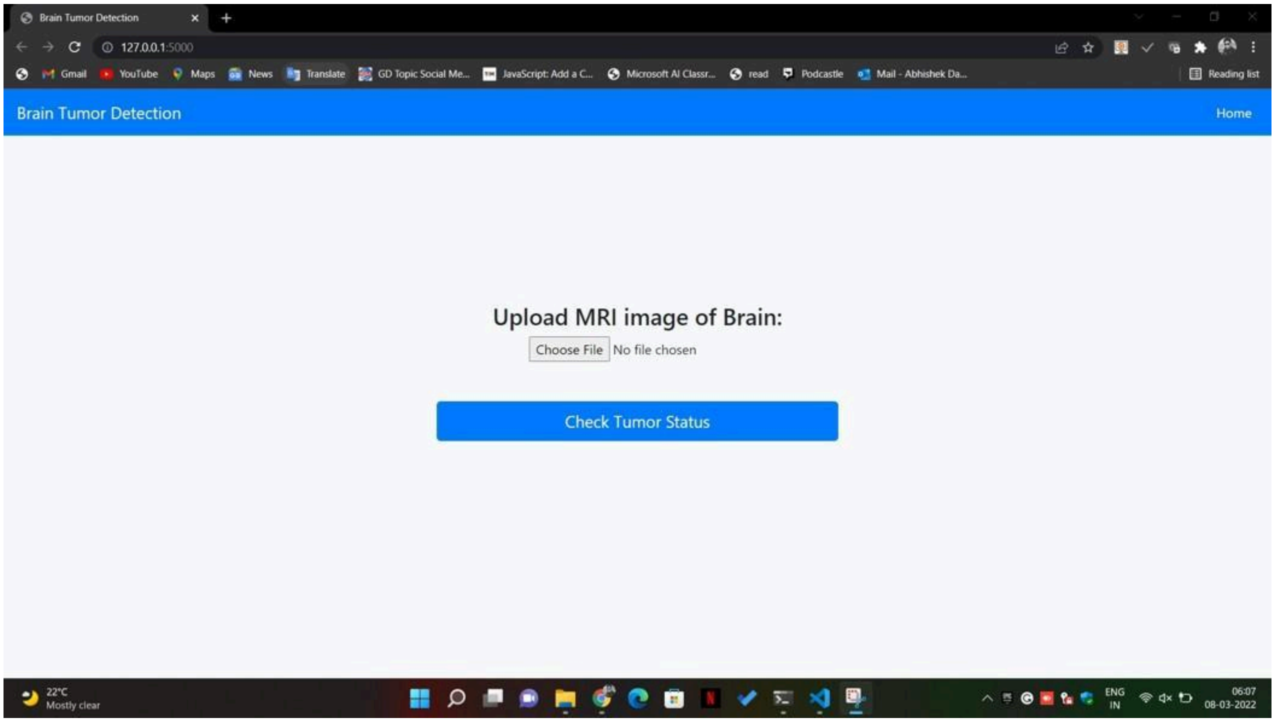
```
In [28]: model.load_weights("GNN_model.h5")
         model.evaluate(test_gen)

         1/1 [==============================] - 1s 1s/step - loss: 0.0386 - accuracy: 1.
         0000

Out[28]: [0.03855770453810692, 1.0]

In [ ]:  training Accuracy:94.95
         testing Accuracy:100
```

fig

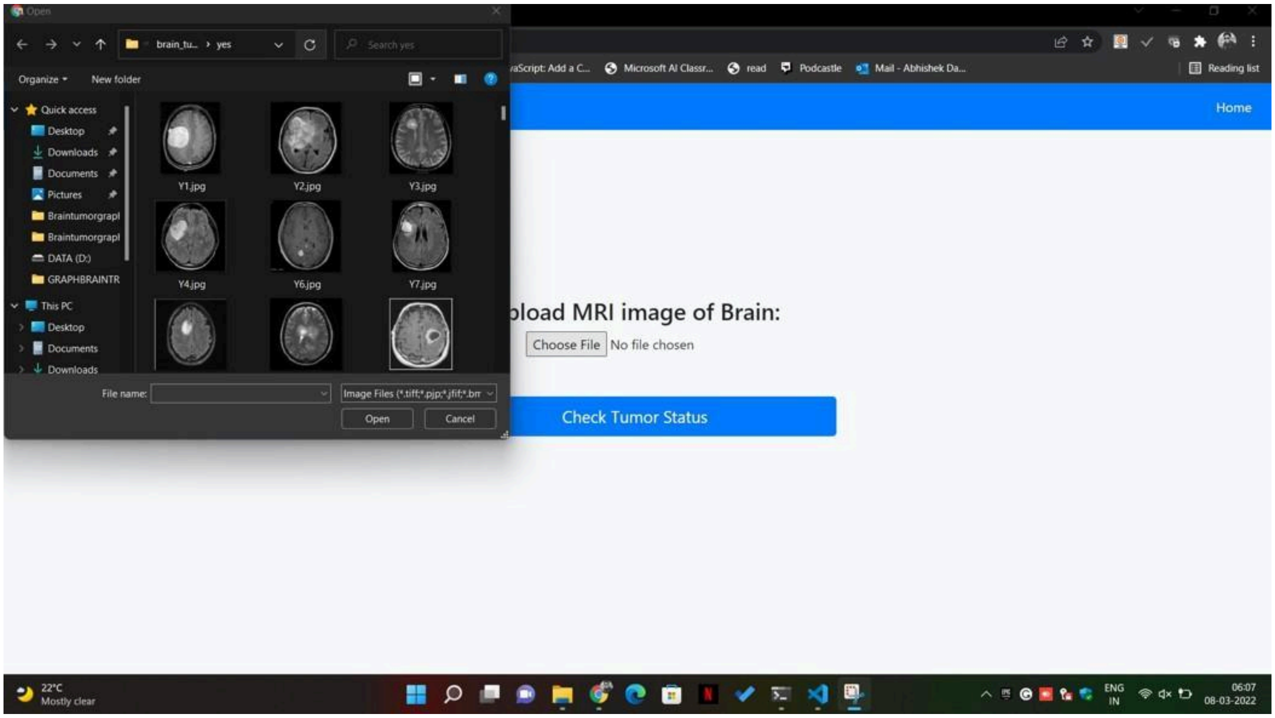
fig- 7.2

CHAPTER-8

Results



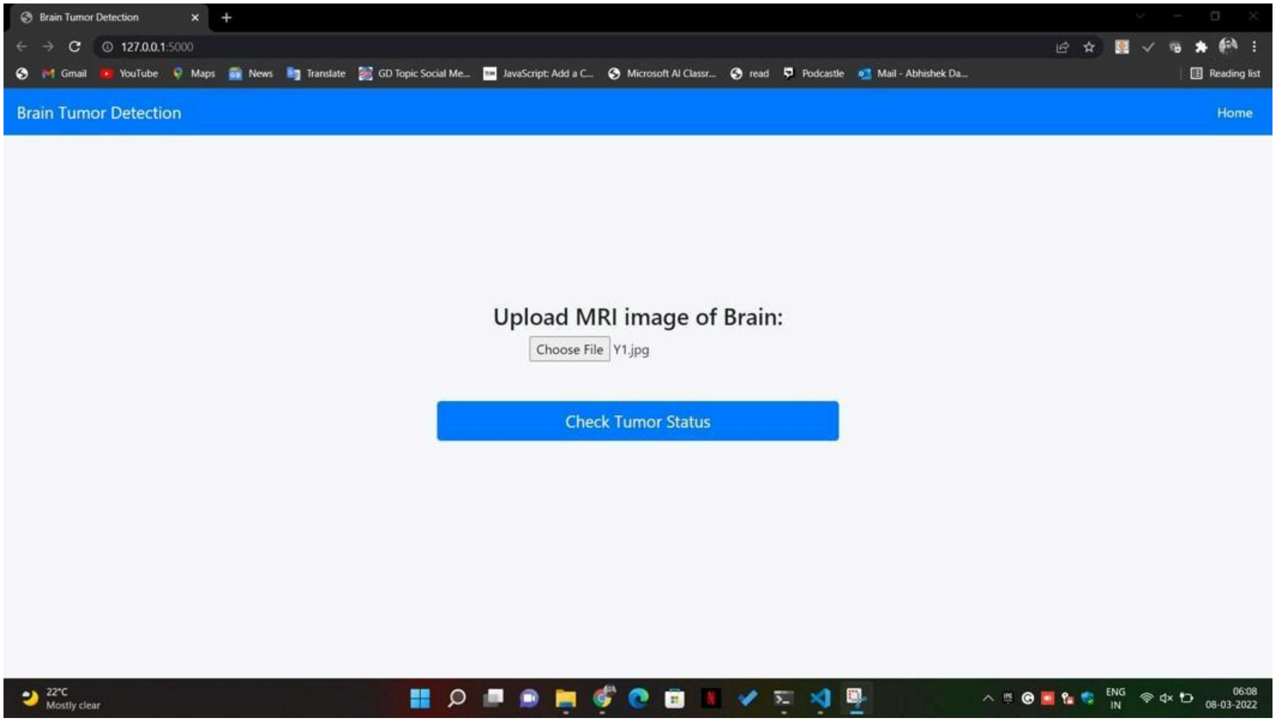
Opening flask application- fig 8.1



pload MRI image of Brain:

No file chosen

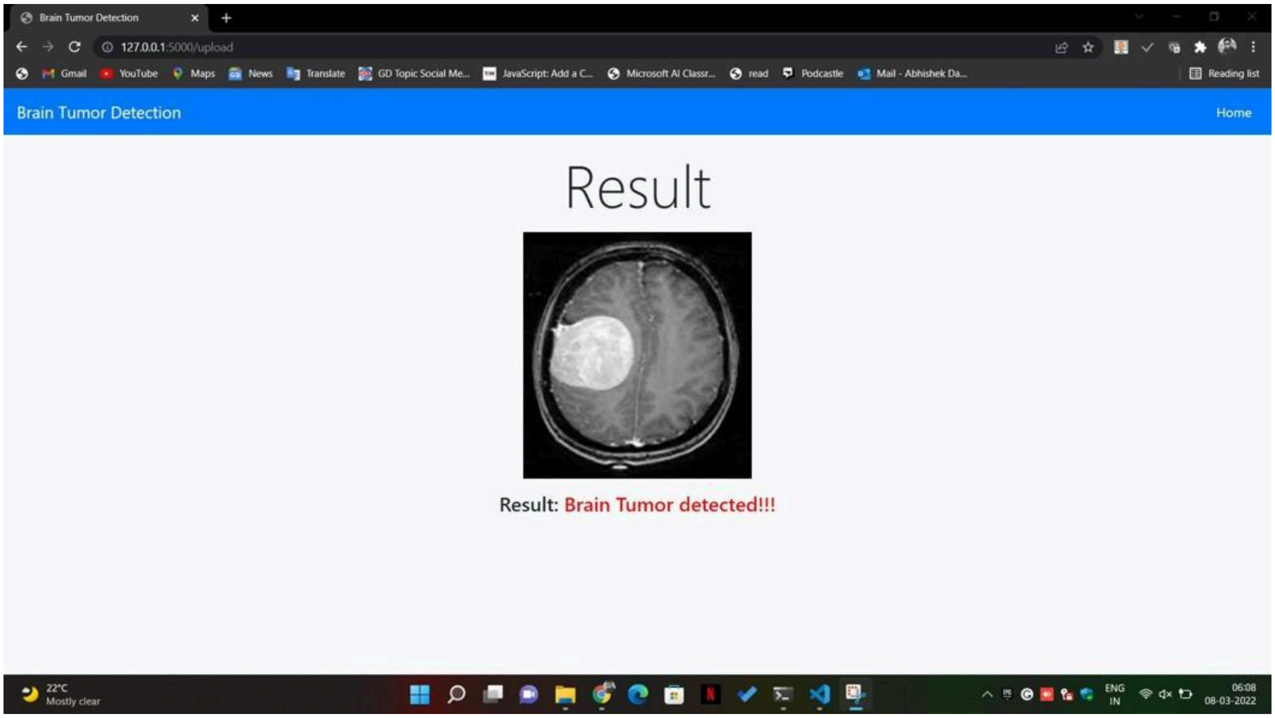
Selecting MRI Images- fig 8.2



Upload MRI image of Brain:

Y1.jpg

Checking Button- fig 8.3



Result

Result: Brain Tumor detected!!!

Results Display- fig 8.4

CHAPTER-9

CONCLUSION

We proposed an automated strategy for distinguishing proof of cerebrum growth utilizing the Brain Organization. The info MRI pictures are perused from the nearby gadget utilizing the document way and changed over into grayscale pictures. These pictures are pre-handled involving a versatile respective separating procedure for the disposal of clamors that are available inside the first picture. The double thresholding is applied to the denoised picture, and Brain Organization is applied, which helps in sorting out the growth area in the MRI pictures. The proposed model had acquired an exactness of 95% above and yields promising outcomes with practically no mistakes and significantly less computational time.

9.1 Future Scope:

It is seen on killing that the proposed approach needs an immense preparation set for better exact outcomes; in the field of clinical picture handling, the social occasion of clinical information is a drawn-out work, and, in a couple of cases, the datasets probably won't be accessible. In every single such case, the proposed calculation should be strong enough for exact acknowledgment of cancer areas from MR Pictures. The proposed approach can be additionally made done through coordinating feebly prepared calculations that can recognize the irregularities with a base preparation information furthermore self-learning calculations would support improving the exactness of the calculation and decrease the computational time.

References

- S. Sinha, Indian Sign Language: Linguistic Analysis of Its Language Program,

- D.C.: Gallaudet Univ. Press, 2018 🎬 T. Kim, J. Keane, W. Wang, H. Tang, J. Riggle, G. Saharovich, D. Brentari, and K. Livescu, “Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation,” Comput. Speech Lang., vol. 46, pp. 209–232, Nov. 2017.
  - [2] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. B. Lakulu, “A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017,” Sensors, vol. 18, no. 7, p. 2208, 2018.
  - [3] R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuous sign language recognition by iterative training,” IEEE Trans. Multimedia, vol. 21, no. 7, pp. 1880–1891, Jul. 2019.
  - [4] P. S. Santhalingam, P. Pathak, J. Košecká, and H. Rangwala, “Sign language recognition analysis using multimodal data,” in Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2019, pp. 203–210.
  - [5] A. C. Duarte, “Cross-modal neural sign language translation,” in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019, pp. 1650–1654.
  - [6] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, Jan. 2019.
  - [7] Q. Xiao, Y. Zhao, and W. Huan, “Multi-sensor data fusion for sign language recognition based on dynamic Bayesian network and convolutional neural network,” Multimedia Tools Appl., vol. 78, no. 11, pp. 15335–15352, Jun. 2019. [
  - 8] E. K. Kumar, P. V. V. Kishore, M. T. K. Kumar, and D. A. Kumar, “3D signs language recognition with joint distance and angular coded color topographical descriptor on a 2–stream CNN,” Neurocomputing, vol. 372, pp. 40–54, Jan. 2020.
  - [9] J. Wu and R. Jafari, “Wearable computers for sign language recognition,” in Handbook of Large-Scale Distributed Computing in Smart Healthcare.