A Mini Project Synopsis
On
MRI brain tumor detection
of

MINI PROJECT
BMC353
MCA-II year/ III Semester

Submitted to the Department of Computer Applications
by

Satyam Trivedi (2401640140164)
Ritesh Saxena (2401640140150)
Parul Yadav (2401640140129)

Under the Supervision of
Mr. Rahul Bajpai

Assistant Professor, MCA

PRANVEER SINGH INSTITUTE OF TECHNOLOGY

To the

Dr. A.P.J. Abdul Kalam Technical University, Lucknow
(2025-2026)

INDEX

Introduction

MRI-based brain tumor detection using machine learning (ML) is a rapidly advancing field that automates and enhances the diagnostic process. By analyzing complex patterns in medical imaging data, ML models—particularly those based on deep learning (DL)—can help radiologists quickly and accurately identify, classify, and segment tumors. This reduces reliance on time-consuming manual processes and can lead to earlier diagnosis, more effective treatment planning, and better patient outcomes.

Brain tumor is one of the most rigorous diseases in the medical science. An effective and efficient analysis is always a key concern for the radiologist in the premature phase of tumor growth. Histological grading, based on a stereotactic biopsy test, is the gold standard and the convention for detecting the grade of a brain tumor. The biopsy procedure requires the neurosurgeon to drill a small hole into the skull from which the tissue is collected. There are many risk factors involving the biopsy test, including bleeding from the tumor and brain causing infection, seizures, severe migraine, stroke, coma and even death. But the main concern with the stereotactic biopsy is that it is not 100% accurate which may result in a serious diagnostic error followed by a wrong clinical management of the disease. Tumor biopsy being challenging for brain tumor patients, non-invasive imaging techniques like Magnetic Resonance Imaging (MRI) have been extensively employed in diagnosing brain tumors. Therefore, development of systems for the detection and prediction of the grade of tumors based on MRI data has become necessary.

Gap in Study

Key gaps in the application of machine learning (ML) to MRI brain tumor analysis include a lack of model explainability, insufficient use of multi-modal data, issues with generalizability, and limitations with data availability. While machine learning models can achieve high accuracy, these limitations hinder their clinical adoption and trustworthiness. Acquiring large, high-quality, and richly annotated datasets of brain tumor MRIs is difficult due to patient privacy concerns, the expense of labeling, and limited access to imaging data. Existing public datasets often lack the diversity needed to cover all types, grades, and variations of brain tumors seen in clinical practice.

Models trained on a single dataset often fail to perform well on data from different clinics or scanners due to variations in imaging protocols, equipment, and patient populations. This phenomenon, known as "domain shift," limits the models' real-world applicability.

MINI PROJECT SCOPE

Project scope

A mini-project can focus on one of the following specific tasks:

Binary classification: The project can be scoped to classify MRI brain scans into one of two categories: "tumour" or "no tumour". This is the most basic approach, suitable for beginners.

Multi-class classification: A more complex version involves training a model to classify the tumour into different types, such as meningioma, glioma, or pituitary tumor.

Tumour segmentation: This task involves identifying the precise location of the tumor and outlining its boundaries within the MRI scan. This is more challenging than a simple classification task but provides a more clinically useful output.

Web application: The project can culminate in a web or mobile application with a user-friendly interface. Users would be able to upload an MRI scan and receive a diagnosis from the model, assisting doctors with their analysis

AIMS AND OBJECTIVE

Aims

Enhance diagnostic speed and accuracy: To significantly reduce the time and potential for human error associated with manual analysis of MRI scans by radiologists. Support clinical decision-making: To provide clinicians with a reliable "second opinion" to help in early diagnosis, treatment planning, and monitoring of tumor progression.

Enable multi-class and multi-modal analysis: To create models that can classify different types of tumors (e.g., glioma, meningioma, pituitary) and integrate data from various MRI sequences (e.g., T1, T2, FLAIR) for a more comprehensive understanding.

Develop automated segmentation: To accurately and efficiently segment (delineate) the tumor region from healthy brain tissue, which is a crucial step for treatment planning.

Machine learning (ML) models aim to automatically and accurately detect and classify brain tumors from MRI scans. The primary objective is to develop a non-invasive diagnostic aid that reduces human error and subjectivity, leading to quicker and more precise diagnoses for improved patient care.

Objectives

Data acquisition and preprocessing: Collect large datasets of MRI brain images and apply techniques like noise reduction, skull stripping, and image normalization to ensure consistent, high-quality inputs for ML models.

Feature extraction and selection: Implement algorithms to automatically extract and select the most relevant features from MRI images, such as texture, shape, and intensity characteristics, that help in differentiating between tumor types and healthy tissue.

Model development and training: Build, train, and fine-tune various ML or deep learning (DL) models, such as Convolutional Neural Networks (CNNs), to perform segmentation and classification tasks.

Performance evaluation: Rigorously evaluate the trained models using standard metrics like accuracy, precision, recall, and the F1-score to compare their effectiveness and reliability.

Model optimization: Address issues like overfitting and class imbalance, and optimize models for computational efficiency to ensure they can be practically deployed in clinical settings.

Model interpretability: Use explainable AI (XAI) techniques to provide insights into how the model makes decisions, which is critical for building trust among medical professionals

Brief Modules Description

Using machine learning (ML) for brain tumor detection in MRI scans involves several distinct modules that form a processing pipeline. These modules handle everything from preparing the raw image data to making a final diagnostic prediction.

1. Data acquisition and preprocessing

This is the foundational step that prepares the raw MRI scans for the ML model. The quality and consistency of the initial data are critical for the accuracy of subsequent stages.

- **Dataset collection**: Gather a collection of MRI images, which often includes multiple image types (modalities) like T1-weighted, T2-weighted, and FLAIR. Standardized public datasets, such as the BraTS challenge, are frequently used.
- **Image enhancement and noise removal**: Apply filters, such as a median filter, to improve image quality by reducing noise while preserving important details like edges.
- **Normalization**: Standardize the intensity values across different MRI scans to ensure that the model is not biased by variations in image brightness from different scanners or patients.
- **Skull stripping:** Remove non-brain tissues, such as the skull and scalp, from the MRI image to isolate the region of interest (the brain) and focus the model's analysis.

- **Data augmentation**: Increase the size and diversity of the dataset by generating modified copies of existing images through rotations, flips, or scaling. This helps improve the model's ability to generalize and reduces overfitting.

2. Segmentation

This module separates the tumor region from healthy brain tissue. It is a critical step that helps the model focus on the relevant area of the image.

- **Techniques:** Common ML techniques for segmentation include fuzzy C-means clustering, region-growing algorithms, or watershed algorithms.
- **Advanced deep learning segmentation**: More sophisticated models, such as the U-Net, are frequently used to perform pixel-level segmentation, which is a key task in biomedical image analysis.

3. Feature extraction and selection

This module identifies and extracts the most relevant information from the segmented image that describes the characteristics of the tumor.

- **Feature extraction:** For traditional ML approaches, this may involve using techniques like Discrete Wavelet Transform (DWT) or a gray-level co-occurrence matrix (GLCM) to capture texture, shape, and intensity details. For deep learning, this step is handled automatically by the convolutional layers of the network.

4. Classification

This module uses the extracted features to determine the final diagnosis.

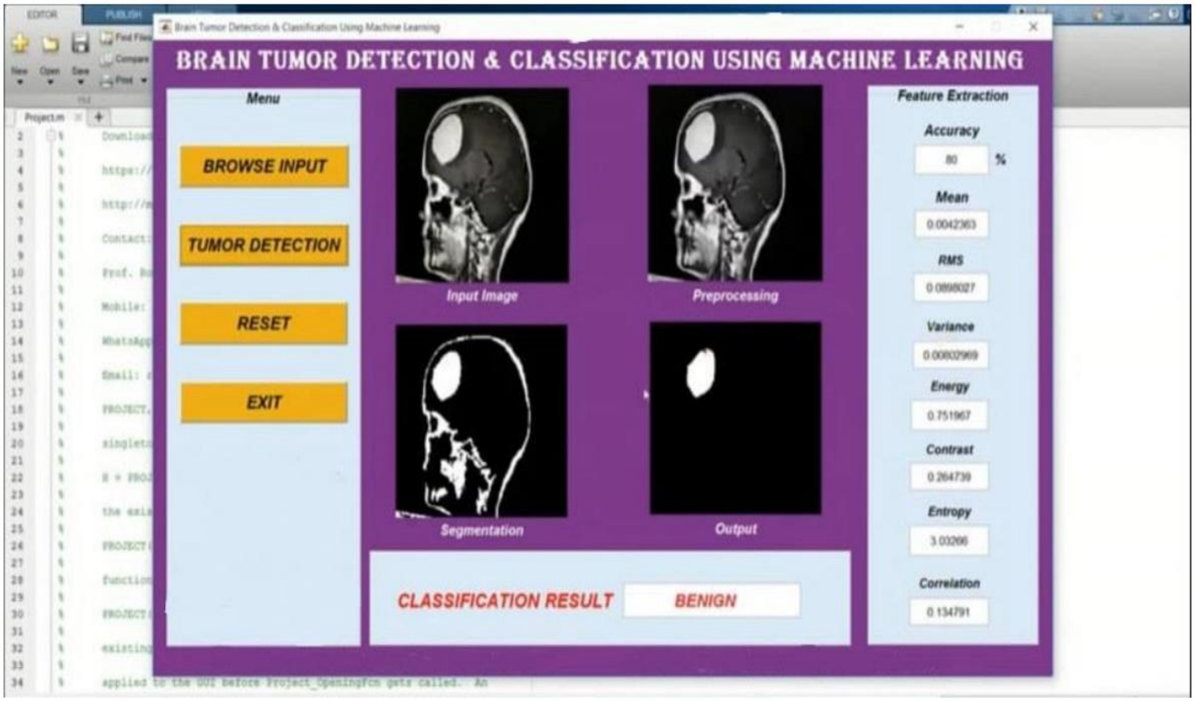
- **Traditional ML classifiers**: Supervised learning models like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) can be used to classify the image as "tumor" or "no tumor" or to categorize tumor types (e.g., benign or malignant).
- **Deep learning classifiers:** Convolutional Neural Networks (CNNs) are the dominant method for image classification in medical imaging. The CNN automatically learns and extracts features, passing them through layers to a final prediction output.
- **Transfer learning:** To overcome small dataset limitations, a pre-trained CNN (like VGG16 or ResNet) can be used. Its weights are fine-tuned on the specific MRI dataset, leveraging knowledge learned from vast general image datasets.

5. Model evaluation and deployment

After training, the model must be rigorously evaluated and prepared for use in a real-world setting.

- **Validation:** Test the trained model against an independent dataset to measure its performance using metrics such as accuracy, precision, recall, and the F1-score.

- **Prediction:** Apply the validated model to new, unseen MRI images to provide predictions that can assist radiologists in their diagnostic process.



Tools and Technology Used

1. Data handling and processing

Imaging modalities: While other technologies like CT and PET scans can be used, MRI is the preferred modality due to its superior soft-tissue contrast, which is essential for visualizing brain structures and tumors. Several MRI sequences are used:

T1-weighted: Shows excellent contrast between gray matter, white matter, and cerebrospinal fluid (CSF). Tumors appear dark or medium gray.

T2-weighted: Sensitive to water content and shows tumors as bright regions.

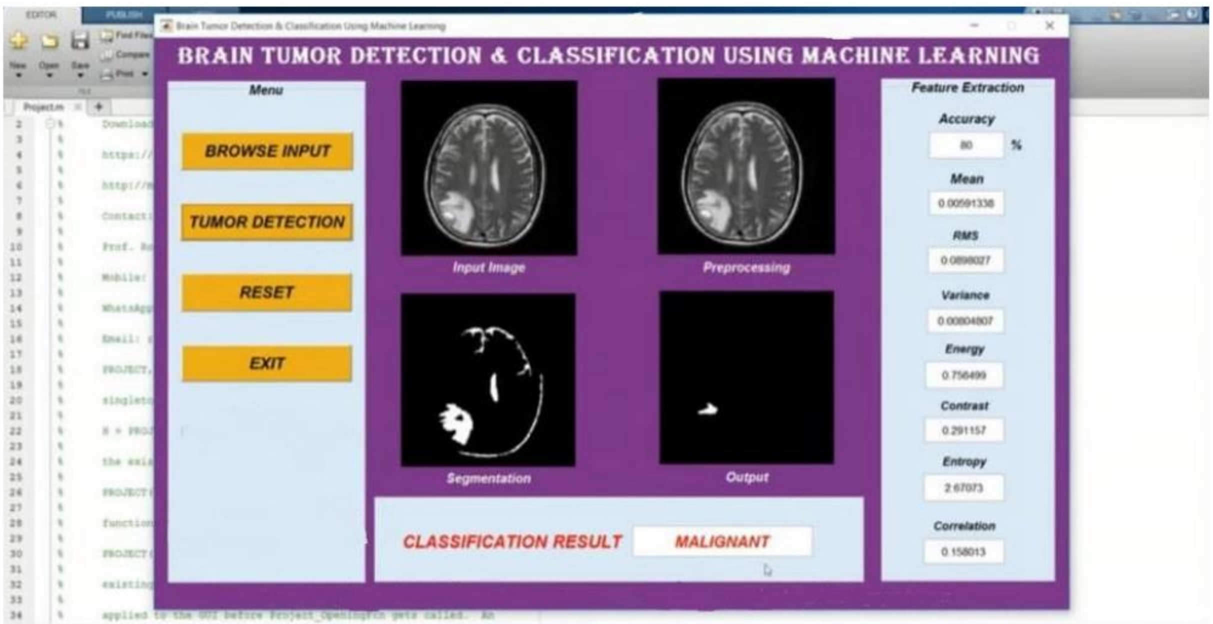
FLAIR: Suppresses the CSF signal, making abnormalities and lesions in the brain stand out.

Image preprocessing: Raw MRI scans often contain noise and artifacts that can interfere with analysis. Standard preprocessing steps include:

Noise reduction: Filters like Gaussian blurring or anisotropic diffusion can be used to smooth the image while preserving important edges.

Skull stripping: This process isolates the brain tissue from non-brain regions, such as the skull and scalp, to focus the analysis on the relevant areas.

Image enhancement: Techniques like histogram equalization improve the contrast of an image, making tumors more visible.



<u>Public datasets:</u> These are vital for training and evaluating models, allowing researchers to compare the performance of different techniques.

BraTS (Brain Tumor Image Segmentation Benchmark): A widely used dataset containing multimodal MRI scans and expert-annotated tumor segmentations for various glioma types.

TCIA (The Cancer Imaging Archive): Provides public medical images, including MRI scans of brain tumors.

<u>File format libraries:</u> These libraries are used for working with standard medical imaging file formats.

Pydicom: Reads, modifies, and writes files in the DICOM (Digital Imaging and Communications in Medicine) format.

Nibabel: Provides tools for reading and writing neuroimaging file formats like NIfTI (Neuroimaging Informatics Technology Initiative).

2. Machine learning and deep learning models

Classical machine learning algorithms: These models require hand-crafted feature extraction but can still be effective, especially with smaller datasets.

> **Support Vector Machine (SVM):** A popular model for classification tasks that works well with complex, non-linear classification problems.

> **K-Nearest Neighbors (KNN):** A simple, non-parametric algorithm that classifies data points based on the majority class of their "k" nearest neighbors.

> **Random Forest:** An ensemble learning method that constructs a multitude of decision trees to make robust predictions and control overfitting.

Deep learning techniques: These approaches automatically learn hierarchical features directly from the image data, eliminating the need for manual feature extraction.

> **Convolutional Neural Networks (CNNs):** A type of deep learning model that is highly effective for image analysis tasks like classification and segmentation. CNNs consist of convolutional and pooling layers to detect features and reduce dimensionality.

> **Transfer learning:** This technique uses pre-trained CNN models, such as VGG16, ResNet, and EfficientNet, that were originally trained on large general datasets (like ImageNet). The pre-trained models are then fine-tuned on the medical image dataset, which is useful when labeled medical data is scarce.

> **U-Net:** A specialized CNN architecture particularly effective for medical image segmentation, where the goal is to identify and delineate specific structures like tumors.

3. Software tools and programming libraries

Programming languages: Python is the dominant language for developing ML models in medical imaging due to its extensive ecosystem of libraries.

Core machine learning frameworks:

> TensorFlow: An open-source library for numerical computation and large-scale machine learning, with a focus on deep learning.

> PyTorch: An open-source ML library for Python known for its flexibility and ease of use in deep learning research.

Image processing libraries:

scikit-image: An image processing library for Python that provides a wide range of functions for preprocessing and analysis.

OpenCV: A library for computer vision tasks, including various image preprocessing functions.

SimpleITK: An image analysis library that offers a user-friendly interface for image segmentation and registration.

Cloud computing platforms: Due to the large computational resources required for deep learning, cloud services are frequently used for training.

Google Colab: A free, web-based platform that offers access to computing resources, including T4 GPUs, which can be used to run Python code and train models.

METHODOLOGY

The methodology for MRI brain tumor detection using machine learning involves several key steps, from data preparation to model deployment. The most common approach uses deep learning models, such as Convolutional Neural Networks (CNNs), for their ability to process image data effectively.

1. **Data acquisition**

- **Source**: The process begins with obtaining MRI scans from publicly available databases (like those from the Brain Tumor Segmentation Challenge (BraTS) and Figshare) or clinical hospital records.
- **Dataset:** The dataset includes MRI images of brains with tumors (e.g., glioma, meningioma, pituitary) and healthy brains. The images can be from different modalities (T1-weighted, T2-weighted, FLAIR) for comprehensive analysis.

2. **Preprocessing and augmentation**

- **Image refinement:** Raw MRI images often contain noise and variations in size, position, and contrast. Preprocessing standardizes the images and includes steps like resizing, normalization of pixel values, and noise removal.
- **Data augmentation**: To prevent model overfitting, especially with small datasets, augmentation techniques increase the size and diversity of the training data. This involves generating new images by applying random rotations, flips, shifts, and zooms to existing images.

3. Model development

- **Feature extraction**: For traditional machine learning, features like texture, shape, and intensity are extracted from the images using methods such as wavelet transforms or gray-level co-occurrence matrices (GLCM). For deep learning, the model automatically learns to extract relevant features through its convolutional layers.
- **Classification:**
    - **Deep Learning**: Convolutional Neural Networks (CNNs) are widely used for their high performance. These networks typically consist of multiple layers, including convolutional, pooling, and fully connected layers, that can learn complex patterns in the image data.
    - **Traditional Machine Learning**: After manual feature extraction, the process uses classifiers like Support Vector Machines (SVM), K-Nearest Neighbors (KNN), or Random Forest to distinguish between tumor and non-tumor images.
- **Model training: The** dataset is divided into training, validation, and testing sets. The model is trained on the training data, and hyperparameters are tuned based on performance on the validation set.
- **Transfer learning:** In cases with limited medical image data, a pre-trained model (e.g., VGG16, InceptionResNetV2) can be used. It is fine-tuned on the brain MRI dataset to improve accuracy and reduce training time.

4. **Evaluation and validation**

- **Performance metrics:** The model's performance is evaluated using standard metrics such as accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).
- **Validation:** Techniques like k-fold cross-validation ensure the model is robust and not biased toward a specific data subset.

5**. Deployment**

- **Clinical application:** The validated model can be integrated into a user-friendly application, potentially with an interface that allows radiologists to upload MRI images and receive real-time predictions.
- **Explainability:** Incorporating Explainable AI (XAI) techniques, such as attention maps, helps clinicians understand how the model arrives at its predictions, which increases trust and facilitates better decision-making

Frontend

HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are fundamental technologies for front-end web development, working in conjunction to create the structure and visual presentation of web pages.

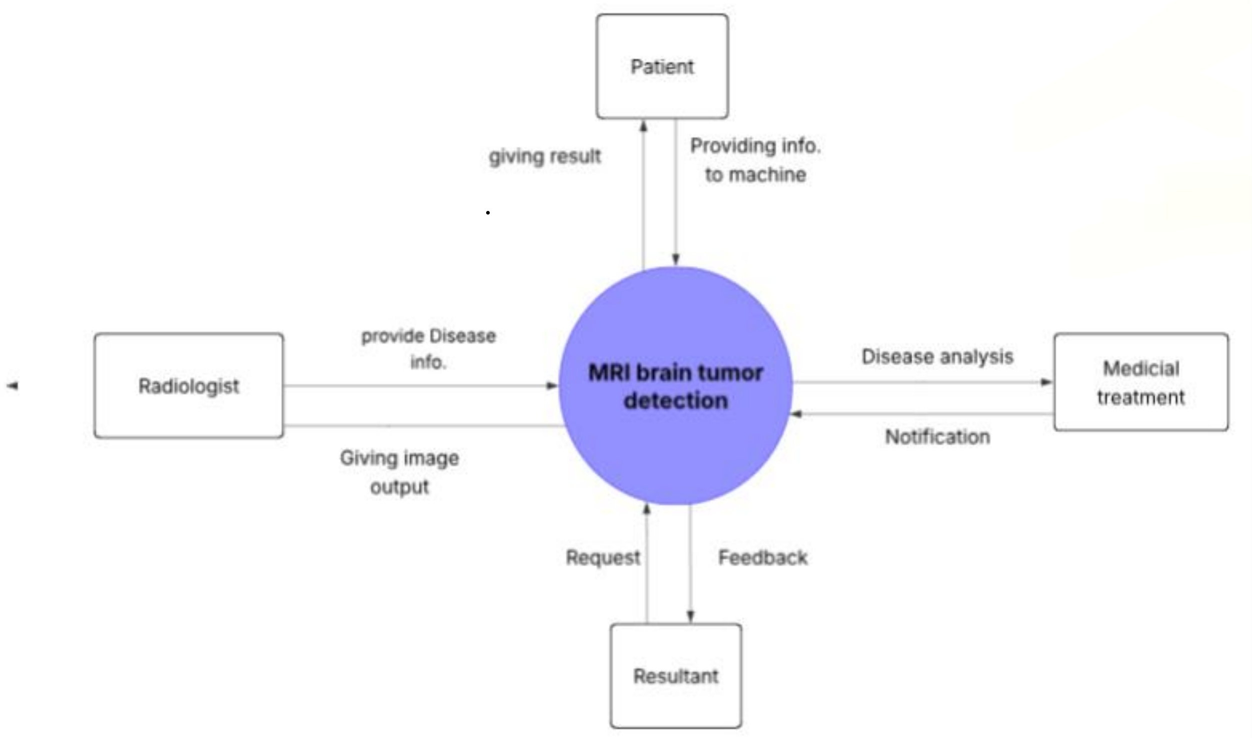
Backend.

- **Extensive libraries**: Python's machine learning libraries simplify complex tasks. Key examples for MRI analysis includes-

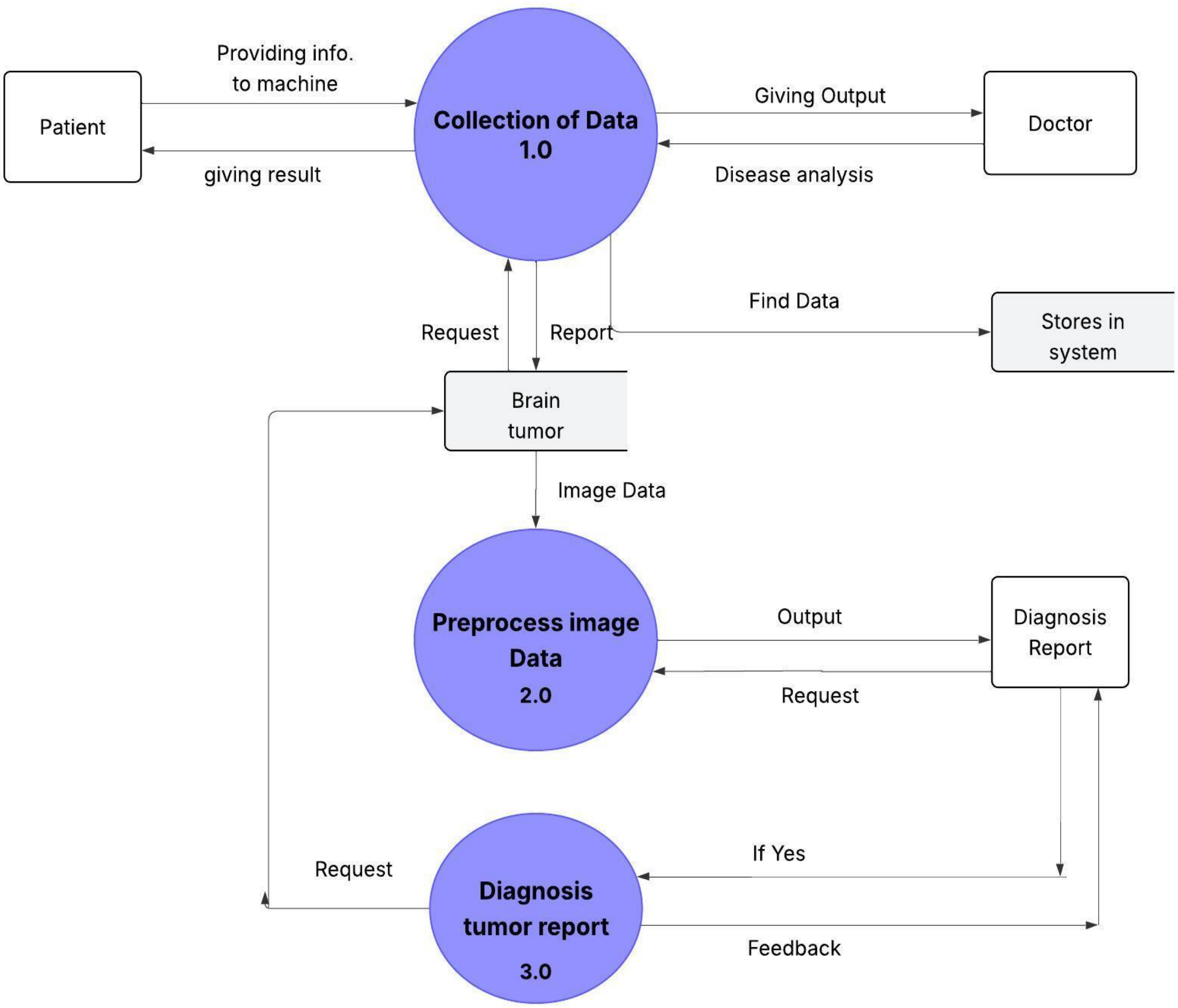
    - **TensorFlow and Keras**: Used for building and training deep neural networks, particularly Convolutional Neural Networks (CNNs) that are effective for medical image analysis.
    - **PyTorch:** Another leading deep learning framework, used to train CNNs for medical image classification.
    - **Scikit-learn**: Provides classic machine learning algorithms for tasks like classification and clustering.
    - **OpenCV**: An open-source computer vision library used for image processing steps, such as resizing and filtering the MRI images.

DFD

Zero level

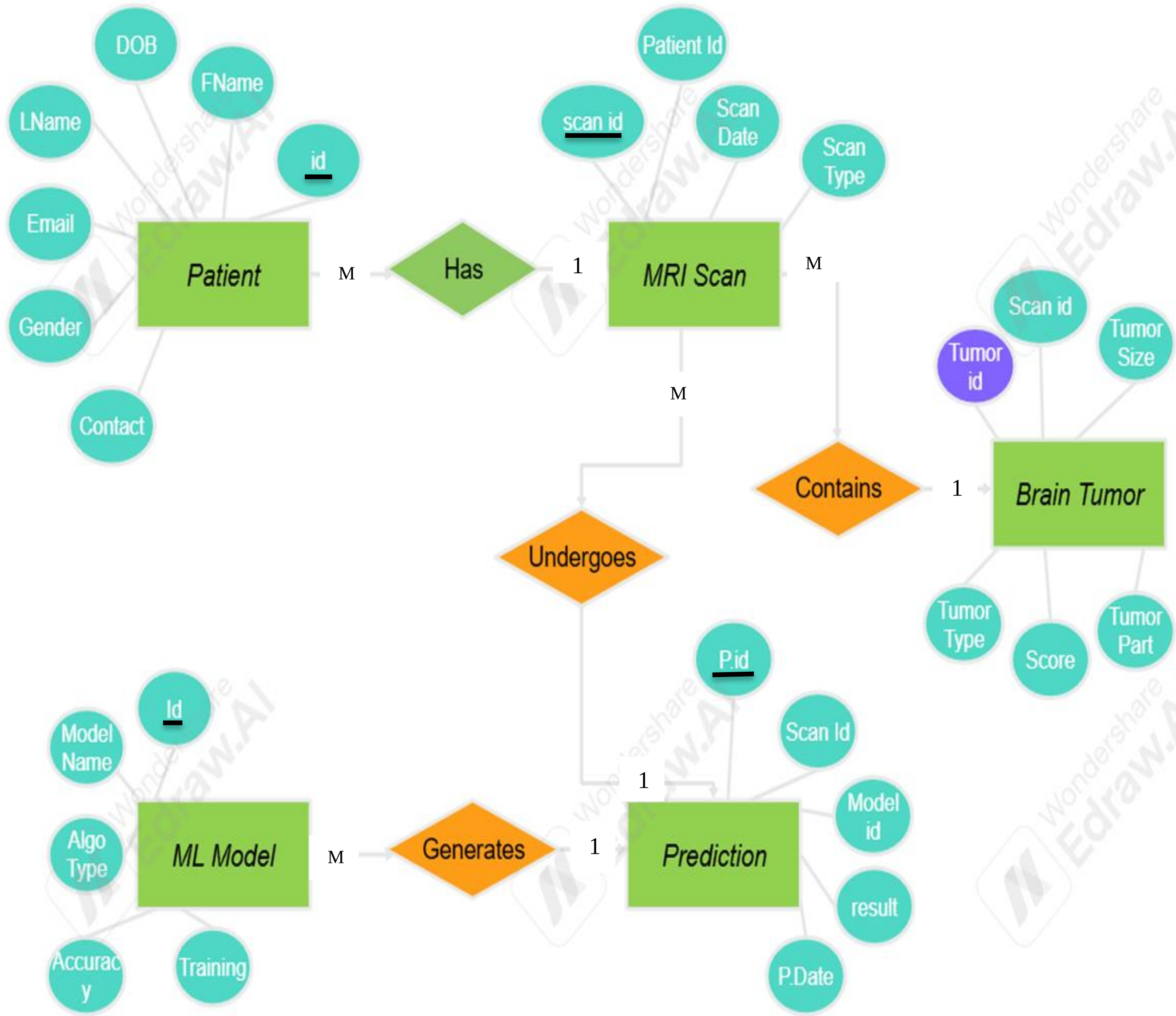


One level



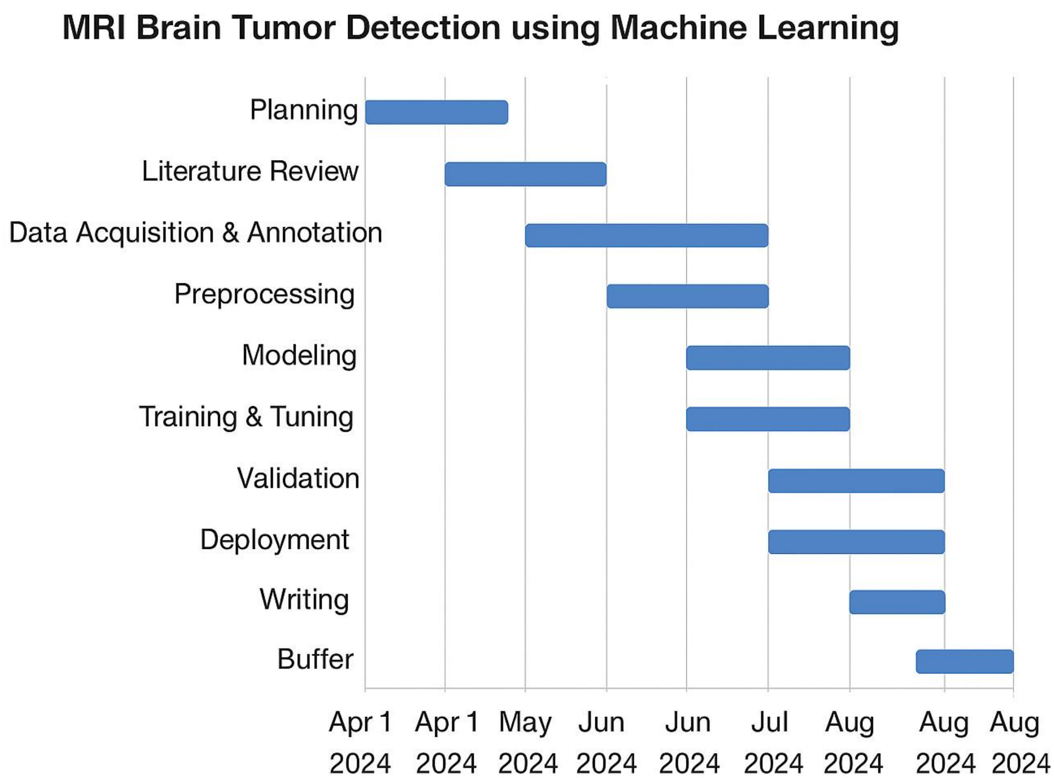
Patient
Providing info.
to machine
giving result
Collection of Data
1.0
Giving Output
Disease analysis
Doctor
Find Data
Stores in
system
Request
Report
Brain
tumor
Image Data
Preprocess image
Data
2.0
Output
Request
Diagnosis
Report
If Yes
Feedback
Request
Diagnosis
tumor report
3.0

ERD



DOB
FName
LName
id
Email
Patient
Gender
Contact
M
Has
1
Patient Id
scan id
Scan Date
Scan Type
MRI Scan
M
M
Scan id
Tumor id
Tumor Size
Contains
1
Brain Tumor
Undergoes
Tumor Type
Score
Tumor Part
P.id
Scan Id
Id
Model Name
Algo Type
ML Model
M
Generates
1
1
Prediction
Model id
result
Accurac y
Training
P.Date

EXPECTED TIME SCHEDULE (Gantt Chart)



IMPACT OF PROPOSED SYSTEM IN ACADEMICS AND INDUSTRY

Impact on academics:

- Faster and more precise research
- Benchmark datasets and collaboration
- Multimodal data integration
- Focus on challenging aspects

Impact on industry:

- Enhanced diagnostic accuracy and speed
- Improved treatment planning
- Predictive analytics
- Workforce augmentation

ROLES AND RESPONSBILITY

1. Ritesh Saxena (Frontend Developer & Researcher)
Primary Responsibility: Ritesh will handle the frontend development of the project, ensuring smooth user interaction.

Tasks:

- **Frontend Development:** Design the user interface using HTML, CSS, and JavaScript, ensuring it is user-friendly and efficient.
- **Collaboration:** Work closely with the backend team to integrate the machine learning model's results into the user interface.
- **Report Writing:** Help document the project's testing process and findings for the final report.

2. Satyam Trivedi (Group Leader, Backend Developer & Model Trainer)
Primary Responsibility: Satyam will manage the backend and machine learning aspects of the project.

- **Backend Development:** Build and maintain the backend using Python and Flask/Django. Develop APIs to link the frontend with the detection model.
- **Model Training:** Implement and train machine learning models, ensuring high accuracy using datasets Kaggle.
- **Leadership:** Oversee the team's progress and ensure timely completion of tasks.

- **Performance Tuning:** Optimize models by adjusting parameters and improving accuracy metrics like precision and recall.

3. Parul Yadav (Backend Developer & Tester)
Primary Responsibility: Himanshu will focus on testing the system and ensuring the model's performance aligns with project goals.

Tasks:

- **Backend Development:** Build and maintain the backend using Python and Flask/Django. Develop APIs to link the frontend with the detection model.

- **System Testing:** Test the entire system, identifying and reporting any bugs or issues.

- **Bug Fixing:** Collaborate with Ayush to resolve errors and improve system reliability

PROS AND CONS

Pros:

- **Increased accuracy and efficiency**: ML models can analyse images with high accuracy, potentially leading to earlier and more precise detection than human analysis alone.
- **Reduced human error**: AI systems eliminate potential human error in the detection process.
- **Consistent and reliable results**: Once trained, models provide consistent analysis, unlike human experts who can have variations.
- **High-resolution image analysis**: Machine learning can process high-resolution MRI scans to find subtle indicators of tumor.

Cons:

- **High computational cost**: Training deep learning models requires powerful and expensive hardware.
- **Large dataset requirement**: A vast amount of labeled MRI data is necessary to train the algorithms effectively.
- **Long training times**: Training these complex models can be a very time-consuming process.
- **Lack of interpretability**: It can be difficult to understand why a model makes a particular prediction, which is a significant hurdle in a medical setting. Explainable AI (XAI) techniques are being developed to address this.
- **Need for physician validation**: The results from ML models still require validation and approval from healthcare professionals.

REFERENCES

• Cheng, J., Huang, W., Cao, S., et al. (2015) — "Enhanced performance of brain tumor classification via tumor region augmentation and partition."
→ Used CNN-based segmentation and classification of MRI images.
• Hossain, M. S., & Muhammad, G. (2019) — "Deep learning-based pathology detection for brain MRI images." Computers in Biology and Medicine.
→ Applied deep CNN models for tumor identification.
• Saba, L., et al. (2021) — "Brain tumor segmentation and classification using deep learning and machine learning approaches: a review." Computers in Biology and Medicine.
→ Comprehensive review of ML and DL techniques for brain MRI tumor analysis.
• Afshar, P., Mohammadi, A., & Plataniotis, K. N. (2020) — "Brain tumor type classification via capsule networks."
→ Introduced CapsuleNet architecture for improved tumor classification accuracy.
• Menze, B. H., et al. (2015) — "The Multimodal Brain Tumour Image Segmentation Benchmark (BRATS)." IEEE Transactions on Medical Imaging.
→ Provided the standard dataset (BRATS) used for training ML models in brain tumour detection.
• Nayak, D. R., Dash, R., & Majhi, B. (2016) — "Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests."
→ Traditional ML approach combining wavelet features with ensemble methods.
• Zhao, L., et al. (2018) — "Brain tumour segmentation from MRI using convolutional neural networks." Biomedical Signal Processing.
→ Used CNN for pixel-wise tumour segmentation.→ Traditional ML approach combining wavelet features with ensemble methods.

CONCLUSION

For the purpose of detecting brain tumors, the suggested EDN-SVM method proposes a novel method of image classification by establishing a direct link between all levels and ensuring that data is freely shared among them. Extensive simulations are conducted to test the effectiveness and viability of the suggested model. It has correctly identified the tumors image with a 97.93 % accuracy rate. Several benefits are shown by the data, suggesting that this model combination is worthwhile. To begin, the model automatically extracted the salient characteristics, making feature extraction far less time-consuming and arduous than it would have been with more conventional classifiers. Second, the suggested EDN-SVM model included the best features of deep NN and SVM, the two most well-known and widely-used classifiers for image recognition and classification. The following are the directions that the work going forward should take: (a) The algorithms that were created need to be included in the software that doctors use, and (b) the methodologies and procedures that were proposed in this research can only be used to grayscale photographs. Color images may be utilized to study the same difficulties and also work with 3D brain scans to obtain more effective brain tumors segmentation in future research.