Brain Tumor Vision AI

AI - Powered Deep Learning System for Brain Tumor Prediction and Classification Using MRI Images

Research Project Report

Dr. Subramani Suresh

1. Abstract

This project introduces **Brain Tumor Vision AI**, an advanced deep-learning–based diagnostic tool designed for accurate and efficient brain tumor detection using MRI scans. The system automatically classifies MRI images into four categories-**Glioma, Meningioma, Pituitary Tumor, and No Tumor**-through a complete pipeline encompassing preprocessing, dataset balancing, augmentation, model selection, training, and model explainability.

Three leading CNN architectures - **VGG16, DenseNet121, and MobileNetV3-Large** - were evaluated under identical training conditions. After resolving significant dataset imbalance through a custom resampling and re-splitting strategy, **MobileNetV3-Large emerged as the best-performing model**, with validation accuracy improving from **78.93% to 95.11%**, demonstrating that *dataset quality, not architecture*, was the primary performance bottleneck.

To ensure clinical transparency, **Grad-CAM heatmaps** were integrated to visually highlight tumor-affected regions, enabling interpretability and supporting clinician trust. The final deployed tool, **Brain Tumor Vision AI**, is accurate, lightweight, explainable, and optimized for deployment via a Django web application-making it suitable for research, prototype clinical use, and real-world diagnostic decision support.

2. Problem Statement

Brain tumor detection using MRI scans is a critical but time-consuming task that requires expert radiologists. Manual analysis can lead to delays, variability in diagnosis, and potential human errors.
Additionally, MRI datasets often suffer from issues such as:

- **Class imbalance** (more images in one class than others)
- **Incorrect train–test splits**
- **Mixed data distributions**
- **Low-quality or noisy validation sets**

These issues cause models to achieve **high training accuracy but poor validation accuracy**, making predictions unreliable in real-world clinical settings.

The core problem:
How can we automatically detect and classify different types of brain tumors from MRI images using AI and deep learning, while ensuring accuracy, reliability, and consistent performance?

3. Training Workflow & Model Creation

A complete deep-learning pipeline was developed to ensure reliable MRI-based brain-tumor classification. The workflow included data preparation, model selection, training, and validation using industry-standard best practices.

3.1. Dataset Preparation

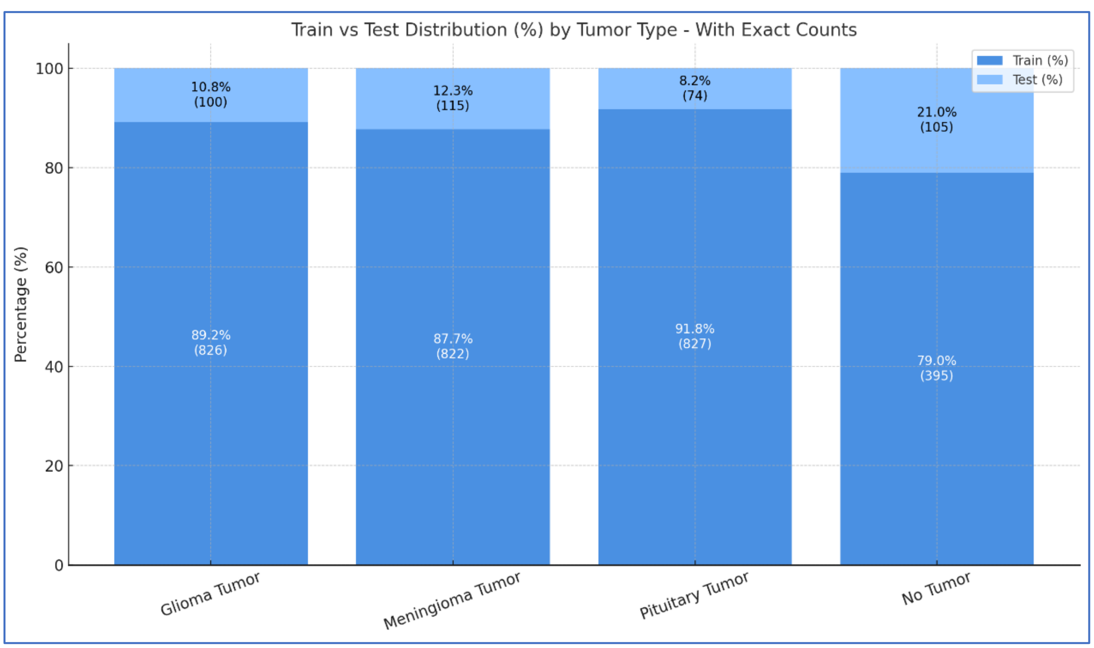
A publicly available Brain MRI Tumor Dataset was downloaded from Kaggle:

🔗 **Kaggle Dataset:** The dataset contains four classes:
https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

Table 1: Brain Tumor MRI Dataset Summary (Kaggle)

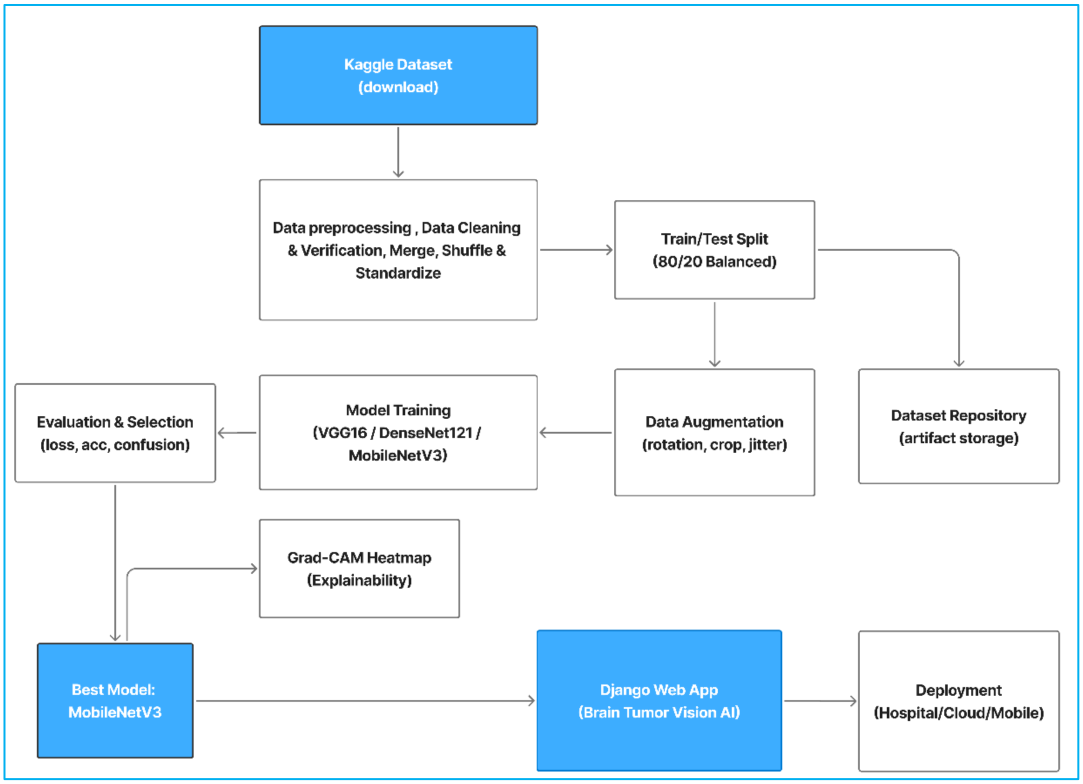
Tumor Type	Train Images	Test Images	Total Images
Glioma Tumor	826	100	**926**
Meningioma Tumor	822	115	**937**
Pituitary Tumor	827	74	**901**
No Tumor	395	105	**500**
Total	**2870**	**394**	**3264**

Figure 1: MRI Dataset Train–Test Distribution by Tumor Type



3.2. Dataset Processing Overview

Figure 2 - System Architecture and methodology workflow for **Brain Tumor Vision AI**, showing data acquisition, preprocessing, augmentation, multi-model training, selection, Grad-CAM explainability and Django deployment.



3.2.1. Original Dataset Issues (Problem Identification)

The initial Kaggle MRI dataset was **highly imbalanced and inconsistently split**, which caused several learning challenges:

- Overfitting with unstable validation accuracy
- High training accuracy but low validation accuracy (46–78%)
- Validation images had a different distribution from training images
- Uneven or incorrect train–test splits
- Significant class imbalance across tumor categories

These issues demonstrated that **dataset quality-not the model-was the main bottleneck** affecting performance.

3.2.2. Dataset Processing Steps (Cleaning & Restructuring)

To correct these issues and prepare the data for reliable model training, the following steps were performed:

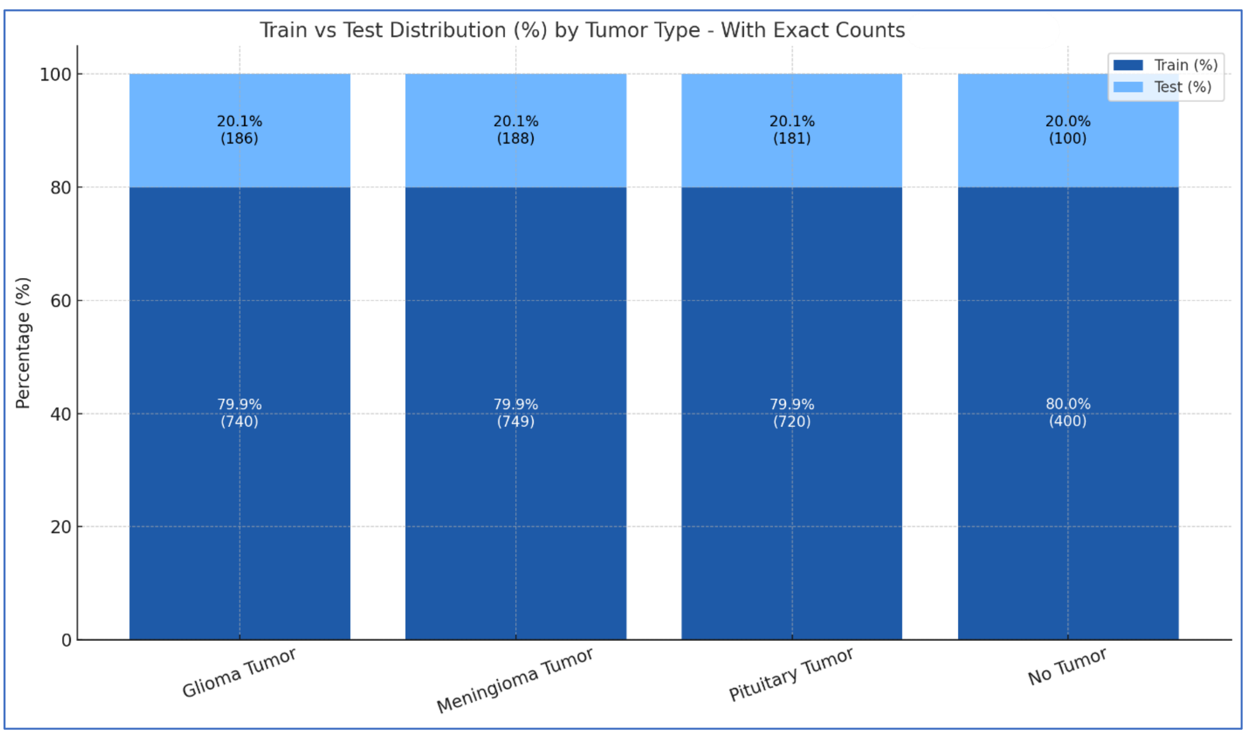
- All MRI images were **verified, cleaned, and standardized**
- Original **train and test folders were merged** to ensure uniform distribution
- The full dataset was **shuffled** to remove ordering bias
- A new **balanced 80/20 train–test split** was generated
- Class imbalance and distribution mismatches were fully corrected

This process produced a **consistent, balanced dataset** that significantly improved model learning and generalization.

Table 2: Final Balanced Dataset Distribution (After Fix)

Tumor class	Train (80%)	Test (20%)	Total
Glioma Tumor	740	186	**926**
Meningioma Tumor	749	188	**937**
Pituitary Tumor	720	181	**901**
No Tumor	400	100	**500**
TOTAL	**2609**	**655**	**3264**

Figure 3: Final Balanced Train–Test Distribution by Tumor Type



3.3. Model Selection

Convolutional Neural Networks (CNNs) were selected for this project because they are the state-of-the-art architecture for medical image analysis. CNNs automatically learn hierarchical spatial features (edges, textures, shapes, patterns) that are essential for accurately detecting tumor regions in MRI scans. Their ability to capture local and global visual structures makes them the most suitable choice for brain tumor classification tasks.

To identify the most effective model,

Table 3. Three widely used CNN architectures were evaluated:

Model	Reason for Selection
VGG16	Classical deep CNN architecture, well-known for strong feature extraction and baseline benchmarking.
DenseNet121	Efficient gradient flow, dense layer connectivity, and high feature reuse-often achieves strong performance on medical datasets.
MobileNetV3-Large	Lightweight, fast, optimized for real-time inference, and ideal for deployment on CPU/GPU environments.

3.4. Model Training

The preprocessed and balanced MRI dataset was used to train all three CNN models-VGG16, DenseNet121, and MobileNetV3-Large-under the same experimental conditions to ensure fair comparison.

Each model was trained using:

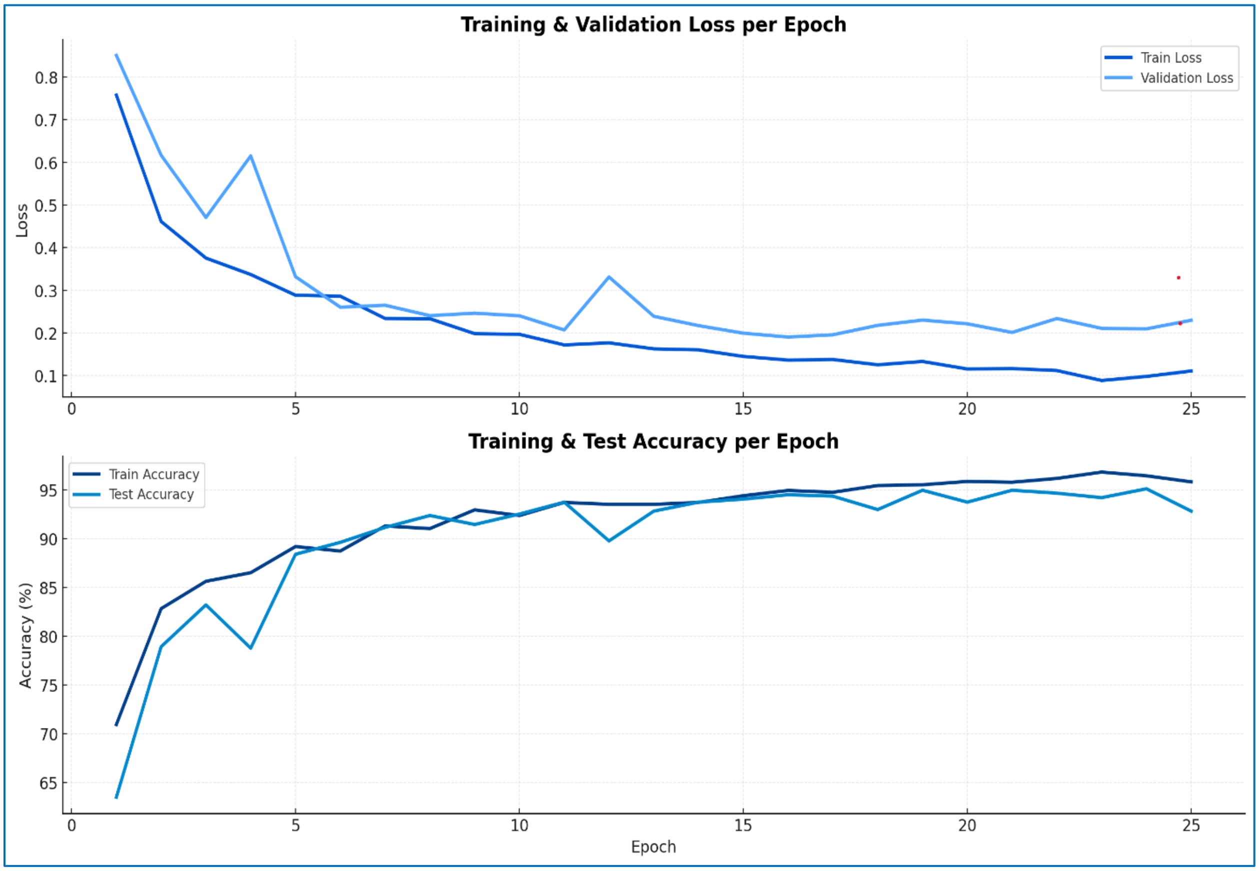
- **Transfer Learning** with pretrained ImageNet weights
- **Data augmentation** (rotation, cropping, jitter, flips) to improve robustness
- **Selective fine-tuning** of the deeper layers
- **Cross-entropy loss + Adam optimizer**
- **Early stopping and best-model checkpointing**

Model performance was evaluated using:

- **Training loss**
- **Validation loss**
- **Validation accuracy**
- **Generalization behavior across epochs**

3.5. Model Validation

Figure 4: Training & Validation Loss and Accuracy Curves for MobileNetV3-Large



After training each CNN model, validation was performed using the balanced 20% test split. The following evaluation techniques were used:

- **Validation accuracy and loss curves** to assess learning stability
- **Confusion-based metrics** to check class-wise performance
- **Grad-CAM heatmaps** to visualize discriminative regions used during prediction
- **Cross-model metric comparison** (VGG16 vs DenseNet121 vs MobileNetV3-Large)

These steps ensured that the best model was selected based on accuracy, generalization strength, and clinical interpretability.

Figure 4 shows a steady decrease in both losses and a consistent rise in accuracy, indicating stable learning, minimal overfitting, and strong generalization. The close alignment between training and validation accuracy demonstrates that dataset balancing significantly improved model performance.

4. Results & Discussion

4.1. Model comparison

A systematic evaluation of VGG16, DenseNet121, and MobileNetV3-Large was performed using identical preprocessing, augmentation, and validation procedures. Table 4 summarizes the comparative performance of all models.

Table 4. Model Performance Summary (VGG16, DenseNet121 and MobileNetV3)

Model	Dataset Condition	Best Validation Accuracy	Best Validation Loss	Notes
VGG16	Original (Imbalanced)	71.32%	15.19	Heavy, slow on CPU, high overfitting, weak generalization
DenseNet121	Original (Imbalanced)	70.56%	1.0570	High-capacity but sensitive to imbalance; requires clean data
MobileNetV3-Large (Before Fix)	Imbalanced Dataset	78.93%	0.6159	Good architecture but limited by dataset mismatch
MobileNetV3-Large (After Balanced Dataset)	**Balanced + Shuffled Dataset**	**95.11%**	**0.2096**	**Best model: fast, stable, high accuracy, strong generalization**

Table 4, The results clearly indicate that VGG16 and DenseNet121 struggled with the imbalanced dataset, producing validation accuracies of only 71.32% and 70.56%. Their heavy architectures and CPU-intensive computations led to slow training and poor generalization.

In contrast, MobileNetV3-Large achieved 78.93% on the imbalanced dataset, but after applying dataset balancing and shuffling, its accuracy dramatically improved to 95.11%, with validation loss dropping from 0.6159 → 0.2096.

These findings confirm that the primary limitation was dataset quality-not the model architecture. After correcting the imbalance and distribution issues, MobileNetV3 delivered superior accuracy, stability, and generalization. With state-of-the-art performance, fast inference speed, and strong robustness, MobileNetV3 stands out as the most reliable and deployment-ready model for MRI brain-tumor classification.

4.2 MobileNetV3 Performances - Before vs After Dataset Balancing

Table 5. MobileNetV3 Performance Comparison – Before vs After Dataset Balancing

Metric	Before (Imbalanced Dataset)	After (Balanced Dataset)	Improvement
Best Train Epoch	22	24	
Train Loss	0.5038	**0.0980**	↓ Major improvement
Train Accuracy	83.51%	**96.44%**	↑ +12.93%
Validation Loss	0.6159	**0.2096**	↓ Stronger generalization
Validation Accuracy	78.93%	**95.11%**	↑ +16.18%
Generalization	Poor - Overfitting	**Excellent - Balanced learning**	Dramatic improvement
Dataset Quality	Unbalanced & mismatched	**Clean, shuffled, balanced**	Fixed distribution

Key Improvements (Integrated in Table 5 Summary)

✓ Training Accuracy increased: 83.51% → 96.44% (↑ +12.93%)
✓ Training Loss reduced drastically: 0.5038 → 0.0980
✓ Validation Accuracy improved significantly: 78.93% → 95.11% (↑ +16.18%)
✓ Validation Loss decreased: 0.6159 → 0.2096
✓ Training became stable and smooth
✓ No overfitting after dataset balancing
✓ Model generalization improved dramatically

Table 5 clearly demonstrates how correcting dataset imbalance drastically enhanced MobileNetV3's performance. After restructuring the dataset with clean, balanced, and properly shuffled splits, the model achieved **much higher accuracy**, **lower loss**, and **excellent generalization**. Training and validation behaviors aligned consistently, confirming that the **balanced dataset enabled MobileNetV3 to reach its full performance potential**.

4.3. Visualization (Grad-CAM Heatmap Integration)

Grad-CAM (Gradient-weighted Class Activation Mapping) was integrated to provide visual explainability for the MRI tumor classification system. This technique highlights the most important regions in the MRI scan that influenced the model's prediction by using gradients from the last convolution layer to generate a heatmap overlay.

In medical imaging, simply predicting *"Tumor"* or *"No Tumor"* is insufficient - clinicians need to understand *why* the model arrived at its decision. Grad-CAM enables this interpretability by clearly indicating tumor-affected brain regions, ensuring that the model focuses on medically relevant structures rather than background noise or artifacts.

In this project, Grad-CAM was applied to the final **MobileNetV3-Large** model to enhance clinical transparency. The resulting heatmaps allow radiologists and researchers to visually confirm detected tumor regions, improving trust, diagnostic confidence, and the practical usability of the AI system.

5. Deployment

The final deployment integrates the optimized MobileNetV3-Large model into a fully functional Django-based web application called "**Brain Tumor Vision AI"**, enabling real-time MRI tumor prediction with clinical-grade interpretability.

5.1. Model Packaging

The final model **best_mobilenetv3_mri_final.pth** was exported after training and validation. Key characteristics:

- PyTorch.pth format
- Supports both CPU and GPU inference
- Lightweight and optimized for real-time classification
- Integrated Grad-CAM module for explainability

This makes the model suitable for clinical prototypes, research workflows, and scalable deployment.

5.2. Integrated Prediction Pipeline

A complete prediction pipeline was implemented within Django, supporting:

Core Capabilities

- **MRI upload** via the web interface
- **Automated preprocessing** (resize, normalization, tensor conversion)
- **Tumor / No-Tumor detection**
- **4-class classification**: Glioma, Meningioma, Pituitary, No Tumor
- **Grad-CAM heatmap generation** to highlight suspected tumor regions
- **Confidence score output** for clinical transparency

Table 6, Tech Stack

Component	Technology	Description
Backend / API	Django (Python)	Handles image upload, processing, and inference
Frontend	HTML, CSS, Bootstrap	Clean and responsive UI
Model Integration	PyTorch	Loads best MobileNetV3 model for prediction
Visualization	Grad-CAM + OpenCV	Generates heatmaps for interpretability
Deployment	Django Web Application	Runs on CPU/GPU; suitable for labs and hospitals

5.3. Real-World Deployment Readiness (Django Web Application)

The final AI system has been fully deployed as a Django-based clinical web application, enabling seamless MRI upload, automated tumor prediction, and heatmap visualization.

Figure 5: Brain Tumor Vision AI - Input page.

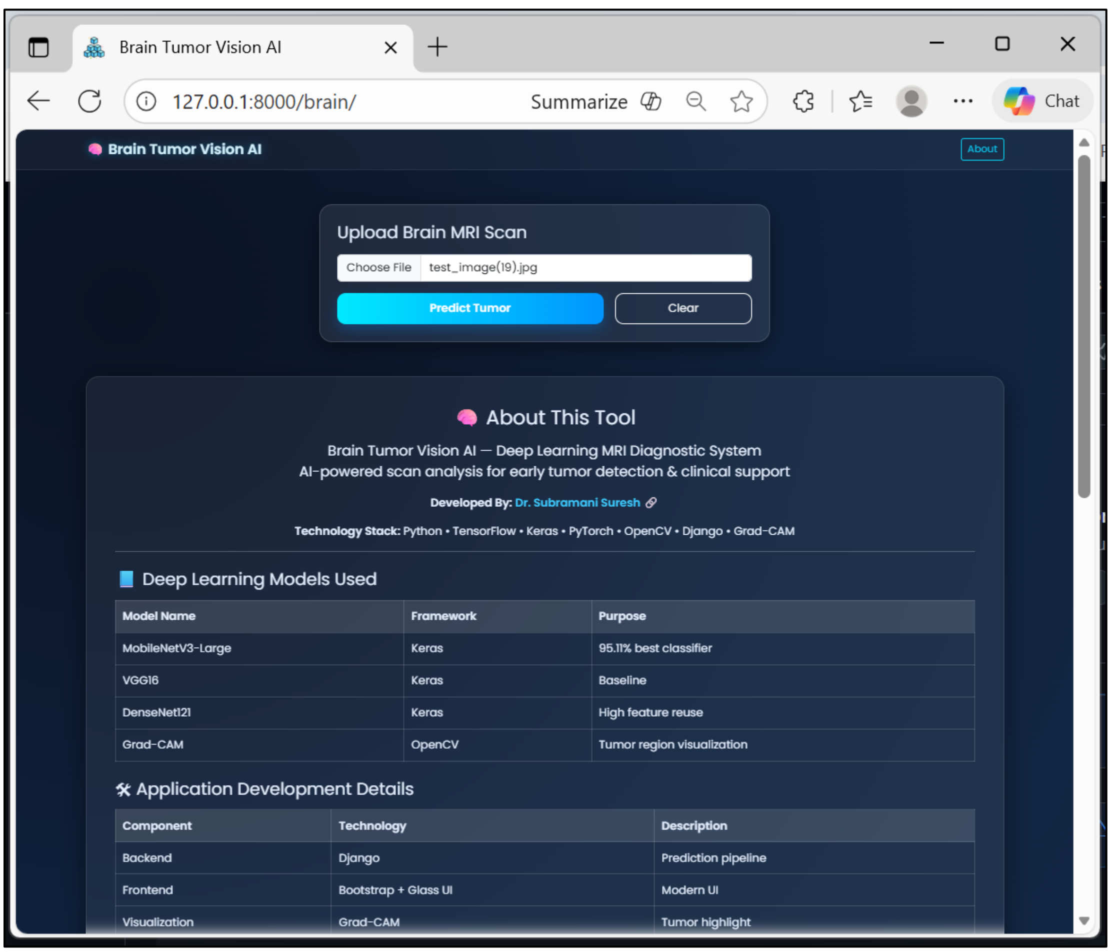
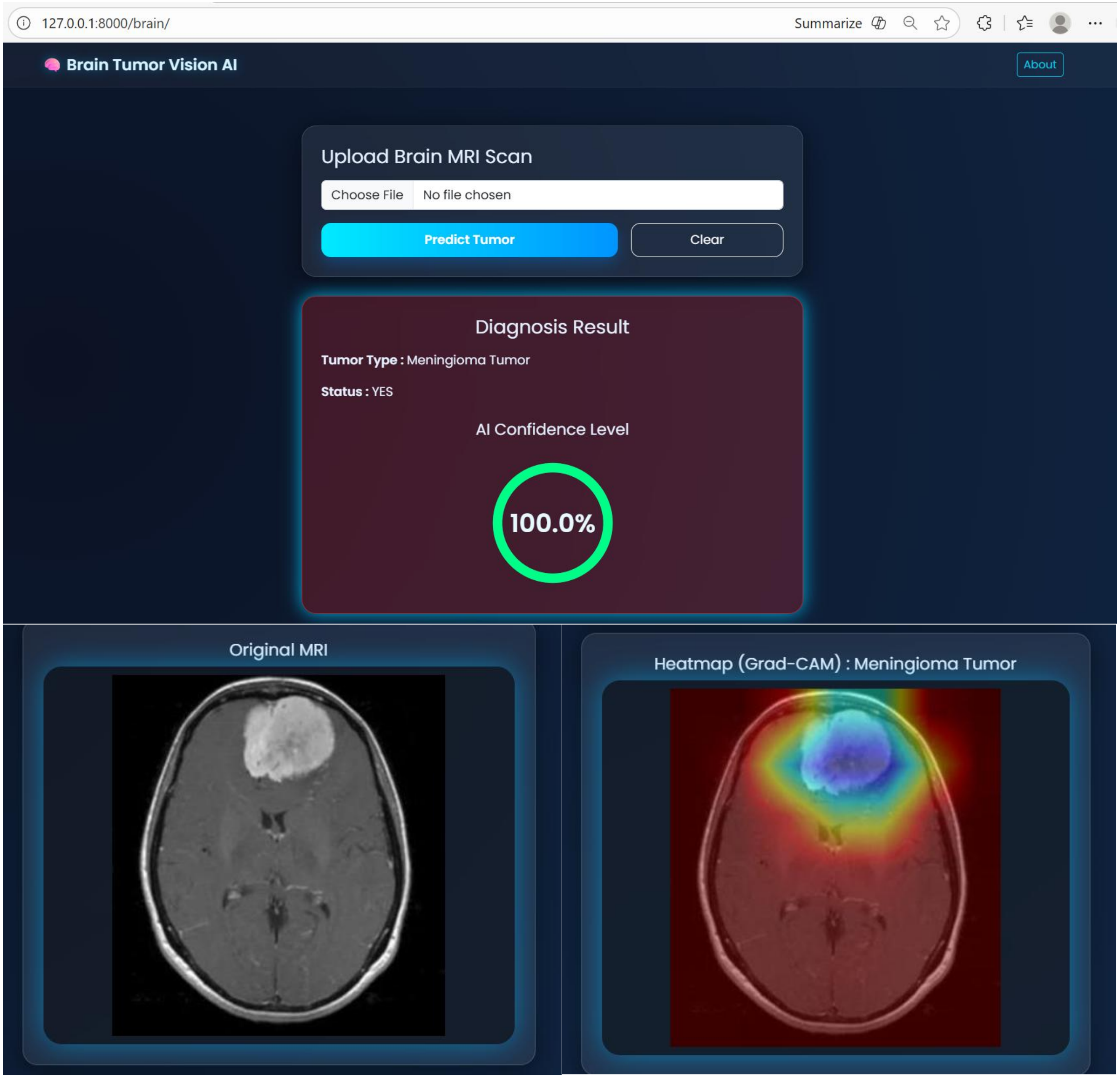


Figure 6: Brain Tumor Vision AI - Result page



6. Conclusion

This project successfully developed **Brain Tumor Vision AI**, an end-to-end deep-learning system capable of accurately detecting and classifying brain tumors from MRI scans. Through systematic dataset cleaning, balancing, and model evaluation, MobileNetV3-Large emerged as the most efficient and reliable architecture, delivering a final validation accuracy of **95.11%** with excellent generalization. The integration of Grad-CAM heatmaps further enhanced clinical interpretability by visually indicating tumor-affected regions. The final model was deployed as a fully functional Django web application, enabling real-time MRI upload, prediction, and explainable visualization. Overall, Brain Tumor Vision AI demonstrates strong potential as a fast, accurate, and transparent AI-based diagnostic support tool suitable for research and prototype clinical environments.