BRAIN TUMOR DETECTION USING DEEP LEARNING

Submitted by

NAGALAKSHMI B	**810422243065**
SUKITHA S	**810422243107**
SHALI SIBIYA J	**810422243098**

Of

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE(AUTONOMOUS)

PERAMBALUR – 621 212

A MINI PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY CHENNAI - 600 025

MAY 2025

BONAFIDE CERTIFICATE

Certified that this mini project report titled "**BRAIN TUMOR DETECTION USING DEEP LEARNING**" is the bonafide work of **"NAGALAKSHMI B (810422243065), SUKITHA S (810422243107), SHALI SIBIYA J (810422243098)"** who carried out the research under my supervision.certified further,that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
MR.SARAVANAN S, M.TECH,M.B.A,(Ph.D)	**Dr. SHREE K.V.M, M.E..., Ph.D**
Assistant professor,	Professor & Head,
Department of Artificial Intelligence and Data Science,	Department of Artificial Intelligence and Data Science,
Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur-621 212.	Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur-621 212.

Submitted for Mini Project Viva-Voce Examination held on _______________

INTERNAL EXAMINER **EXTERNAL EXAMNIER**

ACKNOWLEDGEMENT

It is with immense pleasure that I present my first venture in the field of real application of computing in the form of project work. First, I am indebted to the Almighty for his choicest blessing showered on me in completing this endeavor.

I express my sincere thanks to **Shri. A. SRINIVASAN,** Chancellor, **Dhanalakshmi Srinivasan University,** for having given me an opportunity to study in this Institution.

I would like to express my sincere gratitude to **DR. D. SHANMUGASUNDARAM, M.E., Ph.D., F.I.E., C.Eng.,** Principal, **Dr. K. ANBARASAN, M.E., Ph.D.,** Dean, and **Dr. M. CHELLAPPAN, M.E., Ph.D.,** Controller of Examinations, **Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur,** for their constant moral support and encouragement throughout the course.

I extend my heartfelt thanks to the Head of the Department, **Dr. K.V.M. SHREE, M.E., Ph.D.,** for providing all the necessary facilities and academic support.

I owe my deepest gratitude to my Internal Guide, **Mr. S. SARAVANAN, M.Tech., M.B.A, (Ph.D.),** for his invaluable guidance, encouragement, and constructive suggestions throughout the project work.

We render our thanks to all the staff members and programmers of department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** for their timely assistance.

ABSTRACT

Brain tumor detection is a vital area of research in medical imaging, as early and accurate diagnosis can significantly improve treatment outcomes and patient survival rates. Traditional diagnostic methods rely heavily on manual interpretation of MRI (Magnetic Resonance Imaging) scans by radiologists, which can be time-consuming, subjective, and prone to error due to subtle differences between tumor types. To address these limitations, this study proposes an advanced deep learning-based approach using Convolutional Neural Networks (CNNs) for automated brain tumor detection and classification.

The model is trained on a comprehensive dataset of MRI brain scans categorized into four classes: no tumor, glioma, meningioma, and pituitary tumor. These types represent the most common forms of brain tumors, each requiring distinct treatment strategies. The CNN architecture is designed to extract hierarchical features from the input images, allowing the system to learn complex patterns and spatial hierarchies indicative of different tumor types. Preprocessing techniques such as normalization, data augmentation, and noise reduction are applied to enhance image quality and model generalization.

The proposed system is evaluated based on performance metrics including accuracy, precision, recall, and F1-score, demonstrating high effectiveness in distinguishing between tumor and non-tumor cases. By automating the detection process, this model offers significant advantages in clinical settings, such as reducing diagnostic time, minimizing human error, and providing a second opinion to medical professionals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MRI	-	MAGNETIC RESONANCE IMAGING
CT	-	COMPUTED TOMOGRAPHY
PET	-	POSITRON EMISSION TOMOGRAPHY
DICOM	-	DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE
NIFTI	-	NEUROIMAGING INFORMATICS TECHNOLOGY INITIATIVE
ROI	-	REGION OF INTEREST
AI	-	ARTIFICIAL INTELLIGENCE
ML	-	MACHINE LEARNING
DL	-	– DEEP LEARNING
CNN	-	CONVOLUTIONAL NEURAL NETWORK
DCNN	-	DEEP CONVOLUTIONAL NEURAL NETWORK
R-CNN	-	REGION-BASED CONVOLUTIONAL NEURAL NETWORK
CAPSNET	-	CAPSULE NETWORK
VAE	-	VARIATIONAL AUTOENCODER
GAN	-	GENERATIVE ADVERSARIAL NETWORK
RELU	-	RECTIFIED LINEAR UNIT
FCN	-	FULLY CONVOLUTIONAL NETWORK
U-NET	-	U-SHAPED CNN ARCHITECTURE FOR SEGMENTATION
NNU-NET	-	SELF-CONFIGURING U-NET FRAMEWORK
RF	-	RANDOM FOREST
PCA	-	PRINCIPAL COMPONENT ANALYSIS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Brain tumors are among the most serious and life-threatening health conditions, affecting thousands of people globally each year. Early and accurate detection of brain tumors is crucial for effective treatment planning and can significantly improve patient survival rates. However, manual diagnosis based on MRI (Magnetic Resonance Imaging) scans can be time-consuming, prone to human error, and highly dependent on the radiologist's experience. To overcome these challenges, deep learning—a subset of artificial intelligence—has emerged as a powerful technology that can automate the detection and classification of brain tumors from medical images. In this project, we propose a deep learning-based system that can accurately classify brain MRI images into four categories: No Tumor (healthy brain), Meningioma Tumor, Glioma Tumor, and Pituitary Tumor. This system is designed to assist radiologists by providing a second opinion, thus improving diagnostic accuracy, consistency, and speed.

The primary aim of this project is to develop a deep learning model that can classify brain tumors efficiently and accurately. By using convolutional neural networks (CNNs), which are highly effective in image analysis tasks, the system can automatically learn important features from MRI images without the need for extensive manual feature extraction. The model will be trained and validated on a dataset of brain MRI images categorized into the four mentioned classes. The ultimate objective is to achieve high classification accuracy while maintaining strong precision, recall, and F1-scores—metrics that are especially important in medical diagnosis where false negatives can be critical.

1.2 PURPOSE

The purpose of this study is to apply deep learning techniques, particularly Convolutional Neural Networks (CNNs), to automate the process of brain tumor detection. The study aims to reduce the dependency on radiologists for tumor identification, thereby addressing the challenges of limited availability and diagnostic variability. It seeks to improve both the accuracy and speed of brain tumor diagnosis by analyzing MRI images using advanced deep learning models. Another key objective is to develop a model capable of classifying different types of brain tumors based on MRI scan data, enhancing the precision of diagnostic outcomes. Ultimately, the study aspires to contribute to the development of a reliable tool that can be integrated into clinical settings, providing valuable support to healthcare professionals and improving patient care.

1.3 PROBLEM STATEMENT

Challenges in Traditional Methods: Traditional diagnostic methods involve manual inspection of MRI images, which is time-consuming and highly dependent on the expertise of the radiologists. These methods are also prone to errors, especially in the case of subtle tumors.

Growing Demand for Automation: With the increasing number of MRI scans being produced daily, there is a significant burden on healthcare providers. The need for an automated system that can handle large datasets, detect tumors accurately, and assist in diagnosing brain abnormalities is crucial.

1.4 MOTIVATION

Increased Incidence of Brain Tumors: Brain tumors are becoming more prevalent due to factors like population growth, improved diagnostic techniques, and increased awareness. Early and efficient detection is essential for treatment.

Current Gaps in Diagnostic Methods: Despite advances in medical imaging, current diagnostic methods (manual inspection and traditional machine learning) still face several challenges:

- High dependency on radiologists
- Long processing times
- Limited generalization to diverse datasets

Advantages of Deep Learning: Deep learning algorithms, especially CNNs, have shown remarkable potential in automating tasks like image classification. They can analyze large volumes of MRI images in a fraction of the time it takes human experts and provide highly accurate results.

1.5 OBJECTIVE

The objectives of this project are as follows:

- To develop a CNN-based system for automatic brain tumor detection using MRI images.
- To train the system on a dataset of MRI scans with labeled tumor data.
- To evaluate the model based on accuracy, precision, recall, and F1-score, and compare it with existing detection methods.
- To implement a user-friendly interface (optional) that allows medical professionals to upload MRI images for tumor detection.
- To ensure the system can handle various types of brain tumors, ensuring high robustness and accuracy.

CHAPTER 2

LITERATURE SURVEY

2.1 CNN-Based Classification Models

A notable contribution to brain tumor detection using deep learning was made by Hossain et al. (2021) in their study titled "Brain Tumor Detection using CNN." The authors developed a custom convolutional neural network (CNN) from scratch, consisting of five convolutional layers and two fully connected layers. The model was trained on the Figshare dataset, which includes 3,064 T1-weighted MRI images categorized into glioma, meningioma, pituitary tumor, and no tumor. Despite its relative simplicity compared to state-of-the-art architectures, their CNN achieved an impressive accuracy of approximately 98%. This work underscores the potential of task-specific deep learning models, illustrating that a well-designed and properly regularized CNN can outperform more complex models when applied to a focused medical imaging problem.

2.2 Transfer Learning Approaches

Transfer learning has proven to be highly effective in medical image classification, as shown in the study by Sajjad et al. (2019) titled "Multi-grade Brain Tumor Classification using Deep CNN and Transfer Learning." In this research, the authors utilized the VGG19 model pre-trained on ImageNet and fine-tuned it for classifying brain tumors. Data augmentation and dropout techniques were used to enhance generalization and prevent overfitting. The model achieved an accuracy of 94.58%, with strong performance in identifying glioma and pituitary tumors. Similarly, Islam et al. (2020) evaluated multiple transfer learning models—ResNet50, DenseNet121, and InceptionV3—on the Figshare dataset. Among them, DenseNet121 outperformed others with an accuracy of around 96%, demonstrating that deeper architectures with dense skip connections are particularly effective in extracting nuanced features from medical images. These studies collectively emphasize how transfer learning significantly reduces training time while maintaining high diagnostic accuracy.

2.3 Capsule Networks

To address limitations in traditional CNNs, Afshar et al. (2020) introduced a novel approach using Capsule Networks (CapsNets) in their study "Brain Tumor Type Classification via Capsule Networks." CapsNets are designed to better capture spatial hierarchies and relationships between features, which is particularly beneficial in brain tumor classification where orientation and positioning may vary. Using the Figshare dataset, the authors demonstrated that while CapsNets performed comparably to conventional CNNs in terms of accuracy, they exhibited superior robustness to changes in image orientation and worked well even with relatively small datasets. This was the first known application of CapsNets in this domain, offering a promising direction for more human-like and interpretable image analysis in medical diagnostics.

2.4 Tumor Segmentation Studies

Segmentation of brain tumors is a critical step in diagnosis and treatment planning, and deep learning has made significant strides in this area. The nnU-Net framework, developed by Isensee et al. (2021), revolutionized medical image segmentation by offering a self-configuring system based on the U-Net architecture. nnU-Net automatically adjusts preprocessing, network architecture, training schedules, and post-processing strategies depending on the input data. When applied to the BraTS Challenge dataset—which includes multiple tumor subregions such as the enhancing tumor, tumor core, and edema—nnU-Net achieved top-tier performance across various segmentation challenges. Another influential study by Myronenko et al. (2018)** combined U-Net with a variational autoencoder (VAE) to regularize training and reduce overfitting. This hybrid model won first place in the BraTS 2018 competition and demonstrated the efficacy of combining encoder-decoder structures with regularization techniques for precise tumor segmentation.

2.5 Hybrid and Ensemble Methods

Hybrid models that integrate deep learning with classical machine learning have also gained traction. Deepak and Ameer (2019) proposed a method in which a CNN was used for deep feature extraction from MRI images, followed by classification using a Support Vector Machine (SVM). This approach leveraged the representational power of CNNs with the interpretability and robustness of SVMs, achieving a classification accuracy of approximately 96%, which was higher than using CNNs alone. On the other hand, Pereira et al. (2016) focused on a patch-based segmentation strategy using CNNs. By analyzing smaller regions of the MRI images, the model was able to learn fine-grained tumor structures. The architecture featured small convolutional kernels (3×3), and the method delivered a high Dice coefficient on the BraTS dataset. This study was among the first to show that deep CNNs could significantly outperform traditional image processing techniques in brain tumor segmentation tasks.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing systems for brain tumor detection and classification in clinical practice are largely based on manual interpretation of MRI scans by trained radiologists. This traditional method involves visual examination of multiple slices of brain MRI to identify abnormalities, which is not only time-consuming but also prone to variability depending on the radiologist's experience and expertise. In many hospitals, initial screening is done through T1-weighted or T2-weighted MRI images, and decisions are made based on the visual appearance of tumors such as their shape, size, intensity, and location. While experienced radiologists can often make accurate diagnoses, the process is susceptible to human errors, especially when dealing with small or early-stage tumors that may not be easily noticeable.

To assist radiologists, some semi-automated tools based on classical image processing techniques have been developed. These include methods like thresholding, region growing, edge detection, and histogram analysis to segment tumor areas. However, these techniques are highly sensitive to noise and image quality, and they struggle with variations in tumor shape, texture, and contrast. Furthermore, early machine learning-based approaches have attempted to classify tumors using algorithms such as Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN). These models, however, rely heavily on handcrafted features (such as texture, intensity, or shape descriptors) which must be manually extracted from images before classification. This feature engineering step not only requires domain expertise but also limits the model's flexibility and accuracy. Moreover, these traditional systems do not scale well across large datasets or different MRI modalities, making them less reliable for real-world medical applications. As a result, while these systems provided a foundation, they are now being rapidly replaced by more robust and automated deep learning models that require less manual intervention and provide higher accuracy and generalization.

3.1.2 DISADVANTAGES OF EXISTING SYSTEM

Despite the progress made in applying deep learning and machine learning for brain tumor detection, the existing systems still face several disadvantages:

- Data scarcity and imbalance
- Overfitting and generalization issues
- Interpretability and trust issues
- High computational cost
- Lack of standardization
- Limited clinical adoption
- Tumor localization and segmentation
- Challenges with multi-class classification

3.2. PROPOSED SYSTEM

The proposed system aims to leverage advanced deep learning techniques, specifically a convolutional neural network (CNN)-based model, for automated brain tumor detection and classification from MRI images. The system follows a structured pipeline beginning with the acquisition of pre-processed T1-weighted MRI images, followed by image normalization and resizing to ensure consistency across the dataset. A custom-designed CNN architecture is then applied to extract deep spatial and contextual features from the images. This CNN model comprises multiple convolutional layers with ReLU activation, batch normalization, and pooling layers, followed by fully connected layers for classification. Additionally, dropout layers are included to reduce overfitting. The model is trained and validated on a labeled dataset such as the Figshare or BraTS dataset, which includes MRI scans labeled into categories such as glioma, meningioma, pituitary tumor, or healthy. Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the system. To further enhance robustness, the system may also integrate data augmentation techniques such as flipping, rotation, and contrast adjustments during training. Once trained, the model can classify input MRI images in real time, offering a rapid and reliable diagnostic aid for radiologists and clinicians.

3.2.1. ADVANTAGES OF PROPOSED SYSTEM

- Automated Diagnosis: Reduces the workload on radiologists and minimizes human error.
- Deep Learning Accuracy: Detects complex and subtle patterns in MRI images that manual analysis may miss.
- Efficient Architecture: Custom CNN provides high accuracy with low computational requirements.
- Better Generalization: Uses data augmentation and dropout to perform well even with limited or varied data.
- Consistent Results: Delivers objective and repeatable outputs, reducing variability between clinicians.
- Scalable & Adaptable: Can be retrained for other medical images or tumor types, making it future-ready.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

• Processor	:	Intel i5 / AMD Ryzen 5
• RAM	:	8-16 GB
• Storage	:	256 GB
• Keyboard	:	Standard keyboard
• Monitor	:	15-inch color monitor

4.2 SOFTWARE REQUIREMENTS

• Operating systems	:	Windows 10/11,Ubuntu 20.04+
• Programming language	:	Python 3.8+
• Python libraries	:	Numpy,Pandas,Opencv,Matplotlib,Schikit learn,Tensorflow
• Development tools	:	VS Code/Jupyter Notebook
• Dataset	:	Kaggle Brain tumor MRI dataset
• Web based interface	:	Streamlit/Django/Flask

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 LIST OF MODULES

1. Data Acquisition module

2. Data Preprocessing module

3. Model Definition Module

4. Model Training Module

5. Model Evaluation Module

6. Prediction Module

7. Visualization Module

8. User Interface Module

5.2 MODULE DESCRIPTION

5.2.1. Data Acquisition Module

The Data Acquisition Module is responsible for gathering and managing the dataset required for training and evaluating the deep learning model. This module typically involves loading MRI images from specified directories, which may contain images in various formats such as JPG, PNG, or DICOM. It is essential to ensure that the dataset is well-organized, often structured into subdirectories representing different classes (e.g., Meningioma, Glioma, Pituitary, No Tumor). This organization facilitates the use of data generators that can automatically label the images based on their directory names. Additionally, this module may handle splitting the dataset into training, validation, and testing sets, ensuring that the model can be trained effectively while also being evaluated on unseen data.

5.2.2. Data Preprocessing Module

The Data Preprocessing Module plays a crucial role in preparing the raw MRI images for input into the deep learning model. This module typically includes several steps to ensure that the images are in a suitable format for training. First, pixel values are normalized, often rescaled to a range of [0, 1] to improve model convergence. Next, images are resized to a consistent shape, such as 224x224 pixels, which is a common input size for many CNN architectures. Additionally, data augmentation techniques may be applied to artificially increase the diversity of the training dataset. These techniques can include random rotations, flips, zooms, and shifts, which help the model generalize better by exposing it to various transformations of the input data.

5.2.3. Model Definition Module

The Model Definition Module is responsible for constructing the architecture of the deep learning model used for brain tumor detection. This module typically involves defining a Convolutional Neural Network (CNN) or other architectures, such as U-Net, which are well-suited for image classification and segmentation tasks. The architecture is built using various layers, including convolutional layers (Conv2D) for feature extraction, pooling layers (MaxPooling2D) for downsampling, and fully connected layers (Dense) for classification. The output layer is configured to match the number of classes in the dataset, using a softmax activation function for multi-class classification. This module ensures that the model is designed to effectively learn the features necessary for distinguishing between different types of brain tumors.

5.2.4. Model Training Module

The Model Training Module manages the training process of the deep learning model. Once the model architecture is defined, this module compiles the model using an appropriate optimizer, such as Adam, and a loss function, typically categorical crossentropy for multi-class classification tasks. The training process involves feeding the model with the training dataset, allowing it to learn the underlying patterns and features associated with each class. During training, the model is validated using a separate validation dataset to monitor its performance and prevent overfitting. This module also includes functionality to save the trained model to disk, allowing for future use without the need to retrain, which can be time-consuming.

5.2.5. Model Evaluation Module

The Model Evaluation Module is responsible for assessing the performance of the trained model on a test dataset that it has not seen during training. This module calculates various performance metrics, such as accuracy, precision, recall, and F1-score, to provide a comprehensive understanding of the model's effectiveness. Additionally, it may generate confusion matrices to visualize the model's predictions against the true labels, helping to identify areas where the model may be misclassifying certain classes. This evaluation is crucial for understanding the model's strengths and weaknesses, guiding further improvements or adjustments to the model architecture or training process.

5.2.6. Prediction Module

The Prediction Module handles the process of making predictions on new, unseen MRI images. After the model has been trained and evaluated, this module allows users to input new images for classification. It begins by loading the trained model from disk, ensuring that the most recent version is used for predictions. The new images are preprocessed in the same manner as the training data, including resizing and normalization. Once the images are prepared, the model makes predictions, outputting class probabilities for each category. The module interprets these results, identifying the predicted class and the associated confidence score, which indicates the model's certainty in its prediction.

5.2.7. Visualization Module

The Visualization Module is responsible for presenting the results of the model's predictions in an intuitive and user-friendly manner. This module typically includes functionality to display the uploaded MRI images alongside the model's predictions, allowing users to visually assess the results. It may also show confidence scores, providing insight into how certain the model is about its predictions. Additionally, this module can visualize training and validation metrics over epochs, such as loss and accuracy curves, helping users understand the model's learning process and performance trends. Effective visualization is key to interpreting the results and gaining trust in the model's predictions.

5.2.8. User Interface Module (Streamlit)

The User Interface Module, often built using Streamlit, serves as an interactive platform for users to engage with the brain tumor detection system, allowing clinicians or users to easily upload MRI images for analysis. This module features a user-friendly layout that includes a file uploader for various image formats (e.g., JPG, PNG), and once an image is uploaded, it processes the image through predefined preprocessing steps before making predictions with the trained model. The results, including the predicted class and confidence score, are displayed alongside the uploaded image, providing clear and immediate feedback. Streamlit's dynamic capabilities enable real-time updates, enhancing user experience and trust in the model's predictions. Overall, this module acts as a crucial bridge between the complex deep learning model and end-users, ensuring accessibility and practicality in clinical settings.

CHAPTER 6
SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

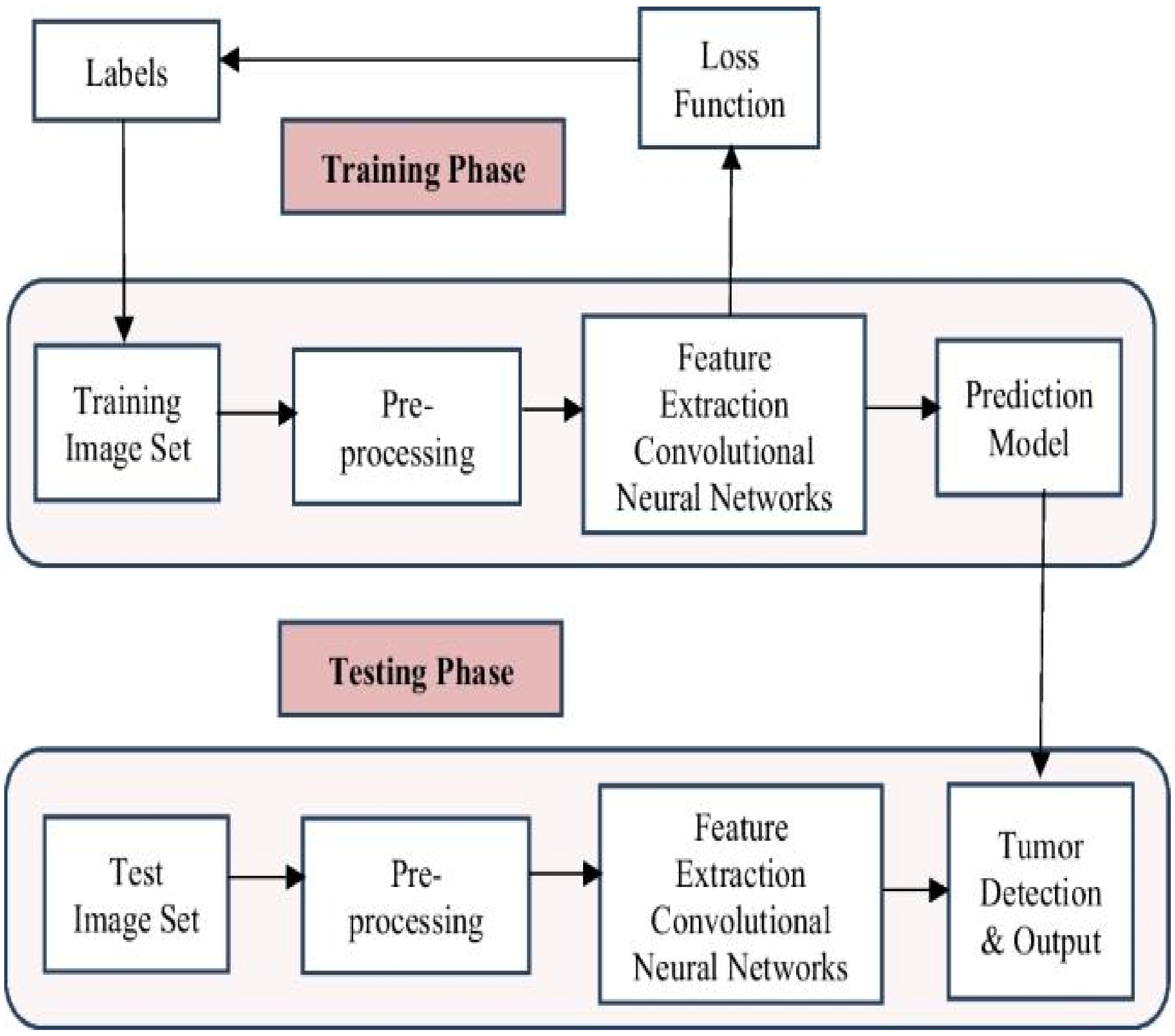


Figure 6.1 System Architecture

6.2 USE CASE DIAGRAM

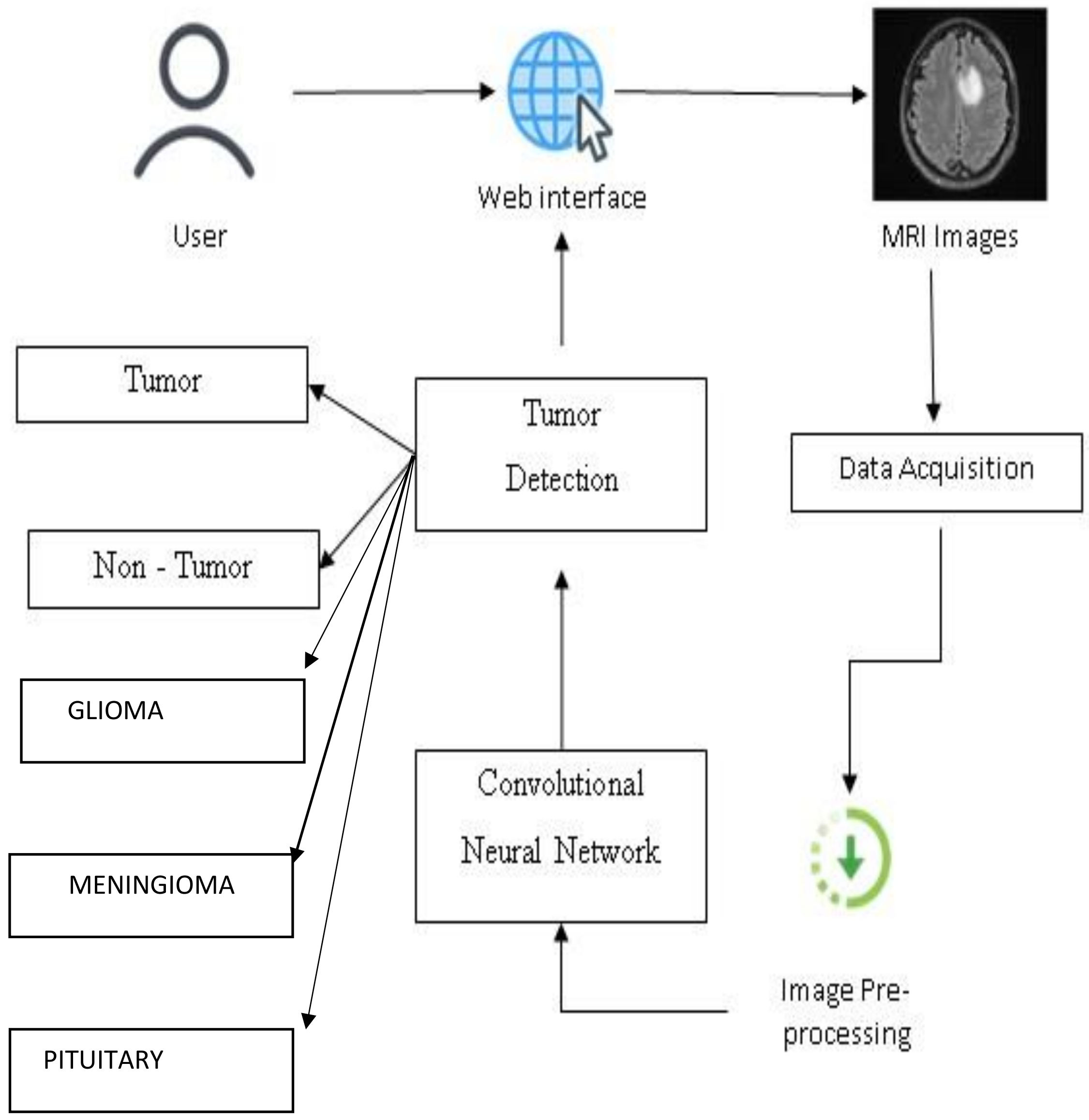


Figure 6.2 Use Case Diagram

6.3 CLASS DIAGRAM

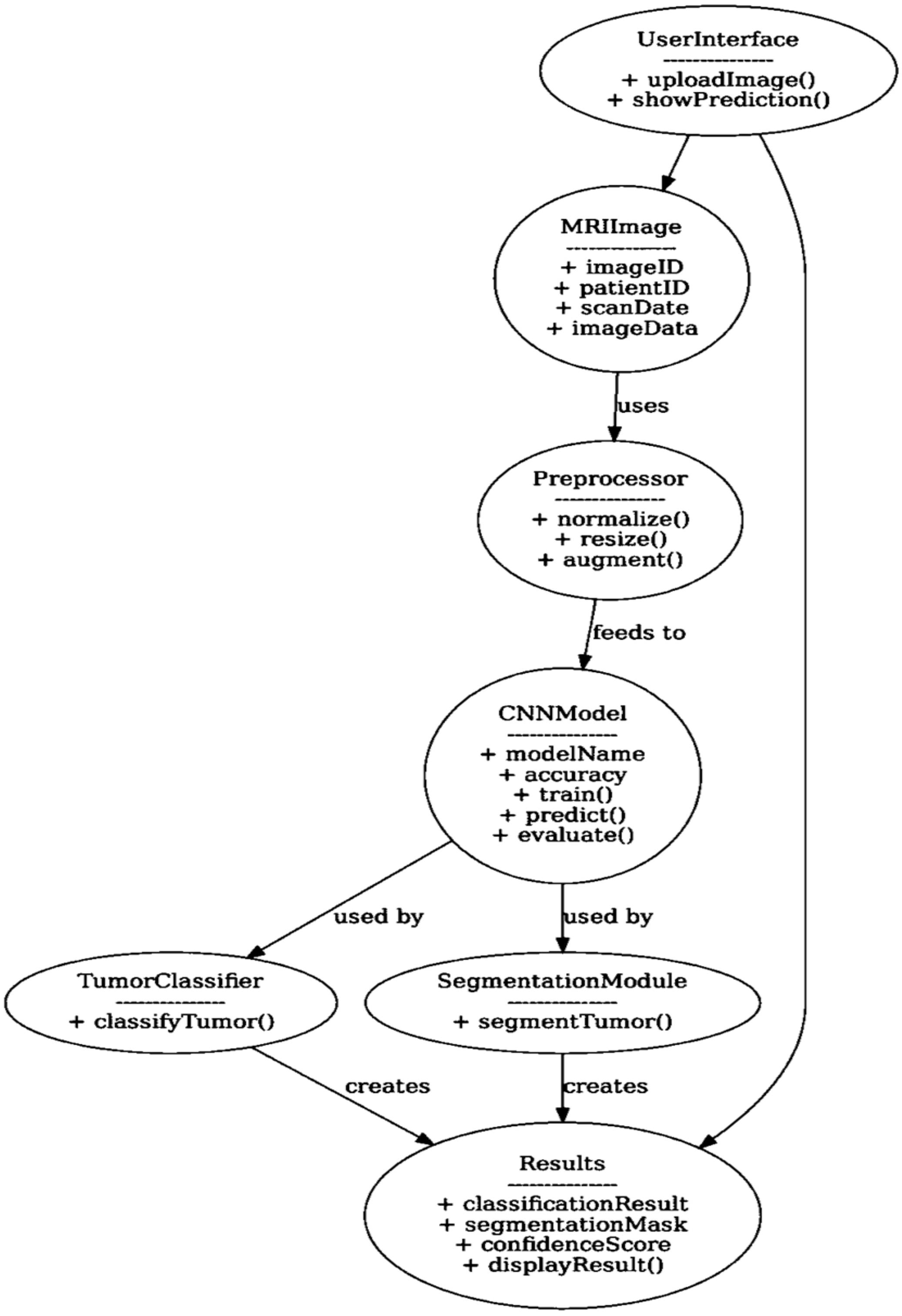


Figure 6.3 Class Diagram

6.4 SEQUENCE DIAGRAM

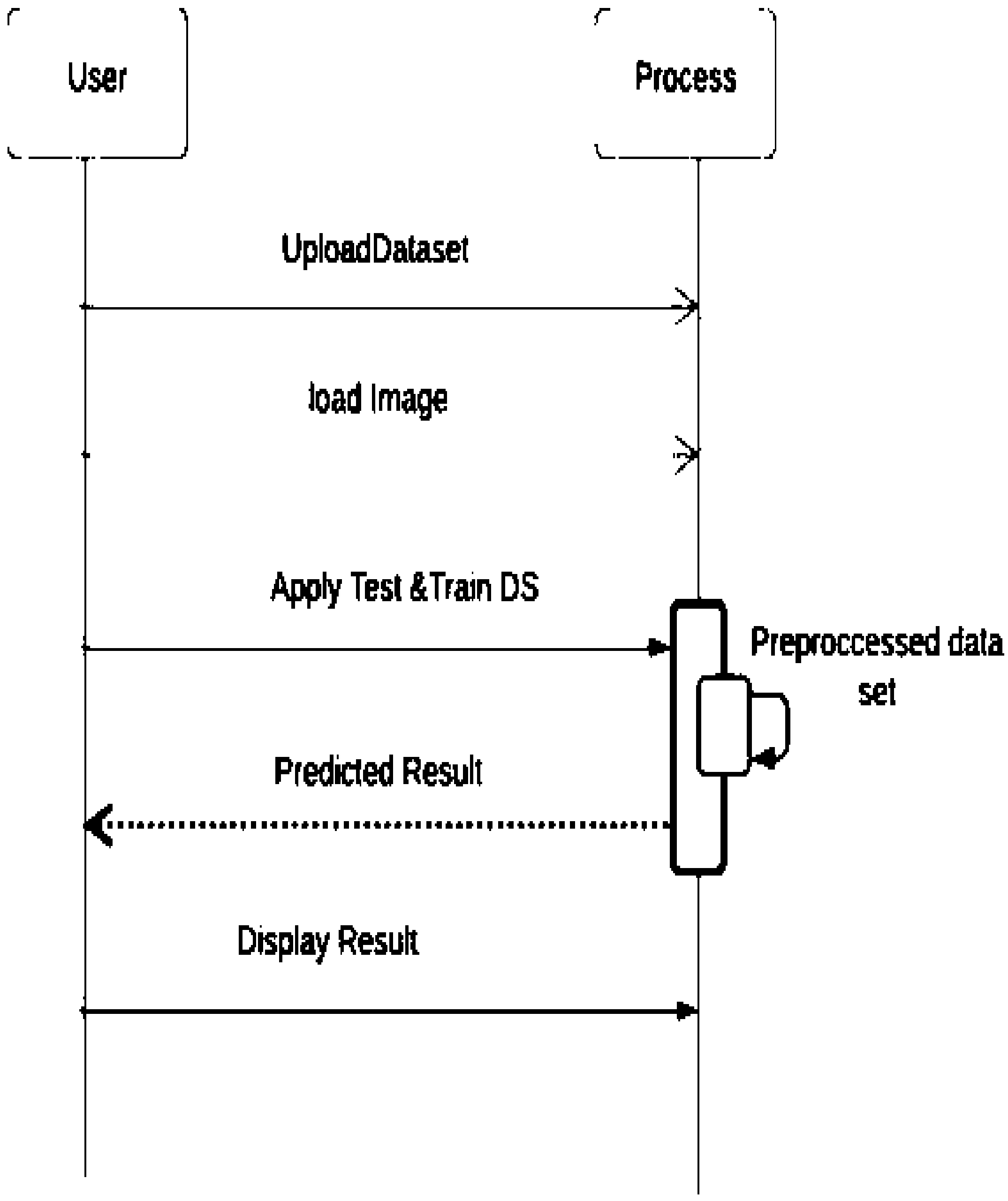


Figure 6.4 Sequence Diagram

6.5 ACTIVITY DIAGRAM

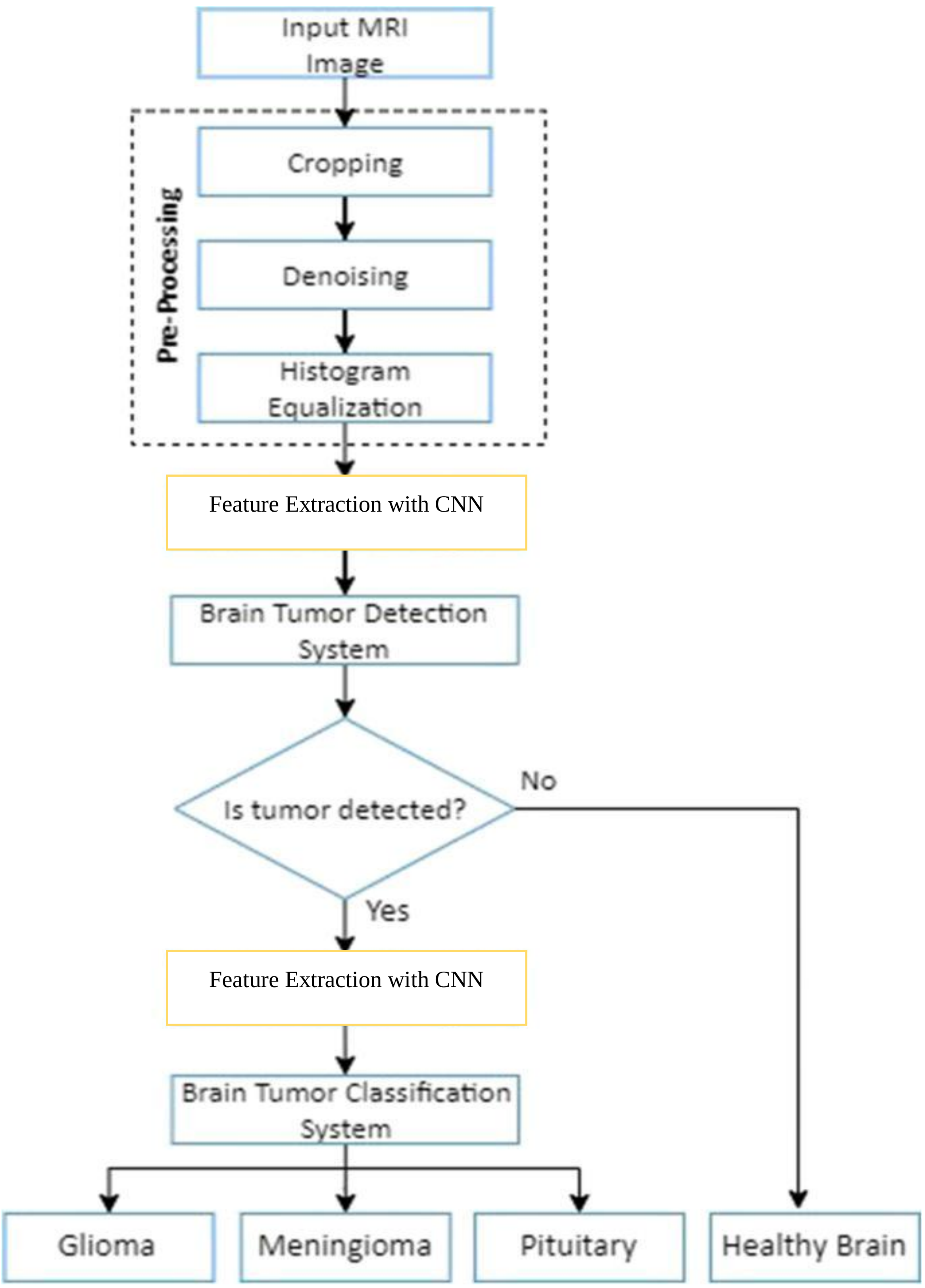


Figure 6.5 Activity Diagram

CHAPTER 7
SOFTWARE DESCRIPTION

7.1 OBJECTIVE OF BRAIN TUMOR DETECTION

The objective of this software is to automate the detection and classification of brain tumors from MRI images using Convolutional Neural Networks (CNNs). The system can accurately predict whether a brain tutor is present and classify its type (if applicable). This software leverages deep learning techniques, specifically CNNs, to analyze MRI scan images for medical purposes.

7.2 KEY FEATURES

MRI Image Preprocessing:

The software preprocesses MRI images to improve model performance. This includes resizing, normalization, and augmentation (rotation, flipping, scaling).

CNN Architecture:

The core of the system is a CNN that extracts features from the MRI scans. The architecture consists of multiple convolutional layers, pooling layers, and fully connected layers that help in the accurate identification of tumor regions.

Segmentation:

Optionally, the software can use segmentation techniques (like U-Net or 3D U-Net) to highlight tumor regions in MRI images for better visualization.

Classification:

The model can classify tumors into different categories, such as benign or malignant.

Evaluation Metrics:

The software evaluates the model's performance using metrics like accuracy, precision, recall, F1 score, and Dice coefficient for segmentation tasks.

Output:

The system provides a detailed output, which includes the classification of the tumor type and, in case of segmentation, a mask showing the detected tumor area.

7.3 SOFTWARE COMPONENTS

7.3.1. INPUT LAYER

MRI Dataset: Input images are typically 2D or 3D MRI scans. Popular datasets for training include BraTS (Brain Tumor Segmentation Challenge) and Kaggle Brain MRI Dataset.

Preprocessing:

Images are resized to fit the input shape of the CNN model (e.g., 224x224 pixels). Normalization is applied to scale pixel values to a range of [0,1].

7.3.2.CNN ARCHITECTURE

Convolutional Layers:

The CNN model uses several convolutional layers to extract features from the MRI scans. Filters (kernels) slide over the input images to detect edges, textures, and patterns associated with tumor regions.

Activation Function:

After each convolutional layer, a ReLU (Rectified Linear Unit) activation function is applied to introduce non-linearity.

Pooling Layers:

Max pooling is applied after convolution layers to reduce the spatial dimensions (height and width) of the image, making the model computationally efficient.

Fully Connected Layers:

These layers are used after the feature extraction process to classify the image based on the learned features.

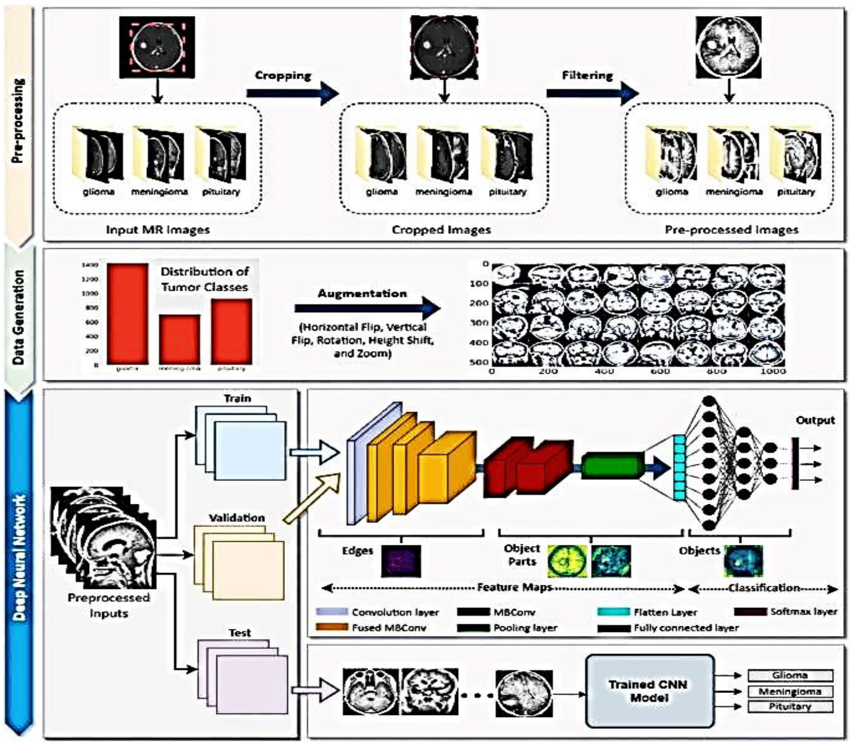


Figure 7.3.2 CNN Architecture

7.3.3. TRAINING AND VALIDATION

Loss Function :

Categorical Cross-Entropy (for multi-class classification) or Binary Cross-Entropy (for binary classification) is used to calculate the error between predicted and true labels.

Optimizer:

Adam optimizer is typically used to update the weights of the CNN model during training.

Metrics:

The model's performance is evaluated using accuracy, precision, recall, F1 score, and Dice coefficient for segmentation tasks.

7.3.4. SEGMENTATION (OPTIONAL.)

U-Net / 3D U-Net:

U -Net is often used in medical image segmentation. It consists of an encoder-decoder structure that helps to segment the tumor region in MRI images accurately.

Dice Coefficient:

A metric used to evaluate the overlap between the predicted tumor mask and the ground truth mask..

7.3.5. Post-Processing

Tumor Localization:

In the segmentation phase, the software generates a mask that highlights the tumor's location in the MRI image.

Overlay:

The detected tumor area is overlaid on the original MRI image for easy visualization by healthcare professionals.

7.4 WORKFLOW

- **Input Data:** Users upload MRI images (single or batch).
- **Preprocessing**: The images are preprocessed for size, normalization, and augmentation.
- **Model Training**: The CNN model is trained on the preprocessed dataset to learn the features of tumors.
- **Evaluation**: The model is validated on a separate dataset, and performance metrics are calculated.
- **Prediction**: For new MRI scans, the model predicts tumor presence and classifies it.
- **Segmentation**: If tumor segmentation is enabled, the model identifies and visualizes the tumor region in the image.
- **Output**: The result is provided as a classification (tumor or no tumor) or a segmented image showing the tumor area.

7.5 TECHNOLOGY USED

- **Deep Learning Frameworks:** TensorFlow, Keras, or PyTorch for building and training the CNN model.
- **Python Libraries:** NumPy and OpenCV for image processing, Matplotlib for visualization.
- **Dataset**: BraTS (for brain tumor segmentation) or Kaggle MRI datasets.

CHAPTER 8

SOFTWARE TESTING

8.1 AIM OF TESTING

The primary aim of testing a brain tumor detection system built using deep learning and Convolutional Neural Networks (CNNs) with MRI datasets is to ensure the system functions correctly, reliably, and efficiently. It is essential to verify that the software can accurately detect and classify tumors (e.g., benign or malignant) from MRI scans. Testing aims to validate the correctness of predictions, robustness against various input conditions, efficiency in processing time, and consistency with expert medical evaluations. Furthermore, testing ensures the system is user-friendly, secure, and meets all functional and non-functional requirements defined in the development process.

8.2 TEST CASE

In the context of this software, a test case is a specific scenario designed to check a particular functionality of the system. For instance, a test case might involve inputting an MRI image to verify if the system can successfully preprocess the image, feed it through the CNN model, and produce the correct classification (such as “tumor detected” or “no tumor”).

Another test case might involve checking the segmentation mask output to ensure it aligns accurately with known tumor regions. Edge cases, like providing a corrupted image file or an unsupported format, are also tested to see if the software handles errors gracefully. Each test case has a defined input, an expected output, and a pass/fail status based on the actual output.

8.3 TYPES OF TESTING

Several types of software testing are applied throughout the development and validation of the brain tumor detection system. These include:

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Performance Testing
- Validation Testing
- Regression Testing
- Usability Testing

8.4 DESCRIPTION OF TESTING

8.4.1. UNIT TESTING:

Unit Testing involves testing the smallest parts of the system, such as individual functions or components. In this project, it includes testing the preprocessing functions that resize and normalize MRI images, or the operations of a single CNN layer, to ensure they work correctly and return the expected results.

8.4.2. INTEGRATION TESTING:

Integration Testing ensures that different modules or components of the software interact correctly with each other. For example, after preprocessing, the output image should be correctly fed into the CNN model. Integration testing verifies that this data flow works as expected and that the system components are properly integrated.

8.4.3. FUNCTIONAL TESTING:

Testing focuses on verifying that the software performs its core functionalities according to the requirements. In brain tumor detection, this includes ensuring that the CNN model correctly classifies MRI scans, and if segmentation is included, that it accurately highlights the tumor region in the image.

8.4.4. SYSTEM TESTING:

System Testing is a comprehensive, end-to-end test of the complete software system. It checks whether the entire pipeline—from image input and preprocessing to prediction and output generation—works as a cohesive unit and meets the software specifications.

8.4.5. PERFORMANCE TESTING:

Performance Testing evaluates how efficiently the system performs under different conditions. It involves measuring the time taken to process images, the speed of inference during prediction, and how the system handles large MRI datasets. It also includes memory usage and computational efficiency, especially important when working with deep learning models.

8.4.6. VALIDATION TESTING:

Validation Testing is conducted to ensure the model's predictions are accurate and medically meaningful. This involves comparing the software's output with a labeled dataset or expert-annotated ground truth. For classification, it ensures the system can reliably distinguish between different tumor types or detect the presence of a tumor. For segmentation, it checks how well the predicted tumor mask overlaps with the actual tumor region using metrics like Dice coefficient or Intersection over Union (IoU).

8.4.7. REGRESSION TESTING:

Regression Testing is essential whenever the model or software is updated, such as retraining the CNN with a larger dataset or adding a new preprocessing step. It ensures that new changes do not break existing functionalities or reduce the system’s accuracy.

8.4.8. USABILITY TESTING:

Testing assesses the ease of use of the software. It ensures that healthcare professionals or researchers using the system can easily upload MRI images, interpret the predictions, view segmentation overlays, and export reports. The goal is to make the interface and output understandable and accessible to non-technical users.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

9.1 CONCLUSION

The development of a brain tumor detection system using deep learning and Convolutional Neural Networks (CNNs) with MRI datasets has shown significant promise in automating and improving the accuracy of tumor diagnosis. This approach leverages the power of CNNs to extract meaningful features from complex MRI images, enabling the system to classify and, in some cases, segment brain tumors with high precision. Through effective preprocessing, model training, and validation, the system can assist medical professionals in early detection and decision-making. The implementation has demonstrated that deep learning not only reduces human error but also accelerates the diagnostic process, leading to better patient outcomes. Overall, the project proves that AI-driven solutions can play a vital role in the field of medical imaging and diagnostics.

9.2 FUTURE ENHANCEMENT

The brain tumor detection system using deep learning and CNNs can be significantly improved through various future enhancements. One major advancement would be the integration of 3D MRI data and volumetric analysis, allowing the model to understand spatial structures more effectively and improve tumor localization accuracy.Additionally, incorporating more advanced segmentation models like U-Net or Mask R-CNN can enhance the precision of tumor boundary detection. To facilitate real-world use, optimizing the model for real-time inference on low-resource devices such as mobile or embedded platforms is essential. Another critical enhancement would be the adoption of explainable AI techniques, which can help visualize the model’s decision-making process and build trust among medical professionals.

APPENDICES

APPENDIX 1

SOURCE CODE

Tumor.Py

```
import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import os

# Define the CNN model

model = Sequential([

    Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),

    MaxPooling2D(pool_size=(2,2)),

    Conv2D(64, (3,3), activation='relu'),

    MaxPooling2D(pool_size=(2,2)),

    Flatten(),

    Dense(128, activation='relu'),

    Dropout(0.5),

    Dense(4, activation='softmax')  # 4 classes: Meningioma, Glioma, Pituitary, No Tumor

])
```

```
# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Data preprocessing and augmentation

datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_generator = datagen.flow_from_directory( r"D:\mini project\archive\Training",
target_size=(224, 224), batch_size=32, class_mode='categorical', subset='training')

val_generator = datagen.flow_from_directory ( r"D:\mini project\archive\Testing",
target_size=(224, 224), batch_size=32, class_mode='categorical', subset='validation')

# Train the model

history = model.fit(train_generator, validation_data=val_generator, epochs=10)

# Save the trained model

if not os.path.exists('saved_model'):

    os.makedirs('saved_model')

model.save('saved_model/brain_tumor_classifier.h5')

print(" Model trained and saved successfully!")
```

APP.PY

```
import streamlit as st

import tensorflow as tf

from tensorflow.keras.preprocessing.image import load_img, img_to_array

import numpy as np

# Load model

model = tf.keras.models.load_model(r"D:\mini
project\saved_model\brain_tumor_classifier.h5")

CLASS_LABELS = ["Glioma","Meningioma","No Tumor", "Pituitary"]

def preprocess_image(image):

    img = image.resize((224, 224))

    img_array = img_to_array(img) / 255.0

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

    return img_array

st.title("Brain Tumor Detection")

uploaded_file = st.file_uploader("Upload MRI Image", type=["jpg", "png", "jpeg"])

if uploaded_file is not None:

    image = load_img(uploaded_file)

    st.image(image, caption="Uploaded Image", use_container_width=True)

    img_array = preprocess_image(image)
```

```
predictions = model.predict(img_array)

predicted_class = CLASS_LABELS[np.argmax(predictions)]

confidence = float(np.max(predictions))

st.write(f"### Prediction: {predicted_class}")

st.write(f"### Confidence: {confidence:.2f}")
```

APPENDIX 2

SCREENSHOTS

Webpage with streamlit interface

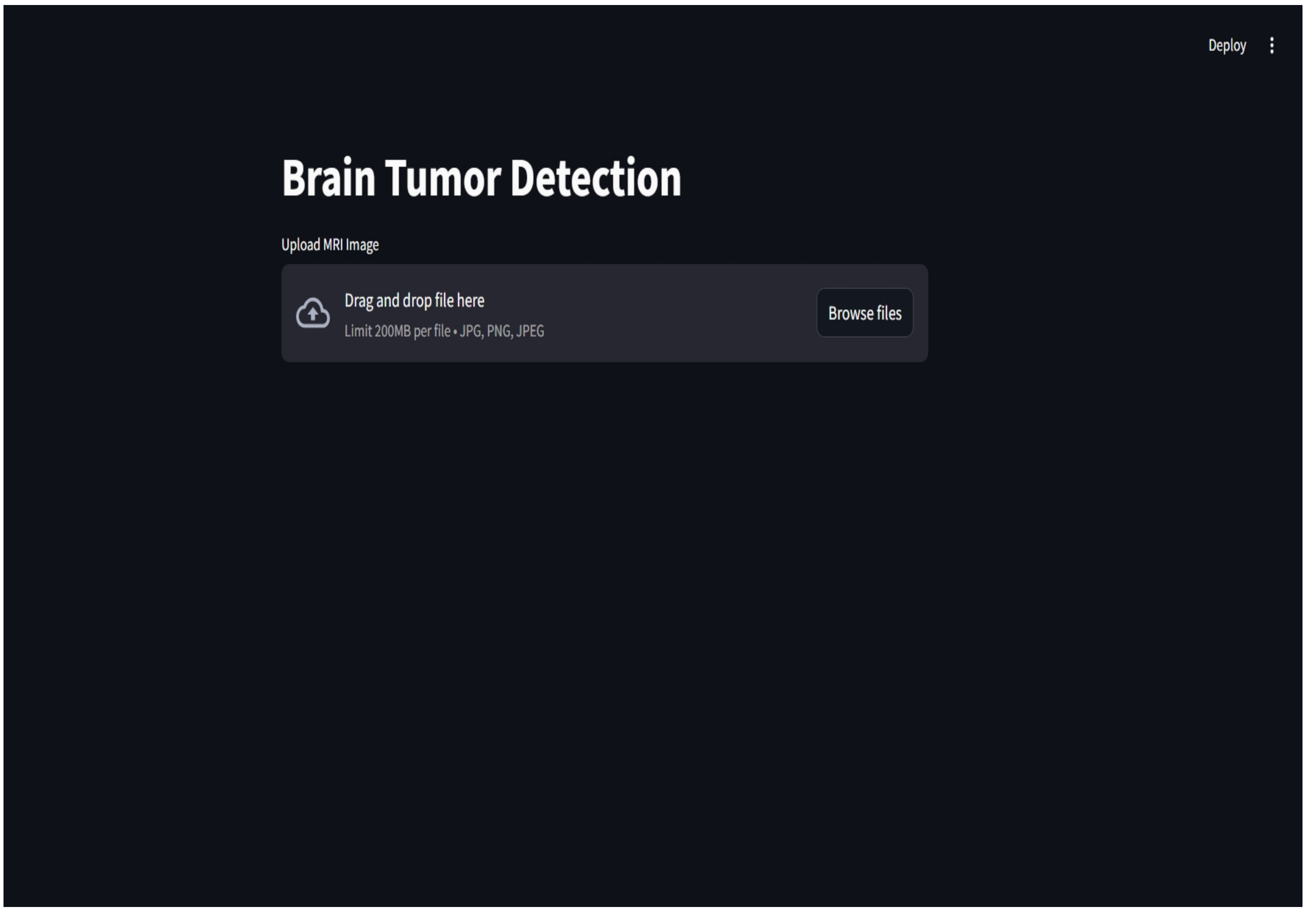
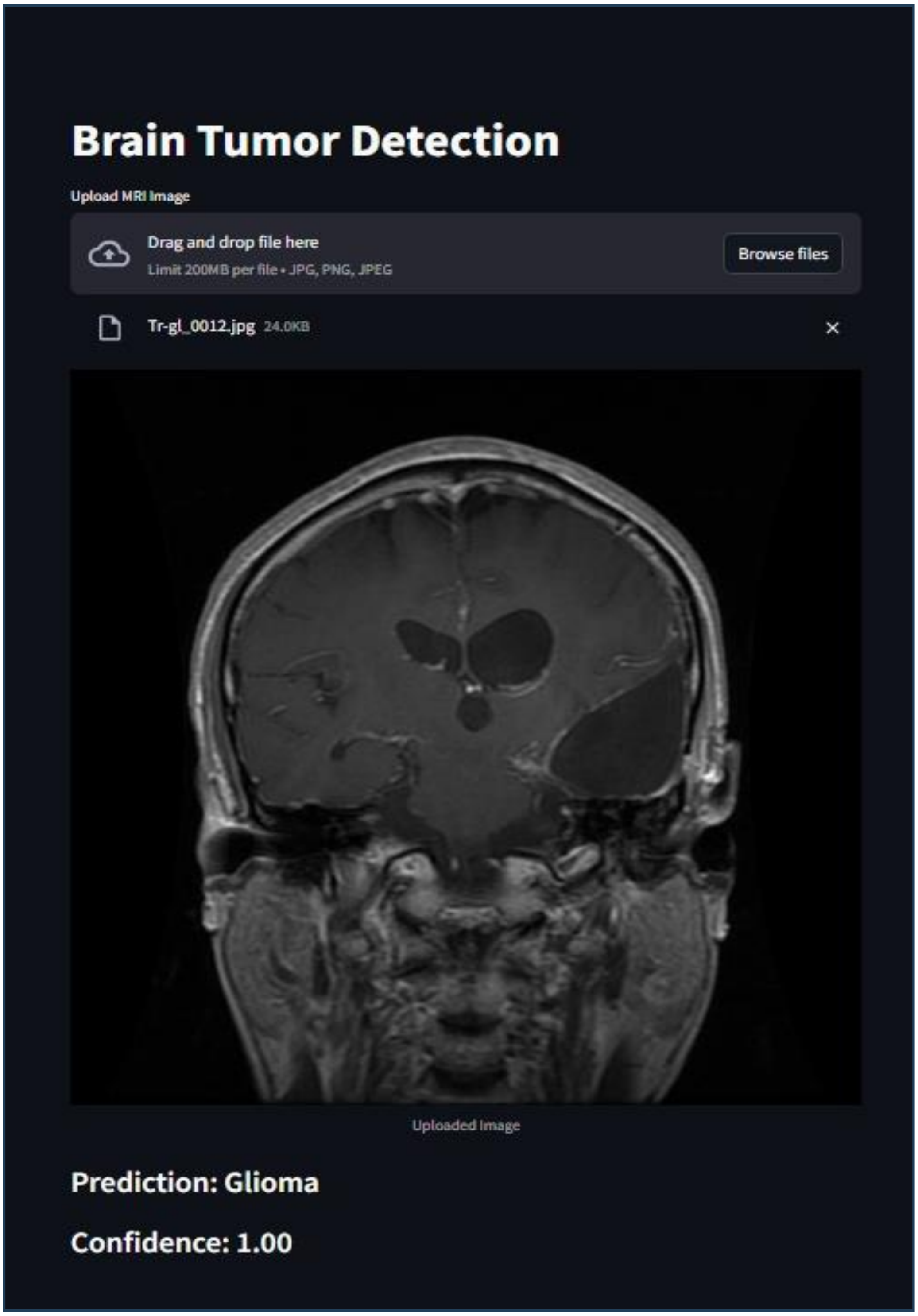


Figure 10.1 Webpage with Streamlit Interface

CLASSIFICATION

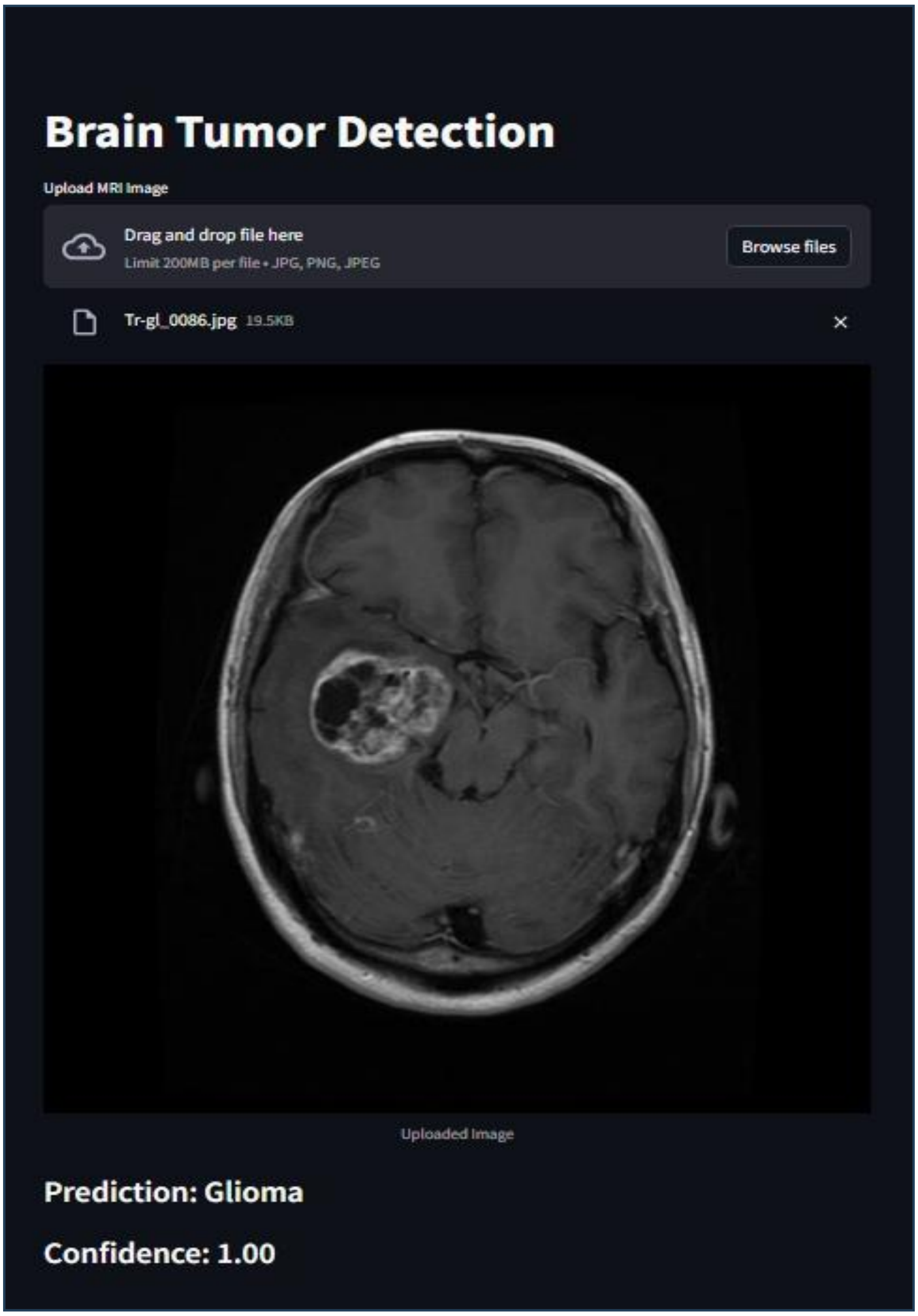
- **GLIOMA**



Prediction: Glioma

Confidence: 1.00

Figure 10.2 Coronal MRI view displaying Glioma

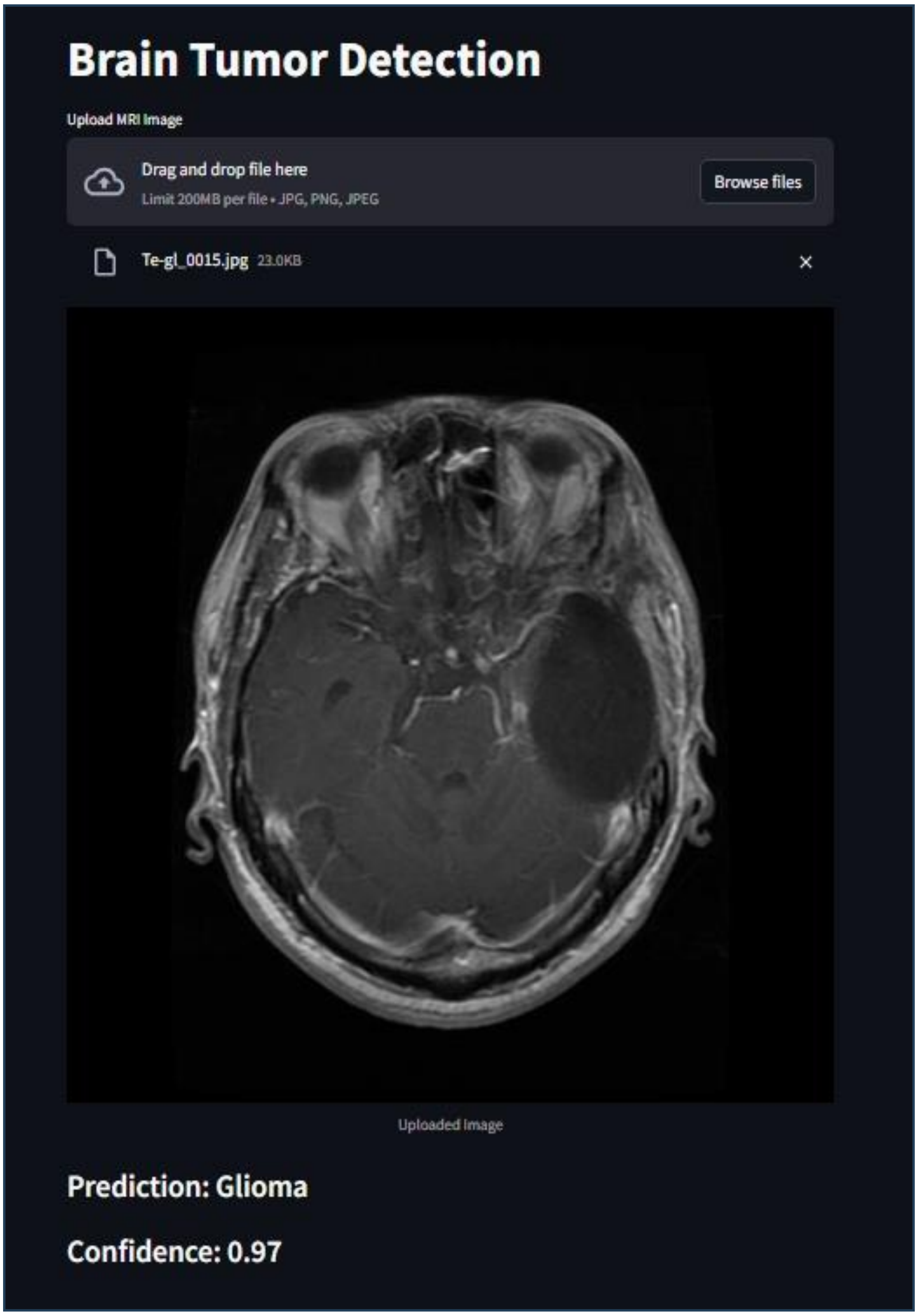

Brain Tumor Detection

Upload MRI Image

Prediction: Glioma

Confidence: 1.00

Figure 10.3 Axial MRI view Displaying Glioma


Brain Tumor Detection

Upload MRI Image

Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Browse files

Te-gl_0015.jpg 23.0KB

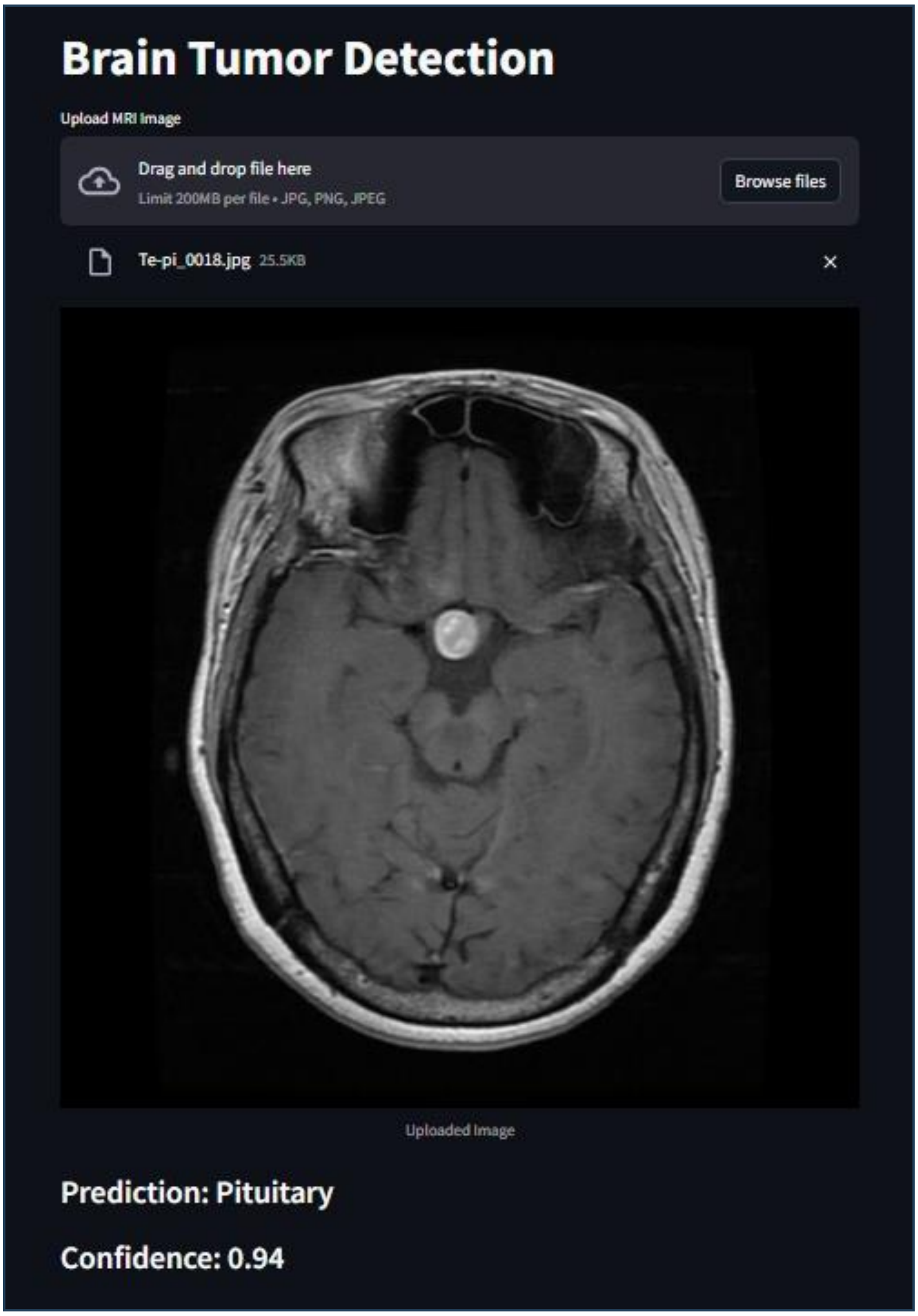
Uploaded Image

Prediction: Glioma

Confidence: 0.97

Figure 10.4 Axial MRI view Displaying Glioma

- **PITUITARY TUMOR**


Brain Tumor Detection

Upload MRI Image

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

Browse files

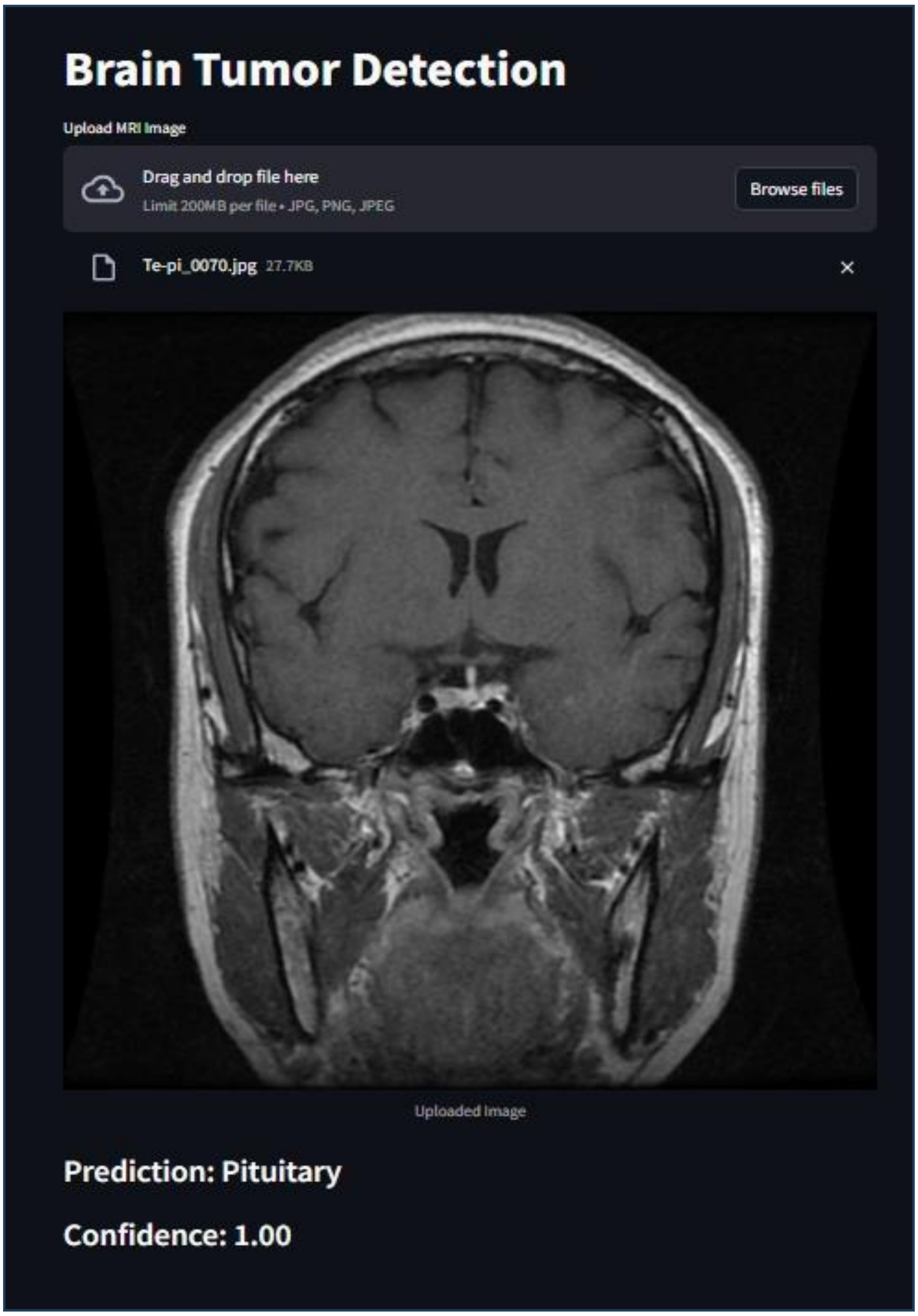
Te-pi_0018.jpg 25.5KB

Uploaded Image

Prediction: Pituitary

Confidence: 0.94

Figure 10.5 Axial MRI view displaying Pituitary



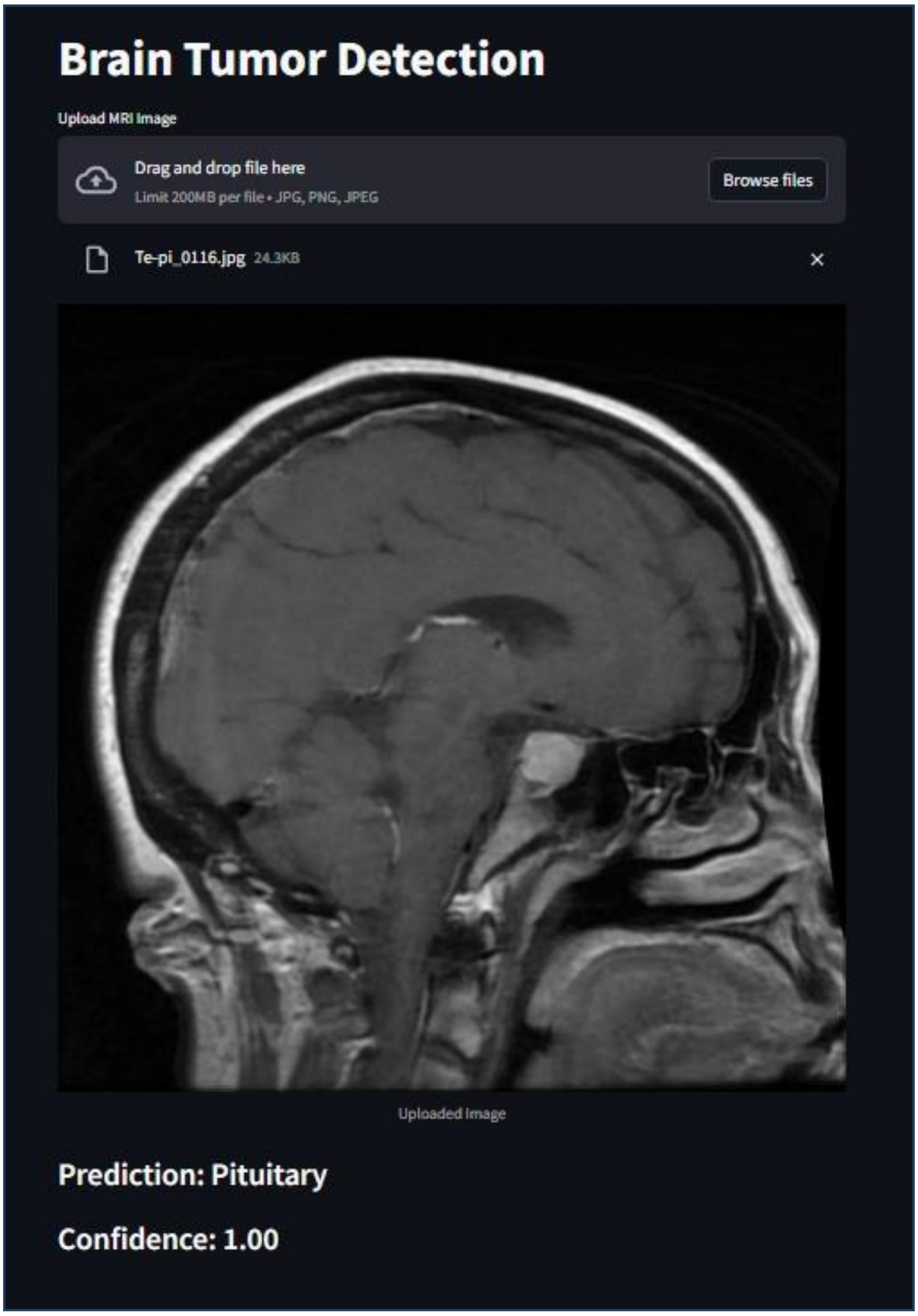
Brain Tumor Detection

Upload MRI Image

Prediction: Pituitary

Confidence: 1.00

Figure 10.6 Coronal MRI view displaying Pituitary



Brain Tumor Detection

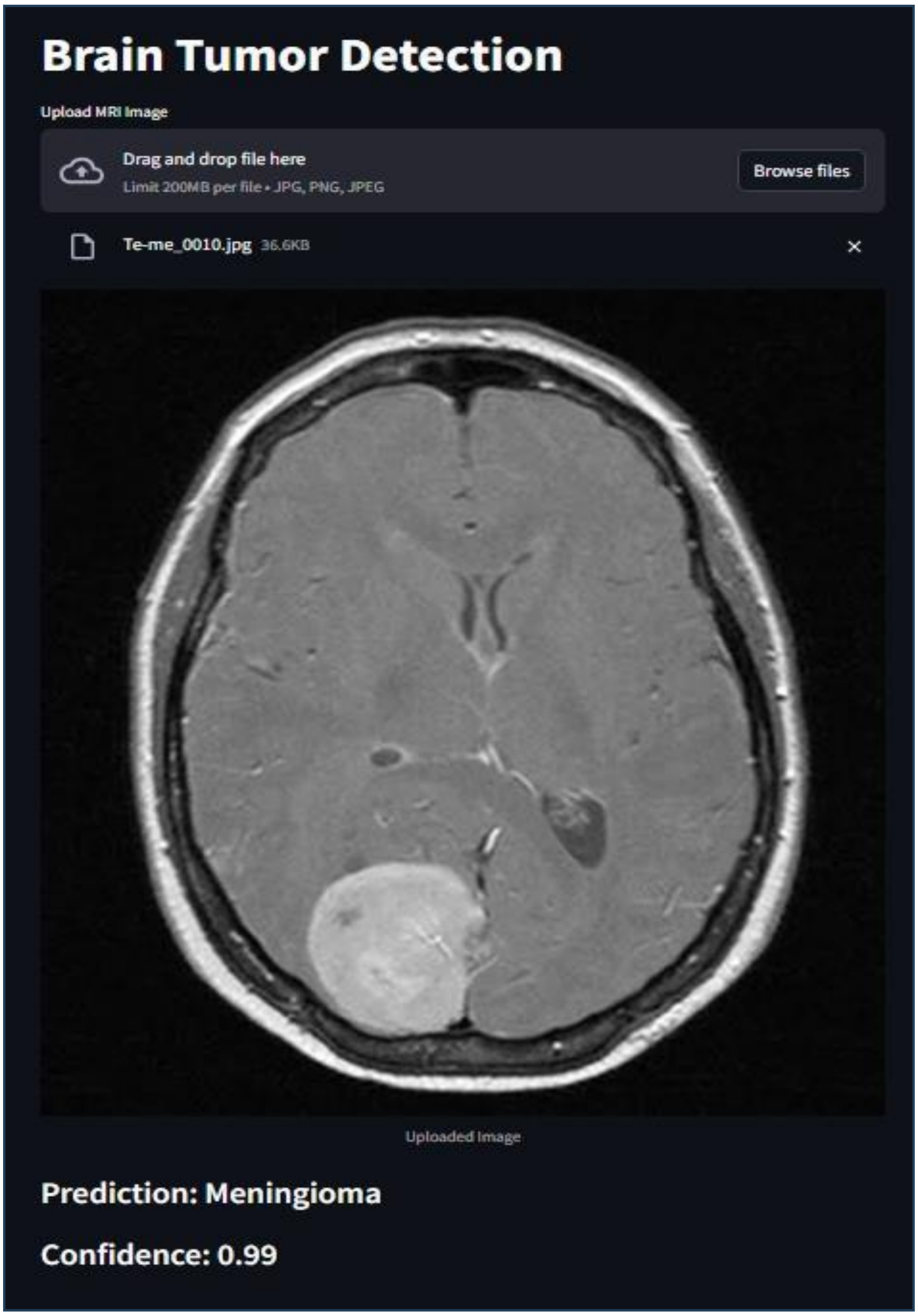
Prediction: Pituitary

Confidence: 1.00

Figure 10.7. Sagittal MRI view displaying Pituitary

- **MENINGIOMA**

2.



Brain Tumor Detection

Upload MRI Image

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

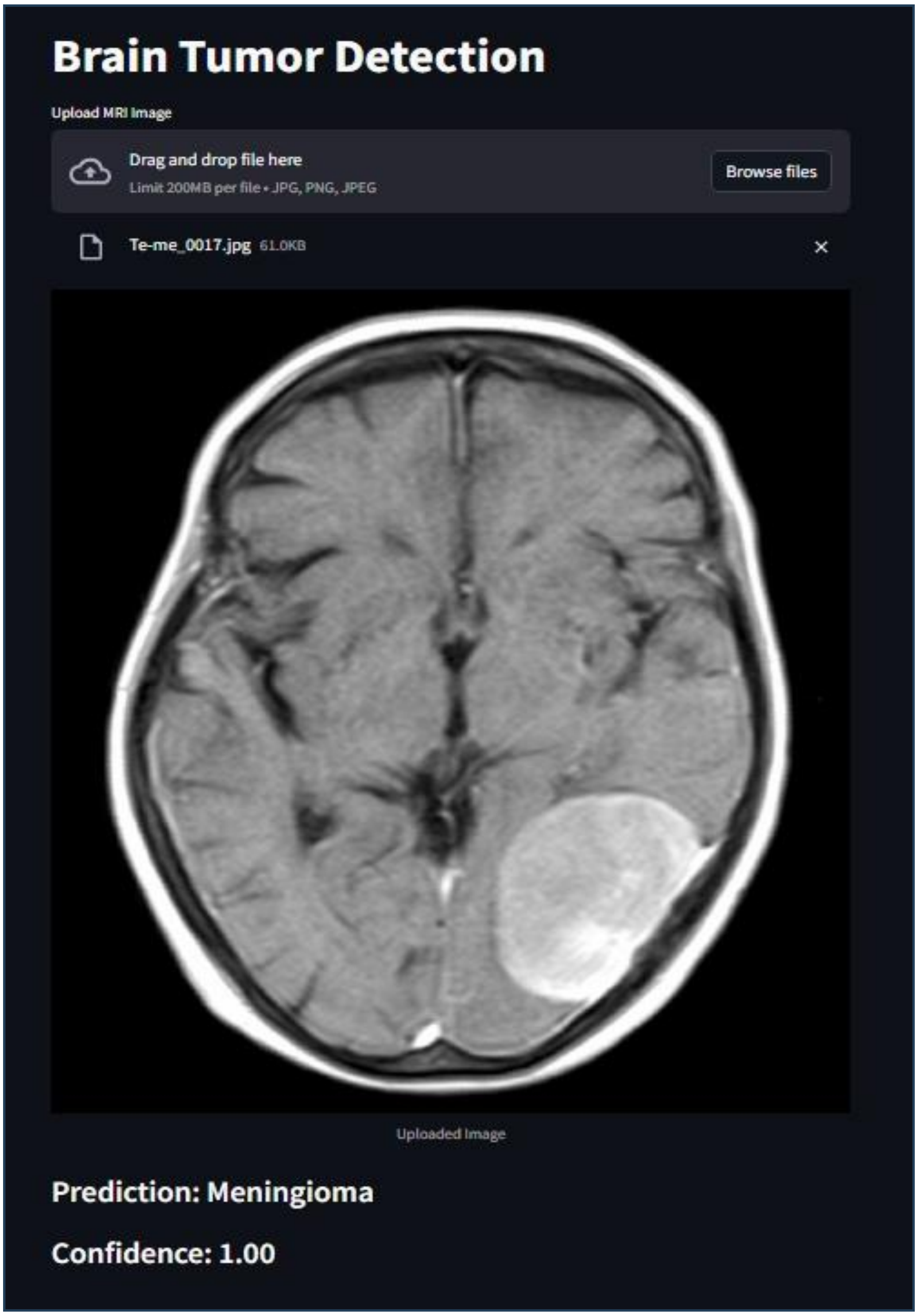
Te-me_0010.jpg 36.6KB

Uploaded Image

Prediction: Meningioma

Confidence: 0.99

Figure 10.8 Axial MRI view displaying Meningioma


Brain Tumor Detection

Upload MRI Image

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

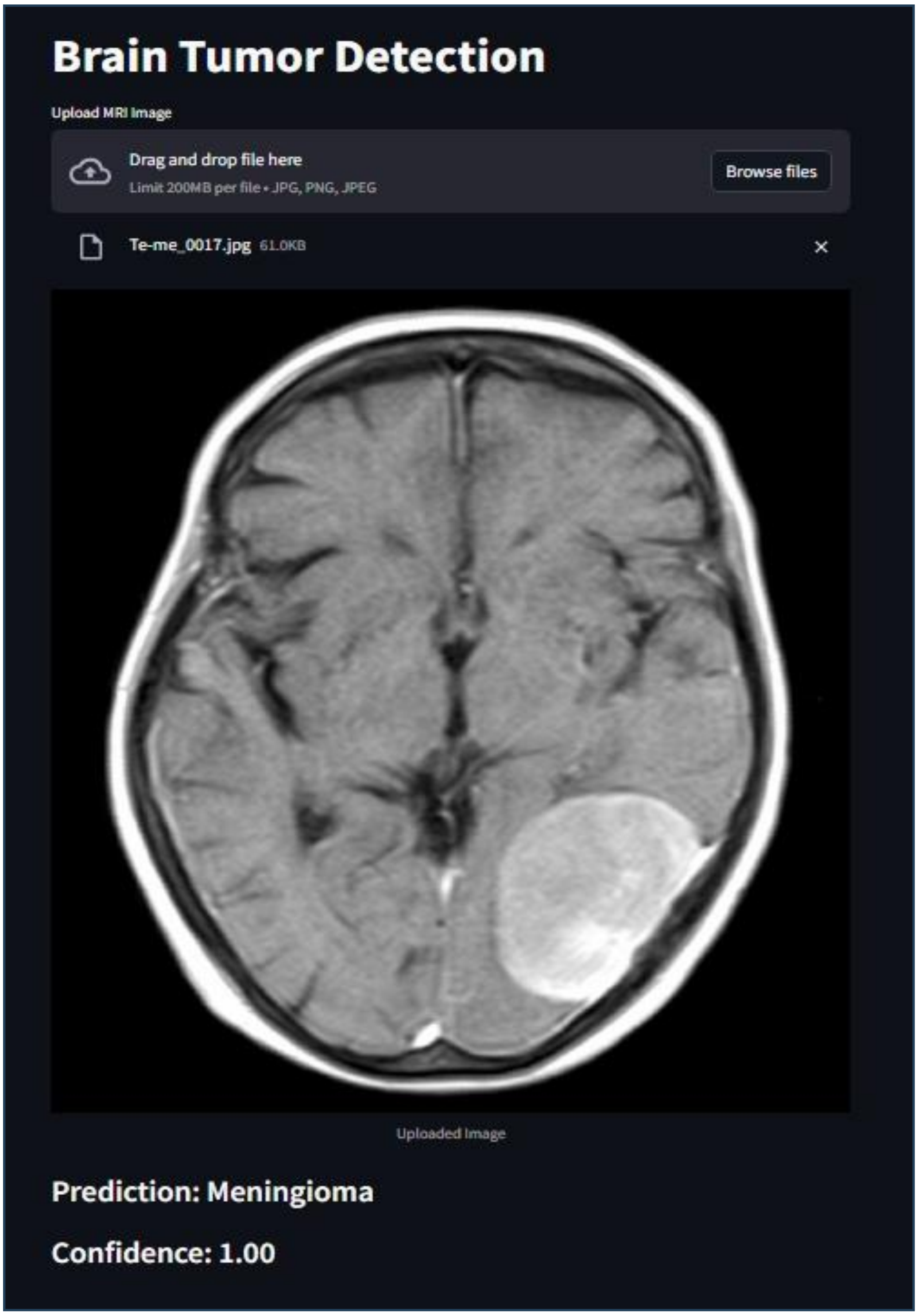
Te-me_0017.jpg 61.0KB

Uploaded Image

Prediction: Meningioma

Confidence: 1.00

Figure 10.9. Coronal MRI view displaying Meningioma



Brain Tumor Detection

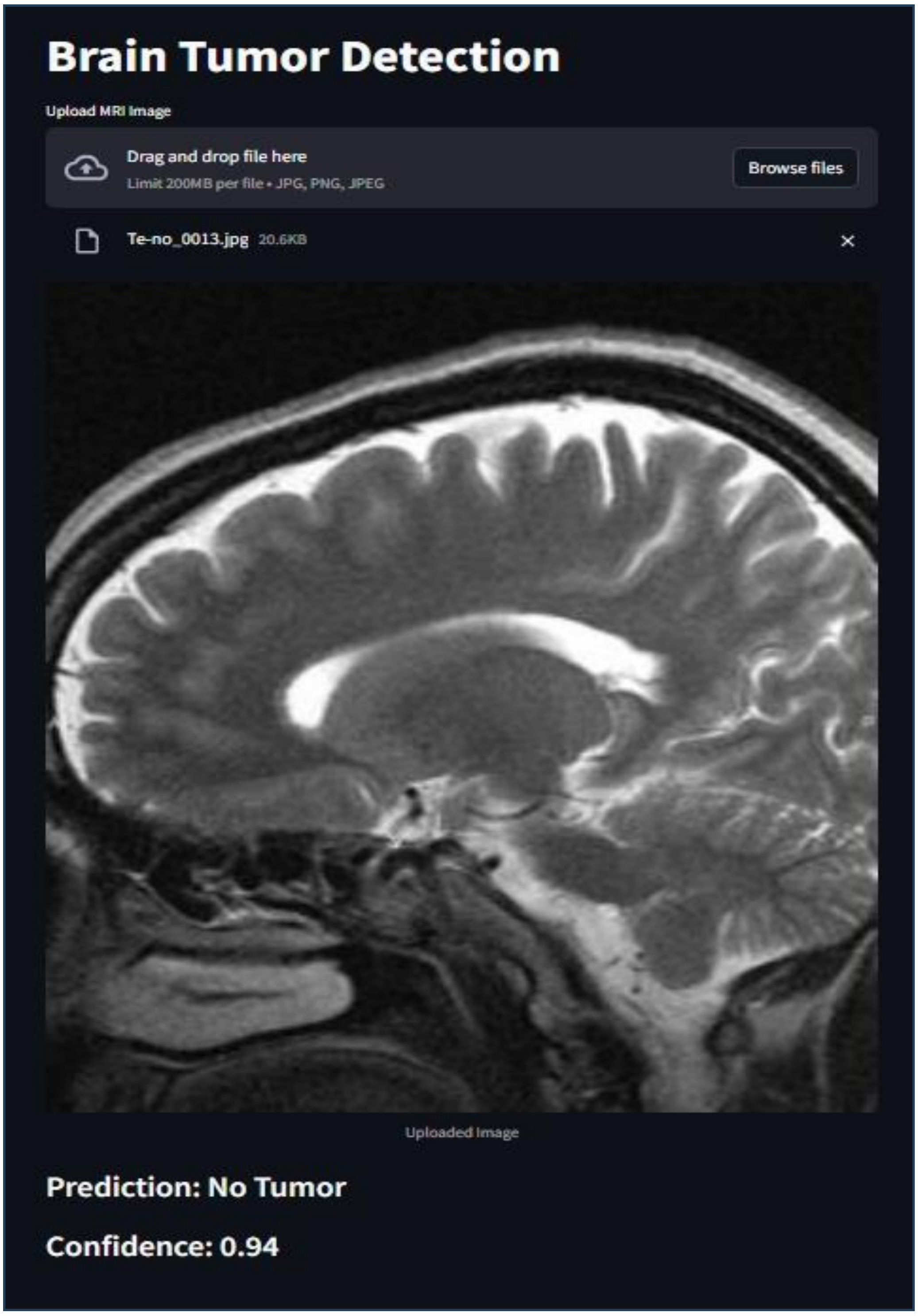
Upload MRI Image

Prediction: Meningioma

Confidence: 1.00

Figure 10.10. Coronal MRI view displaying Meningioma

- **NO TUMOR**

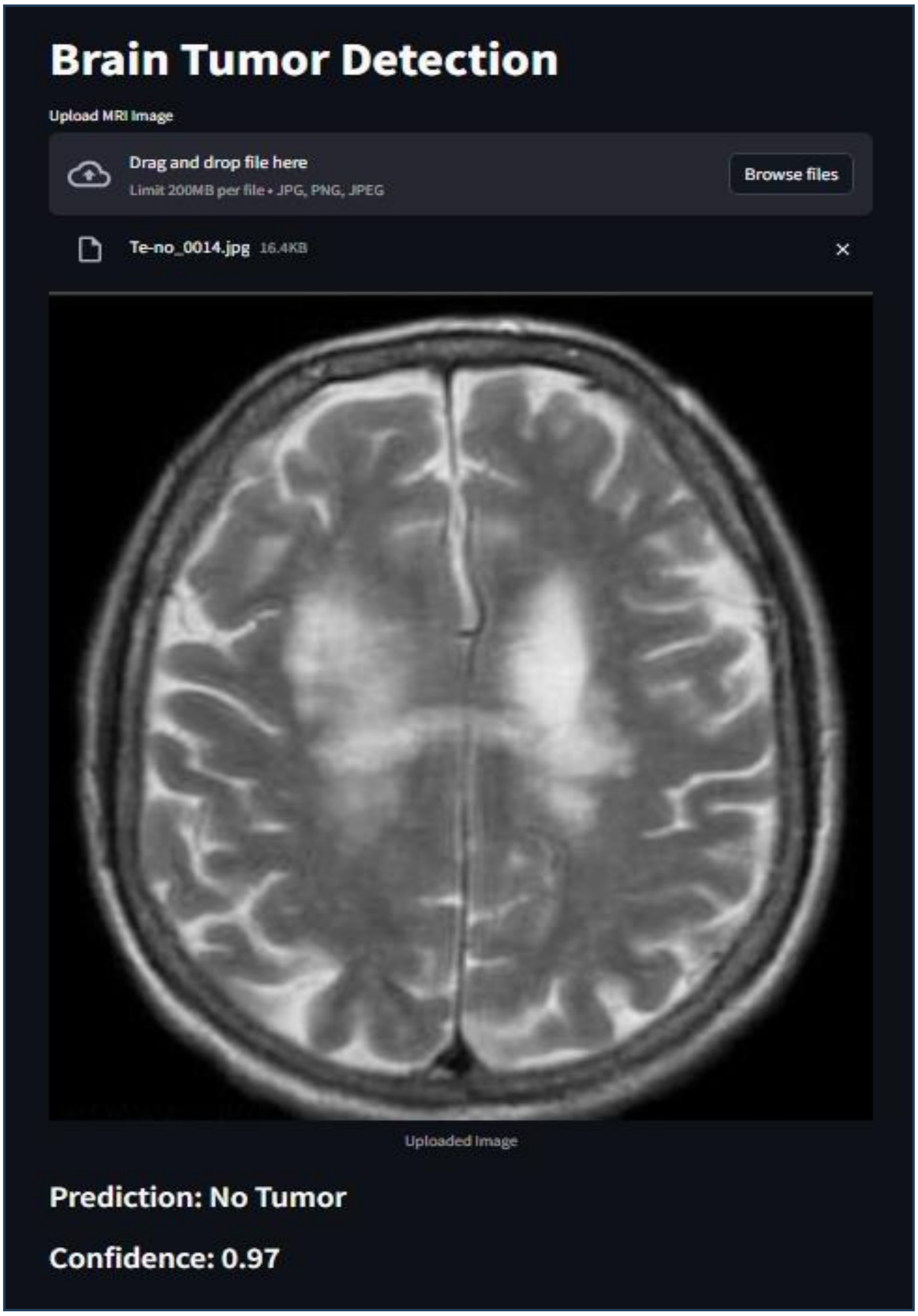


Brain Tumor Detection

Prediction: No Tumor

Confidence: 0.94

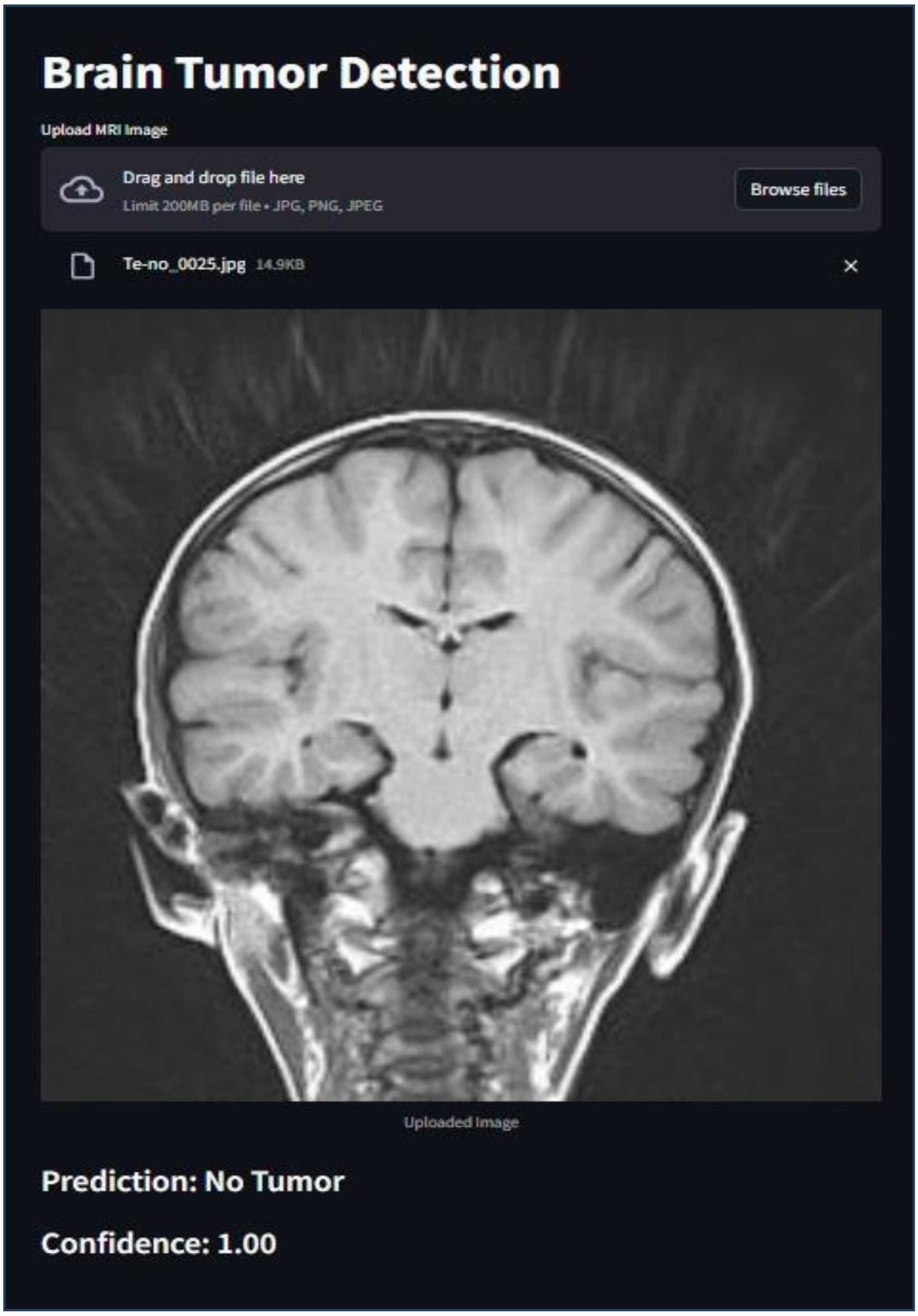
Figure 10.11. Sagittal MRI view displaying No Tumor



Prediction: No Tumor

Confidence: 0.97

Figure 10.12. Axial MRI view displaying No Tumor



Brain Tumor Detection

Upload MRI Image

Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG

Te-no_0025.jpg 14.9KB

Uploaded Image

Prediction: No Tumor

Confidence: 1.00

Figure 10.13. Coronal MRI view displaying No Tumor

REFERENCES

1. Prabukumar M, Agilandeeswari L, Ganesan K. An intelligent lung cancer diagnosis system using cuckoo search optimization and support vector machine classifier. J Ambient Intell Humaniz Comput. 2019;10(1):267–93.
2. El-Dahshan ESA, Mohsen HM, Revett K, Salem ABM. Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. Expert Syst Appl. 2014;41(11):5526–45.
3. Meng Y, Tang C, Yu J, Meng S, Zhang W. Exposure to lead increases the risk of meningioma and brain cancer: a meta-analysis. J Trace Elem Med Biol. 2020;60:126474.
4. McFaline-Figueroa JR, Lee EQ. Brain tumors. Am J Med. 2018;131(8):874–82.
5. Badža MM, Barjaktarović MČ. Classification of brain tumors from MRI images using a convolutional neural network. Appl Sci. 2020;10(6):1999.
6. Shen D, Wu G, Suk H-I. Deep learning in medical image analysis. Annu Rev Biomed Eng. 2017;19(1):221–48.
7. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal.
8. Suzuki K. Overview of deep learning in medical imaging. Radiol Phys Technol. 2017;10(3):257–73.
9. Hijazi S, Kumar R, Rowen C. Using convolutional neural networks for image recognition. San Jose: Cadence Design Systems Inc; 2015. p. 1–12.
10. O’Shea K, Nash R. An introduction to convolutional neural networks. 2015. arXiv:1511.08458.