

BRAIN TUMOR VISION AI

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

AI

DR. SUBRAMANI



Mentored By:
Ramisha Rani K
Ramya dinesh



Problem Statement

Brain tumor detection from MRI scans is slow and requires expert radiologists. Manual diagnosis may cause delays, variation, and human error.

Common dataset issues in MRI classification:

- Class imbalance
- Incorrect train–test splits
- Mixed data distributions
- Unclean/noisy validation images

👉 **How to automatically detect and classify types of brain tumors using AI & deep learning with high accuracy, reliability, and clinical usefulness?**

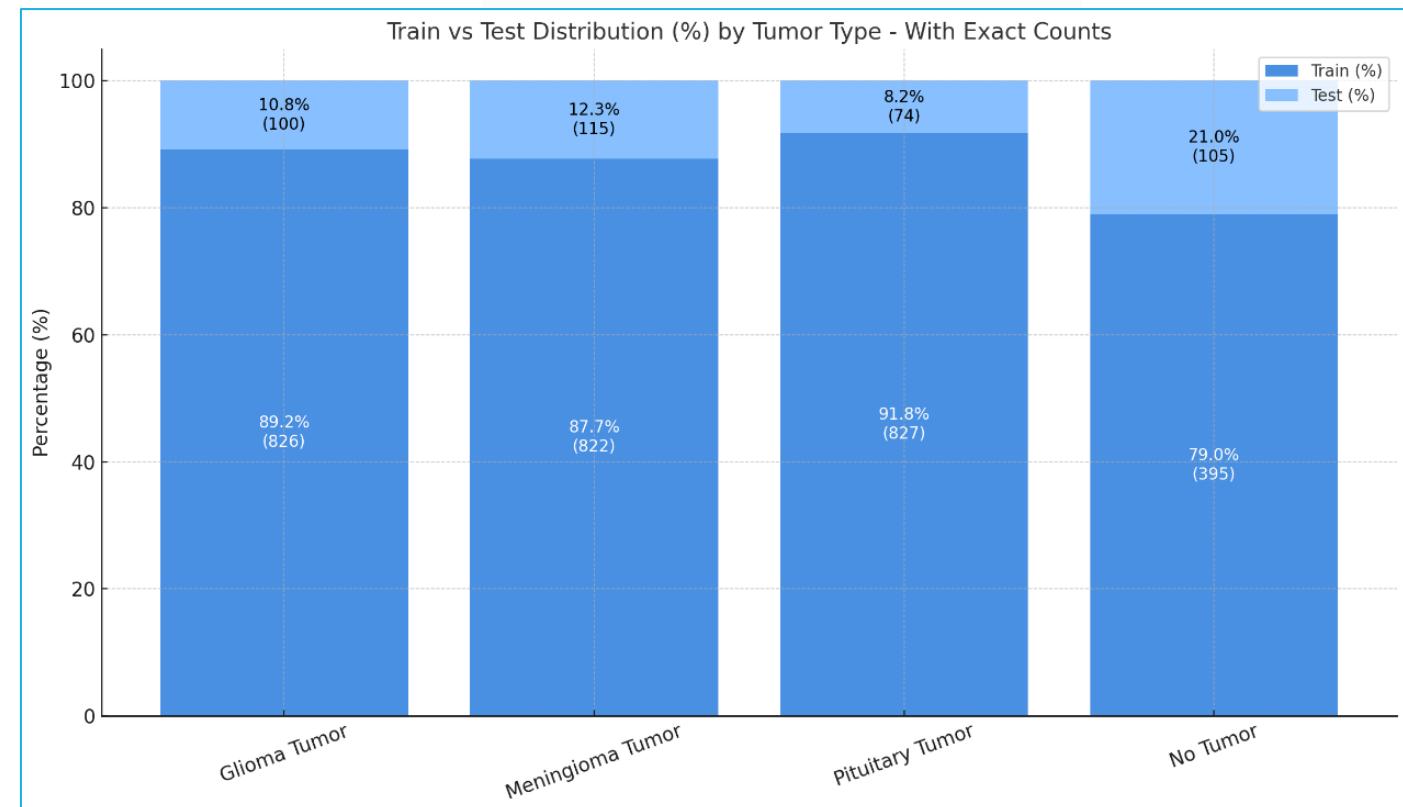
Dataset Preparation

Kaggle Dataset: A publicly available Brain MRI Tumor Dataset was downloaded from Kaggle:
<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

Brain Tumor MRI Dataset Summary (Kaggle)

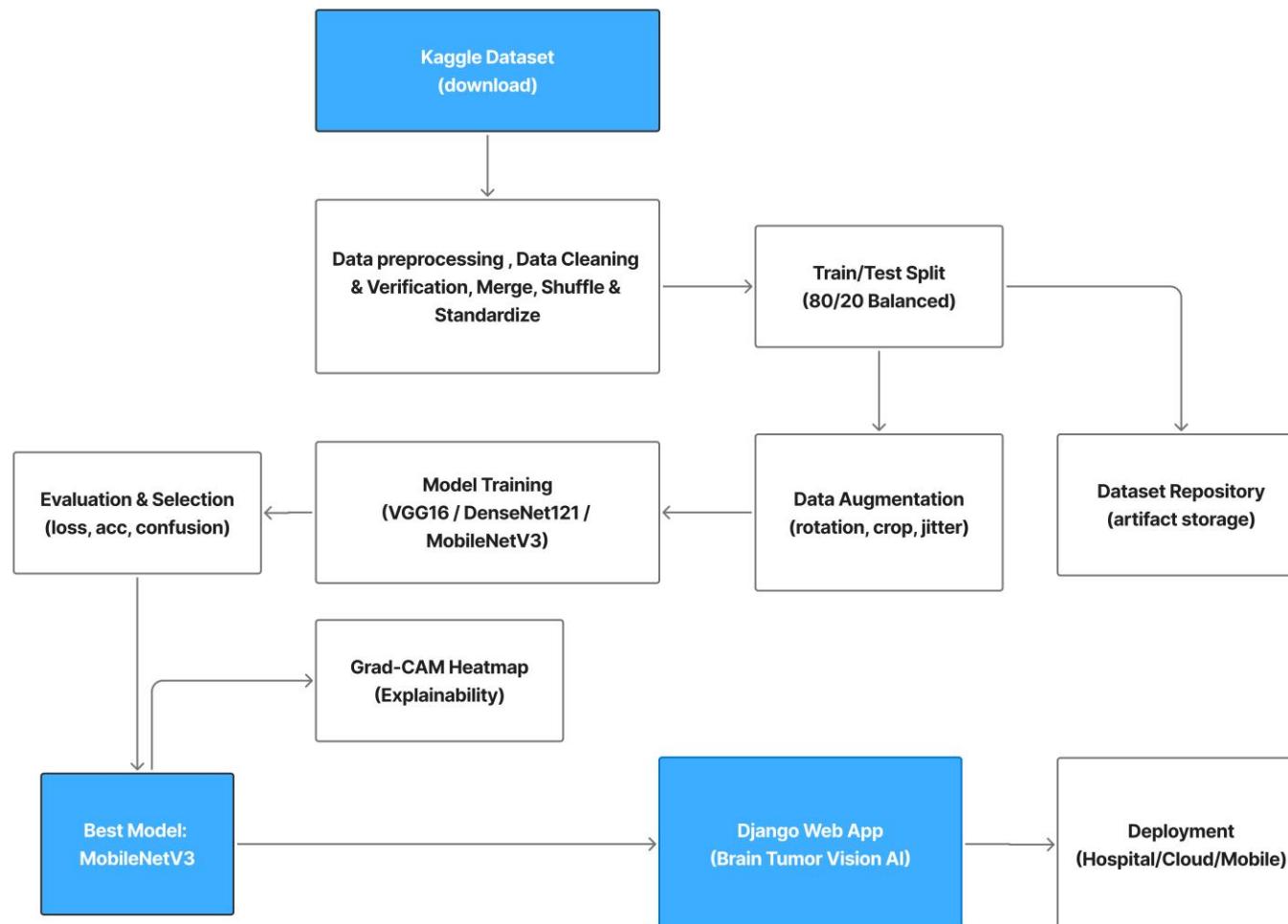
Tumor Type	Train Images	Test Images	Total Images
Glioma Tumor	826	100	926
Meningioma Tumor	822	115	937
Pituitary Tumor	827	74	901
No Tumor	395	105	500
Total	2870	394	3264

MRI Dataset Train–Test Distribution by Tumor Type



System Architecture and methodology workflow

Brain Tumor Vision AI



Data Preprocessing

The initial Kaggle MRI dataset was **highly imbalanced**

■ Before dataset imbalanced

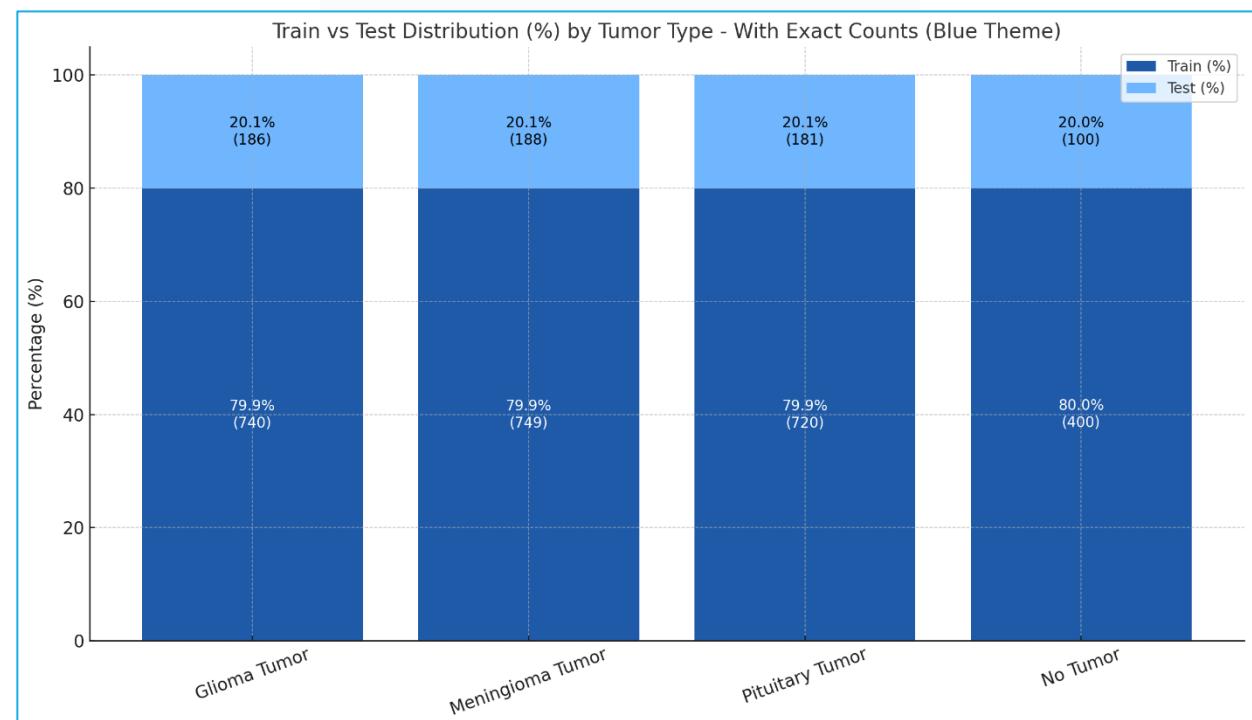
- Overfitting with unstable validation accuracy
- High training accuracy but low validation accuracy (46–78%)
- Validation images had a different distribution from training images
- Uneven or incorrect train–test splits
- Significant class imbalance across tumor categories

■ After dataset balanced

- 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

Balanced dataset

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



Model Selection & Training

Convolutional Neural Networks (CNNs) were selected for this project because they are the state-of-the-art architecture for medical image analysis.

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.

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**

Model Comparison

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, no overfitting, strong generalization
Grad-CAM	Balanced + Shuffled Dataset			Shows tumor region

Model Comparison

MobileNetV3 Performance: Both losses and a consistent rise in accuracy, indicating stable learning, minimal overfitting, and strong generalization.

**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

Training & Validation Loss and Accuracy Curves for MobileNetV3-Large



Conclusion & Deployment

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 with **95.11% accuracy** using MobileNetV3-Large. Dataset balancing and preprocessing were key to high performance. Grad-CAM enabled clinical interpretability

Deployment:

- ✓ Django Web Application
- ✓ MRI Upload → Preprocess → Predict → Heatmap
- ✓ Fast, accurate, explainable diagnostic support tool

Final Output Example:

- **Tumor / No-Tumor detection**
- **Predicted Class** (Glioma, Meningioma, Pituitary, No Tumor)
- **Ai Confidence Score**
- **Grad-CAM heatmap:** highlight suspected tumor regions

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

Brain Tumor Vision AI - Result

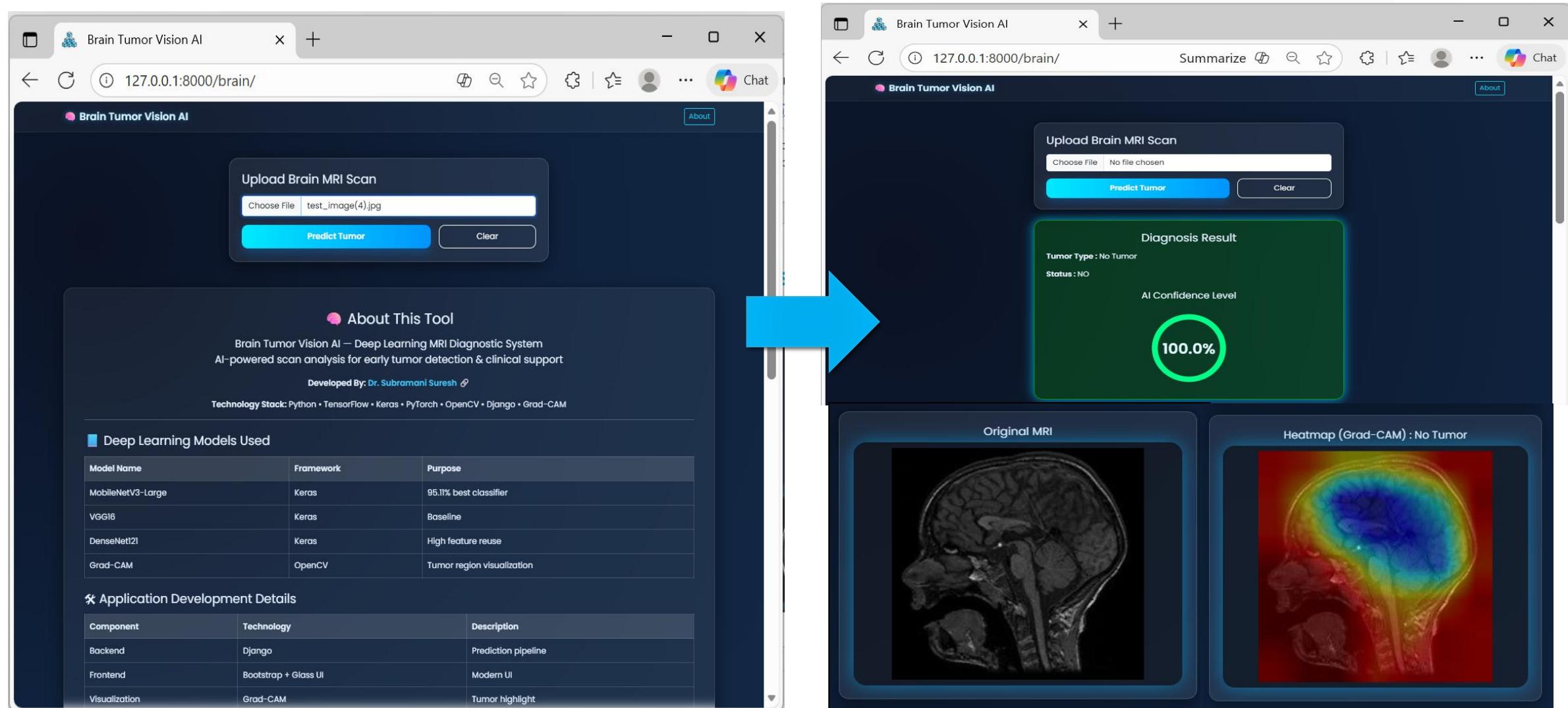
The screenshot shows the initial state of the application. At the top, there's a header bar with the title "Brain Tumor Vision AI". Below it is a navigation bar with links for "Summarize", "Chat", and "About". The main area features a "Upload Brain MRI Scan" form with a file input field containing "test_image(19).jpg", a blue "Predict Tumor" button, and a "Clear" button. To the right of this is a large blue arrow pointing to the right. Below the form is a section titled "About This Tool" which includes a brief description of the tool, developer information (Dr. Subramani Suresh), and technology stack (Python, TensorFlow, Keras, PyTorch, OpenCV, Django, Grad-CAM). There's also a table titled "Deep Learning Models Used" comparing five models based on Model Name, Framework, and Purpose. The last section, "Application Development Details", provides a breakdown of the components and their technologies.

Model Name	Framework	Purpose
MobileNetV3-Large	Keras	95.11% best classifier
VGG16	Keras	Baseline
DenseNet[2]	Keras	High feature reuse
Grad-CAM	OpenCV	Tumor region visualization

Component	Technology	Description
Backend	Django	Prediction pipeline
Frontend	Bootstrap + Glass UI	Modern UI
Visualization	Grad-CAM	Tumor highlight

The screenshot shows the result page after the prediction was made. The top part has a "Summarize" button and an "About" link. The main area starts with an "Upload Brain MRI Scan" form where no file is chosen. Below it is a "Diagnosis Result" section indicating a "Meningioma Tumor" with a "Status : YES". A large green circle displays an "AI Confidence Level" of "100.0%". At the bottom, there are two side-by-side brain MRI scans: the left one is labeled "Original MRI" and the right one is labeled "Heatmap (Grad-CAM) : Meningioma Tumor", showing a color-coded heatmap overlay on the MRI image.

Brain Tumor Vision AI - Result





Thank You

