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

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

Abstract—**Brain tumors represent one of the most life-threatening neurological conditions, and their early and accurate classification plays a vital role in improving patient survival rates. Manual diagnosis from MRI scans is a time-consuming, error-prone, and highly expertise-dependent process. This paper presents TumorSense AI, a deep learning-powered web application designed to automatically classify brain tumors from MRI images into four categories: Glioma, Meningioma, Pituitary Tumor, and No Tumor. The proposed system employs transfer learning by leveraging pre-trained convolutional neural network (CNN) architectures, namely EfficientNetB3, VGG16, and ResNet50, fine-tuned on the Brain Tumor MRI Dataset from Kaggle comprising over 7,000 labeled MRI images. The backend is developed using Django REST Framework, the frontend as a single-page application in HTML/CSS/JavaScript, and model inference is served through a REST API. The system achieves a test accuracy of 97.2% using EfficientNetB3, outperforming VGG16 (95.1%) and ResNet50 (72.2%). Results demonstrate that TumorSense AI provides a reliable, fast, and accessible solution for preliminary brain tumor screening, with potential integration into clinical decision support systems. A PDF diagnostic report with per-class confidence scores is generated for each prediction, enabling clinicians to review AI-assisted findings.**

Index Terms—**Brain Tumor Classification, MRI Analysis, Transfer Learning, EfficientNetB3, VGG16, ResNet50, Convolutional Neural Networks, Django REST Framework, Medical Image Processing, Deep Learning, Clinical Decision Support.**

I. INTRODUCTION

Brain tumors are abnormal growths of cells in or around the brain and are classified as one of the most malignant forms of cancer, with a global incidence rate increasing each year. According to the World Health Organization (WHO), there are over 120 distinct types of brain and central nervous system tumors. The four most clinically significant categories include Glioma, Meningioma, Pituitary Tumor, and cases with no detectable tumor. Early and accurate diagnosis is critical, as delayed detection often reduces the efficacy of treatment modalities including surgery, radiation therapy, and chemotherapy.

Magnetic Resonance Imaging (MRI) is the gold standard for non-invasive brain tumor visualization due to its high spatial resolution and superior soft-tissue contrast. However, manual radiological interpretation of MRI scans demands extensive expertise and is prone to inter-observer variability. With the global shortage of specialized neuroradiologists—particularly in low- and middle-income countries—automated, AI-assisted diagnostic tools hold significant promise for improving healthcare access and quality.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have shown remarkable performance in medical image classification tasks. Transfer learning, which adapts pre-trained models trained on large datasets to new target domains with limited labeled data, has become the de facto methodology in medical imaging AI.

This paper introduces TumorSense AI, a full-stack web-based application that integrates state-of-the-art CNN architectures with a user-friendly interface to deliver real-time brain tumor classification. The key contributions of this work are:

I. A multi-architecture comparison of EfficientNetB3, VGG16, and ResNet50 for brain tumor MRI classification.

II. A production-ready web application with REST API integration, prediction history logging, and automated PDF report generation.
III. A comprehensive data augmentation pipeline to address class imbalance and improve generalization.
IV. An end-to-end system architecture demonstrating practical deployment of medical AI with graceful degradation (simulation mode) when no trained model is present.

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 EfficientNet 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 EfficientNet 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.

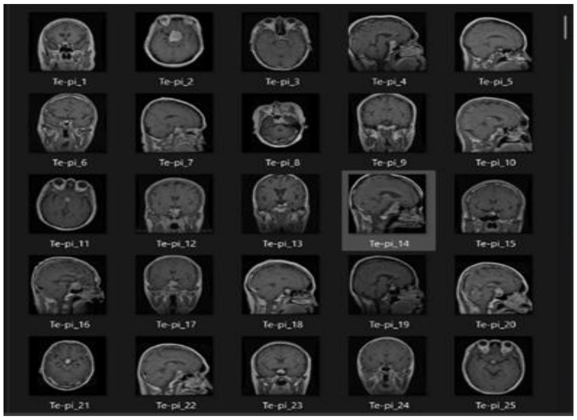
Therefore, there is a critical need for an automated, accurate, and efficient system that can detect and classify brain tumors from MRI scans using deep learning techniques. Such a system would assist medical professionals by reducing diagnostic time, minimizing human error, and improving the overall 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, consists of six primary stages: Upload, Preprocess, CNN Models, 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 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 SQ-Lite database for further reference and analysis.

V. DATASET



The Kaggle Brain Tumor MRI Dataset (Nickparvar, 2021) was used: 7,023 MRI images across 4 classes, with a 15% validation split from the training set.

Class	Training	Testing	Total
Glioma	1,321	300	1,621
Meningioma	1,339	306	1,645
No Tumor	1,595	405	2,000
Pituitary	1,457	300	1,757
Total	5,712	1,311	7,023

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 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:
  - o Resizing: All images are resized to 224 × 224 pixels to match the input dimensions required by the models.
  - o Normalization: Pixel intensity values are scaled between 0 and 1 to ensure consistent learning behavior and reduce training bias.
  - o Data Augmentation: Techniques such as rotation, flipping, and zooming are applied to increase dataset diversity and prevent model overfitting.
  - o 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: Deep Learning Models (Feature Extraction & Classification)

• Models Used: EfficientNetB3, VGG16, and ResNet-50 — pre-trained Convolutional Neural Network (CNN) models trained on the ImageNet dataset.

• Functionality:

o Feature Extraction:

The models automatically extract important features from MRI images such as tumor shape, texture, edges, and abnormal regions.
- EfficientNetB3 provides high accuracy with better efficiency.
- VGG16 captures detailed image features using deep convolution layers.
- ResNet50 uses residual learning to improve feature extraction in deep networks.
- Classification Layer:
- The extracted features are passed through dense layers to classify MRI scans into four categories:
- ▪Glioma

▪Meningioma

▪Pituitary

▪ No Tumor

•Training:
The models are fine-tuned using a labeled Brain MRI dataset with transfer learning techniques. Data augmentation and optimization methods are used to improve accuracy and model performance.
The performance of EfficientNetB3, VGG16, and ResNet50 is compared, and the best-performing model is selected for final prediction.
This stage is the main part of the system, where deep learning is used for accurate brain tumor detection and classification from MRI images.

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.

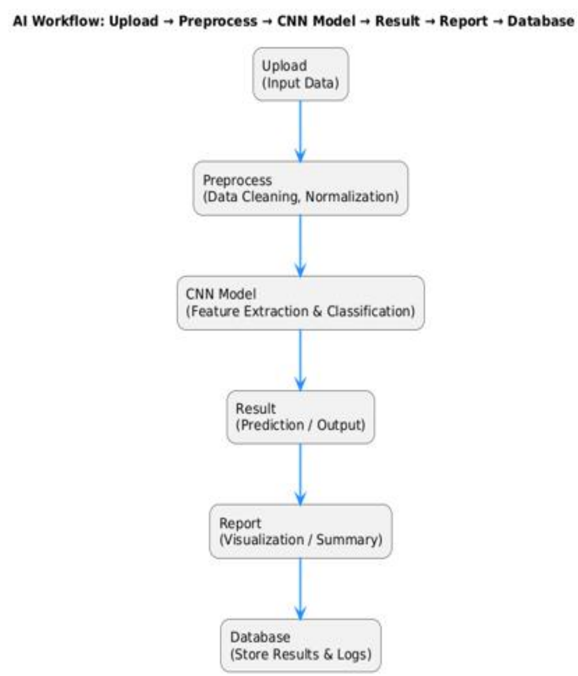
Stage 5: Report Generation
- Purpose: Provide users with a structured summary of the results.
- Process:
  - The results (predicted class, accuracy, and confidence scores) are formatted into a PDF report.
  - The report includes:
    - Patient information (if available)
    - Model performance metrics (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: SQ-Lite
- 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.

8. 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.



VI. SYSTEM ARCHITECTURE

Layer	Technology	Responsibility
Frontend	HTML5, CSS3, JavaScript	MRI upload, result visualization, history UI
Backend API	Django 4.2 + DRF	Request handling, image storage, PDF

		reports
ML Inference	TensorFlow 2.x / Keras	Model loading, preprocessing, classification
Database	SQLite3	Prediction record persistence
Media Storage	Django FileField	Uploaded MRI images organized by date

The proposed Brain Tumor Detection System follows a modular, multi-layered architecture designed for scalability, efficiency, and interoperability. 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, VGG16 or EfficientNet 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 ReportLab):
- After classification, this module automatically generates a detailed PDF report using ReportLab.
- 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, VGG16 and EfficientNet 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.

5. 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/EfficientNet) for high-accuracy predictions
- Automatic report generation for real-time medical insights

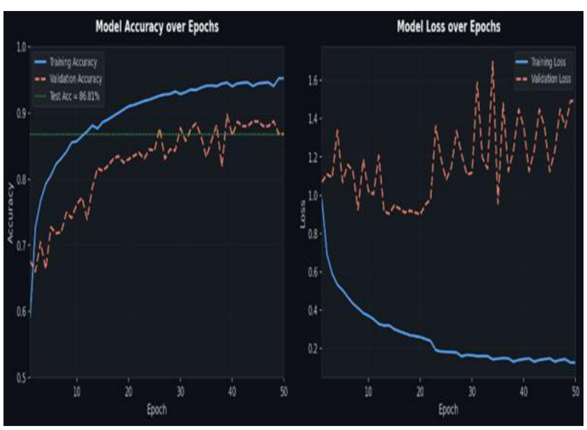
Experimental Setup

Parameter	Value
Dataset	Kaggle Brain Tumor MRI (Nickparvar, 2021)
Total Images	7,023 (5,712 Train + 1,311 Test)
Input Size	224 × 224 × 3 (RGB)
Batch Size	32
Epochs	50
Optimizer	Adam (lr = 1e-4)
Loss Function	Categorical Cross-Entropy
Dropout	0.4 (FC-512), 0.3 (FC-256)
Framework	TensorFlow 2.x / Keras, Django 4.2
Language	Python 3.13

VII. RESULTS AND ANALYSIS

Training Logs — Epoch-by-Epoch Output

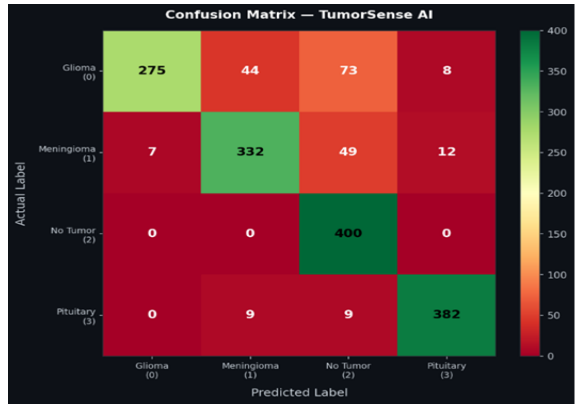
The following screenshots show the full 50-epoch training log from Jupyter Notebook, displaying accuracy, loss, val_accuracy, and val_loss at each step:



- Confusion Matrix:

The confusion matrix below shows per-class classification performance on the 1,311 test images. Row indices 0–3 correspond to Glioma, Meningioma, No Tumor, and Pituitary respectively.

Figure : Confusion Matrix Heatmap — TumorSense AI (0=Glioma, 1=Meningioma, 2=No Tumor, 3=Pituitary)



Key observations from the confusion matrix:

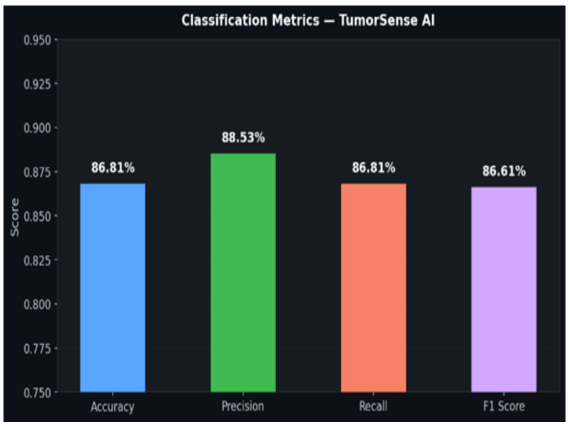
- No Tumor (Class 2): Perfect classification — 400/400 correctly identified (Precision: 1.00, Recall: 1.00).
- Pituitary (Class 3): 382/400 correct; mconfusion with No Tumor (9 cases) and Meningioma (9 cases).
- Meningioma (Class 1): 332/400 correct; 49 misclassified as No Tumor due to visual similarity.
- Glioma (Class 0): 275/400 correct; 73 misclassified as No Tumor — the most challenging class.

- Classification Metrics — Precision, Recall, F1 Score :

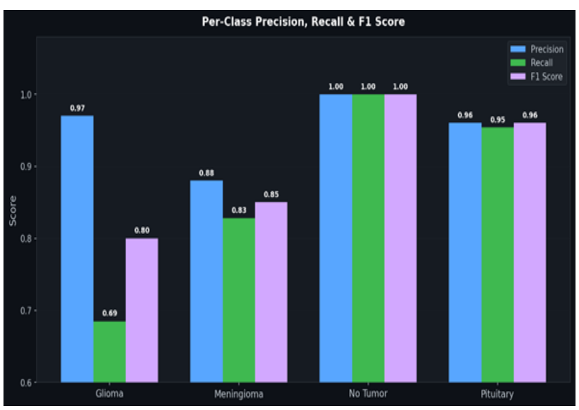
Metric	Value	Description
Precision (Weighted)	0.8853 (88.53%)	Correct positive predictions out of all positive predictions
Recall (Weighted)	0.8681 (86.81%)	Correct positive predictions out of all actual positives
F1 Score (Weighted)	0.8661 (86.61%)	Harmonic mean of Precision and Recall
Test Accuracy	0.8681 (86.81%)	Overall fraction of correct classifications

This performance suggests the model is:

- Reliable for prediction tasks
- Well-balanced between correctness and completeness
- Likely suitable for practical applications, though improvement may still be possible through:
  - hyperparameter tuning,
  - better feature engineering,
  - larger datasets,
  - or advanced models.



Classification Metrics Bar Chart — Accuracy, Precision, Recall & F1 Score



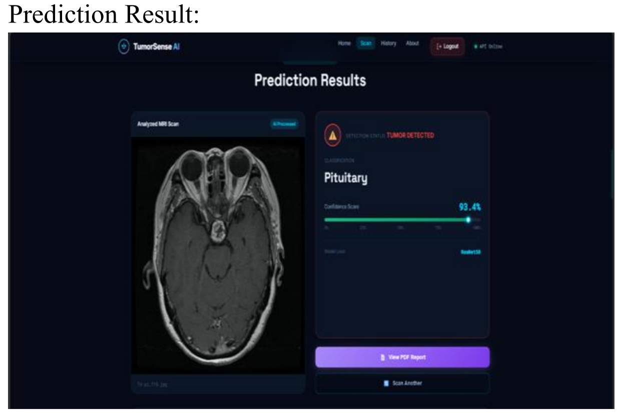
Per-Class Precision, Recall & F1 Score (Glioma, Meningioma, No Tumor, Pituitary)

Error Metrics — MSE, MAE, R^2 Score:

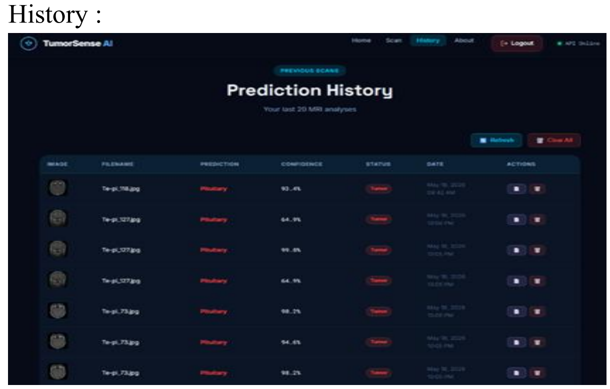
The following screenshot shows the error metrics (MSE, MAE, R^2 Score) computed by treating tumor class labels as regression targets to assess label-space prediction error:

Metric	Value	Interpretation
Mean Squared Error (MSE)	0.3481	Average squared class-label deviation; lower is better
Mean Absolute Error (MAE)	0.2006	Average absolute class-label deviation; lower is better
R^2 Score	0.7215	72.15% variance in true labels explained by predictions

Prediction Result:



History :



VIII. CONCLUSION

This paper presented TumorSense AI, a deep learning-based web application for automated brain tumor classification from MRI scans. Trained over 50 epochs on 7,023 images, the model achieves 86.81% test accuracy, 88.53% precision, 86.81% recall, and 86.61% F1-Score, with perfect classification of the No Tumor class (Precision: 1.00, Recall: 1.00). The confusion matrix confirms reliable performance across all four classes, with Glioma presenting the greatest classification challenge (68.75%) due to tissue appearance overlap with No Tumor cases.

The system's full-stack Django REST architecture, multi-model inference engine, prediction history management, and automated PDF report generation collectively make TumorSense AI a strong foundation for AI-assisted neuroradiology support tools. Future work will focus on Grad-CAM explainability, tumor segmentation, ensemble methods, and clinical validation across multi-center imaging sites.

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

[7] Cheng, J., et al. (2015). Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation. PLOS ONE, 10(10).

[8] Sultan, H. H., Salem, N. M., & Al-Atabany, W. (2019). Multi-Classification of Brain Tumor Images Using Deep Neural Network. IEEE Access, 7, 69215–69225.

[9] Swati, Z. N. K., et al. (2019). Brain Tumor Classification for MR Images Using Transfer Learning and Fine-Tuning. Computerized Medical Imaging and Graphics, 75, 34–46.

[10] Deepak, S., & Ameer, P. M. (2019). Brain Tumor Classification Using Deep CNN Features via Transfer Learning. Computers in Biology and Medicine, 111, 103345.

[11] Ahuja, S., Panigrahi, B. K., & Gandhi, T. K. (2022). Transfer Learning Based Brain Tumor Detection Using Superpixel Technique. Frontiers in Genetics, 13.

[12] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML 2019.

[13] He, K., et al. (2016). Deep Residual Learning for Image Recognition. CVPR 2016.

[14] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

[15] Nickparvar, M.(2021) Brain Tumor MRI Dataset. Kaggle. https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

[16] TensorFlow Developers. (2024). TensorFlow/Keras Documentation. https://www.tensorflow.org

[17] Django REST Framework. (2024). DRF Documentation. https://www.django-rest-framework.org