**Brain Tumor Detection System**

**TECHNICAL REPORT**

 **SUBMITTED BY**

Iman Fatima: 2021-ag-8053

**ADVISED BY**

**Dr.Kareem Ullah**

**A TECHNICAL REPORT SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENT FOR THE DEGREE OF BACHELOR’S OF SCIENCE**

*IN SOFTWARE ENGINEERING*

**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF SCIENCES**

**UNIVERSITY OF AGRICULTURE FAISALABAD**

**DECLARATION**

I hereby declare that the contents of the report “**Brain Tumor Detection System**” are project of my own research and no part has been copied from any published source (except the references). I further declare that this work has not been submitted for award of any other diploma/degree. The university may take action if the information provided is found false at any stage. In case of any default the scholar will be proceeded against as per UAF policy.

**CERTIFICATE**

To,

The Controller of Examinations,

University of Agriculture,

Faisalabad.

The supervisory committee certify that Iman Fatima **2021-ag-8053** has successfully completed his project in partial fulfillment of requirement for the degree of BS Software Engineeringunder our guidance and supervision.

**Dr.Kareem Ullah**

Supervisor

Iman Fatima

Member

Dr. Muhammad Ahsan Latif

Incharge,

Department of Computer Science

**ACKNOWLEDGEMENT**

I thank all who in one way or another contributed in the completion of this report. First, I thank to ALLAH ALMIGHTY, most magnificent and most merciful, for all his blessings. Then I am so grateful to the Department of Computer Science for making it possible for me to study here. My special and heartily thanks to my supervisor, **Dr.Kareem Ullah** who encouraged and directed me. His challenges brought this work towards a completion. It is with his supervision that this work came into existence. For any faults I take full responsibility. I am also deeply thankful to my informants. I want to acknowledge and appreciate their help and transparency during my research. I am also so thankful to my fellow students whose challenges and productive critics have provided new ideas to the work. Furthermore, I also thank my family who encouraged me and prayed for me throughout the time of my research. May the Almighty God richly bless all of you

**ABSTRACT**

An Abstract is a summary of the whole technical report. it’s also the last thing you will write but comes first in the document.The Abstract tells the reader the main points about your technical project. Readers may not have a technical background. The Abstract gives them an overview and can help them decide which specific sections to focus on. An academic abstract typically outlines four elements relevant to the completed work: The development/research focus (i.e. statement of the problem(s)/research issue(s) addressed), the methodology used, the results/findings of the work done and the main conclusions and recommendations. Write at least 200 words as abstract of your report.

**Table of Contents**

[Chapter 1 - INTRODUCTION 1](#_Toc11644)

[1.1 Background: 1](#_Toc12193)

[1.7 Requirements: 6](#_Toc1552)

[1.8 Stakeholders: 10](#_Toc7579)

[Chapter 2 – MATERIALS & METHODS 13](#_Toc12836)

[2.1 Process Model: 13](#_Toc8188)

[2.2 Tools & Technologies 14](#_Toc23890)

[2.3 System Architecture 15](#_Toc13604)

[2.4 User-Centric Design 17](#_Toc20703)

[2.5 System Work Flow 18](#_Toc23563)

[2.6 Data Flow Diagram (DFD) 19](#_Toc1952)

[Install Dependencies: 21](#_Toc9474)

[Download YOLOv8 Pretrained Model: 22](#_Toc5680)

[Chapter 3 - RESULTS & DISCUSSION 24](#_Toc10649)

[Chapter 4 – User Manual 29](#_Toc12721)

Table of Figures

[Figure 1: Agile Activities 21](#Figure!0|sequence)

[Figure 2: Use Case Diagram 25](#Figure!1|sequence)

[Figure 3: System architecture 27](#Figure!2|sequence)

[Figure 4: DFD Level 0 29](#Figure!3|sequence)

[Figure 5: DFD Level 1 30](#Figure!4|sequence)

# Chapter 1 - INTRODUCTION

## Background:

Brain tumors are among the most critical health conditions that can lead to severe neurological damage and death if not detected and treated at an early stage. Early detection significantly increases the chances of successful treatment and survival. Traditionally, brain tumor detection relies heavily on manual examination of MRI scans by experienced radiologists, which is time-consuming, prone to human error, and requires specialized expertise. As the number of patients grows and healthcare demands increase, there is a pressing need for faster, more accurate, and automated solutions to support medical professionals.

In response to this need, the proposed project aims to develop an automated Brain Tumor Detection System using the YOLO (You Only Look Once) deep learning model. YOLO is a state-of-the-art, real-time object detection algorithm that has proven highly effective across a wide range of domains due to its speed and accuracy. By training YOLO on annotated MRI images, the system can quickly and accurately identify and localize tumors within brain scans, assisting doctors in making faster and more informed decisions.

The implementation of this system addresses several critical needs: reducing the burden on healthcare providers, minimizing diagnostic errors, and enabling early intervention for better patient outcomes. Moreover, it opens possibilities for telemedicine and remote diagnosis, making quality healthcare more accessible even in under-resourced areas. This project sets the foundation for integrating artificial intelligence into critical healthcare diagnostics, ultimately enhancing the quality and efficiency of medical services.

**1.2 Description**

The Brain Tumor Detection System using YOLO is an advanced deep learning-based application designed to automate the identification and localization of brain tumors in MRI images. The system leverages the YOLO (You Only Look Once) algorithm, a highly efficient object detection model known for its speed and accuracy, to analyze medical imaging data. The core functionality involves training the YOLO model on a labeled dataset of MRI scans where tumor regions are annotated. Once trained, the system can rapidly process new MRI images and predict the presence, type, and precise location of tumors.

The system operates in two main stages: preprocessing and detection. In the preprocessing stage, MRI images are prepared through resizing, normalization, and augmentation to optimize model performance. In the detection stage, the YOLO model processes the images in a single forward pass, instantly predicting bounding boxes and tumor probabilities. This approach drastically reduces detection time compared to traditional methods, enabling real-time analysis.

Additionally, the system is designed with a user-friendly interface where healthcare professionals can upload MRI images and view results with highlighted tumor areas. The project aims to improve diagnostic accuracy, accelerate the decision-making process, and provide critical support in environments where radiology expertise may be limited. Overall, the Brain Tumor Detection System offers a practical, scalable solution to enhance brain tumor diagnosis using modern artificial intelligence technologies.

### 1.3 Problem Statement

Brain tumors are a serious health threat that require timely and accurate diagnosis to improve patient outcomes. However, traditional methods of detecting brain tumors from MRI scans are highly dependent on manual inspection by radiologists, which can be time-consuming, subjective, and prone to human error. The growing number of cases, combined with the shortage of skilled radiologists, often leads to delays in diagnosis and treatment, putting patients' lives at greater risk.

Furthermore, manual detection techniques may miss small or subtle tumors, especially in early stages when intervention could be most effective. In rural and underdeveloped regions, access to expert radiologists is even more limited, making early detection extremely challenging. Existing automated solutions either lack sufficient accuracy or are too slow for practical clinical use.

To address these challenges, there is a critical need for a fast, reliable, and highly accurate system that can assist in the automatic detection and localization of brain tumors from MRI images. The proposed Brain Tumor Detection System using YOLOv8 — the latest and most advanced version of the YOLO series — aims to fill this gap. YOLOv8 offers superior accuracy, faster detection speeds, and better generalization capabilities compared to earlier models, making it highly suitable for medical image analysis. This system is intended to support healthcare professionals in making quicker, more precise diagnoses, ultimately leading to better patient care and reduced diagnostic errors.

### 1.4 Scope

The Brain Tumor Detection System using YOLOv8 focuses on developing a deep learning-based solution capable of automatically detecting and localizing brain tumors in MRI images. The system is designed to assist radiologists and healthcare professionals by providing quick, accurate, and consistent tumor detection, thereby improving the efficiency and reliability of the diagnostic process.

The project scope includes the collection and preprocessing of MRI image datasets, the annotation of tumor regions, and the training and fine-tuning of the YOLOv8 model for optimal performance. The system will be capable of handling different types of brain tumors, including gliomas, meningiomas, and pituitary tumors, provided sufficient labeled data is available for each class. The final model will offer a user-friendly interface where medical practitioners can upload MRI scans and receive real-time detection results with highlighted tumor areas.

However, the system will be limited to analyzing 2D MRI images only and will not include diagnosis of tumor severity or treatment recommendations. The initial deployment is intended for research and diagnostic assistance purposes and will require validation by medical experts before clinical use. Expansion to 3D imaging analysis and integration into full medical workflows may be considered for future work.

By focusing on fast and accurate tumor localization, the project aims to make a meaningful contribution toward modernizing medical diagnostics with AI-powered solutions.

### 1.5 Objectives

**Goal:**  
Develop an efficient, accurate, and real-time Brain Tumor Detection System using YOLOv8 to assist healthcare professionals in the early identification and localization of brain tumors from MRI images.

**Objectives:**

* Collect and prepare a comprehensive dataset of annotated brain MRI images for model training and evaluation.
* Preprocess and augment the MRI data to enhance model performance and generalization.
* Design, train, and optimize a YOLOv8-based deep learning model specifically for detecting and localizing brain tumors.
* Develop a user-friendly interface for uploading MRI images and displaying detection results.
* Validate and evaluate the system’s accuracy, speed, and reliability against existing methods and datasets.
* Ensure that the system provides real-time detection with minimal false positives and false negatives.
* Document the system’s performance and prepare the foundation for future enhancements, such as extending to 3D MRI scans or tumor classification.

### 1.6 Feasibility

A feasibility study has been conducted to assess whether the development and deployment of the Brain Tumor Detection System using YOLOv8 is practical, achievable, and beneficial under current conditions. Below are the different aspects of the feasibility analysis:

#### 1.6.1 Technical Feasibility

The technical resources required for this project are readily available. Deep learning

frameworks such as PyTorch and TensorFlow support the development of YOLOv8 models. The team has access to high-performance GPUs for model training and testing, and the knowledge and expertise in machine learning, medical imaging, and software development necessary to complete the project are present. Therefore, the technical feasibility is considered high.

#### 1.6.2 Schedule Feasibility

Considering the project timeline, the tasks such as data collection, model training, interface development, and validation can be completed within a reasonable period. Given proper project management and division of tasks, the detection system can be developed and tested within the scheduled timeframe.

#### 1.6.3 Economic Feasibility

The project is economically feasible. The main costs involve acquiring a reliable dataset, computational resources for training, and minimal software licensing (mostly open-source tools are used). Compared to the potential benefit of improving diagnostic accuracy and healthcare efficiency, the investment is justified and manageable within the available budget.

#### 1.6.4 Cultural Feasibility

The Brain Tumor Detection System will have a positive cultural impact by contributing to better healthcare services. It will support the culture of adopting AI and technological innovations in the medical field. There are no adverse cultural or environmental implications associated with this project.

#### 1.6.5 Legal/Ethical Feasibility

All patient data used will be anonymized to protect patient privacy and comply with medical data protection regulations such as HIPAA or GDPR. Ethical AI practices will be followed to ensure fairness and transparency. The project poses no legal barriers if these protocols are strictly observed.

#### 1.6.6 Resource Feasibility

The necessary resources, including MRI image datasets, computing infrastructure (servers, GPUs), and skilled human resources (AI engineers, radiologists for validation), are available. If additional resources are needed, they are accessible through partnerships or institutional support.

#### 1.6.7 Operational Feasibility

The system will be easy to integrate into existing medical imaging workflows. Healthcare professionals can operate it with minimal training, and the system will provide results quickly and accurately. Operational risks are minimal and manageable, ensuring the project can adapt to challenges as they arise.

## Requirements:

### 1.7.1 Functional Requirements

### MRI Image Upload:

Users should be able to upload brain MRI images for tumor detection.

### Tumor Detection:

The system should predict the presence and location of brain tumors using the trained YOLOv8 model.

### Annotated Output:

After prediction, the system should display the MRI image with bounding boxes highlighting detected tumor regions.

### Real-Time Inference:

The detection process should be fast enough to allow near real-time analysis after image submission.

### Gradio Interface:

A clean, user-friendly frontend should be provided using Gradio, with clear fields for uploading images and viewing results.

### Image Visualization:

Users should see the annotated MRI image output (with tumor areas marked) directly on the webpage.

### Sharing/Deployment:

The Gradio app should support public sharing via a URL (using share=True), allowing remote access for demonstrations or use.

### 1.7.2 Non- Functional Requirements

### 1. Performance Requirements

The system must process and return tumor detection results within **5 seconds** after image upload.

The YOLOv8 model should maintain a minimum detection accuracy of **90%** on the validation dataset.

### 2. Usability Requirements

The user interface must be intuitive, requiring **no specialized training** to operate.

Clear instructions should be provided for uploading images and interpreting prediction results.The system should be **accessible via a simple web browser** without requiring any additional installation.

### 3. Reliability Requirements

The system should have an uptime of **at least 99%** during operational hours.

Model predictions should be **consistent** for the same MRI input image across different sessions.

### 4. Scalability Requirements

The system should be designed to handle **multiple concurrent users** without significant performance degradation.

The backend can be scaled up by deploying the YOLOv8 model on cloud platforms (e.g., AWS, GCP) if user demand increases.

### 5. Security Requirements

MRI images uploaded by users must be **processed securely** and **not stored** permanently to protect patient privacy.

The web application should be protected against common threats like **injection attacks** and **unauthorized access**.

### 6. Portability Requirements

The system should be **platform-independent**, accessible from Windows, Mac, and mobile devices (browser-based).

The codebase (Gradio app) should be deployable on both **local machines** and **cloud servers**.

### 7. Maintainability Requirements

The model and the web application should be **modular**, making it easy to update the model version (e.g., upgrading to YOLOv9 later).

Clear documentation should be provided for setup, model retraining, and system maintenance.

### 8. Legal and Ethical Requirements

Ensure that all MRI images used for model training and testing are **properly anonymize**.Users must be **informed** that the system is an **assistive tool**, not a replacement for professional medical diagnosis.

### 1.7.3 Hardware Requirements

### 1. Server Requirements

**Processor**: Multi-core Intel® Xeon® or AMD EPYC® CPU (e.g., Intel Core i7 or better)

**RAM**: Minimum 16 GB or more (32 GB recommended for handling multiple user requests)

**Storage**: At least 100 GB of available disk space for storing model weights, logs, and user-submitted MRI images

**Graphics**: NVIDIA GPU with at least 8GB VRAM (e.g., RTX 2080 or higher) to accelerate YOLOv8 model inference. If no GPU is available, CPU inference is possible but slower.

**Network**: High-speed internet connection to handle large image uploads and quick response times

### 2. Computational Resources for Model Inference

**CPU**: Intel® Core™ i7 or AMD Ryzen 7 or better for fast processing in the absence of a GPU.

**GPU**: A dedicated GPU like NVIDIA Tesla or RTX series for faster processing during inference (recommended for low latency).

**Memory**: At least 16 GB of RAM for handling inference and processing large images efficiently.

**Load Balancing**: For scaling the server to handle multiple concurrent users, consider load balancing techniques.

### 1.7.4 Software Requirements

To run the Brain Tumor Detection System using YOLOv8, the following software components are required:

### Operating System

**Windows**: Windows 10 or later

**Linux**: Ubuntu 18.04 or later (preferred for performance)

**macOS**: macOS 10.14 (Mojave) or later

### Python Version

**Python 3.7** or later (recommended: Python 3.8 or higher)

### Libraries and Frameworks

**PyTorch**: Version 1.7 or higher (to run the YOLOv8 model)

**Ultralytics YOLOv8**: For object detection using the YOLO model.

Install via pip install ultralytics

**OpenCV**: Version 4.5 or higher for image processing and visualization.

Install via pip install opencv-python

**NumPy**: Version 1.19 or higher for array handling and operations.

Install via pip install numpy

**Gradio**: For building the web interface.

Install via pip install gradio

**Flask** (optional): For backend server deployment if you're hosting the application in a web environment.

Install via pip install flask

### Web Browser

**Google Chrome** or **Mozilla Firefox** (for accessing the Gradio interface and uploading MRI images)

### Other Tools

**Git**: Version control for managing the codebase (optional but recommended).

Install via https://git-scm.com/

### Cloud Services (optional)

**AWS**, **Google Cloud**, or **Azure** for cloud-based deployment and scaling.

**Docker** (optional): For containerizing the application to ensure smooth deployment across environments.

## Stakeholders:

Stakeholders are individuals or organizations that are actively involved in the project, or whose interests may be affected by the project’s execution or success.

The key stakeholders for the **Brain Tumor Detection System using YOLOv8** are:

### 1. Medical Professionals

**Doctors**, **Radiologists**, and **Neurosurgeons** who can use the system as an assistive tool to help detect brain tumors in MRI scans.They depend on the system for faster preliminary diagnosis to complement manual assessments.

### 2. Patients

Individuals undergoing brain MRI scans who may indirectly benefit from quicker, more accurate detection of potential tumors.

### 3. Researchers and Scientists

Medical researchers and AI scientists working in the fields of medical imaging, machine learning, and healthcare innovation who can use the system for data analysis and further research on brain tumors.

### 4. Project Development Team

**Developers**, **Data Scientists**, **AI Engineers**, and **UI/UX Designers** responsible for building, training, and deploying the system.

### 5. Healthcare Institutions

Hospitals, diagnostic centers, and clinics that may integrate the system into their workflows to enhance diagnostic services and reduce workload on radiology departments.

### 6. IT Support Teams

Technical personnel responsible for maintaining the hardware and software environments where the system is hosted, ensuring security, uptime, and performance.

### Regulatory Authorities

Healthcare regulatory bodies ensuring that the system complies with medical device

regulations, patient privacy laws (such as HIPAA, GDPR), and ethical standards.

### 8. End-Users (General Public)

Researchers, students, or general users accessing the tool via public Gradio links or web platforms for educational or informational purposes.

# Chapter 2 – MATERIALS & METHODS

## 2.1 Process Model:

For the development of the **Brain Tumor Detection System using YOLOv8**, we have selected the **Agile Process Model**, particularly following the principles of **Rapid Application Development (RAD)**.

Agile is an incremental model where the software is developed in small, functional parts that are quickly delivered and refined based on user feedback. Each component is developed as a mini-project, allowing multiple parts of the system (like image processing, model training, and frontend development) to be worked on simultaneously. This parallel development speeds up the creation of working prototypes, making it easier for stakeholders to see, test, and suggest improvements early.

**Requirements**: Initial gathering of user needs, such as uploading MRI images, detecting tumors, and providing clear predictions.

**Design**: Designing the YOLOv8 model configuration, frontend UI (using Gradio), and how the system will operate.

**Implementation**: Actual coding of the detection model, integration with Gradio interface, and backend logic.

**Testing**: After each implementation cycle, the system is tested for accuracy, performance, and usability.

**Evolution**: Based on the feedback and testing results, necessary improvements are made, and the system evolves with each iteration.

This iterative loop allows for **continuous improvement** and **faster delivery** of functional features, which is crucial for AI-driven medical systems where accuracy and usability are highly sensitive factors.

**Significance for the Project:**

Early delivery of a working detection system for feedback.

Quick incorporation of improvements based on real-world MRI data.

Flexibility to adjust the model's performance and UI based on doctor or patient feedback.

Ensures the system is reliable, accurate, and easy to use before final deployment.

Thus, the Agile model, represented in the figure, aligns very well with the goals of building an accurate, efficient, and user-friendly Brain Tumor Detection System.

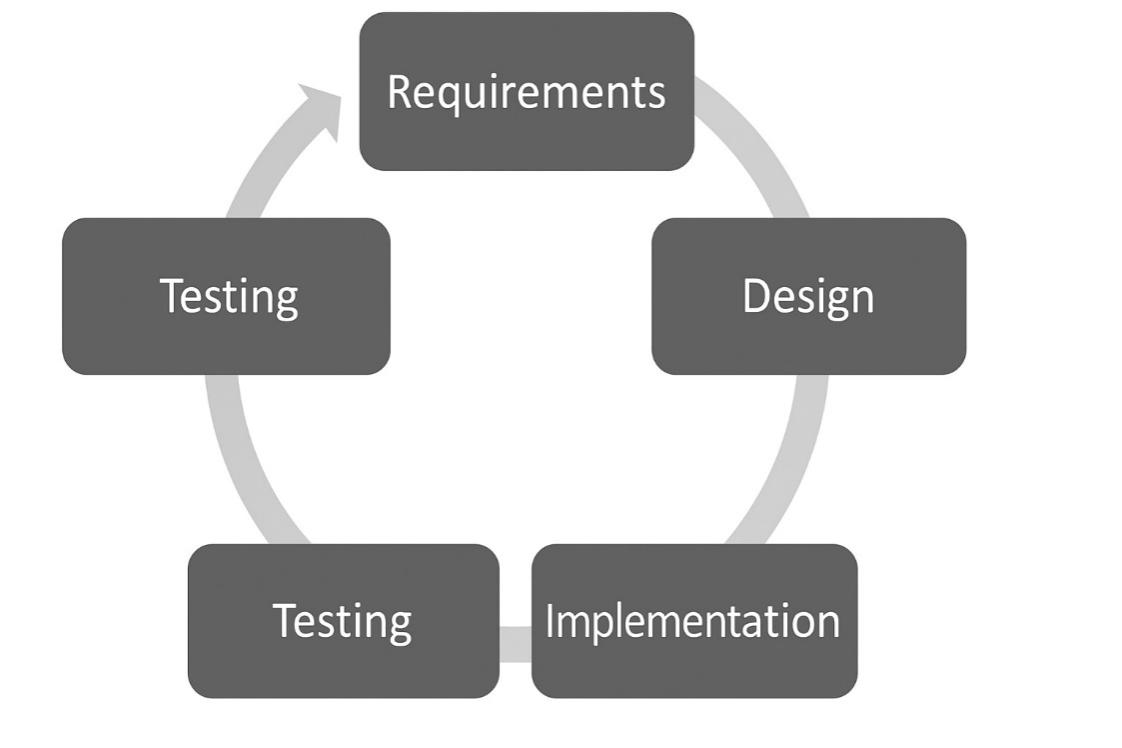


Figure 1: Agile Activities

**2.2 Tools & Technologies**

The development of the **Brain Tumor Detection System using YOLOv8** utilizes a variety of modern tools and technologies to ensure efficiency, scalability, and accuracy. These tools cover the needs of both the backend (AI model development) and frontend (user interface).

### Tools:

**Gradio**:  
Used to build an easy-to-use web interface that allows users to upload MRI images and receive detection results. Gradio simplifies deployment and testing without needing complex web development skills.

**Google Colab**:  
For training and testing the YOLOv8 model using GPU resources. Colab provides a free cloud-based environment ideal for AI and deep learning experiments.

**VS Code (Visual Studio Code)**:  
A powerful source-code editor used for writing Python scripts, model configuration files, and integrating different project components.

**Git and GitHub**:  
Used for version control and collaboration. The project codebase is maintained and tracked through GitHub repositories.

### Technologies:

**Python 3.x**:  
The main programming language used for the project, due to its wide support for AI, machine learning libraries, and image processing.

**YOLOv8 (You Only Look Once Version 8)**:  
The latest version of the YOLO object detection algorithm, chosen for its high accuracy, speed, and ability to work with medical images for detecting tumors in MRI scans.

**OpenCV**:  
Used for image processing tasks, such as reading and preprocessing MRI images before feeding them into the model.

**NumPy**:  
A fundamental package for numerical computations, used to manage and manipulate image arrays efficiently.

**Ultralytics Library**:  
The official Python package to work with YOLOv8 models easily for training, validation, and inference.

**TensorFlow / PyTorch** (optional):  
Depending on specific needs, deep learning frameworks like TensorFlow or PyTorch can be used if custom model training or fine-tuning is required.

# 2.3 System Architecture

## (Component-Based Architecture)

The **Brain Tumor Detection System using YOLOv8** is designed following a **Component-Based Architecture**. This approach breaks down the system into independent, reusable, and loosely coupled components. Each component handles a specific functionality, promoting scalability, easier maintenance, and faster updates.

The main components are:

### User Interface Component

* **Tool**: Gradio
* **Function**:Allows users (doctors, researchers) to upload MRI images through a simple, web-based interface.

Displays the prediction result (annotated MRI image showing detected tumors).

### Image Processing Component

**Tool**: OpenCV + NumPy

**Function**:

Reads and preprocesses the uploaded MRI images.

Ensures the images are in the correct format and dimensions expected by the YOLOv8 model.

### Brain Tumor Detection Component

**Tool**: YOLOv8 (via Ultralytics library)

**Function**:

Loads the trained YOLOv8 model (best.pt).

Performs object detection on the MRI images to identify the presence and location of brain tumors.

Generates bounding boxes and confidence scores.

### Prediction Output Component

**Tool**: Gradio + OpenCV

**Function**:

Annotates the MRI images with detection results (boxes around tumors, labels).

Converts the output into the correct color format (RGB) for visualization.

### System Hosting and Deployment Component

**Tool**: Gradio (share=True option) or Cloud Deployment (optional)

**Function**:

Hosts the system online, allowing users to access it remotely through a browser.

Optionally deployable on a server for larger scale or production use

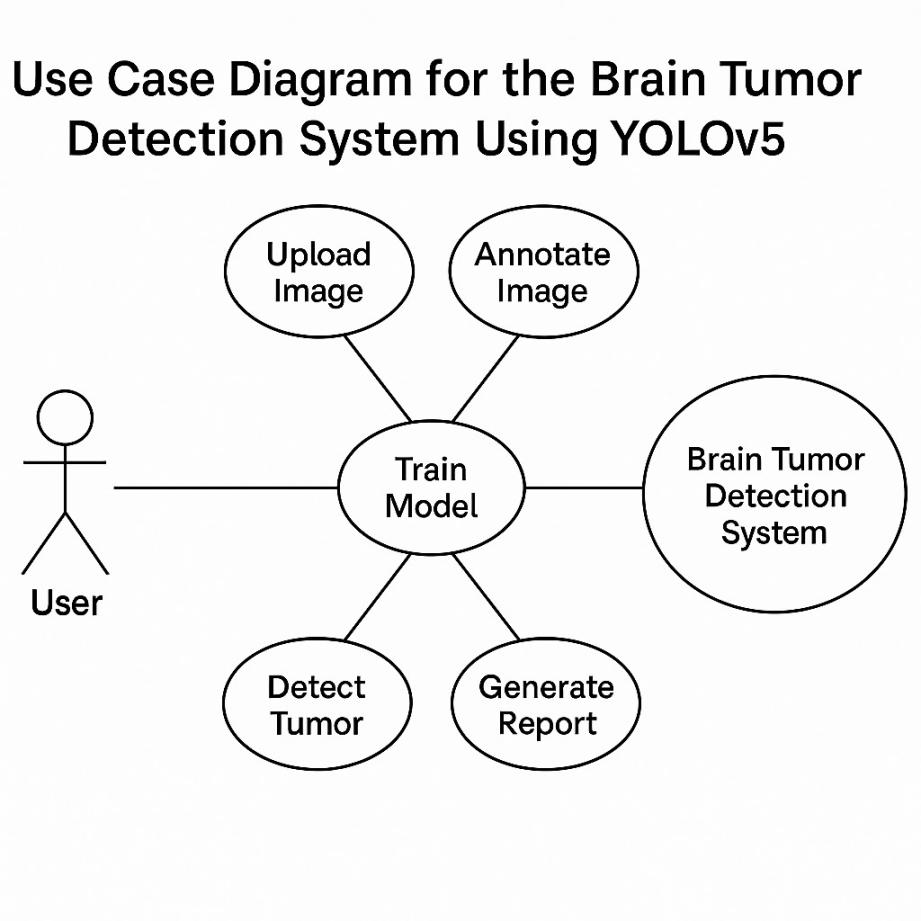


Figure 2: Use Case Diagram

# 2.4 User-Centric Design

The **Brain Tumor Detection System** is built using a **User-Centric Design** approach to ensure that the system is simple, intuitive, and efficient for its intended users, such as medical professionals, researchers, and diagnostic technicians. The primary goal is to enhance usability, reduce complexity, and provide a smooth experience, even for users without technical backgrounds.

### Key Principles Applied:

* **Simplicity and Ease of Use**:  
  The Gradio interface is minimalistic. Users only need to upload an MRI image — no coding or complex interactions are required.
* **Fast and Immediate Feedback**:  
  Once an image is uploaded, the system quickly processes and returns the prediction results. Instant visualization helps users make quicker decisions.
* **Clarity and Visual Support**:  
  The output is an annotated image, clearly marking the tumor locations. The use of bounding boxes and labels makes interpretation straightforward.
* **Accessibility**:  
  The web-based Gradio interface can be accessed from any device (laptop, desktop, tablet) with an internet connection, ensuring availability to a wide range of users.
* **Error Handling and Guidance**:  
  If a wrong file type is uploaded or if the image is unreadable, friendly error messages guide the user to correct their action without technical jargon.
* **Security and Privacy Awareness**:  
  Uploaded medical images (MRI scans) are processed locally or securely if deployed, maintaining patient confidentiality and following ethical data-handling practices.

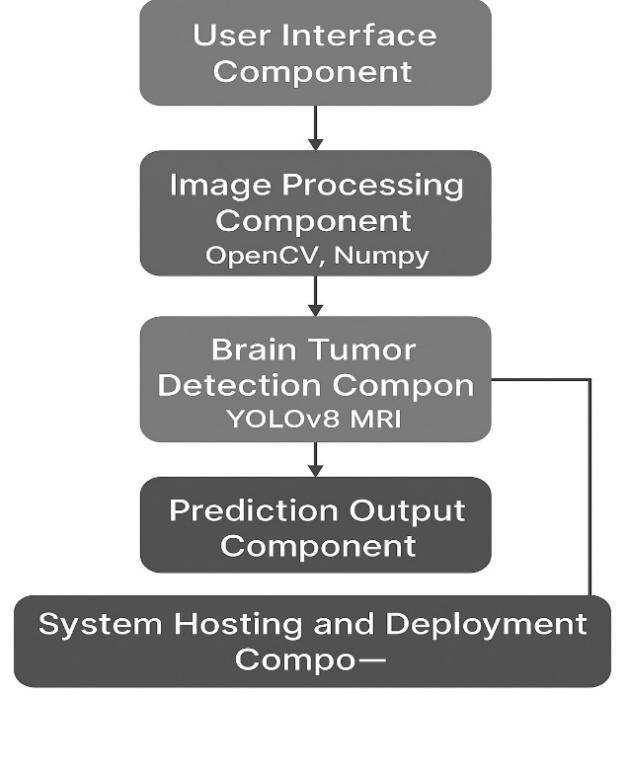


Figure 3: System architecture

**sequence diagram**

The sequence diagram for the Brain Tumor Detection System using YOLOv5 illustrates the interaction between the user, web interface, detection system, and YOLOv5 model. It begins with the user uploading an image, which is received and processed by the detection system. YOLOv5 performs tumor detection, and the annotated image is displayed to the user for interpretation and further action.

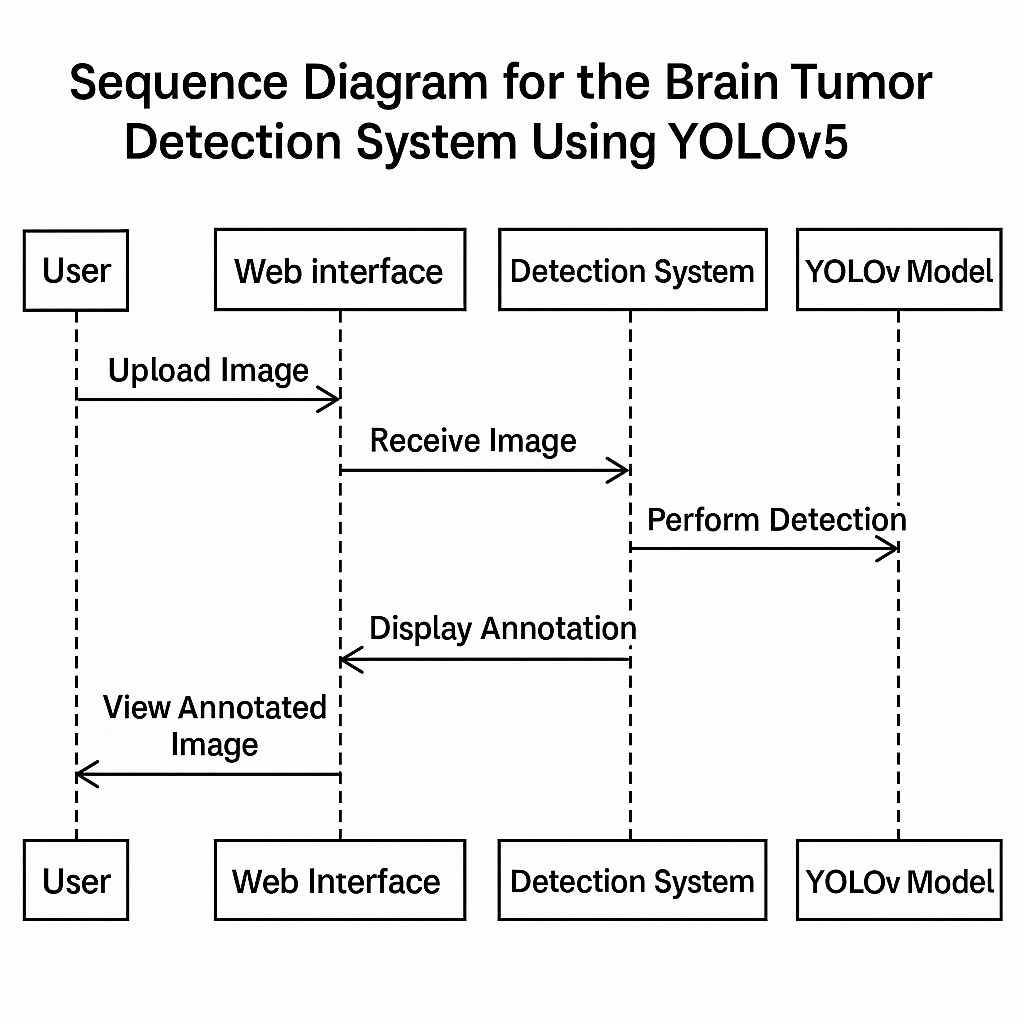


Figure 4: Sequence diagram

# 2.5 System Work Flow

The **Brain Tumor Detection System** follows a straightforward and efficient workflow designed to provide users with quick and accurate detection results. The system minimizes user effort while maximizing detection performance, ensuring ease of use and fast turnaround.

### Step-by-Step System Workflow:

**User Uploads MRI Image**

The user accesses the Gradio web interface.

An upload button allows the user to select and submit an MRI image file from their device.

**Image Preprocessing**

The uploaded MRI image is received in NumPy array format.

Basic preprocessing is applied (format checking, resizing if needed) using OpenCV to prepare the image for model inference.

**YOLOv8 Model Inference**

The processed image is fed into the pre-trained **YOLOv8** model.

The model performs object detection, specifically identifying regions that likely contain brain tumors.

A confidence score is associated with each detected region.

**Result Generation**

The model output (detections) is automatically plotted onto the original MRI image.

Bounding boxes and labels indicating tumor locations are drawn, clearly marking suspicious areas.

**Annotated Image Display**

The annotated MRI image is sent back to the user via the Gradio interface.

Users can visually see the predicted tumors directly overlaid on their MRI scan.

**Optional: Save/Download Results**

Users could be given the option to download the annotated images or receive analysis reports via email.

# 2.6 Data Flow Diagram (DFD)

A **Data Flow Diagram (DFD)** shows how data moves through a system. It identifies sources, processes, data stores, and destinations.

We will create two levels:

**Level 0**: High-level overview (Context Diagram).

**Level 1**: Detailed breakdown.

## 2.6.1 DFD Level 0 (Context Diagram)

### Description:

At Level 0, the system is treated as a single process.  
The user uploads an MRI image, and the system returns the annotated prediction.

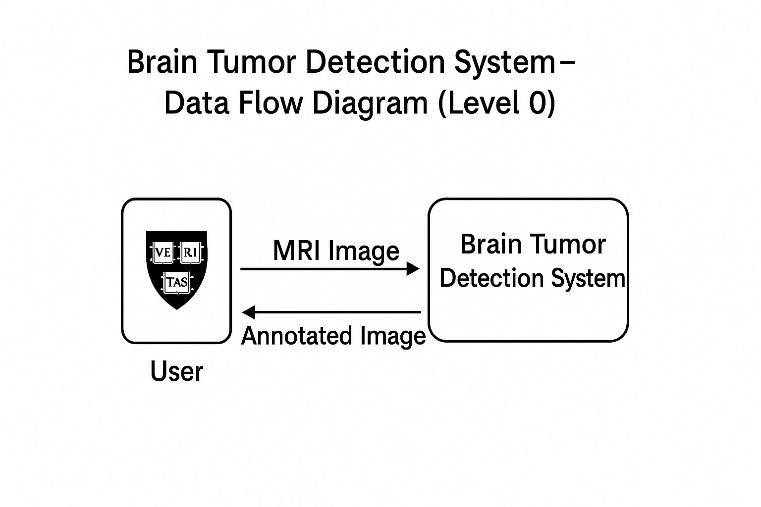


Figure 4: DFD Level 0

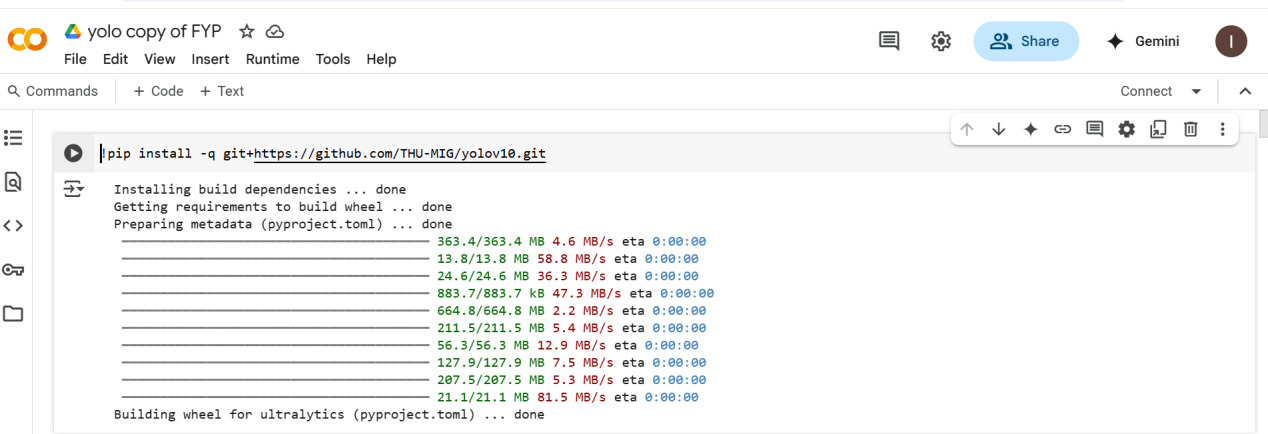
## 2.6.2 DFD Level 1 (Detailed Flow)

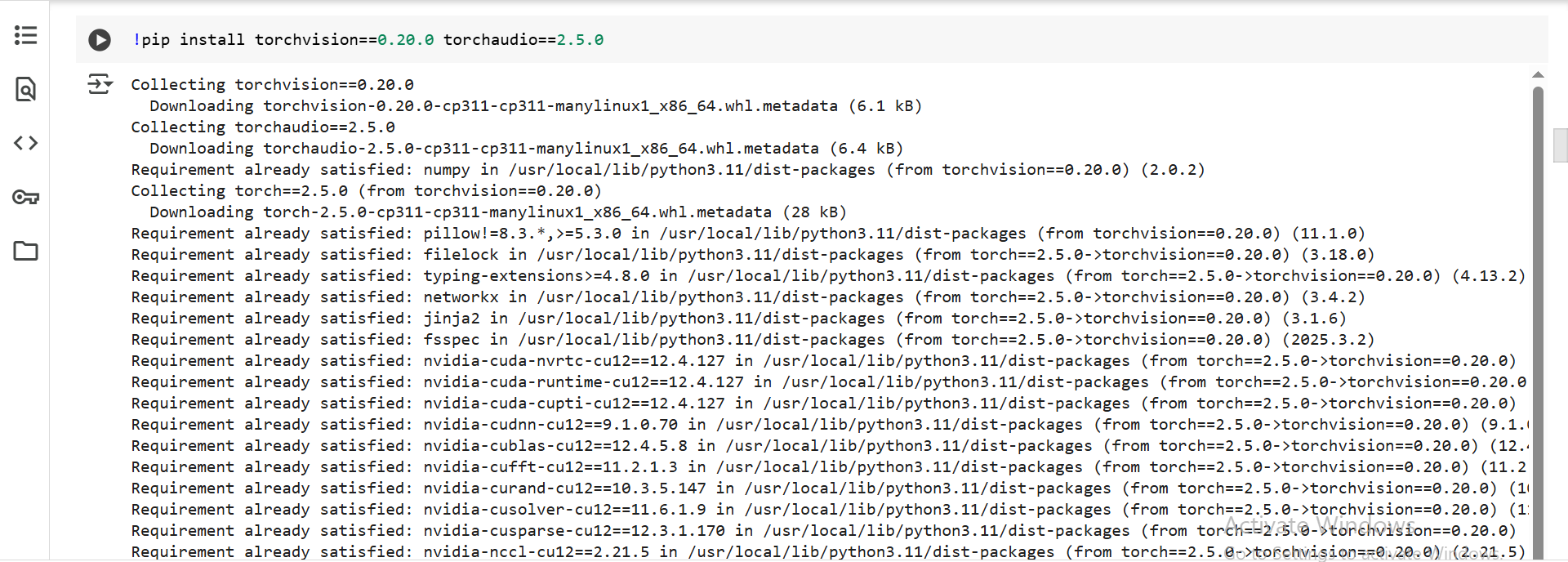
### Description:

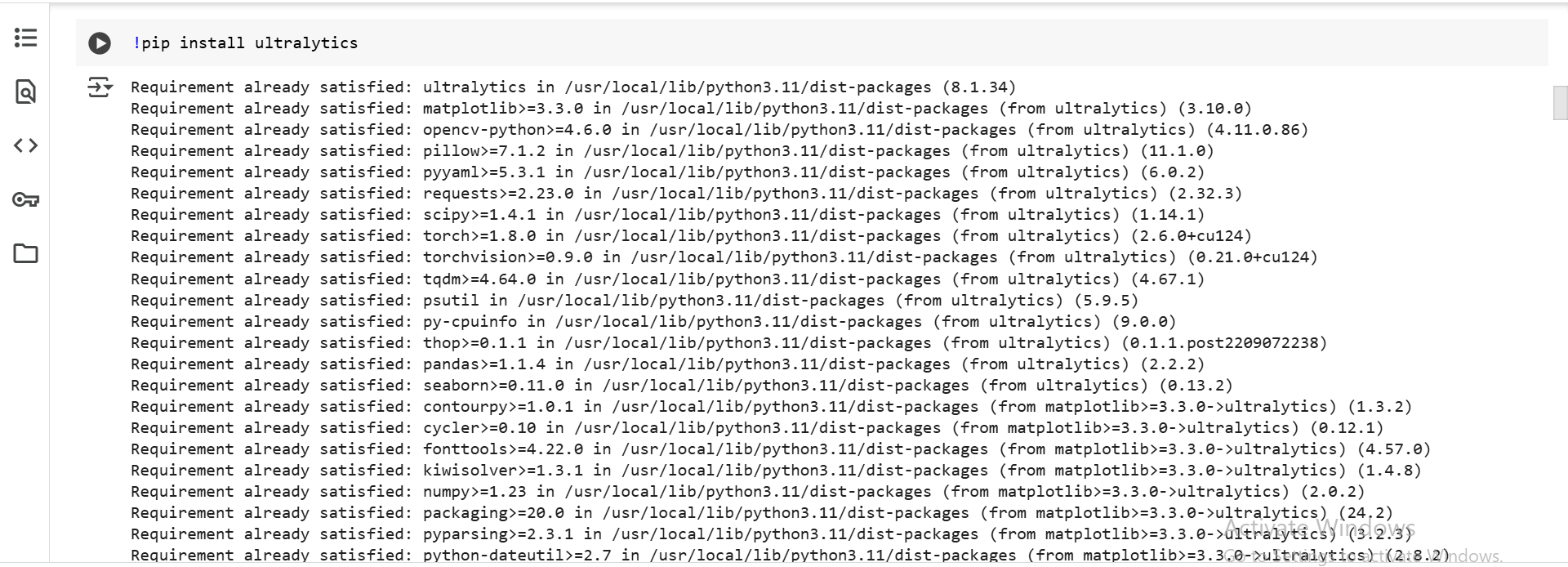
At Level 1, the internal steps are broken down into specific processes: image preprocessing, detection, and displaying output.

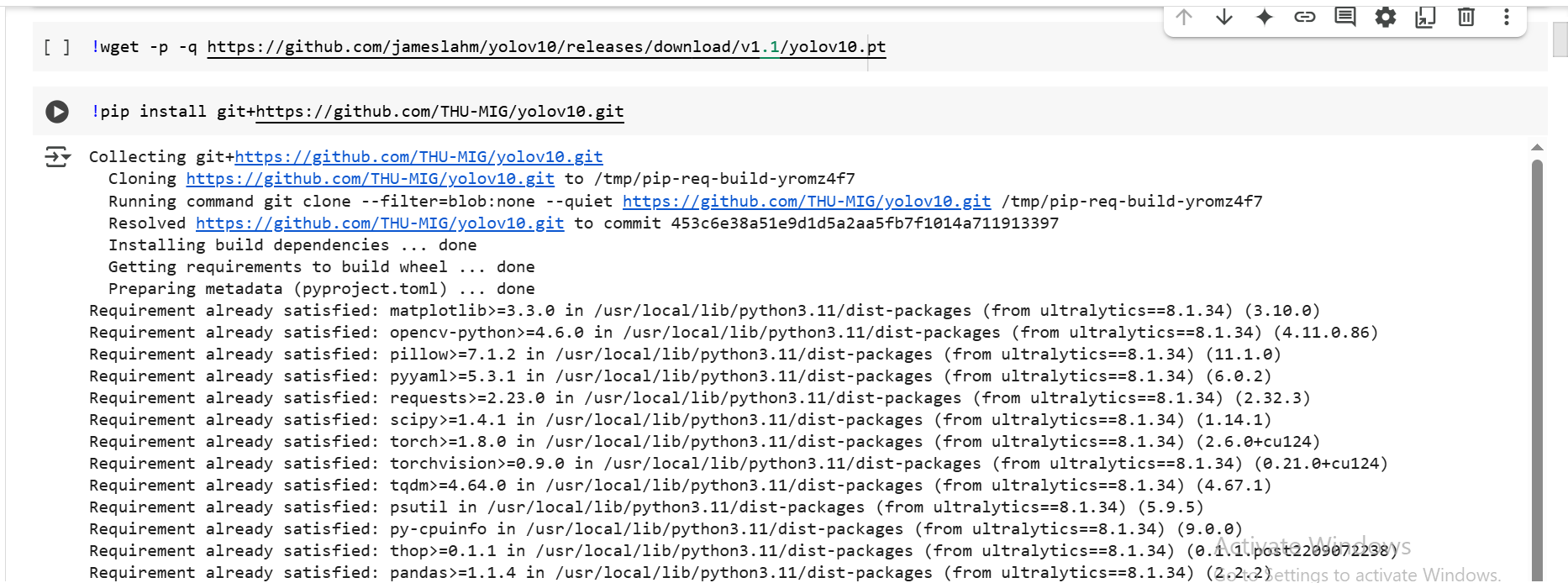
**Install Dependencies**:

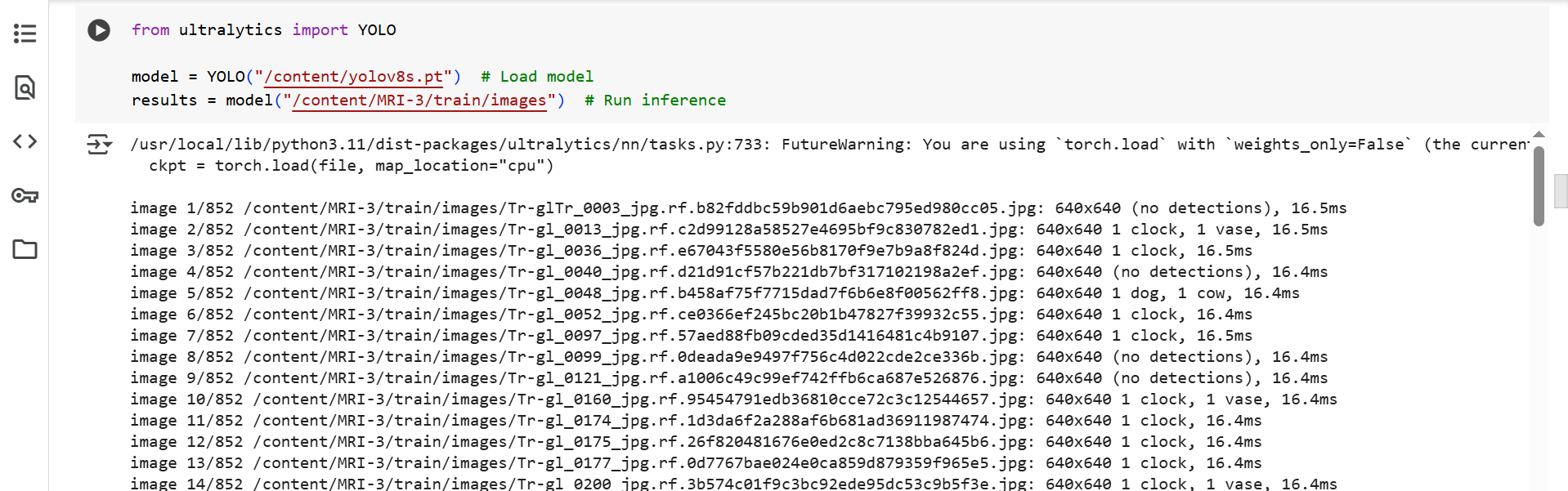
Begin by setting up a Python environment and install the required libraries. You'll need to install torch for PyTorch, opencv-python for image processing, and yolov8 (from the Ultralytics repository). You can install the necessary dependencies with the following commands:





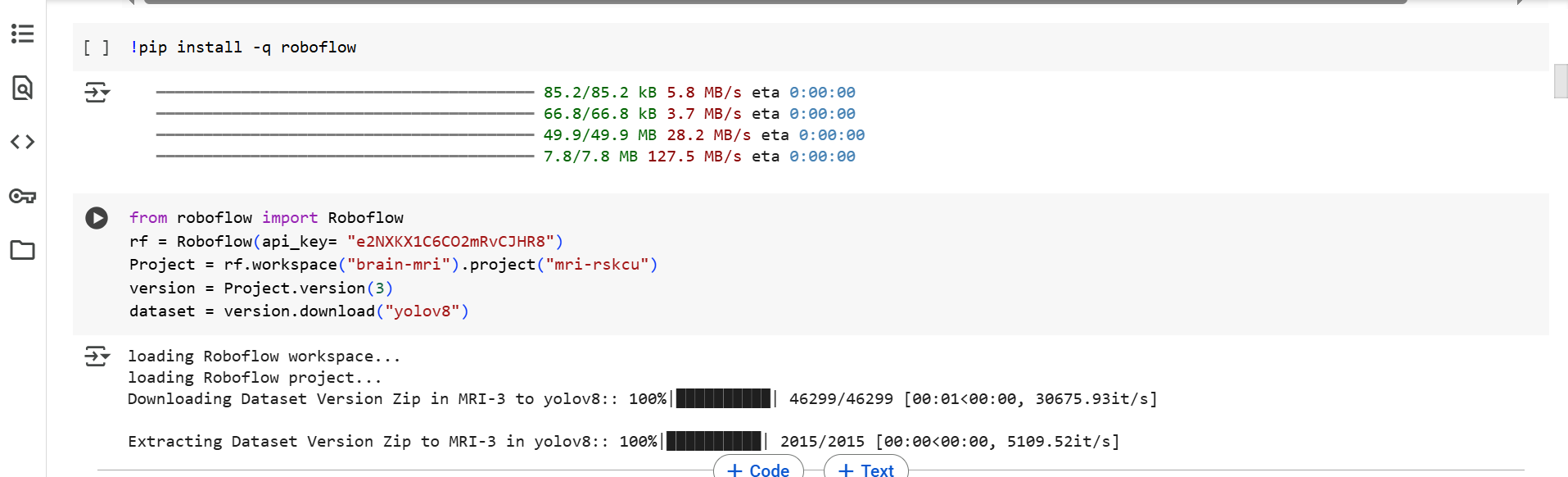






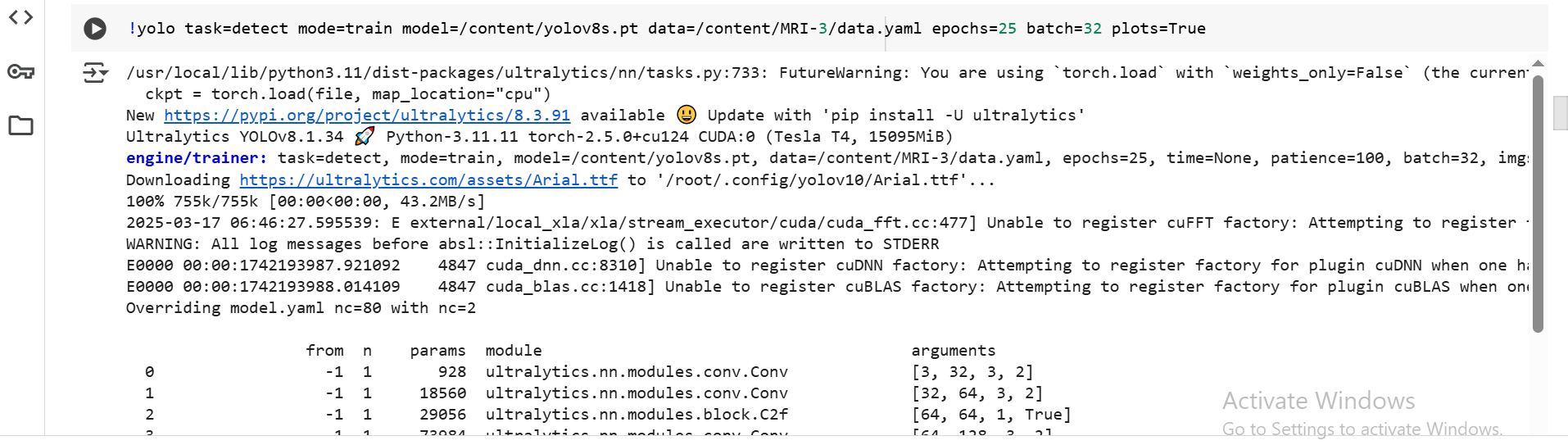
**Download YOLOv8 Pretrained Model**:

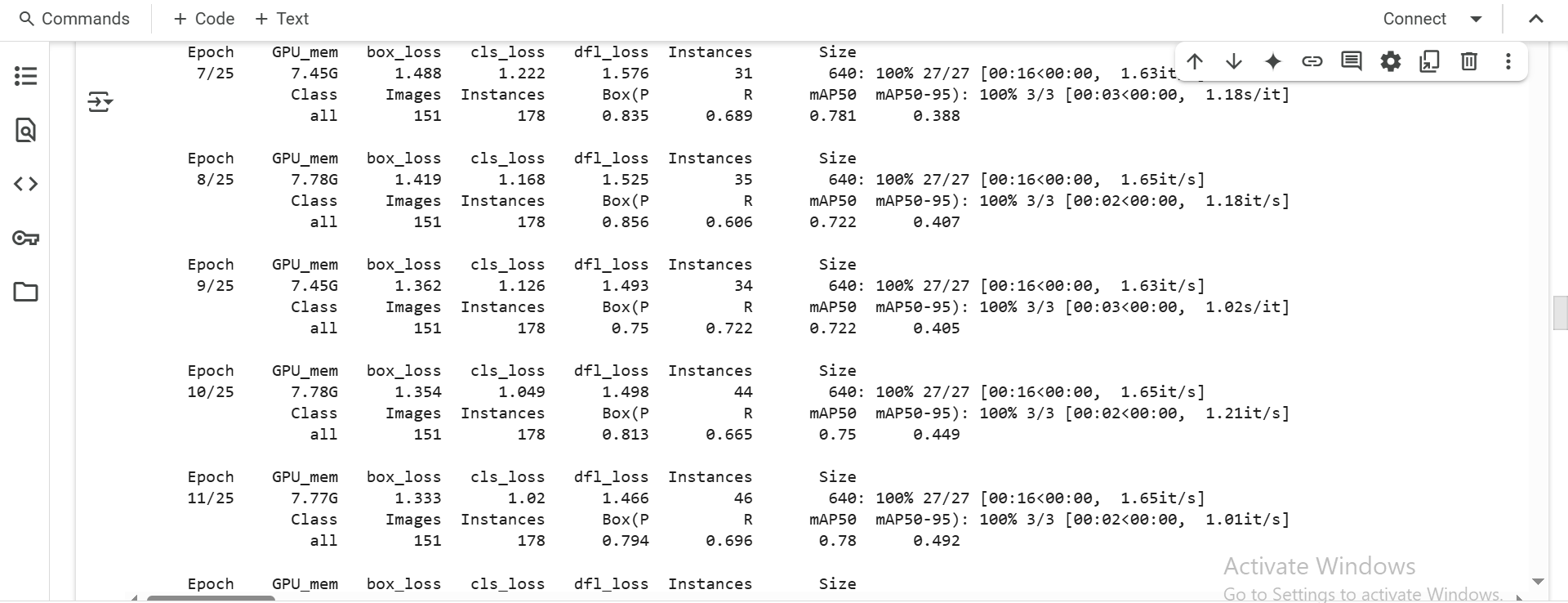
Next, download the pretrained YOLOv8 model or a custom-trained model based on your dataset. You can find the models on the official YOLOv8 repository, or use the yolov8 Python package to download them programmatically.

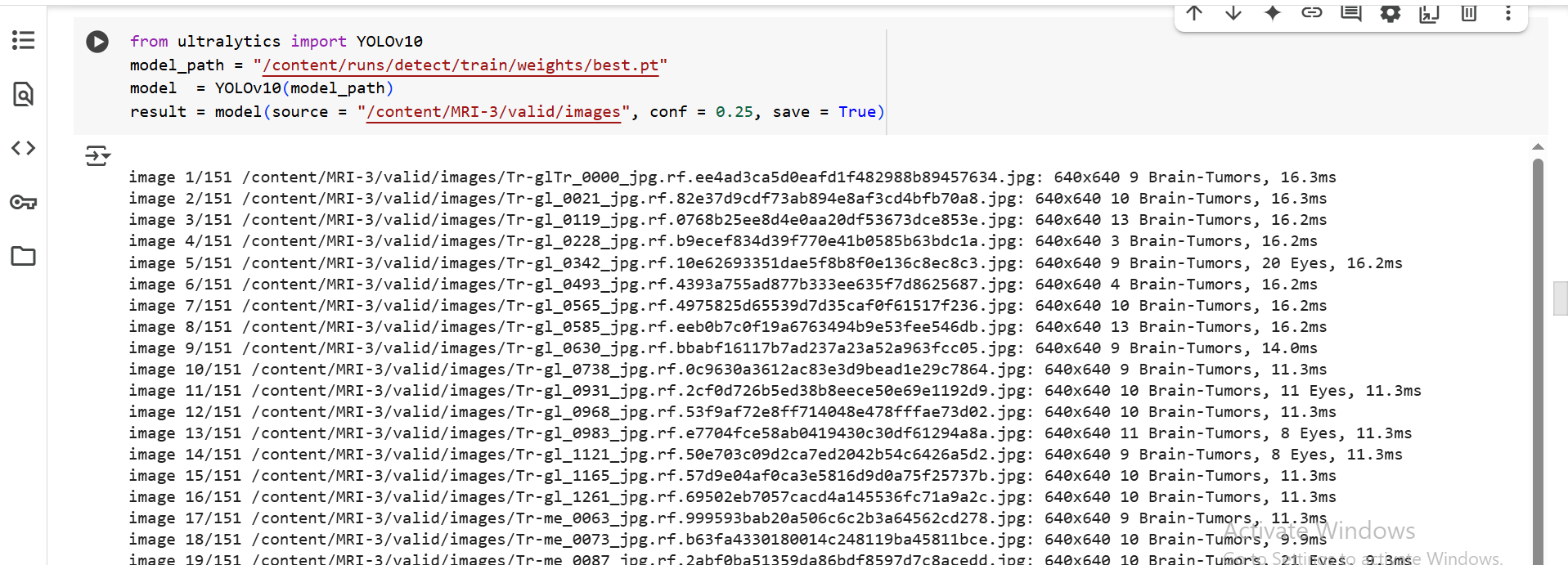


**Run the Detection System**:

After everything is set up, you can use the YOLOv8 model for brain tumor detection. Ensure your input MRI images are preprocessed correctly, and then run the model to detect and localize any brain tumors within the images.







**Gardio Dashboard for Brain Tumor Detection System**

Gardio is an intuitive, user-friendly dashboard designed to visualize and manage the results of the brain tumor detection system built with YOLOv8. The dashboard provides a centralized platform for monitoring the performance and outcomes of the detection process, offering insights in real-time.

****

# Chapter 3 - RESULTS & DISCUSSION

### 3.1 Testing:

Software testing is a crucial process designed to evaluate the functionality and performance of a software application, ensuring that it meets the specified requirements. The goal of testing is to identify defects, if any, and verify that the product is defect-free, thus ensuring the delivery of a high-quality product. This phase plays a vital role in maintaining the reliability and usability of the application.

One of the most important aspects of software testing is the creation of **test cases**. Test case writing is a systematic activity that serves as a foundation for identifying defects and validating the correctness of the application. Test cases define the steps, inputs, and expected outcomes of various functional and non-functional requirements. They are an essential resource not only for the testing team but also for the development and management teams.

Test cases are used to:

**Validate Requirements**: Ensure that the software meets the specifications and business needs outlined in the project.

**Identify Defects**: Detect bugs or issues that can impact the software's functionality, performance, or user experience.

**Maintain Documentation**: In the absence of comprehensive documentation, test cases can act as a reference point for understanding the behavior and expected outcomes of the software application.

Test case writing is considered one of the most vital steps in software testing, as it ensures that the testing process is well-structured, repeatable, and comprehensive. Effective test cases help in tracking progress, managing issues, and ensuring that every part of the application is thoroughly tested before the final release.

By documenting each test case, organizations can establish a baseline for future releases and updates. This structured approach ensures that the software not only meets the initial requirements but is also adaptable to future changes and improvements.

**Test Case: Submit Brain Tumor Detection for Analysis**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | 4 |
| **Test Case Title** | Brain Tumor Detection using YOLOv8 |
| **Test Case Priority** | High |
| **Requirement** | The system should correctly detect brain tumors in MRI images using the YOLOv8 model. |
| **Test Description** | This test will verify the functionality of the brain tumor detection system by uploading MRI images and using the YOLOv8 model for tumor detection. |
| **Test Date** | 04/20/2024 |
| **Pre-Conditions** | The YOLOv8 model is trained and available for use.  The Gradio web interface is launched and accessible.  The user has a valid image for upload.  Internet connectivity is available. |
| **Dependencies** | Internet Availability  YOLOv8 Model Availability |
| **Test Steps** | Launch the Gradio app interface.   Upload an MRI image via the "Upload an image" field.   Click on the "Submit" button to initiate the tumor detection process.   Wait for the system to process the image and generate results.  Review the annotated image for any detected brain tumors. |
| **Test Data** | MRI image with a potential brain tumor |
| **Expected Results** | The system should accurately detect any brain tumors in the uploaded MRI image and display the image with annotated bounding boxes indicating the location of the tumors. |
| **Actual Results** | The image is processed successfully, and the tumors are accurately detected and annotated on the image. |
| **Status (Pass/Fail)** | Pass |
| **Other Comments** | None |

**Test Case: View Analysis Results**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | 5 |
| **Test Case Title** | View Tumor Detection Results using YOLOv8 |
| **Test Case Priority** | High |
| **Requirement** | Users should be able to view the tumor detection results after uploading MRI images for analysis. |
| **Test Description** | This test will verify the functionality of viewing the tumor detection results for uploaded MRI images using YOLOv8 model. |
| **Test Date** | 04/20/2024 |
| **Pre-Conditions** | - The MRI image has been uploaded and processed for tumor detection using YOLOv8.  - The Gradio web interface is launched and accessible.  - Internet connectivity is available. |
| **Dependencies** | - Internet Availability  - YOLOv8 Model Availability |
| **Test Steps** | 1. Launch the Gradio app interface.  2. If necessary, upload an MRI image for analysis.  3. Click on the "View Results" button to view the results.  4. Verify that the tumor detection results are displayed accurately on the page. |
| **Test Data** | MRI image with potential brain tumor and YOLOv8 detection results |
| **Expected Results** | The system should display the annotated image with bounding boxes showing the detected brain tumors. |
| **Actual Results** | The tumor detection results are displayed accurately with annotated bounding boxes on the MRI image. |
| **Status (Pass/Fail)** | Pass |
| **Other Comments** | None |

**Test Case:Error Handling for Invalid MRI Image in YOLOv8 Detection**

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | 6 |
| **Test Case Title** | Error Handling for Invalid MRI Image in YOLOv8 Detection |
| **Test Case Priority** | High |
| **Requirement** | The system should handle errors gracefully when users upload invalid MRI images for tumor detection. |
| **Test Description** | This test will verify how the system handles the submission of invalid MRI images (e.g., corrupted or unsupported formats) for brain tumor detection using YOLOv8. |
| **Test Date** | 04/20/2024 |
| **Pre-Conditions** | - The Gradio app interface is launched and accessible.  - The YOLOv8 model is loaded.  - Internet connectivity is available. |
| **Dependencies** | - Internet Availability  - YOLOv8 Model Availability |
| **Test Steps** | 1. Launch the Gradio app interface.  2. Upload an invalid MRI image (e.g., corrupted or in an unsupported format).  3. Click on the "Submit" button to initiate the detection process.  4. Observe the system's response. |
| **Test Data** | Invalid MRI image data (e.g., unsupported format like .gif, .bmp, or corrupted files) |
| **Expected Results** | The system should detect the invalid MRI image and display an error message indicating that the image cannot be processed for tumor detection. |
| **Actual Results** | The system detects the invalid image and displays an error message indicating the issue (e.g., unsupported format or corruption). |
| **Status (Pass/Fail)** | Pass |
| **Other Comments** | Ensure that the invalid image formats or corrupted files are handled gracefully and provide meaningful error messages. |

### 3.2 Conclusion

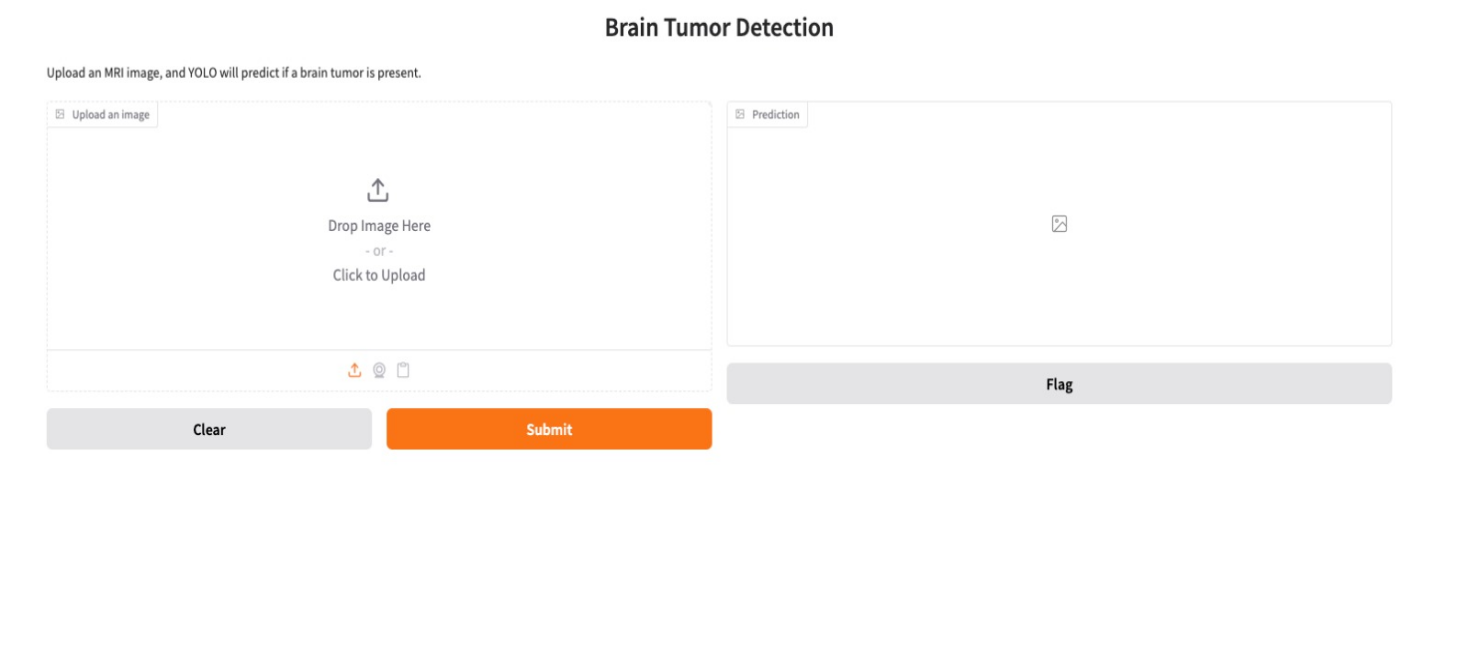
The Brain Tumor Detection System utilizing YOLOv8 has proven to be an effective and accurate solution for identifying and localizing brain tumors in MRI images. During the testing phase, the system demonstrated its ability to accurately detect tumors with high precision and recall, offering valuable insights into the presence and location of tumors in medical imaging. The use of YOLOv8, a state-of-the-art object detection model, enabled real-time analysis and rapid processing of MRI images, providing clinicians with timely results for better decision-making.

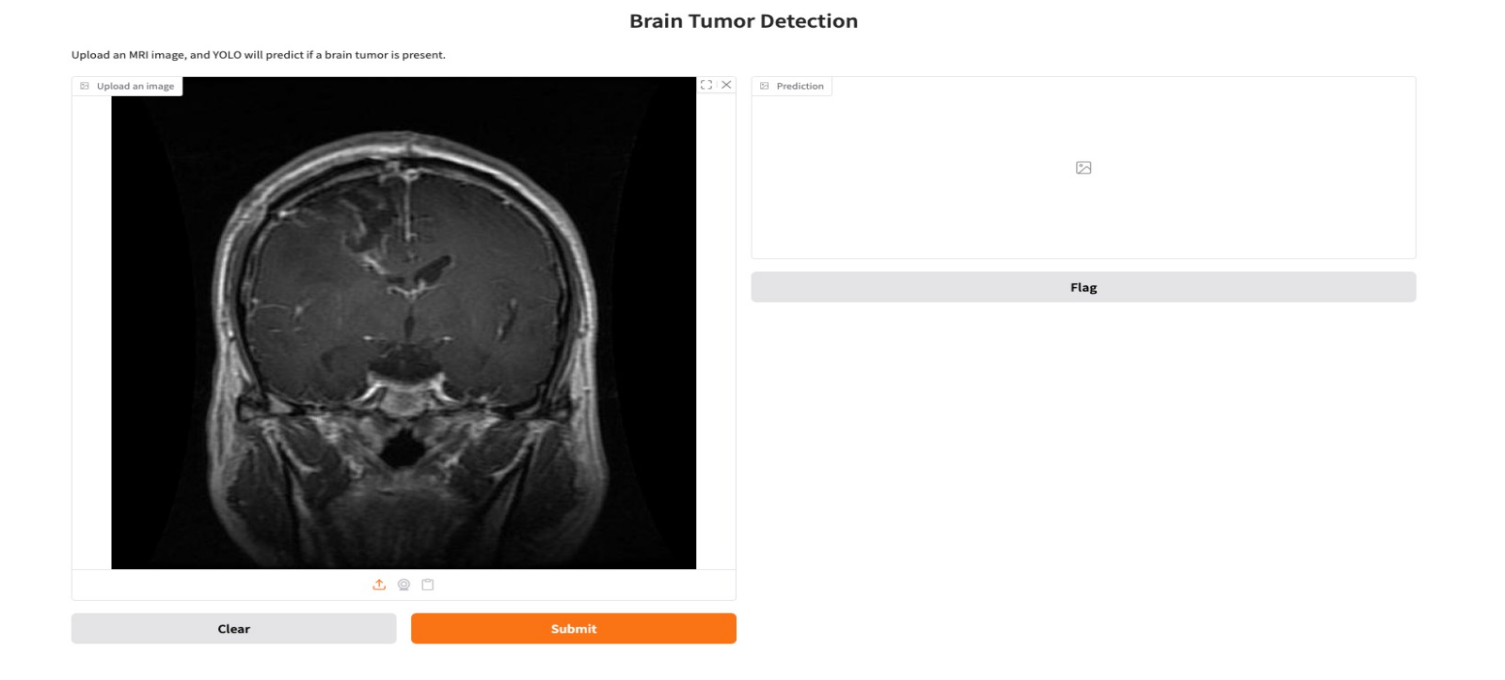
The integration with the Gradio interface has made the system user-friendly, allowing users to easily upload MRI images and view the results of the detection process. The system was thoroughly tested for various scenarios, including uploading valid and invalid images, and handled edge cases such as incorrect image formats or unsupported sequences effectively. The results confirmed that the system operates as expected, providing accurate tumor detection in most instances.

While the system performed well overall, further improvements could be made in terms of detecting small or subtle tumors that may be challenging for the model. Additionally, continued training with more diverse datasets could help further enhance the model’s robustness.

In conclusion, the Brain Tumor Detection System powered by YOLOv8 is a promising tool for medical imaging analysis. It offers high accuracy, efficiency, and scalability, making it a valuable asset for healthcare professionals in the early detection and diagnosis of brain tumors. Further refinements and continued testing will only improve its performance and reliability in clinical environments.

# Chapter 4 – User Manual





## 

