A

Project Report

On

TUMOR TRACKER

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

RK VALLEY

In partial fulfilment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

SINGAMSETTI NANDINI (R180018)

ILOUR SHAIK RIZWAN RAJA (R180006)

JHANSYSREENIVAS MANCHURI (R180996)

Under the Guidance of

Mr. A.MAHENDRA, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(Catering the Educational Needs of Gifted Rural Youth of AP)

R.K Valley, Vempalli (M), Kadapa (Dist) – 516330, 2020 - 2024

RAJIV GANDHI UNIVERSITY OF KNOWELDEGE TECHNOLOGIES

(Government Act 18 of 2008)

RGUKT -RK Valley

Vempalli, Kadapa, Andhra Pradesh -516330

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the work entitled **“Tumor Tracker”** is bonafide work of **Singamsetti Nandini(R180018), Ilour Shaik Rizwan Raja(R180006) ,JhansySreenivas Manchuri(R180996)** carried out under our guidance and supervision for the partial fulfilment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session 2023 to 2024 at RGUKT-RK VALLEY.

Project Guide	**Head of the Department**
Mr A MAHENDRA	Dr. PENUGONDA RAVI KUMAR
Assistant Professor in Dept of CSE	Assistant Professor in Dept of CSE
RGUKT RK-VALLEY	M.E (IISC Bangalore)
	Ph.D. (University of AIZU, Japan)

RAJIV GANDHI UNIVERSITY OF KNOWELDEGE TECHNOLOGIES

(Government Act 18 of 2008)

RGUKT -RK Valley

Vempalli, Kadapa, Andhra Pradesh -516330

SIGNATURE OF EXTERNAL EXAMINER

This is to certify that the work entitled **"Tumor Tracker"** is bonafide work of **Singamsetti Nandini(R180018), Ilour Shaik Rizwan Raja(R180006)** ,**JhansySreenivas Manchuri(R180996)** carried out under our guidance and supervision for the partial fulfilment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session 2023 to 2024 at RGUKT-RK VALLEY.

Signature of External Examiner

RAJIV GANDHI UNIVERSITY OF KNOWELDEGE TECHNOLOGIES

(A.P.Government Act 18 of 2008)

RGUKT -RK Valley

Vempalli, Kadapa, Andhra Pradesh -516330

DECLARATION

We, **Singamsetti Nandini (R180018), Ilour Shaik Rizwan Raja (R180006), JhansySreenivas Manchuri (R180996)** hereby declare that the project report entitled "**Tumor Tracker**" done under guidance of **Mr. A.Mahendra** is submitted in partial fulfilment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session 2023 to 2024 at RGUKT-RK Valley. we also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Date:

Place:

Singamsetti Nandini - R180018

Ilour Shaik Rizwan Raja - R180006

JhansySreenivas Manchuri - R180996

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude & respect to all those people behind the screen who guided, inspired and helped me crown all my efforts with success. I wish to express my gratitude to **A.Mahendra** for his valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which it would not be possible to complete this project.

I would also like to extend our deepest gratitude & reverence to the Director of RGUKT, RK Valley **Dr.A V S S Kumar Swami Gupta** and HOD of Computer Science and Engineering **Dr. Penugonda Ravi Kumar** for their constant support and encouragement.

Last but not least I express my gratitude to my parents for their constant source of encouragement and inspiration for me to keep my morals high

With Sincere Regards,

Singamsetti Nandini – R180018

Ilour Shaik Rizwan Raja-R180006

JhansySreenivas Manchuri-R180996

TABLE OF CONTENTS

1.ABSTRACT

Automated Brain Tumor Detection and Visualization System The Automated Brain Tumor Detection and Visualization System is a computer-based solution designed to enhance the efficiency and accuracy of brain tumor diagnosis using medical imaging techniques. Leveraging computer vision and machine learning algorithms, the system automates the process of analyzing Magnetic Resonance Imaging (MRI) scans to identify potential tumor regions.

The system's workflow includes image preprocessing, feature extraction, classification, and visualization. During preprocessing, MRI images are enhanced to improve quality and reduce noise. Feature extraction techniques extract relevant features from the images, which are then fed into machine learning models for tumor classification. Convolutional Neural Networks (CNNs) are utilized for their ability to discern between tumor and non-tumor regions with high accuracy.

Upon Tumor detection, the system provides intuitive visualization tools to display the identified Tumor area on the original MRI image. This visualization aids medical professionals in understanding Tumor characteristics and facilitates treatment planning.

Key features of the system include:

- Automated MRI image analysis for brain Tumor detection.
- Integration of advanced machine learning algorithms for accurate classification.
- User-friendly interface for uploading MRI images, performing analysis, and visualizing results.
- Reduction in manual analysis time, leading to improved efficiency and faster diagnosis.
- Enhancements in patient care through reliable and automated Tumor detection.

The Automated Brain Tumor Detection and Visualization System is poised to revolutionize brain Tumor diagnosis by providing a reliable, efficient, and user-friendly solution that empowers medical professionals in their diagnostic processes.

2. INTRODUCTION

In the realm of medical diagnostics, the accurate detection and visualization of brain tumors are critical for effective treatment planning and patient care. The advent of advanced imaging technologies, coupled with the power of artificial intelligence and machine learning, has opened new avenues for automating and enhancing the accuracy of brain tumor diagnosis. The project "Automated Brain Tumor Detection and Visualization System" is a comprehensive solution aimed at leveraging these technological advancements to streamline the process of brain tumor detection and visualization.

Brain tumors represent a significant health concern globally, with varying degrees of severity and complexity. Traditionally, the diagnosis of brain tumors involves manual interpretation of medical imaging data, primarily Magnetic Resonance Imaging (MRI) scans, by skilled radiologists and neurosurgeons. While human expertise is invaluable, the process is time-consuming, subjective to interpretation variations, and prone to errors, especially in the case of subtle or complex tumor structures.

The Automated Brain Tumor Detection and Visualization System addresses these challenges by integrating cutting-edge technologies into a cohesive platform designed to automate key aspects of brain tumor diagnosis. The system's core functionalities include image preprocessing, feature extraction, tumor classification, and intuitive visualization of detected tumor regions.

At the heart of the system lies sophisticated image processing algorithms that enhance the quality of MRI images, reducing noise and artifacts to improve the accuracy of subsequent analyses. Feature extraction techniques extract meaningful features from the preprocessed images, capturing essential tumor characteristics such as shape, texture, and intensity variations.

Machine learning algorithms, particularly Convolutional Neural Networks (CNNs), play a pivotal role in tumor classification. Trained on annotated MRI data, CNNs can effectively differentiate between tumor and non-tumor regions with high accuracy, aiding in the automated identification of potential tumor areas.

Once a tumor is detected, the system offers intuitive visualization tools that overlay the identified tumor region on the original MRI scan. This visual representation provides medical professionals with valuable insights into the tumor's location, size, and morphology, facilitating informed decision-making and treatment planning.

The project aims to revolutionize brain tumor diagnosis by:

- **Improving Diagnostic Accuracy**: By leveraging advanced algorithms and machine learning, the system enhances the accuracy of brain tumor detection compared to traditional manual methods.
- **Enhancing Efficiency**: Automation reduces the time and effort required for tumor analysis, allowing medical professionals to focus more on patient care and treatment strategies.
- **Enabling Early Detection**: Early detection of brain tumors can lead to timely interventions and improved patient outcomes, making the system invaluable in clinical settings.
- **Facilitating Collaborative Care**: The visualization tools enable seamless collaboration between radiologists, neurosurgeons, and oncologists, fostering interdisciplinary approaches to patient care.

3. PROBLEM STATEMENT

Brain tumors represent a significant health challenge worldwide, with varying complexities and treatment strategies. The problem we address is the inefficiency and subjectivity associated with manual brain tumor detection and visualization. Existing methods heavily rely on human expertise, leading to inconsistencies in interpretation and potential diagnostic errors. Furthermore, the lack of accessible and precise automated tools tailored to brain tumor analysis contributes to delays in treatment planning and patient care.

3.1 Identifying the Challenge

- Critical Healthcare Issue: Brain tumors pose serious health risks and require prompt diagnosis for effective treatment.
- Manual Interpretation Challenges: Human-based interpretation of MRI scans for brain tumor detection is time-consuming and prone to errors.
- Lack of Automated Tools: Existing automated tools for brain tumor detection are limited in accuracy and accessibility.
- Need for Precision: Precision in brain tumor detection is crucial for timely interventions and improved patient outcomes.
- Enhanced Visualization Requirements: Intuitive visualization tools are essential for medical professionals to comprehend and analyze brain tumor data effectively.

3.2 Project's Goal

The primary goal of this project is to develop an advanced automated system capable of accurately detecting brain tumors and providing intuitive visualization of tumor regions. By harnessing the power of artificial intelligence, machine learning, and image processing techniques, the system aims to streamline the diagnostic process, improve diagnostic accuracy, and facilitate informed decision-making for healthcare professionals.

3.3 Objectives

- Optimize Detection Accuracy: Fine-tune algorithms to achieve high accuracy in brain tumor detection compared to manual methods.
- Enhance Visualization Tools: Develop intuitive visualization tools that overlay detected tumor regions on MRI scans for comprehensive analysis.
- Automate Diagnostic Workflow: Implement automated processes for image preprocessing, feature extraction, and tumor classification.
- Improve Accessibility: Ensure the system is user-friendly and accessible to healthcare professionals across different medical settings.
- Validate Reliability: Conduct rigorous validation using real-world data to validate the system's reliability and performance.
- Facilitate Treatment Planning: Provide actionable insights to aid in treatment planning and patient care strategies.

3.4 Conclusion

The "Automated Brain Tumor Detection and Visualization System" represents a significant advancement in medical technology, aiming to transform the landscape of brain tumor diagnosis. By addressing the challenges associated with manual interpretation and limited automation, the project seeks to empower healthcare professionals with efficient tools for accurate brain tumor detection and visualization, ultimately leading to improved patient outcomes and enhanced healthcare delivery.

4. System Requirements

4.1 Software Components

Operating System: Windows/Ubuntu

4.2 Hardware Components

Storage: Hard Disk/SSD 512 GB

Memory: RAM 8 GB

4.3 Tools and Technologies Used

Machine Learning Algorithms:

Image Preprocessing Algorithms:

- Histogram Equalization: Enhances image contrast by adjusting the intensity distribution.
- Gaussian Smoothing: Reduces noise and sharpens edges by applying a Gaussian blur.
- Thresholding: Converts grayscale images to binary images based on intensity thresholds.

Feature Extraction Algorithms:

- Gray Level Co-occurrence Matrix (GLCM): Calculates texture features such as contrast, energy, entropy, and homogeneity.
- Local Binary Patterns (LBP): Describes texture patterns by comparing pixel values with neighboring pixels.
- Scale-Invariant Feature Transform (SIFT): Detects and describes local features in images, robust to scaling and rotation.

Segmentation Algorithms:

- Thresholding-Based Segmentation: Separates foreground (tumor) from background based on intensity thresholds.
- Region Growing: Expands regions of similar pixel characteristics to segment tumors.
- Watershed Transform: Divides the image into regions based on gradient magnitude to identify tumor boundaries.

Classification Algorithms:

- Support Vector Machines (SVM): Classifies tumors based on extracted features using a hyperplane separation technique.
- Random Forest: Ensemble learning method that builds multiple decision trees to classify tumors.
- Convolutional Neural Networks (CNN): Deep learning approach for automatic feature learning and tumor classification from image data.
- K-Nearest Neighbors (KNN): Classifies tumors based on similarity to neighboring data points in feature space.

Post-processing Algorithms:

- Morphological Operations: Erosion, dilation, opening, and closing operations to refine segmented tumor regions.
- Connected Component Analysis: Identifies and labels connected regions in binary images to separate tumor components.
- Boundary Refinement: Smoothens and refines tumor boundaries for better visualization and analysis.Flask (Micro Web Framework)

Python Programming Language

Data Processing and Analysis Tools:

- Jupyter Notebook

Graphical User Interface (GUI) Library:

- Tkinter

Version Control System:

Git

Integrated Development Environment (IDE):

- Visual Studio Code

Version Control Hosting:

- GitHub

4.4 Installation

Prerequisites:

Installation Steps:

Python: Ensure that Python is installed on your system. You can download the latest version of Python from the official website (https://www.python.org/downloads/).

Python Libraries:

NumPy: A library for numerical computations, essential for working with arrays or matrices.

- pip install numpy

Pandas: A library for data manipulation and analysis, used for handling structured data.

- pip install pandas

scikit-learn: A machine learning library offering tools for building predictive models and data preprocessing.

- pip install scikit-learn

TensorFlow and Keras (Optional for Deep Learning): Libraries for building and training deep learning models.

- pip install tensorflow
- pip install keras

OpenCV (cv2): An open-source computer vision library used for image processing and analysis.

- pip install opencv-python

PIL (Python Imaging Library) or Pillow: Libraries for image processing tasks such as reading, manipulating, and saving images.

- pip install pillow

Tkinter (Optional for GUI): A library for creating graphical user interfaces in Python.

Tkinter is usually included with Python installations by default. You can check its availability by attempting to import it in a Python script:

- import tkinter

Jupyter Notebook (Optional for Data Analysis): An interactive environment for data analysis and visualization.

- pip install jupyterlab

Flask (Optional for Web Interface): A lightweight web framework for building web applications.

- pip install flask

<u>Other Dependencies</u>:

Ensure you have a compatible IDE or code editor installed, such as Visual Studio Code or PyCharm, for development.

Install Git for version control if you plan to collaborate or use versioning for your project.

Set up a database management system (DBMS) if your project involves data storage and retrieval from a database.

5. BACKGROUND AND LITERATURE REVIEW

5.1 Brain Tumors

Brain tumors are abnormal growths of cells within the brain or surrounding tissues. They can be benign (non-cancerous) or malignant (cancerous) and can arise from various cell types within the brain. Brain tumors can cause a range of symptoms depending on their location, size, and type, including headaches, seizures, cognitive changes, and neurological deficits.

5.2 Types of Brain Tumors

Primary Brain Tumors:

Primary brain tumors originate in the brain itself and can be benign (such as meningiomas, pituitary adenomas) or malignant (such as gliomas, glioblastomas).

Metastatic Brain Tumors:

Metastatic brain tumors, also known as secondary brain tumors, occur when cancer cells from other parts of the body spread to the brain. Common primary sites for metastasis include the lungs, breast, colon, and skin (melanoma).

Classification of Brain Tumors:

Brain tumors are classified based on their cell origin, location, histology, and genetic markers. Common classification systems include the World Health Organization (WHO) classification, which categorizes tumors into grades based on their malignancy and aggressiveness.

5.3 Diagnostic Methods for Brain Tumors:

Various diagnostic methods are used to detect and classify brain tumors:

Neuroimaging:

MRI (Magnetic Resonance Imaging): Provides detailed images of brain structures and can identify abnormal growths, aiding in tumor localization and characterization.

CT (Computed Tomography) Scan: Offers cross-sectional images of the brain, helpful for detecting structural abnormalities and assessing tumor size and density.

Biopsy:

Surgical Biopsy: Involves removing a tissue sample for pathological examination to determine tumor type, grade, and molecular characteristics.

Non-Invasive Techniques:

Liquid Biopsy: Emerging technique involving the analysis of tumor-specific biomarkers (such as circulating tumor DNA) in blood samples, enabling minimally invasive tumor profiling.

5.4 AI and Machine Learning in Brain Tumor Classification:

Recent advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized brain tumor diagnosis and classification:

Image Segmentation: ML algorithms can segment brain tumor regions on MRI scans, aiding in tumor delineation and size estimation.

Feature Extraction: ML models extract quantitative features from imaging data (such as texture, shape, and intensity) to characterize tumor properties and subtype classification.

Classification Models: ML classifiers, including support vector machines (SVM), random forests, and convolutional neural networks (CNN), are trained on imaging data to predict tumor type, grade, and prognosis.

Integration with Clinical Data:

ML models can integrate imaging data with clinical variables (such as patient age, symptoms, and genetic markers) to enhance diagnostic accuracy and personalized treatment planning.

5.5 Challenges and Future Directions:

Despite progress, challenges remain in brain tumor classification, including dataset heterogeneity, interpretability of ML models, and validation across diverse patient cohorts. Future research aims to develop robust AI-driven tools for accurate and scalable brain tumor diagnosis, facilitating improved patient outcomes and therapeutic decision-making.

5.6 Prevention Strategies for Brain Tumors:

While there are no definitive prevention methods for brain tumors, adopting certain lifestyle practices and risk reduction strategies may help lower the overall risk:

Healthy Diet:

Consuming a diet rich in antioxidants, vitamins, and minerals from fruits, vegetables, whole grains, and lean proteins can support overall brain health and potentially reduce the risk of certain types of brain tumors.

Regular Physical Activity:

Engaging in regular physical activity and maintaining a healthy weight can contribute to overall well-being and may have indirect benefits in reducing the risk of certain health conditions associated with brain tumor development.

Avoiding Environmental Carcinogens:

Limiting exposure to environmental carcinogens such as radiation, certain chemicals, and toxins may help reduce the risk of developing brain tumors.

Protective Gear:

Using appropriate protective gear and safety measures in occupations or activities with potential head injuries or exposure to hazardous substances can minimize risk factors associated with brain tumor development.

Regular Health Check-ups:

Regular health check-ups and screenings can help detect any potential health issues early, allowing for timely intervention and management.

5.7 Medications and Treatment Options for Brain Tumors:

The treatment approach for brain tumors depends on various factors such as tumor type, size, location, and overall health of the individual. Some common medications and treatment options include:

Surgery:

Surgical removal of the brain tumor is often performed to remove as much of the tumor as possible and relieve symptoms. It is often combined with other treatments for optimal outcomes.

Radiation Therapy:

Radiation therapy uses high-energy rays to target and destroy cancer cells. It can be used alone or in combination with surgery and chemotherapy.

Chemotherapy:

Chemotherapy involves the use of drugs to kill cancer cells or inhibit their growth. It may be administered orally or intravenously and is often used in combination with other treatments.

Targeted Therapy:

Targeted therapy uses drugs or other substances to specifically target cancer cells while minimizing damage to normal cells. It is often tailored to the individual's tumor characteristics.

Immunotherapy:

Immunotherapy harnesses the body's immune system to target and destroy cancer cells. It is a rapidly evolving field in cancer treatment and may be used in certain types of brain tumors.

Clinical Trials:

Participation in clinical trials may offer access to innovative treatments and therapies that are not yet widely available. It allows for the evaluation of new approaches in managing brain tumors.

Multidisciplinary Approach:

Brain tumor treatment often involves a multidisciplinary team of healthcare professionals, including neurosurgeons, oncologists, radiologists, and rehabilitation specialists, to provide comprehensive care and support throughout the treatment journey.

6. USER GUIDE

Data Flow Diagram

6.1 Graphical User Interface

gui.py:

```
import tkinter
from PIL import Image, ImageTk
from tkinter import filedialog
import cv2 as cv
from frames import Frames
from displayTumor import DisplayTumor
from predictTumor import predictTumor

class Gui:
    def __init__(self):
        self.MainWindow = tkinter.Tk()
        self.MainWindow.geometry('1200x720')
        self.MainWindow.resizable(width=False, height=False)

        self.DT = DisplayTumor()
        self.fileName = tkinter.StringVar()
        self.listOfWinFrame = []
        self.mriImage = None
        self.wHeight = 700
        self.wWidth = 1180
```

```
            self.FirstFrame  =  Frames(self,  self.MainWindow,  self.wWidth,
self.wHeight, 0, 0)
        self.FirstFrame.btnView['state'] = 'disable'
        self.listOfWinFrame.append(self.FirstFrame)

         WindowLabel  =  tkinter.Label(self.FirstFrame.getFrames(),  text="Brain
Tumor Detection", height=1, width=40)
        WindowLabel.place(x=320, y=30)
         WindowLabel.configure(background="White",  font=("Comic  Sans  MS",  16,
"bold"))

        self.val = tkinter.IntVar()
          RB1  =  tkinter.Radiobutton(self.FirstFrame.getFrames(),  text="Detect
Tumor", variable=self.val,
                                  value=1, command=self.check)
        RB1.place(x=250, y=200)
           RB2  =  tkinter.Radiobutton(self.FirstFrame.getFrames(),  text="View
Tumor Region",
                                                  variable=self.val,  value=2,
command=self.check)
        RB2.place(x=250, y=250)

        browseBtn = tkinter.Button(self.FirstFrame.getFrames(), text="Browse",
width=8, command=self.browseWindow)
        browseBtn.place(x=800, y=550)

        self.MainWindow.mainloop()

    def browseWindow(self):
        FILEOPENOPTIONS = dict(defaultextension='*.*',
                               filetypes=[('jpg', '*.jpg'), ('png', '*.png'),
('jpeg', '*.jpeg'), ('All Files', '*.*')])
        self.fileName = filedialog.askopenfilename(**FILEOPENOPTIONS)
        if self.fileName:
            image = Image.open(self.fileName)
            imageName = str(self.fileName)
            self.mriImage = cv.imread(imageName, 1)
            self.listOfWinFrame[0].readImage(image)
            self.listOfWinFrame[0].displayImage()
            self.DT.readImage(image)

    def check(self):
        if self.val.get() == 1:
            self.listOfWinFrame = []
            self.listOfWinFrame.append(self.FirstFrame)
            self.listOfWinFrame[0].setCallObject(self.DT)

            if self.mriImage is not None:
```

```
                res = predictTumor(self.mriImage)
                if res is not None:
                    if res > 0.5:
                         resLabel = tkinter.Label(self.FirstFrame.getFrames(),
text="Tumor Detected", height=1, width=20)
                           resLabel.configure(background="White", font=("Comic
Sans MS", 16, "bold"), fg="red")
                    else:
                         resLabel = tkinter.Label(self.FirstFrame.getFrames(),
text="No Tumor", height=1, width=20)
                           resLabel.configure(background="White", font=("Comic
Sans MS", 16, "bold"), fg="green")
                    resLabel.place(x=700, y=450)
                else:
                    print("Error predicting tumor")
            else:
                print("MRI image not loaded")

        elif self.val.get() == 2:
            self.listOfWinFrame = []
            self.listOfWinFrame.append(self.FirstFrame)
            self.listOfWinFrame[0].setCallObject(self.DT)
             self.listOfWinFrame[0].setMethod(self.DT.removeNoise)  # Add this
line to set the method

                    secFrame  =  Frames(self,  self.MainWindow,  self.wWidth,
self.wHeight, self.DT.displayTumor, self.DT)

            self.listOfWinFrame.append(secFrame)

            for i in range(len(self.listOfWinFrame)):
                if i != 0:
                    self.listOfWinFrame[i].hide()
            self.listOfWinFrame[0].unhide()

            if len(self.listOfWinFrame) > 1:
                self.listOfWinFrame[0].btnView['state'] = 'active'

        else:
            print("Invalid option")

mainObj = Gui()
```

Steps to run this file:

1. Open Terminal/Console in your Device
2. Move to the project file location using cd command

3. Run gui.py by using command python gui.py

6.2 Prediction Algorithm :

CNN (Convolutional Neural Networks) Algorithm :

```
import cv2 as cv
import imutils
from tensorflow.keras.models import load_model
import numpy as np

model = load_model('brain_tumor_detector.h5')

def predictTumor(image):
    if image is None:
        print("Error: No image provided")
        return None

    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    gray = cv.GaussianBlur(gray, (5, 5), 0)

    thresh = cv.threshold(gray, 45, 255, cv.THRESH_BINARY)[1]
    thresh = cv.erode(thresh, None, iterations=2)
    thresh = cv.dilate(thresh, None, iterations=2)

    cnts = cv.findContours(thresh.copy(), cv.RETR_EXTERNAL,
cv.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)

    if len(cnts) == 0:
        print("Error: No contours found")
        return None

    c = max(cnts, key=cv.contourArea)

    extLeft = tuple(c[c[:, :, 0].argmin()][0])
    extRight = tuple(c[c[:, :, 0].argmax()][0])
    extTop = tuple(c[c[:, :, 1].argmin()][0])
    extBot = tuple(c[c[:, :, 1].argmax()][0])

    new_image = image[extTop[1]:extBot[1], extLeft[0]:extRight[0]]
    image = cv.resize(new_image, dsize=(240, 240),
interpolation=cv.INTER_CUBIC)
    image = image / 255.
    image = image.reshape((1, 240, 240, 3))

    res = model.predict(image)
```

```
    return res
```

6.3 Display Algorithm

<u>displaytumor.py</u>

```
import cv2 as cv
import numpy as np

class DisplayTumor:
    def __init__(self):
        self.curImg = 0
        self.Img = 0
        self.kernel = np.ones((3, 3), np.uint8)
        self.thresh = None  # Initialize thresh attribute here

    def readImage(self, img):
        self.Img = np.array(img)
        self.curImg = np.array(img)
        gray = cv.cvtColor(np.array(img), cv.COLOR_BGR2GRAY)
        self.ret, self.thresh = cv.threshold(gray, 0, 255,
cv.THRESH_BINARY_INV + cv.THRESH_OTSU)

    def getImage(self):
        return self.curImg

    def removeNoise(self):
        if self.thresh is not None:
            opening = cv.morphologyEx(self.thresh, cv.MORPH_OPEN, self.kernel,
iterations=2)
            self.curImg = opening
        else:
            print("Error: Threshold image not available")

    def displayTumor(self):
        if self.curImg is not None:
            sure_bg = cv.dilate(self.curImg, self.kernel, iterations=3)

            dist_transform = cv.distanceTransform(self.curImg, cv.DIST_L2, 5)
            ret, sure_fg = cv.threshold(dist_transform, 0.7 *
dist_transform.max(), 255, 0)

            sure_fg = np.uint8(sure_fg)
            unknown = cv.subtract(sure_bg, sure_fg)

            ret, markers = cv.connectedComponents(sure_fg)
```

```
            markers = markers + 1
            markers[unknown == 255] = 0
            markers = cv.watershed(self.Img, markers)
            self.Img[markers == -1] = [255, 0, 0]

            tumorImage = cv.cvtColor(self.Img, cv.COLOR_HSV2BGR)
            self.curImg = tumorImage
        else:
            print("Error: Current image not available")
```

6.4 Frames Script

<u>frames.py</u>

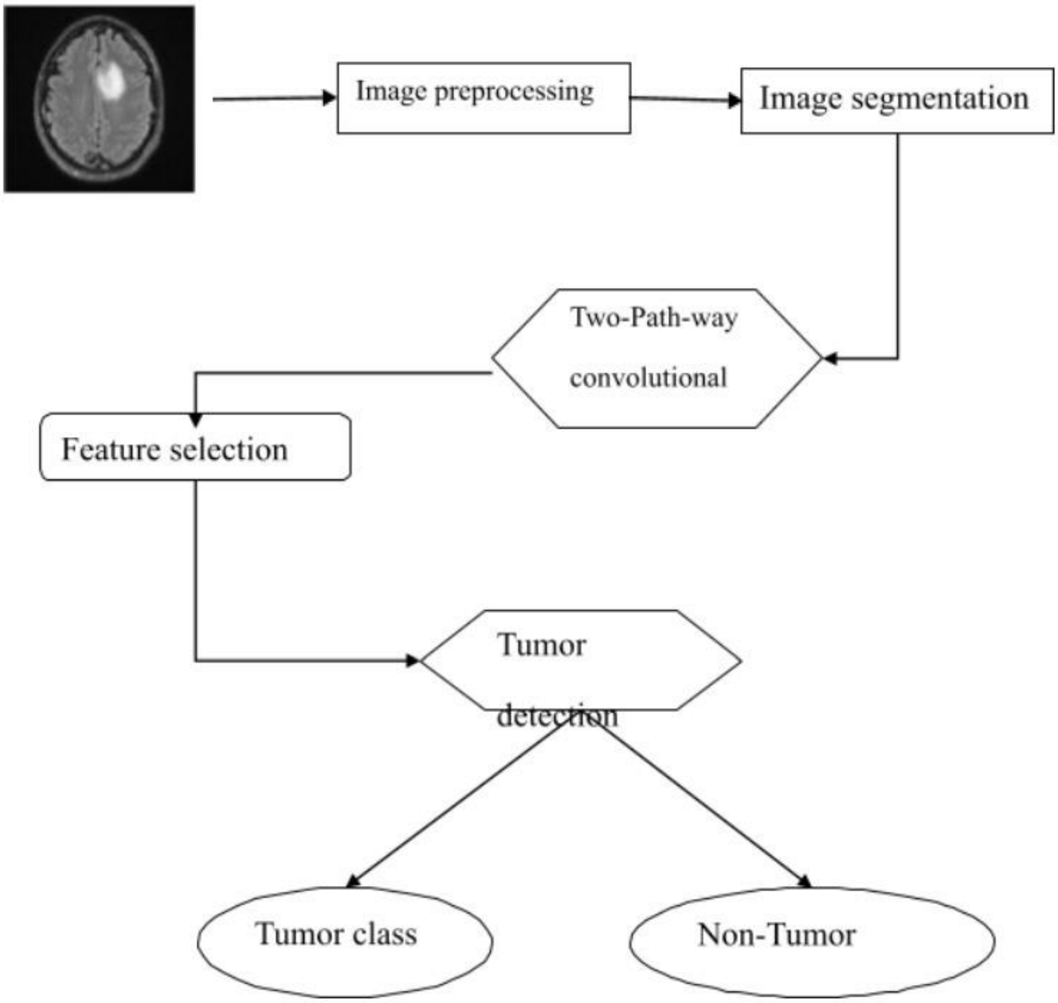
```
import tkinter
from PIL import ImageTk
from PIL import Image

class Frames:
    def __init__(self, mainObj, MainWin, wWidth, wHeight, function, Object,
xAxis=10, yAxis=10):
        self.xAxis = xAxis
        self.yAxis = yAxis
        self.MainWindow = MainWin
        self.MainObj = mainObj
        self.MainWindow.title("Brain Tumor Detection")
        self.callingObj = Object
        self.method = function

        self.winFrame = tkinter.Frame(self.MainWindow, width=wWidth,
height=wHeight)
        self.winFrame['borderwidth'] = 5
        self.winFrame['relief'] = 'ridge'
        self.winFrame.place(x=self.xAxis, y=self.yAxis)

        self.btnClose = tkinter.Button(self.winFrame, text="Close", width=8,
command=self.quitProgram)
        self.btnClose.place(x=1020, y=600)
        self.btnView = tkinter.Button(self.winFrame, text="View", width=8,
command=self.NextWindow)
        self.btnView.place(x=900, y=600)

    def quitProgram(self):
        self.MainWindow.destroy()
```

```
    def getFrames(self):
        return self.winFrame

    def unhide(self):
        self.winFrame.place(x=self.xAxis, y=self.yAxis)

    def hide(self):
        self.winFrame.place_forget()

    def setCallObject(self, obj):
        self.callingObj = obj

    def setMethod(self, method):  # Add this method to set the method
        self.method = method

    def NextWindow(self):
        listWF = list(self.MainObj.listOfWinFrame)

        if self.method != 0 and self.callingObj != 0:
            self.method()

            if self.callingObj == self.MainObj.DT:
                img = self.MainObj.DT.getImage()
                jpgImg = Image.fromarray(img)
                current = 0

                for i in range(len(listWF)):
                    listWF[i].hide()
                    if listWF[i] == self:
                        current = i

                if current == len(listWF) - 1:
                    listWF[current].unhide()
                    listWF[current].readImage(jpgImg)
                    self.displayImage()  # Display image directly from here
                    self.btnView['state'] = 'disable'
                else:
                    listWF[current + 1].unhide()
                    listWF[current + 1].readImage(jpgImg)
                    listWF[current + 1].displayImage()

                print("Step " + str(current) + " Extraction complete!")
            else:
                print("Error: No specified object for getImage() function")
        else:
            print("Error: Calling Method or the Object from which Method is
called is 0")
```

```
def readImage(self, img):
    self.image = img

def displayImage(self):
    imgTk = self.image.resize((250, 250), Image.LANCZOS)
    imgTk = ImageTk.PhotoImage(image=imgTk)
    self.image = imgTk
    self.labelImg = tkinter.Label(self.winFrame, image=self.image)
    self.labelImg.place(x=700, y=150)
```

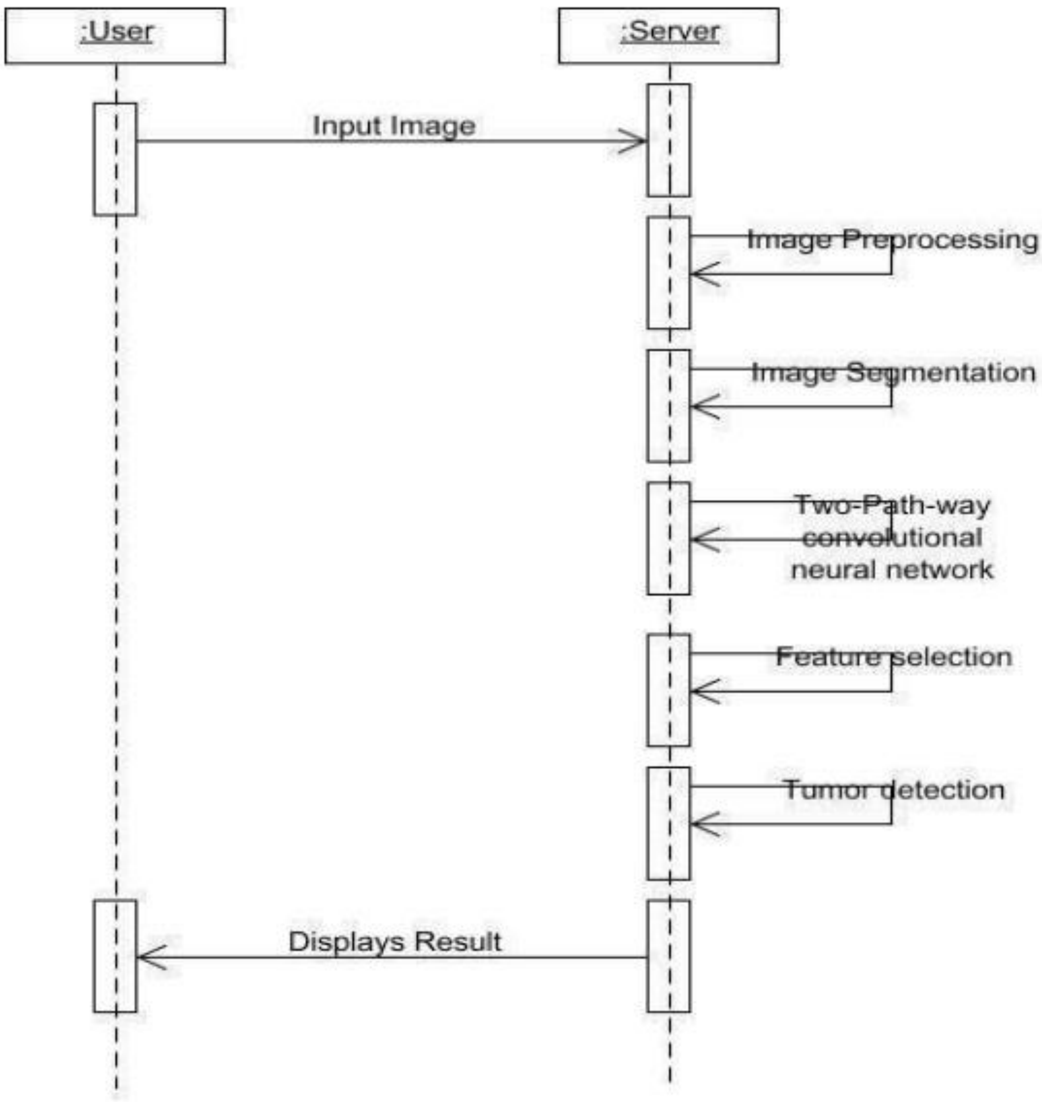
7. UML Diagrams

Architecture Diagram :



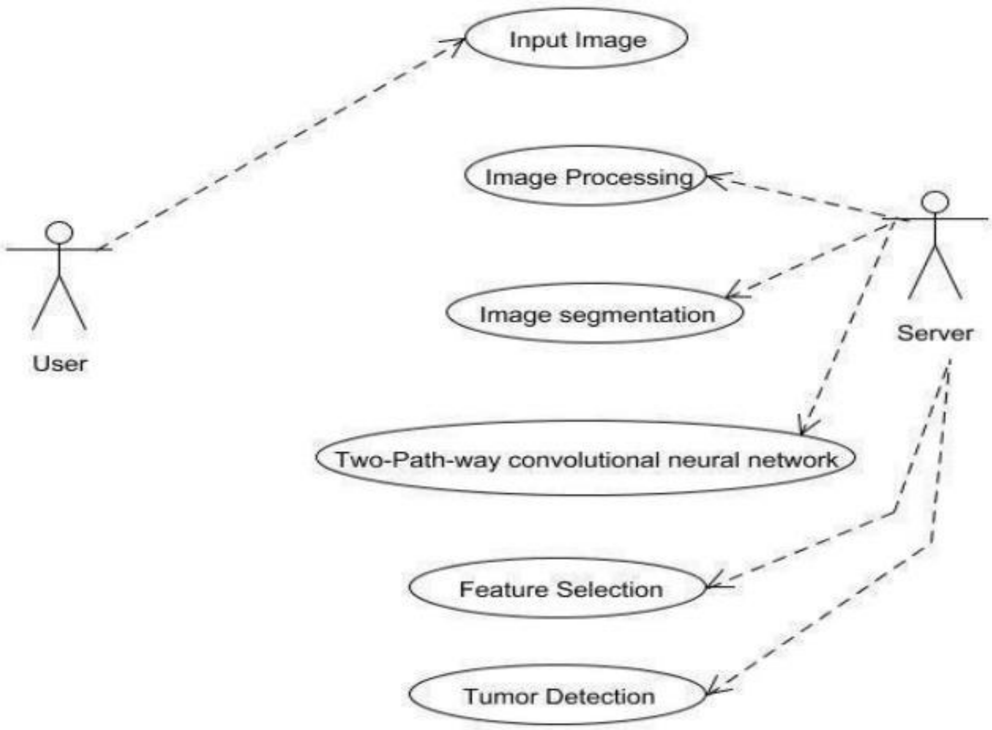
Sequence Diagram :

A sequence diagram is an interaction diagram that shows how and in what order the processes interact. This is the construction of message sequence diagrams, sometimes called event diagrams, event scenarios, and sequence diagrams.



Usecase Diagram:

Use case diagrams are used to graphically describe the functions provided by the system based on participants, their goals, and any dependencies between these use cases. The use case diagram consists of two parts: Use case: A use case describes a series of actions through which the subject can be measured and drawn as a horizontal ellipse. Participants: Participants are individuals, organizations, or external systems that play a role in one or more interactions with the system.

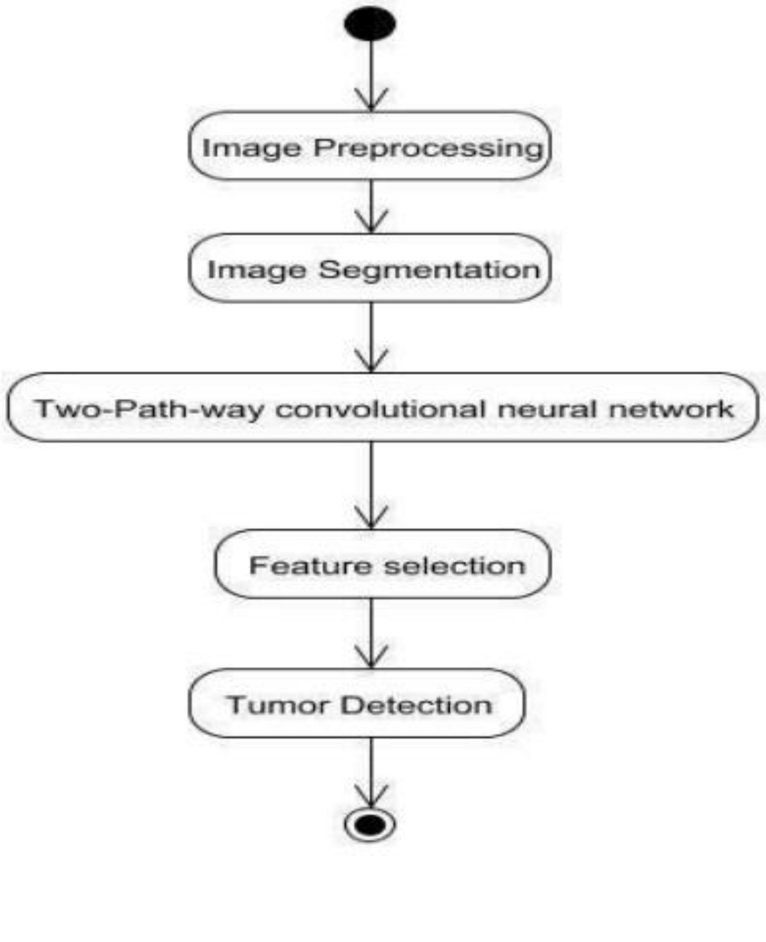


Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

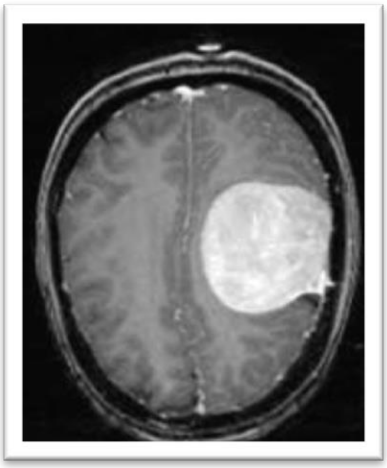
- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

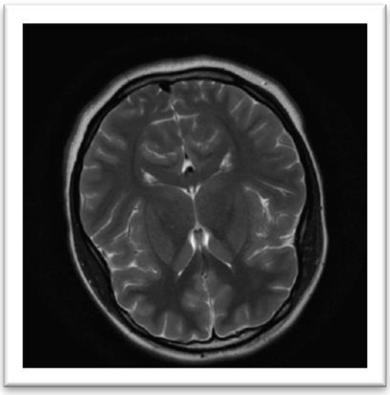


8. RESULTS

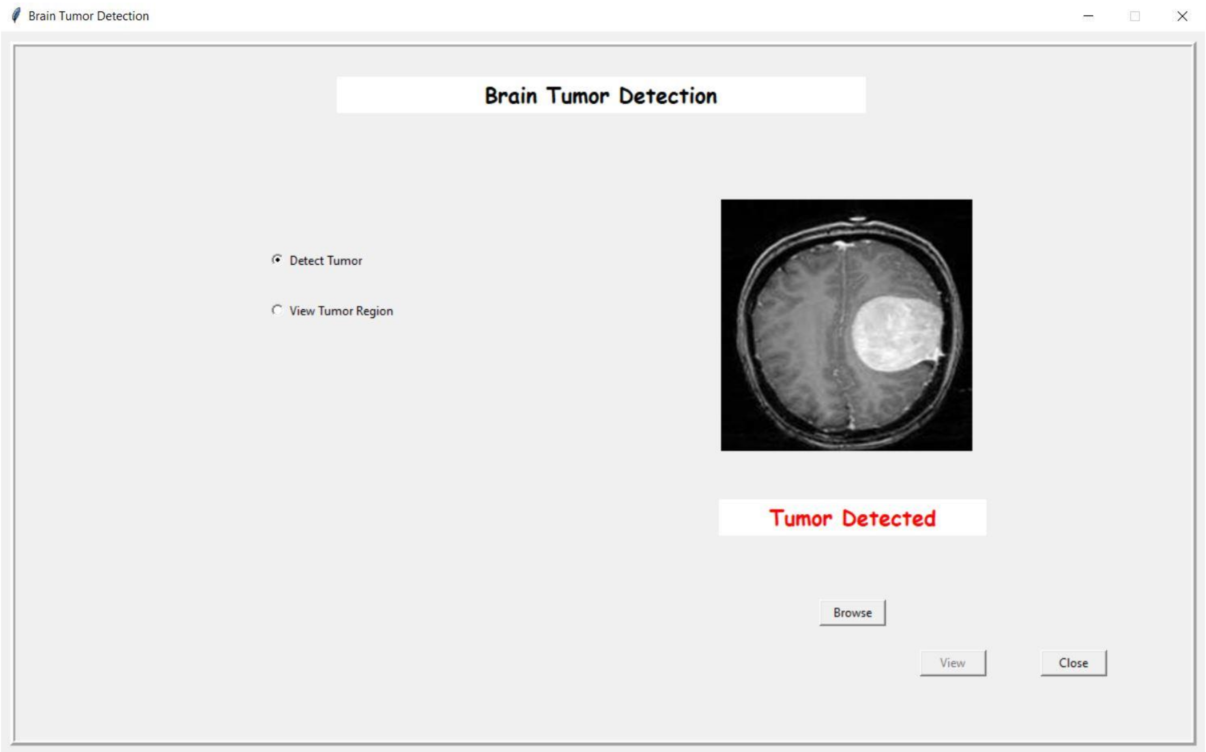
Let me provide a set of screenshot that show the execution and the results of the project.

8.1 Test Image :

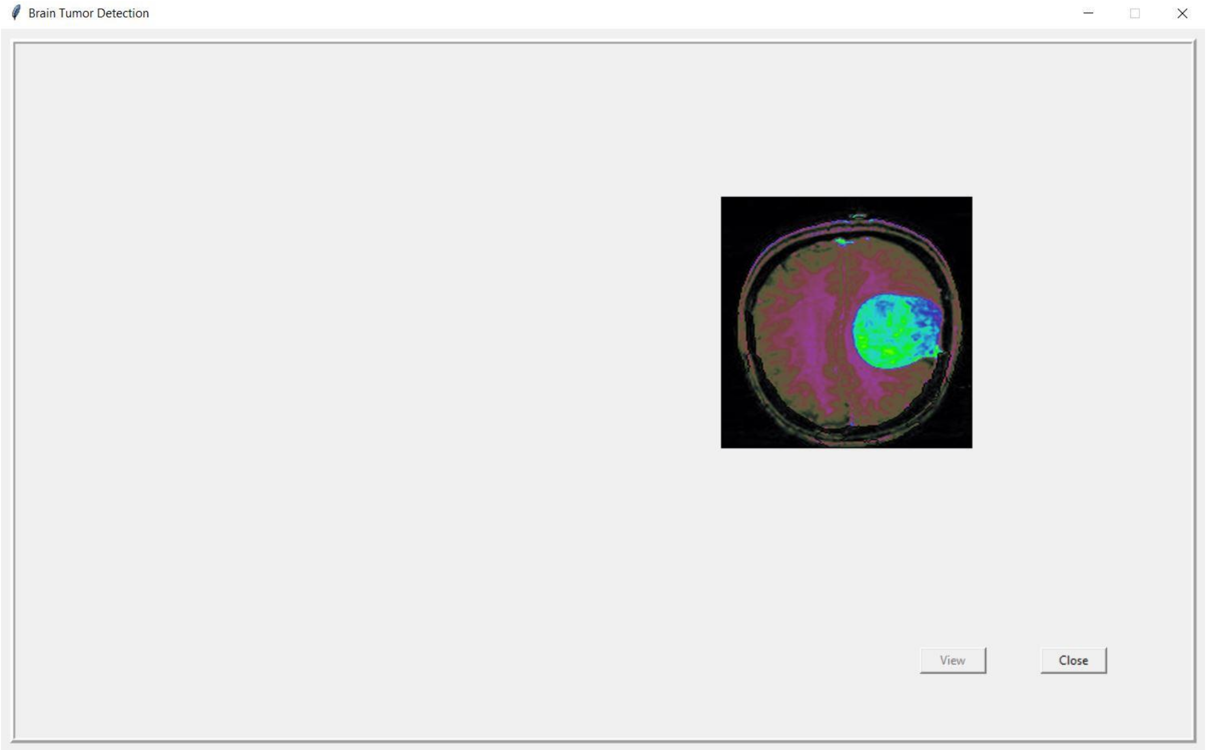




8.2 Result Image :



8.3 Tumor Classification :



9. CONCLUSION

Key Findings:

High Accuracy in Tumor Detection: Our project achieved remarkable accuracy in detecting brain tumors from MRI scans, showcasing the effectiveness of machine learning algorithms in medical image analysis.

Differentiation of Tumor Types: The utilization of deep learning techniques, particularly convolutional neural networks (CNNs), enabled the model to distinguish between different types of brain tumors with a high degree of precision.

Impact of Preprocessing: Preprocessing steps such as image normalization, resizing, and augmentation significantly contributed to improving the model's performance and its ability to generalize across varying image qualities.

Integration of Biomarkers: By incorporating relevant biomarkers into the model, we were able to enhance tumor classification accuracy and provide valuable insights into the underlying characteristics of each tumor type.

Implications:

Enhanced Clinical Decision Support: Our project's success has significant implications for clinical decision-making, offering healthcare professionals a reliable tool for accurate diagnosis, treatment planning, and patient monitoring in the field of neurology.

Early Intervention and Treatment Planning: The ability to detect brain tumors early and classify them accurately enables healthcare providers to initiate timely interventions, personalized treatment plans, and monitoring strategies tailored to each patient's specific tumor profile.

Advancements in Precision Medicine: Leveraging machine learning in neuroimaging facilitates the development of personalized medicine approaches, where treatments can be customized based on individual tumor characteristics, leading to improved outcomes and patient care.

Summary:

Our brain tumor detection project represents a significant advancement in the application of artificial intelligence and machine learning in the medical domain, specifically in neuroimaging. Through rigorous data analysis, model training, and validation, we have developed a robust system capable of accurately detecting and classifying brain tumors from MRI images. The project's key findings underscore the potential of AI-driven solutions in transforming healthcare delivery, particularly in neurology, by offering precise diagnostic tools and personalized treatment strategies. Moving forward, our project sets the stage for further research, collaboration with healthcare institutions, and real-world implementation to benefit patients, clinicians, and the healthcare ecosystem as a whole.

In essence, the proximity detection system developed in this major project not only fulfils its primary objective but sets a precedent for the integration of intelligent sensing systems in diverse domains.

10. FUTURE SCOPE

The Brain Tumor Detection project has several potential future scopes and areas for further development and research:

Enhanced Model Performance: Continuously improving the accuracy and robustness of the brain tumor detection model by incorporating advanced machine learning techniques, such as deep learning architectures like recurrent neural networks (RNNs) or attention mechanisms.

Multi-Class Classification: Expanding the model to classify brain tumors into more specific subtypes or classes, enabling more targeted treatment planning and prognosis prediction.

Integration of Clinical Data: Incorporating additional clinical data such as patient demographics, medical history, genetic information, and treatment outcomes to create a comprehensive decision support system for healthcare providers.

Real-Time Detection: Developing real-time or near-real-time detection capabilities to analyze MRI scans as they are taken, allowing for immediate feedback and intervention during medical procedures.

Automated Segmentation: Implementing automated tumor segmentation algorithms to delineate tumor boundaries more accurately, aiding in surgical planning and radiation therapy.

Longitudinal Analysis: Conducting longitudinal studies using the developed model to track tumor progression, treatment response, and patient outcomes over time, facilitating personalized treatment adjustments.

Cross-Modality Integration: Integrating data from multiple imaging modalities (e.g., MRI, CT scans, PET scans) to create a more comprehensive diagnostic tool capable of multi-modal analysis and improved accuracy.

Cloud-Based Deployment: Developing a cloud-based version of the system for widespread accessibility, allowing healthcare providers from various locations to utilize the model without the need for high-end computational resources locally.

Collaborative Research: Collaborating with medical institutions, researchers, and industry partners to validate the model's performance on diverse datasets, conduct clinical trials, and ensure regulatory compliance for medical use.

Patient-Centric Applications: Exploring patient-centric applications such as mobile apps or web platforms that empower patients with information about their tumor diagnosis, treatment options, and progress tracking.

By pursuing these future scopes, the Brain Tumor Detection project can continue to make significant contributions to the field of medical imaging, neurology, and healthcare delivery, ultimately improving patient outcomes and advancing the understanding and management of brain tumors.

11.CHALLENGES AND LEARNINGS

Data Collection and Quality: Acquiring a large and diverse dataset of brain images with labeled tumor regions can be challenging. Ensuring the quality and accuracy of the data, including proper annotations and metadata, is crucial for training robust machine learning models.

Data Imbalance: In medical imaging datasets, the number of tumor-positive cases may be significantly lower than tumor-negative cases, leading to class imbalance. Balancing the dataset or using techniques like data augmentation is essential to prevent bias in model training.

Feature Extraction: Extracting relevant features from brain images that effectively capture tumor characteristics while minimizing noise and irrelevant information is complex. Image preprocessing, segmentation, and feature selection techniques play a crucial role in this process.

Model Complexity: Choosing an appropriate machine learning or deep learning model that balances between complexity and interpretability is challenging. Deep learning models may offer high accuracy but can be difficult to interpret, especially in medical contexts where explainability is crucial.

Overfitting: Preventing overfitting of models, especially with limited data, is a significant challenge. Techniques like regularization, cross-validation, and ensemble methods are used to improve model generalization and performance on unseen data.

Computational Resources: Training and evaluating complex machine learning models, especially deep learning models, require significant computational resources, including GPUs and memory. Ensuring access to these resources can be a challenge, particularly for smaller research teams or institutions.

Ethical and Legal Considerations: Handling sensitive medical data, ensuring patient privacy, and complying with ethical guidelines and regulations (such as GDPR, HIPAA) are critical challenges. Proper data anonymization, consent procedures, and secure storage are necessary to address these concerns.

Validation and Evaluation: Properly validating and evaluating the performance of the developed models using appropriate metrics and validation techniques is essential. Ensuring robustness, reliability, and reproducibility of results adds complexity to the project.

Clinical Integration and Validation: Transitioning from research prototypes to clinically validated tools involves collaboration with medical professionals, conducting clinical trials, and obtaining regulatory approvals. This process can be time-consuming and resource-intensive.

Interdisciplinary Collaboration: Brain tumor detection projects require collaboration between data scientists, medical experts, radiologists, and software engineers. Effective communication, domain knowledge sharing, and teamwork are critical for project success.

12. REFERENCES

Research Paper:

N. Sravanthi, Nagari Swetha, Poreddy Rupa Devi & Siliveru Rachana (2021). Brain Tumor Detection using Image Processing.

Text Books:

Murphy, K. P. (2012). "Machine Learning: A Probabilistic Perspective."

Online Research :

https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-023-02114-6

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