**Project Assignment List**

**Assignment -1**

**Problem Statement :**

Developing an accurate and efficient system for detecting, classifying, and segmenting brain tumors in MRI scans.

**Objective :**

The objective of this project is to develop an automated system for brain tumor detection and classification using MRI scans, leveraging deep learning models to accurately identify the presence of a tumor and predict its severity level. This system aims to assist medical professionals in diagnosing brain tumors early and effectively, providing a reliable tool for segmenting tumor regions and assessing tumor grade.

* **Tumor Detection**: Accurately detecting whether a brain MRI image contains a tumor or not, using advanced image processing and machine learning techniques.
* **Tumor Segmentation**: segmenting the tumor levels from the MRI scans for precise analysis.
* **Tumor Level Prediction**: Classifying the detected tumors into different severity levels, such as low, medium, or high grade, based on the tumor's characteristics and growth pattern.

**Motivation :**

One of the most serious medical disorders that can be fatal is brain tumors, which calls for prompt diagnosis and careful planning of the course of therapy to enhance patient condition. The following are the main reasons this system:

* **Early Detection for Better cure**: In their early stages can be identified with the aid of automated detection technologies, giving patients and physicians a significant time advantage.
* **Releasing Radiologists' Diagnostic Stress**: Manual MRI scan analysis takes a lot of time and is prone to human mistake, especially when processing big amounts of data. Radiologists might benefit from an automated system's dependable.
* **Improved Accuracy with Deep Learning**: The application of deep learning to image analysis tasks has demonstrated encouraging outcomes, especially in the medical domain.

**Assignment -2**

1. **Algorithms Analysis (NP complete, NP Hard) :**

**Image Segmentation (NP-Hard):** Tumor segmentation from MRI scans is NP-hard due to the complexity of isolating tumor, but deep learning models like U-Net provide efficient approximations

**Tumor Classification (NP):** Tumor classification into severity levels is an NP problem, solvable in polynomial time using machine learning models like CNNs, making it computationally feasible.

**Heuristic Solutions for NP-Hard Problems:** Deep learning models serve as heuristics, offering practical approximations to NP-hard tasks like segmentation without needing exact solutions.

1. **Algorithm Used**
2. **U-Net**

U-Net is a convolutional neural network architecture designed for biomedical image segmentation, especially effective with images. The U-Net architecture processes each MRI slice individually, using a fully convolutional structure to identify and localize tumor regions accurately.

**Key Principles**

* **Encoder-Decoder Structure**: The U-Net has a symmetric encoder-decoder structure, capturing spatial context in the encoder and refining spatial details in the decoder for precise segmentation.
* **Skip Connections**: Skip connections between corresponding layers in the encoder and decoder paths help retain fine details, ensuring high segmentation accuracy.

**Applications**

* Segmenting tumors into subregions (e.g., core and surrounding edema) based on individual MRI slices.

1. **Convolutional Neural Network (CNN)**

CNNs are effective for feature extraction from each MRI slice independently, making them suitable for brain tumor classification tasks where images are used. The model processes each slice to extract spatial features relevant for identifying tumor characteristics.

**Key Principles**

* **Feature Extraction from Slices**: The CNN captures spatial patterns within individual slices, enabling tumor classification based on texture, shape, and intensity patterns.
* **Simplified Architecture**: The model’s convolutional layers allow for efficient processing of single MRI slices, reducing the computational load compared to models.

**Applications**

* Classifying brain tumors by type or severity based on characteristics found in MRI slices.

1. **Mathematical Model**

**1. Dice Coefficient (Dice Similarity Coefficient, DSC)**

The Dice Coefficient is a measure of overlap between two sets. It is commonly used to evaluate the performance of image segmentation models.  
  
Formula:  
Dice Coefficient = (2 \* |A ∩ B|) / (|A| + |B|)  
Where:  
- A is the set of predicted pixels (from the model),  
- B is the set of ground truth pixels.  
  
This ranges from 0 (no overlap) to 1 (perfect overlap).

# **2. Accuracy**

Accuracy is a basic metric, calculating the proportion of correctly classified pixels.  
  
Formula:  
Accuracy = (True Positives + True Negatives) / Total Pixels

# **3. Precision**

Precision measures the proportion of correctly predicted positive pixels out of all predicted positive pixels.  
  
Formula:  
Precision = True Positives / (True Positives + False Positives)

**4. Recall (Sensitivity)**

Recall measures the proportion of correctly predicted positive pixels out of all actual positive pixels.  
  
Formula:  
Recall = True Positives / (True Positives + False Negatives)

**5. F1-Score**

The F1-Score is the harmonic mean of precision and recall. It is a balanced measure between precision and recall, providing a single score to evaluate the model’s performance.  
  
Formula:  
F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall)

**6. Confusion Matrix**

A confusion matrix is a table used to describe the performance of a classification model, including true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). For segmentation, it is applied pixel-wise to compare the predicted segmentation with the ground truth.  
  
Example:

|  |  |  |
| --- | --- | --- |
|  | Predicted background | Predicted Object |
| Actual Background | TN | FP |
| Actual Object | FN | TP |

**7. Mean Absolute Error (MAE)**

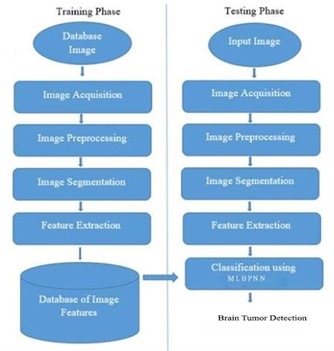
MAE measures the average of the absolute differences between the predicted and ground truth pixel values, useful for continuous value prediction tasks.  
  
Formula:  
MAE = (1/N) \* ∑|y\_i - p\_i|  
Where y\_i is the ground truth pixel value, and p\_i is the predicted pixel value.

1. **Intersection over Union (IoU)**

IoU is another popular metric for segmentation tasks, measuring the overlap between the predicted and ground truth regions.  
  
Formula:  
IoU = |A ∩ B| / |A ∪ B|  
Where:  
- A is the set of predicted pixels,  
- B is the set of ground truth pixels.  
  
IoU gives a ratio of the intersection to the union of the predicted and ground truth sets, providing insight into how well the predicted segmentation matches the ground truth.

**Assignment – 3  
Architectural Diagrams & Modules**

1. **Architecture Diagram:**



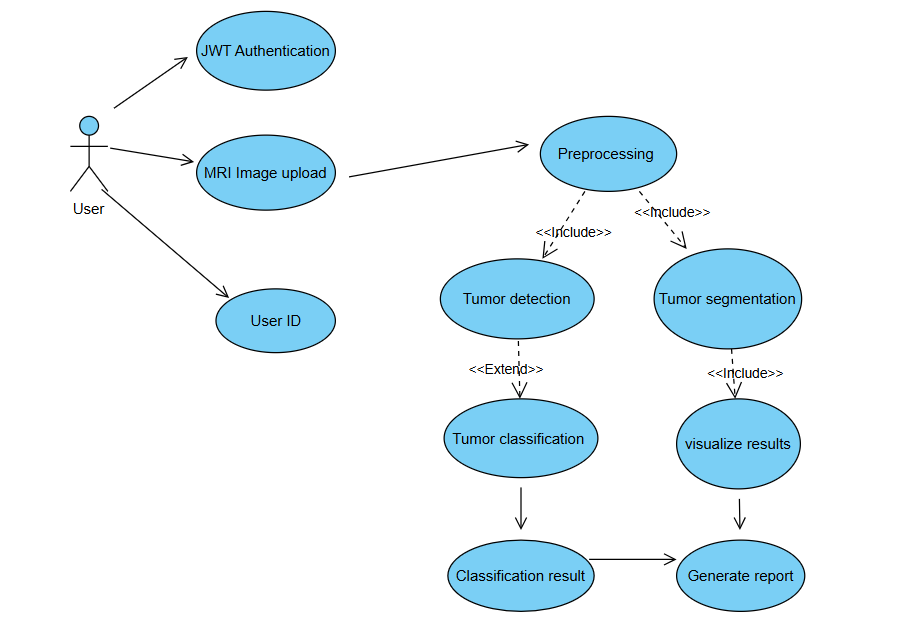
**Description:**

**Training Phase**

1. **Database Image**:
   * A collection of labeled images is used to form a database for training the model. These images contain both healthy and tumor-affected brain scans.
2. **Image Acquisition**:
   * Images from the database are acquired for processing, typically involving the loading of images into the system for analysis.
3. **Image Preprocessing**:
   * This step involves enhancing the quality of the images, which may include noise reduction, normalization, resizing, and other image enhancement techniques to prepare the images for segmentation.
4. **Image Segmentation (Using UNet)**:
   * The preprocessed images are segmented using the **UNet architecture**, a popular convolutional network designed for biomedical image segmentation. UNet helps isolate the tumor regions from the rest of the brain tissue, producing precise tumor masks.
5. **Feature Extraction**:
   * Features are extracted from the segmented images, capturing important characteristics such as texture, shape, and intensity that are indicative of tumor presence.
6. **Database of Image Features**:
   * The extracted features are stored in a database, which will be used to train the CNN for classification.

**Testing Phase**

1. **Input Image**:
   * A new, unseen brain scan image is provided to the system for testing to determine if it contains a tumor.
2. **Image Acquisition**:
   * The input image is loaded into the system for analysis.
3. **Image Preprocessing**:
   * The input image undergoes the same preprocessing steps as those applied during the training phase.
4. **Image Segmentation (Using UNet)**:
   * The preprocessed image is segmented using the trained UNet model to identify potential tumor regions.
5. **Feature Extraction**:
   * Features are extracted from the segmented image, similar to the training phase.
6. **Classification (Using CNN)**:
   * The extracted features are then fed into a **Convolutional Neural Network (CNN)**, which classifies the image as either tumor-positive or tumor-negative based on learned patterns from the training data.
7. **Use Case Diagram:**



**User Interaction**:

The user initiates the process by interacting with the system.

* **JWT Authentication**:
  + The user is authenticated using **JSON Web Tokens (JWT)**, ensuring secure access to the system.
* **MRI Image Upload**:
  + Once authenticated, the user uploads an MRI image of the brain for analysis.
* **User ID**:
  + The system associates the uploaded image with a unique User ID for tracking and reporting purposes.

**Preprocessing**:

After image upload, the system performs **preprocessing** on the MRI image to enhance quality. This may include noise reduction, normalization, or resizing.

**Tumor Detection and Segmentation**:

The preprocessing phase branches into two parallel processes:

* + **Tumor Detection**:

The system includes a step for detecting the presence of a tumor within the brain image. This could involve initial screening to determine if further segmentation is necessary.

* + **Tumor Segmentation (using UNet)**:

If a tumor is detected, the system uses **UNet** to segment the tumor region from the rest of the brain. This step is crucial for accurately identifying the tumor boundaries.

**Feature Extension and Visualization**:

* **Tumor Classification (using CNN)**:
  + After detection, the system uses a **Convolutional Neural Network (CNN)** to classify the type of tumor. The classification result can indicate whether the tumor is benign or malignant.
* **Visualize Results**:
  + The system visualizes the segmented tumor, allowing users to see the affected region clearly.

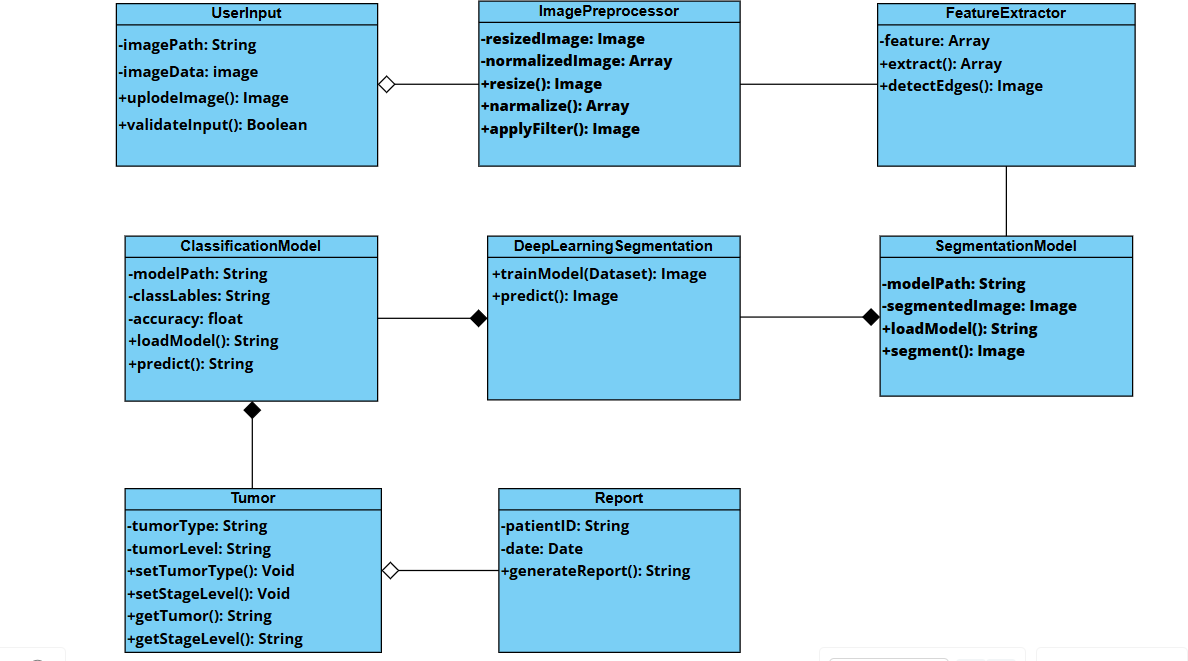
**Final Outputs**:

* **Classification Result**:

The outcome of the CNN classification is presented to the user, indicating the tumor type.

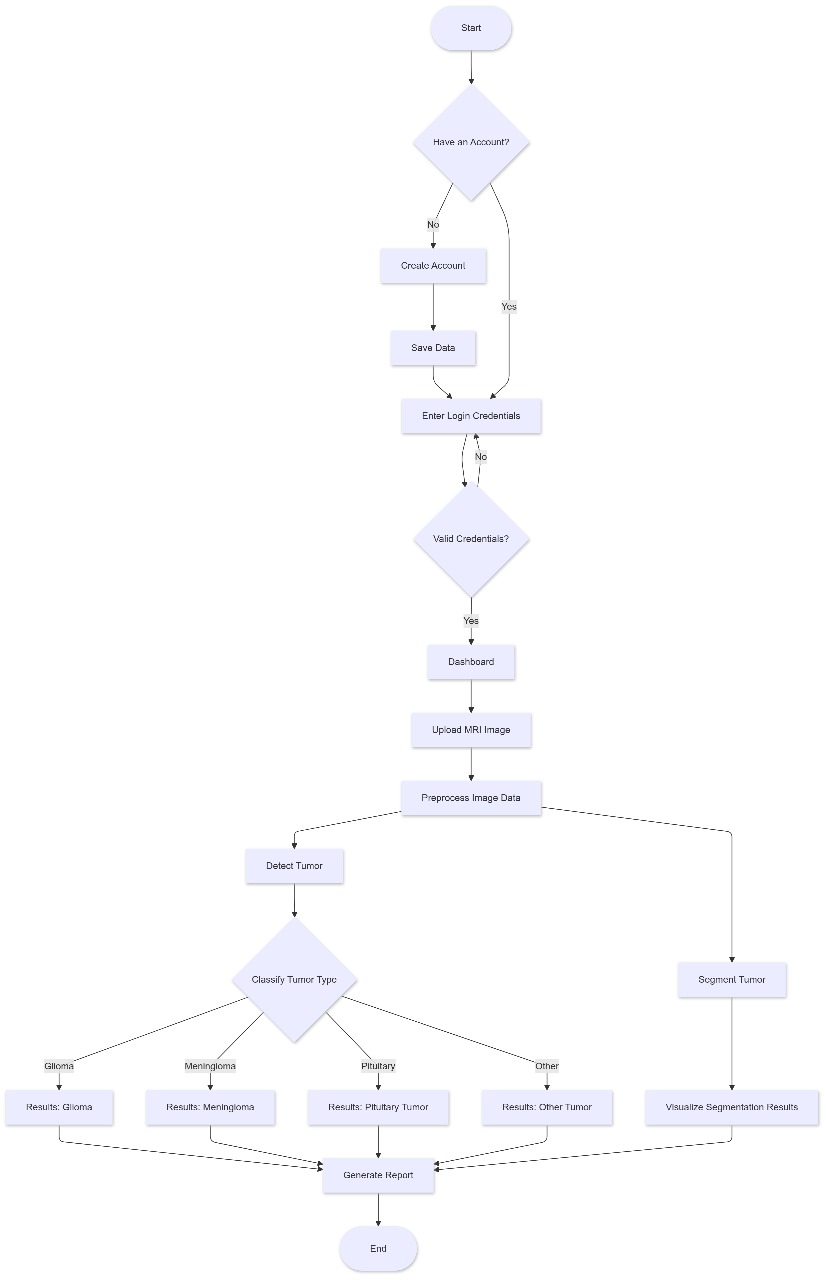
1. **Class Diagram:**

**Description:** The class diagram represents the structure of a brain tumor classification and segmentation system. It starts with the UserInput class, which handles the uploading and validation of MRI images. The ImagePreprocessor class is responsible for resizing, normalizing, and applying filters to the images. The FeatureExtractor class extracts features like edges from the images. The ClassificationModel and DeepLearningSegmentation classes are central to the system, where the former handles tumor classification based on pre-trained models, and the latter handles segmentation of the tumor using deep learning techniques. The Tumor class stores the tumor type and stage information, while the SegmentationModel class processes the segmentation of the tumor. Finally, the Report class generates patient-specific reports based on the classification and segmentation results. Each class has attributes and methods relevant to their roles in the system.



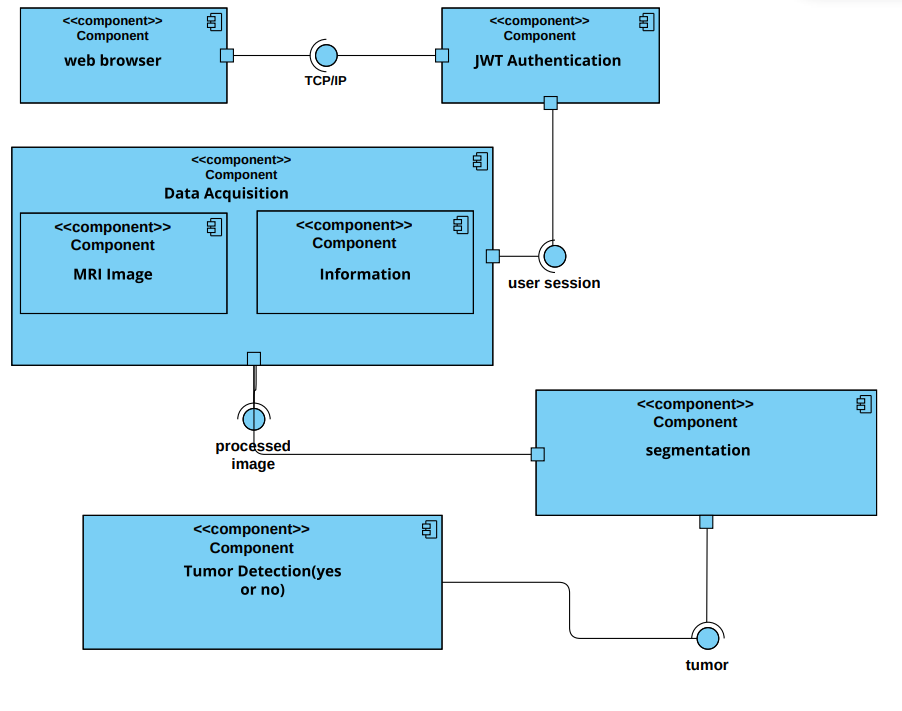
1. **Activity Diagram:**

**Description:** The activity diagram shows the steps involved in a brain tumor classification and segmentation system. It starts by checking if the user has an account. If not, the user creates one. After logging in with valid credentials, they go to the dashboard. The user uploads an MRI image, which is processed. The system can then either detect the tumor or segment it. If the tumor is detected, it is classified into types like Glioma, Meningioma, Pituitary, or others, and the results are shown. For segmentation, the system shows the segmented tumor. Finally, the system generates a report based on the results, and the process ends.



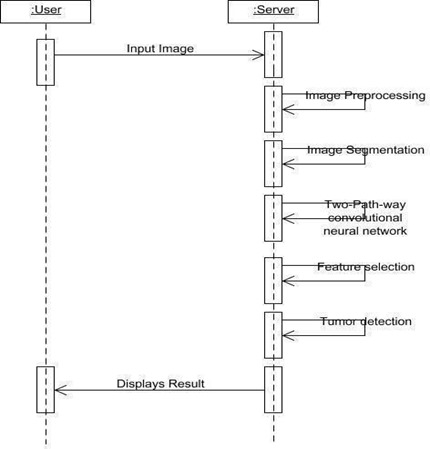
1. **Component Diagram:**

**Description:** This component diagram outlines the architecture for a brain tumor classification and segmentation system. The web browser serves as the user interface, with Authentication ensuring secure access. The Data Acquisition component collects MRI images and patient information. These images are processed and sent to the Segmentation component to identify potential tumor regions. The segmented output is passed to the Tumor Detection component, which determines if a tumor is present (Yes/No). User sessions maintain connectivity between components, ensuring smooth data flow and reliable tumor analysis.

****

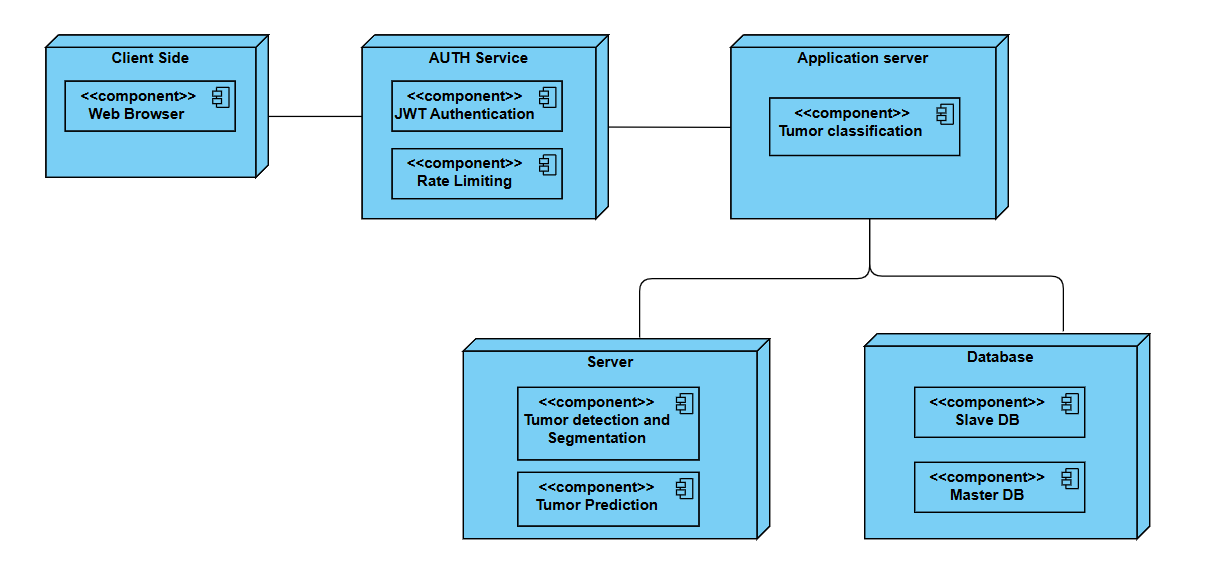
1. **Sequence diagram:**

**Description:** This sequence diagram outlines the step-by-step flow for brain tumor classification and segmentation. The process begins when the user provides an input MRI image to the server. The server first performs image preprocessing to enhance the image quality and reduce noise. Next, image segmentation is applied to isolate potential tumor regions. The segmented data is then processed through a two-pathway convolutional neural network (CNN), which analyzes the image from multiple perspectives. Following this, feature selection identifies the most relevant features for classification. Finally, the tumor detection module determines if a tumor is present, and the result is sent back to the user for display. This sequence ensures an organized workflow from image acquisition to diagnosis.



1. **Deployment Diagram:**

**Description:** This deployment diagram illustrates the infrastructure for a brain tumor classification and segmentation system, showing how components are deployed across different nodes. The client side uses a web browser as the interface for users to input data and receive results. The AUTH service manages secure access with authentication and implements rate limiting to control the traffic. The application server handles tumor classification, analyzing input data to identify tumor types. A dedicated server performs tasks like tumor detection and segmentation and tumor prediction based on the processed images. The system relies on a database with two parts: the Master DB, which stores primary data, and the Slave DB, used for fast read operations to ensure smooth performance. This deployment ensures secure, efficient processing and accurate tumor diagnosis with seamless interaction between the client, servers, and databases.



**BRAIN TUMOR DETECTION AND LEVEL PREDICTION SYSTEM**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)**

**SUBMITTED BY**

**ROHAN KHATODE B1909504264**

**ARANAV MAHALPURE B1909504273**

**ASHISH CHANDEKAR B1909504209**

**AAYUSH BHALAVI**  **B1909504217**



**DEPARTMENT OF COMPUTER ENGINEERING**

**INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, PUNE,**

**HINJEWADI, PUNE, MAHARASHTRA 411057**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2024 -2025**

**INDEX**

1. Introduction 1
2. Description 2
3. Functional Requirements 2
4. Non-Functional Requirements 3
5. Interface Requirements: 4
   1. . User Interface 4

5.2. API Interface 4

* 1. . Database Interface 5

1. Performance Requirements 6
2. Preliminary Schedule 7

**Software Requirements Specification (SRS) :**

* 1. **Introduction**

The purpose of this Software Requirement Specification (SRS) document is to define the software requirements for the Brain Tumor Detection and Level Prediction System. This system aims to automate the detection of brain tumors and predict their severity levels based on MRI scans using machine learning techniques.

* 1. **Scope of the project:**

The brain tumor detection and level prediction system is designed to assist medical professionals by analyzing MRI images to identify and classify brain tumors and predict their severity level. This system will leverage machine learning techniques to perform accurate tumor classification and segmentation, focusing on improving diagnostic efficiency and supporting clinical decision-making.

Key components of the project include:

* **Data Collection and Preprocessing**: Utilizing existing MRI datasets for model training, the system will preprocess the MRI images to enhance clarity and standardize input formats. Data preprocessing will involve resizing, normalization, and augmenting images to improve model robustness.
* **Image Segmentation and Classification Models**: The project will employ machine learning and deep learning models such as 2D U-Net for tumor segmentation and ResNet for tumor classification. The models will be trained to accurately identify tumor regions, classify tumor types, and predict the tumor's level based on MRI slices.
* **Level Prediction and Severity Assessment**: The system will analyze segmented tumor areas to assess the grade level of the tumor. This prediction helps to determine the aggressiveness of the tumor, supporting timely and appropriate treatment decisions.
* **User-Friendly Interface**: A web-based interface will allow users, such as radiologists and oncologists, to upload MRI images, view segmentation outputs, and receive detailed reports on tumor classification and severity levels. The interface will also enable users to review model predictions and adjust parameters as needed.
  1. **Purpose of Document:**

The purpose of this System Requirement Specification (SRS) document is to clearly outline the functional and non-functional requirements of the brain tumor detection and level prediction system. It serves as a comprehensive guide for stakeholders, including developers, medical professionals, and project managers, to understand the system’s objectives, features, and limitations. This document ensures that all requirements are addressed and serves as a reference for development, testing, and deployment phases. It also provides a structured framework for validating the system’s functionality and ensuring it meets the needs of its end users, such as radiologists and oncologists.

* 1. **Overview:**

The project aims to develop an automated system for detecting brain tumors and predicting their severity levels using deep learning techniques. The primary objective is to create a model that can accurately detect the presence of tumors from MRI scans and classify their severity into various levels (such as low-grade or high-grade tumors). This system will utilize image processing and machine learning algorithms to assist medical professionals in making quicker and more accurate diagnoses, ultimately improving patient outcomes.

1. **Description**

The brain tumor detection and level prediction system uses machine learning to analyze MRI images for identifying, segmenting, and classifying brain tumors. By leveraging models like U-Net for segmentation and ResNet for classification, the system predicts tumor severity, aiding in diagnosis and treatment planning. A web interface allows medical professionals to upload images, view segmentation results, and access detailed reports, providing actionable insights that support early diagnosis and improved patient care.

1. **Functional Requirements**
2. **Data Collection**

* **MRI Data Access**: The system shall access MRI datasets containing labeled brain tumor images from reliable medical imaging sources.

1. **Data Preprocessing**

* **Image Standardization**: The system shall standardize MRI images by resizing and normalizing them to ensure consistency in input format.
* **Data Augmentation**: The system shall apply augmentation techniques (e.g., rotation, flipping) to increase dataset diversity and improve model robustness.

1. **Tumor Detection and Segmentation**

* **Image Segmentation Model**: The system shall employ a U-Net model for segmenting tumor regions in 2D MRI slices.
* **Tumor Boundary Detection**: The system shall highlight and delineate the boundaries of identified tumor regions on each MRI slice.

1. **Tumor Classification and Level Prediction**

* **Classification Model Training**: The system shall support the training of classification models (e.g., ResNet) on labeled MRI data to categorize tumor types.
* **Level Prediction**: The system shall predict the tumor’s severity level, providing insights into its aggressiveness.

1. **User Interface**

* **Input Functionality**: The system shall provide a user-friendly interface for users to upload MRI images for analysis.
* **Output Display**: The system shall display segmented images with highlighted tumor regions, classification results, and severity predictions in a clear and accessible format.

1. **Non-Functional Requirements**
2. **Performance**

* **Response Time**: The system shall process MRI images and return segmentation and classification results within 10 seconds under normal load conditions.
* **Throughput**: The system shall handle a minimum of 50 concurrent user sessions without performance degradation.

1. **Scalability**

* The system shall be designed to accommodate an increasing number of MRI images and user access without requiring major re-architecture, allowing for efficient scaling as data and user demand grow.

1. **Reliability**

* The system shall maintain an uptime of 99.5%, ensuring it is consistently available for healthcare professionals. Scheduled maintenance or downtime shall be communicated to users in advance.

1. **Security**

* **Data Protection**: The system shall implement data encryption for both storage and transmission to protect sensitive patient data and MRI images.
* **User Authentication**: The system shall require secure user authentication to prevent unauthorized access and protect patient confidentiality.

1. **Usability**

* The system shall have an intuitive user interface, allowing users to upload MRI images, view segmentation results, and access reports easily, with minimal training required.

1. **Maintainability**

* The system shall be designed with modular architecture and well-documented code to facilitate easy maintenance, updates, and future enhancements.

1. **Compatibility**

* The system shall be compatible with major web browsers (e.g., Chrome, Firefox, Safari) and optimized for responsiveness to ensure usability across different devices (e.g., desktops, tablets, smartphones).

1. **Data Integrity**

* The system shall ensure data integrity by implementing validation checks and error handling to prevent corruption or loss of image data and results during processing and storage.

1. **Interface Requirements:**

**5.1. User Interface**

**Description**:  
The system will provide a web-based user interface (UI) that allows healthcare professionals to interact with the application, upload MRI images, and view results.

**Features**:

* **Input Forms**: The system will allow users to upload MRI images for analysis and specify the type of analysis (e.g., tumor detection, classification, severity prediction).
* **Display of Results**: The system will display segmented MRI images with highlighted tumor regions, along with predicted tumor type and severity levels.
* **Report Generation and Export**: Users can generate and export detailed reports (e.g., PDF, CSV) summarizing tumor detection, classification, and severity predictions.
* **User Authentication and Account Management**: The system will support secure user login, role-based access, and account management for healthcare professionals.

**5.2. API Interfaces**

**External Data Sources**:

* **Medical Imaging Data APIs**: The system may integrate with external medical imaging APIs to retrieve additional MRI image datasets for model training and validation.
* **Historical Data Integration**: If necessary, the system will integrate with APIs that provide supplementary medical data (e.g., tumor type or severity references).

**RESTful API**:  
The system will expose a RESTful API for integration with other medical applications or platforms, allowing seamless interaction with external systems.

**Endpoints**:

* **POST v1/api/analyze**: Submits MRI images for analysis, triggering tumor detection, segmentation, and classification, and returns results.
* **GET v1/api/reports**: Retrieves generated reports on tumor segmentation, classification, and severity predictions for further analysis or sharing.
* **GET v1/api/models**: Fetches the status and version of the models used for tumor detection and classification.
  1. **Database Interface**

**Database Management System**:  
The system will interact with a database (e.g., PostgreSQL, MySQL) to store MRI images, patient data (when applicable), model training data, and analysis results.

**Functions**:

* **Data Storage**: Store MRI images, model predictions, and reports securely with proper indexing for fast retrieval.
* **Data Retrieval**: Efficient retrieval of processed results and reports for display to users.
* **Data Management**: Support for data backup, recovery, and data integrity checks to ensure reliability.

1. **Performance Requirements**
2. **Response Time**

* The system shall provide segmentation, classification, and level prediction results within 10 seconds of receiving an MRI image input.

1. **Throughput**

* The system shall support at least 50 concurrent users analyzing MRI images simultaneously without degradation in performance.

1. **Scalability**

* The system shall be designed to scale horizontally to accommodate an increasing number of MRI images and user access without requiring significant re-architecture.

1. **Model Training and Update Frequency**

* The machine learning models shall be retrained with new labeled MRI data at least once a month to maintain accuracy and relevance. The retraining process shall complete within 1 hour to ensure minimal downtime for diagnostic services.

1. **Availability**

* The system shall maintain an uptime of 99.5%, ensuring consistent reliability and availability for healthcare professionals to access predictions and reports.

1. **Preliminary Schedule**

**Phase 1: Problem Definition and Literature Review (1-2 weeks)**

* Define the scope of brain tumor detection, segmentation, and severity prediction.
* Review current research on brain tumor analysis using MRI and explore relevant machine learning techniques.
* Identify successful models and approaches (e.g., U-Net, ResNet) and finalize the exact project objectives.

**Phase 2: Data Collection (2-3 weeks)**

* Collect labeled MRI data from reliable sources, such as open medical datasets.
* Gather MRI images covering various tumor types, levels, and labels for training, validation, and testing.

**Phase 3: Data Preprocessing (2-3 weeks)**

* Preprocess MRI images by resizing, normalizing, and enhancing quality to ensure consistency in inputs.
* Apply data augmentation techniques (e.g., rotations, flips) to increase data diversity.
* Prepare labels for classification and segmentation tasks.

**Phase 4: Tumor Detection and Segmentation Model Development (3-4 weeks)**

* Implement the U-Net model for tumor segmentation to accurately identify and localize tumor regions in MRI slices.
* Train and test the model using a subset of the dataset, fine-tuning the architecture and parameters for optimal segmentation results.

**Phase 5: Tumor Classification and Level Prediction Model Development (2-3 weeks)**

* Develop and train a ResNet-based model for classifying tumor types and predicting severity levels.
* Integrate segmentation and classification models to create a cohesive pipeline for end-to-end prediction.
* Define evaluation metrics (e.g., Dice coefficient for segmentation accuracy, accuracy for classification).

**Phase 6: Model Testing and Validation (2-3 weeks)**

* Test the segmentation and classification models on unseen MRI images.
* Evaluate model performance on metrics like accuracy, precision, and segmentation overlap.
* Compare predictions with radiologist-verified ground truth to validate clinical relevance.

**Phase 7: Optimization and Fine-Tuning (1-2 weeks)**

* Tune hyperparameters to improve model accuracy and reduce response time.
* Incorporate feedback from validation results to enhance model performance and adjust configurations for better results.

**Phase 8: Project Finalization and Documentation (1 week)**

* Prepare the final project report, presentation, and documentation for the code, models, and data.
* Organize all assets for a smooth demo and user training, ensuring clarity in results and ease of understanding for medical professionals.

**Gantt Chart:**

