Automated Brain Tumor Detection from MRI Scans using Deep Learning with Django Web Deployment

Prof. Neha Rai[1], Harsha Dhumal[2], Prasad Gaikwad[3], Siddhesh Ghode[4], Gauri Kotekar[5]
[1]Associate Professor, Department of Computer Engineering,
[1,2,3,4,5]AISSMS College of Engineering, Savitribai Phule Pune University, Pune, India

***Abstract*—Brain tumor detection is a critical task in medical imaging. Manual diagnosis using MRI scans is time-consuming and error-prone. This paper presents an automated deep learning-based system for detecting and classifying brain tumors using Convolutional Neural Networks (CNNs). Transfer learning models such as VGG16, Efficient Net and ResNet50 are applied to improve classification accuracy on limited datasets. The system is deployed as a web application using Django framework, enabling users to upload MRI scans, obtain classification results, and generate diagnostic reports. Experimental results demonstrate that the system achieves reliable accuracy, assisting radiologists in early detection and treatment planning.**

***Index Terms*—Brain Tumor, Deep Learning, CNN, ResNet50, Efficient Net, Django, MRI Classification**

I. INTRODUCTION

In today's healthcare domain, brain tumors represent a serious medical challenge. Brain tumors are abnormal growths of cells in the brain that can be life-threatening. Early detection and classification are crucial for successful treatment planning. Traditional diagnosis through MRI scans depends heavily on expert radiologists, which is both time-consuming and susceptible to human error. With advances in Artificial Intelligence (AI), Deep Learning (DL) techniques have proven highly effective for medical image analysis. This study focuses on developing an automated web-based brain tumor detection system integrating Deep Learning and Django web framework.

II. LITERATURE REVIEW

Brain tumor detection and classification using Artificial Intelligence (AI) and Deep Learning (DL) have been explored extensively in recent years. Several studies demonstrate the potential of machine learning and transfer learning models in improving diagnostic accuracy and efficiency.

1. Brain Tumor Detection and Classification Using Intelligence Techniques: An Overview (2023) by Shubhangi Solanki, Uday Pratap Singh, Siddharth Singh Chouhan, and Sanjeev Jain provides a comprehensive study of MRI-based brain tumor detection using AI. The authors compared Deep Learning (DL), Transfer Learning (TL), and Machine Learning (ML) models, focusing on methodologies, datasets, and emerging trends. Their work emphasizes how intelligent systems enhance diagnostic precision and streamline tumor analysis.

2. An Efficient Brain Tumor Detection and Classification Using Pre-Trained Convolutional Neural Network Models (2024) by K. Nishanth Rao, V. Krishnasree, Aruru Sai Kumar, and S. Siva Priyanka proposed the integration of MRI imaging with CNN-based architectures. Their research highlighted that deep learning approaches significantly improve detection speed and accuracy. The study demonstrated that transfer learning models such as ResNet50 and Efficient Net yield higher performance compared to traditional CNNs.

3. Breakthroughs in Brain Tumor Detection: Leveraging Deep Learning and Transfer Learning for MRI-Based Classification (2025) by Kiana Kiashemshaki, Alireza Golkarieh, Sajjad Rezvani Boroujeni, Maryam Deldadehasl, Hamed Aghayarzadeh, and Azita Ramezani explored advanced deep learning techniques for MRI-based tumor classification. Their results confirmed that Efficient Net achieves superior accuracy among modern

architectures, proving the importance of transfer learning in medical image classification tasks.

III. PROBLEM STATEMENT

The detection and classification of brain tumors from Magnetic Resonance Imaging (MRI) scans remain complex, time-consuming, and highly dependent on the expertise of radiologists. Manual interpretation of MRI images is often prone to human error, leading to delayed or inaccurate diagnoses that can adversely affect patient treatment outcomes. With the increasing volume of medical imaging data, traditional diagnostic methods are insufficient to ensure timely and consistent analysis. Therefore, there is a critical need for an automated, accurate, and efficient system that can detect and classify brain tumors from MRI scans using deep learning techniques. Such a system would assist medical professionals by reducing diagnostic time, minimizing human error, and improving the overall reliability and accessibility of early brain tumor detection.

IV. METHODOLOGY

The proposed system employs a Convolutional Neural Network (CNN) based architecture for automatic brain tumor classification from MRI scans. The workflow, as illustrated in Figure [X], consists of six primary stages: Upload, Preprocess, CNN Model, Result, Report, and Database. MRI images are first uploaded through a web interface and preprocessed via resizing, normalization, and augmentation. The processed images are then fed into a ResNet50 CNN model for feature extraction and classification into four categories: Glioma, Meningioma, Pituitary, and No Tumor. The prediction results are then compiled into a visualized PDF report including accuracy metrics. Finally, all results and logs are stored securely in a MySQL database for further reference and analysis.

1. System Overview

The proposed AI-based brain tumor detection system follows a structured pipeline that automates the complete process from MRI image upload to result storage. The workflow (see Figure [X]) comprises six sequential stages: Upload → Preprocess → CNN Model → Result → Report → Database.

Each stage plays a crucial role in ensuring the efficiency, accuracy, and reproducibility of the tumor classification process.

Stage 1: Image Upload

- Input Source: Users upload MRI scan images through a web-based interface.
- Format Support: The system accepts standard medical image formats (e.g., PNG, JPG, JPEG, DICOM).
- Functionality: Once uploaded, the images are temporarily stored in the application's server for preprocessing.

This stage serves as the entry point for the system, allowing users such as radiologists or researchers to provide input images for automated analysis.

Stage 2: Image Preprocessing

- Objective: Enhance image quality and standardize data before feeding it into the CNN model.
- Processes Involved:
- Resizing: All images are resized to 224 × 224 pixels to match the input dimensions required by the ResNet50 model.
- Normalization: Pixel intensity values are scaled between 0 and 1 to ensure consistent learning behavior and reduce training bias.
- Data Augmentation: Techniques such as rotation, flipping, and zooming are applied to increase dataset diversity and prevent model overfitting.
- Noise Removal: Optional filters can be applied to remove artifacts or noise from the MRI scans.

This preprocessing ensures uniformity and improves the generalization ability of the neural network.

Stage 3: CNN Model (Feature Extraction & Classification)

- Model Used: ResNet50, a deep Convolutional Neural Network pre-trained on ImageNet.
- Functionality:
- Feature Extraction: The CNN automatically identifies and extracts important features (edges, textures, and tumor regions) from MRI images.
- Classification Layer: The extracted features are passed through fully connected layers that classify the MRI scan into one of four categories:
- Glioma
- Meningioma
- Pituitary
- No Tumor

- Training: The model is fine-tuned using a labeled brain MRI dataset to improve accuracy and robustness.

This stage is the core of the system, where deep learning is leveraged for accurate medical image analysis.

Stage 4: Result Generation

- Output: The trained CNN model outputs a prediction label and a confidence score indicating the probability of each class.
- Interpretation: The system interprets these probabilities to determine the most likely tumor type.
- Example Output: Predicted Class: Glioma Confidence: 96.5%

This automated result generation allows for fast and accurate medical assessment.

Stage 5: Report Generation

- Purpose: Provide users with a structured summary of the results.
- Process:
  - The results (predicted class, accuracy, and confidence scores) are formatted into a PDF report.
  - The report includes:
- Patient information (if available)
- Model performance metrics (accuracy, precision, recall)
- Graphical visualizations (e.g., bar charts or confusion matrices)

This report serves as a diagnostic reference for healthcare professionals.

Stage 6: Database Integration

- Database Used: MySQL
- Functionality:
  - Stores uploaded images, prediction results, reports, and log data.
  - Maintains a secure and retrievable record for future use or research validation.
  - Enables scalability and integration with hospital management systems.

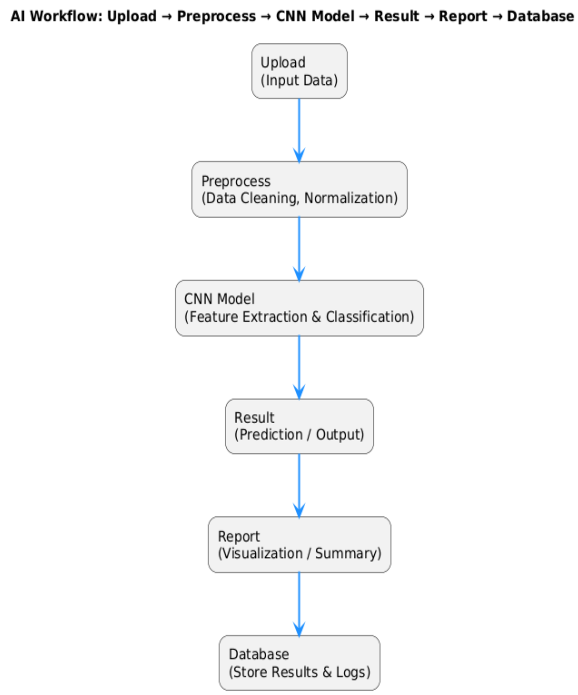
This stage ensures data persistence, traceability, and auditability of all predictions and reports.

Workflow Summary

The overall workflow can be visualized as follows (referencing your provided diagram):

Upload → Preprocess → CNN Model → Result → Report → Database

Each block represents a modular step, ensuring clarity, efficiency, and smooth transition between stages. The architecture supports scalability and can be integrated with cloud-based deployment for real-time medical use.



V. SYSTEM ARCHITECTURE

The proposed Brain Tumor Detection System follows a modular, multi-layered architecture designed for scalability, efficiency, and interoperability. As illustrated in Figure [X], the system consists of four main layers:

1. Client Layer
2. Application Layer
3. Data Layer
4. Training Environment (Offline/GPU System)

Each layer performs specific functions that collectively enable automated brain tumor detection, classification, and reporting using MRI images. The overall system is implemented using Django (Python framework) on the backend, with TensorFlow/Keras for deep learning model training and MySQL/SQLite for data management.

1. Client Layer

The Client Layer acts as the user interaction interface, allowing doctors, patients, and administrators to communicate with the system. It provides both web-based and mobile-based access.

- Web Browser Interface (HTML/CSS/JS): The web interface allows users to upload MRI scans, view reports, and access diagnostic results. Doctors can log in to review patient results, while patients can track their medical reports.
- Mobile Interface (Responsive UI): A mobile-friendly version of the application enables users to interact conveniently from any device. The responsive design ensures accessibility and consistent functionality across platforms.
- Authentication Service: The system includes a secure login and registration mechanism to verify user credentials. It prevents unauthorized access and maintains patient data privacy.
- Functions:
  - Upload MRI scans
  - Register/Login
  - View diagnostic reports and predictions
  - Manage user accounts (admin, doctor, patient roles)

This layer ensures an intuitive user experience while maintaining secure and structured access to the system.

2. Application Layer (Django Framework)

The Application Layer serves as the core processing unit of the system. It handles image preprocessing, model inference, prediction generation, and report compilation. Built using the Django framework, this layer ensures modular development, scalability, and efficient communication between client and data layers.

Key Modules:

- a) Image Upload & Preprocessing (OpenCV / NumPy): This module handles uploaded MRI images, performing essential preprocessing tasks such as:
  - Image resizing to 224×224 pixels
  - Normalization and augmentation
  - Noise reduction and grayscale conversion (if applicable)

The preprocessed images are then forwarded to the AI model for classification.

- b) AI Prediction Service (ResNet50 / VGG16): This component represents the core inference engine. It uses deep learning models such as ResNet50 or VGG16, pre-trained on medical imaging datasets and fine-tuned for brain tumor classification.
- Inputs: Preprocessed MRI images
- Outputs: Predicted tumor class (Glioma, Meningioma, Pituitary, No Tumor) and confidence scores
- Frameworks: TensorFlow and Keras
- c) Report Generator (PDF via Report Lab): After classification, this module automatically generates a detailed PDF report using Report Lab. The report includes:
- Patient ID and image details
- Prediction results and confidence levels
- Model accuracy metrics
- Visual elements such as bar graphs or confusion matrices
- d) Dashboard / Result Viewer: Provides a user-friendly visualization of results. The dashboard allows doctors and patients to view previous scans, download reports, and analyze trends.

The Application Layer acts as the intermediary between the user-facing interface and the backend systems, orchestrating the flow of data and predictions.

3. Data Layer

The Data Layer is responsible for secure data storage and retrieval. It ensures persistent management of user information, uploaded images, prediction results, and generated reports.

Key Components:

- a) File Storage (Images / Reports): All uploaded MRI scans and generated PDF reports are stored securely in the file system. This structure allows for efficient retrieval and backup.
- b) Database (MySQL / SQLite): The database stores structured data such as:
- User credentials and authentication tokens
- Prediction logs and metadata
- File references and timestamps

MySQL is preferred for deployment environments due to its scalability and robustness, while SQLite is used for local development and testing.

- Data Flow:
- Stores prediction details and report metadata from the Application Layer
- Retrieves data for user dashboards and report viewing
- Maintains audit logs for medical compliance

This layer ensures the reliability, security, and traceability of all patient and system data.

4. Training Environment (Offline / GPU System)
The Training Environment operates as an independent offline module, primarily used for training and fine-tuning the deep learning models before deployment.
Components:

- a) Model Trainer (TensorFlow / Keras): This component is responsible for training the CNN model using labeled MRI datasets. The ResNet50 and VGG16 architectures are fine tuned to achieve high classification accuracy. Training is performed on a GPU-based system to accelerate computation.
- b) Model Repository (Saved Weights / Metadata): Once training is complete, the trained weights and metadata are stored in a Model Repository. During inference, the Application Layer loads these weights to perform predictions in real time.

Workflow

1. Model training and validation occur offline.
2. Trained weights are saved in the repository.
3. During deployment, the AI Prediction Service loads these trained models for inference.

This separation between training and inference ensures system stability, scalability, and the ability to update models without interrupting the live environment.

System Flow Summary
The complete process flow of the system is as follows:

1. User Interaction: A doctor or patient accesses the system through the web or mobile interface.
2. Image Upload: The user uploads an MRI image.
3. Preprocessing: The image undergoes resizing, normalization, and enhancement.
4. Model Inference: The preprocessed image is analyzed by the trained CNN model (ResNet50/VGG16).
5. Prediction Output: The system classifies the tumor type and generates a confidence score.
6. Report Generation: A PDF report is automatically generated and stored.
7. Data Storage: All results, reports, and logs are securely saved in the database.
8. Result Viewing: The doctor or patient can view or download the report through the dashboard.

6. Advantages of the Proposed Architecture

- Modular design for easy maintenance and updates
- Scalable cloud-ready backend (Django + MySQL)
- Secure user authentication and data storage
- Integration of deep learning (ResNet50/VGG16) for high-accuracy predictions
- Automatic report generation for real-time medical insights

VI. CONCLUSION

The proposed Brain Tumor Detection System integrates Artificial Intelligence and Web Technology to create an efficient and accessible diagnostic platform. It automates the process of brain tumor detection, reducing manual effort and providing faster and more reliable results.

I. The system uses advanced deep learning models such as Efficient Net, ResNet50, and VGG16 to accurately classify MRI brain scans into Glioma, Meningioma, Pituitary, or No Tumor categories. These models enhance diagnostic accuracy and provide confidence scores that help doctors make better-informed decisions.

II. Through the Django-based web interface, users can easily upload MRI images, view real-time predictions, and download detailed PDF reports containing patient and diagnostic information. The inclusion of a database ensures that all patient records, scan details, and reports are securely stored and easily +retrievable when needed.

III. Overall, this project demonstrates how combining AI-driven image analysis with a web-based platform can support radiologists, minimize diagnostic time, and move healthcare closer to a smart, technology-powered future.

REFERENCES

[1] Brain Tumor Detection and Classification Using Intelligence Techniques: An Overview (2023)
[2] An efficient brain tumor detection and classification using pre-trained convolutional neural network models. (2024)
[3] Breakthroughs in Brain Tumor Detection: Leveraging Deep Learning and Transfer Learning for MRI-Based Classification. (2025)

[4] S. Bhuvaji et al., “Brain Tumor Classification (MRI),” Kaggle Dataset, 2020. Available: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

[5] S. Deepak and P.M. Ameer, “Brain Tumor Classification Using Deep CNNs with Transfer Learning and Data Augmentation,” Int. J. Imaging Systems and Technology, 2019.

[6] I. Ali et al., “ResNet-Based Deep Learning Framework for Detection of Brain Tumors in MRI Images,” IEEE Access, vol. 9, 2021, pp. 162712–162723.