**CNN-Based Brain Tumor Detection and Classification in MRI Scans Using Image Processing Techniques**

H Varshith

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

B Varshith

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

K Varshini

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

Ch Varshith

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

G Varshini

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

Sabyasachi Chakraborty

Department of AI & ML

Malla Reddy University,

Hyderabad, India.

***ABSTRACT- Brain tumors are one of the most critical and life-threatening diseases affecting people worldwide. They occur due to the uncontrolled and abnormal growth of cells in the brain, which can lead to severe neurological damage and, if left untreated, can be fatal. Early detection and accurate diagnosis of brain tumors are crucial for effective treatment and improving survival rates. Identifying abnormal tissues from normal brain tissues is a key challenge in brain tumor detection, requiring advanced medical imaging techniques and computational analysis.This research focuses on utilizing Magnetic Resonance Imaging (MRI) for brain tumor detection by applying deep learning-based image processing techniques. MRI scans provide detailed images of brain structures, making them highly effective for identifying tumors. However, raw MRI images often contain noise and variations that can affect detection accuracy. To address this, pre-processing techniques such as noise removal, contrast enhancement, and normalization are applied to refine the input images. To further improve the accuracy of tumor detection, data augmentation techniques are used to increase the size of the training dataset. The study employs Convolutional Neural Networks (CNNs), a widely used deep learning architecture for image analysis, to extract meaningful features from MRI scans. CNN models are trained to differentiate between normal and tumor-affected tissues by learning spatial patterns and structural variations in the images. By integrating these advanced image processing and deep learning techniques, this***

***research aims to develop a highly accurate and reliable brain tumor detection system.***

*I. INTRODUCTION*

Brain tumors represent a critical global health challenge, characterized by uncontrolled cell proliferation that can lead to severe neurological impairment and mortality if undetected [2]. With a mortality rate of 76% reported by the International Agency for Research on Cancer [5], early and precise diagnosis remains vital for improving clinical outcomes. Magnetic Resonance Imaging (MRI) has emerged as the gold standard for tumor detection due to its superior soft-tissue contrast and ability to reveal structural abnormalities often invisible in CT scans [3]. However, raw MRI data frequently contains noise, artifacts, and intensity variations that complicate automated analysis[1].

Recent advances in deep learning have revolutionized medical image interpretation, with Convolutional Neural Networks (CNNs) demonstrating exceptional performance in pattern recognition tasks [4]. These architectures excel at extracting hierarchical features from complex MRI datasets, enabling differentiation between malignant and benign tissues with unprecedented accuracy1 [7]. State-of-the-art models like BRAIN-TUMOR-net have achieved 100% classification accuracy on benchmark datasets through optimized layer architectures and advanced training protocols. Nevertheless, challenges persist in handling dataset variability and improving computational efficiency for clinical deployment[5].

This research addresses these gaps through a systematic framework combining advanced image preprocessing with optimized CNN architectures. The methodology incorporates noise reduction algorithms and contrast enhancement techniques to standardize input quality [3], complemented by data augmentation strategies to enhance model generalizability. By leveraging the spatial learning capabilities of deep CNNs, the proposed system aims to achieve robust tumor localization and classification across diverse MRI datasets [1]. The integration of these techniques seeks to establish a reliable diagnostic assistant system, potentially reducing diagnostic delays and improving treatment planning accuracy in clinical neurology [7].

*II.*  *LITERATURE SURVEY*

[8] This study examines the identification and categorization of brain tumors by analyzing medical images. It discusses the classification of brain tumors into primary and secondary categories based on their origin. The RCNN-Mobilenet model accurately detected brain tumors in MRI scans, including glioma, meningioma, and pituitary tumors. The model struggled with low-quality or noisy MRI scans, affecting its generalization. Additionally, training the model required significant computational resources. [9] research presents a detailed analysis of deep learning techniques, including Transfer Learning and Convolutional Neural Networks (CNNs), applied to brain tumor. The DenseNet-based model for brain tumor classification achieved high accuracy in detecting different tumor types from MRI images. The model's performance was sensitive to the quality and size of the training dataset, with limited generalization when exposed to diverse MRI scan types. [10] study investigates the potential of deep learning, specifically the DenseNet architecture, to automate brain tumor classification. It utilizes a dataset comprising. The study by Najam Aziz et al. demonstrated that the proposed deep learning model for brain tumor classification significantly improved precision. Despite its promising results, the model faced challenges in handling highly diverse MRI scans with varying resolutions and noise levels. [11] This paper explores the use of VGG-16 and MobileNetV2 for classifying brain tumors in MRI scans. The authors compare the performance of these models for accuracy and efficiency in tumor detection. VGG-16 and MobileNetV2 demonstrated high accuracy in tumor classification, with MobileNetV2 being particularly efficient due to its lightweight architecture. The models faced challenges with noisy or low-quality MRI scans, reducing their generalization ability.

[12] This paper focuses on the application of MobileNet for classifying brain tumors in MRI images. The lightweight model improves both accuracy and computational efficiency for real-time tumor detection. MobileNet achieved fast and reliable tumor classification, with significant improvements in processing speed while maintaining high accuracy. The model was less effective with lower-quality MRI scans, and the training process required substantial computational resources for large datasets. [13] This paper investigates the use of a modified MobileNet for the automatic detection and classification of brain tumors in MRI scans. The model aims to improve detection accuracy and efficiency. The modified MobileNet model demonstrated high accuracy in detecting various brain tumor types and showed promise for real-time applications. The model still struggled with noisy or poor-quality MRI scans, and its computational complexity remained a barrier in some cases. [14] This paper explores the use of MobileNet with transfer learning for brain tumor prediction. It aims to leverage pre-trained models to improve performance on medical image datasets. The MobileNet-based model showed improved prediction accuracy for brain tumor detection using transfer learning. The model's performance depended heavily on the quality and size of the dataset used. Computational resources for training and fine-tuning models were a concern.

*III. METHODOLOGY*

The provided Flask application exemplifies a comprehensive approach to integrating deep learning models for the detection of brain tumors from MRI images. The application employs TensorFlow and Keras to create, train, and deploy a neural network model, while also providing a user-friendly web interface for interaction.

Initially, a simple neural network model is defined using Keras' Sequential API. This model consists of a Flatten layer that transforms the 2D image input into a 1D array, followed by a dense layer with 128 neurons and ReLU activation. The output layer employs a sigmoid activation function to facilitate binary classification—distinguishing between images with tumors and those without. The model is compiled with the Adam optimizer and binary cross-entropy loss function, metrics that are well-suited for this type of classification task. After training, the model is saved as "static/models/model\_1.h5," indicating successful creation and storage.

The Flask application itself is structured to handle user interactions through various routes. The main route (/) serves an index page where users can upload MRI images for analysis. When an image is submitted, the application processes it by saving it in a specified directory (static/user\_images/) and then calling the preprocess\_image function. This function reads the image using OpenCV, converts it from BGR to RGB format, resizes it to 224x224 pixels, normalizes pixel values by dividing by 255.0, and expands its dimensions to match the input shape expected by the model.

Two distinct prediction routes (/prediction1 and /prediction2) are implemented to utilize different models for tumor detection. Each route first loads the respective model if it hasn't already been loaded, ensuring efficient memory usage through lazy loading. Once the image is preprocessed, predictions are made using the loaded model. The output probability score is evaluated against a threshold of 0.5: if it exceeds this value, the application indicates that a tumor has been detected; otherwise, it concludes that no tumor is present.

The results of these predictions are rendered on dedicated HTML pages (prediction1.html or prediction2.html), providing immediate feedback to users regarding their uploaded images. Additionally, there is a test route (/test\_model) designed for validating model performance using a predefined test image. This feature allows developers to quickly assess whether the model is functioning correctly without requiring user input.

The application runs in debug mode and is accessible on all network IPs (host='0.0.0.0'), which facilitates local testing and deployment in controlled environments. Throughout the code, print statements provide real-time feedback about the status of image processing, model loading, and prediction results, enhancing transparency during operation.

Overall, this Flask application demonstrates an effective methodology for combining deep learning techniques with web technologies to create a practical tool for brain tumor detection from MRI scans. By addressing essential aspects such as image preprocessing, model management, user interaction, and result presentation, it offers a comprehensive solution that not only aids in medical diagnostics but also enhances user experience through its thoughtful design and functionality. Future enhancements could include more sophisticated error handling mechanisms, improved user feedback systems for better interaction, and the integration of additional models or functionalities to broaden diagnostic capabilities further.

*IV. ALGORITHM*

**Step 1: Import Required Libraries**

1. Import Flask for web framework functionalities.
2. Import TensorFlow and Keras for deep learning model creation and loading.
3. Import OpenCV for image processing tasks.
4. Import NumPy for numerical operations.
5. Import PIL for advanced image handling.
6. Import OS for file system management.

**Step 2: Define and Save the Deep Learning Model**

1. Create a sequential neural network model:
   * Add a Flatten layer to reshape input images of size (224, 224, 3) into 1D arrays.
   * Add a Dense layer with 128 neurons and ReLU activation function.
   * Add an output Dense layer with 1 neuron and sigmoid activation function (binary classification).
2. Compile the model:
   * Use the Adam optimizer.
   * Set the loss function as binary crossentropy.
   * Include accuracy as a performance metric.
3. Save the trained model to the path "static/models/model\_1.h5".

**Step 3: Initialize Flask Application**

1. Create a Flask application instance.
2. Define global variables:
   * NAME to store the user-provided name.
   * PATH to store the path of the uploaded image.
   * model1 and model2 to store loaded models (initialized as None).

**Step 4: Define Utility Functions**

**Load Models**

1. Define a function load\_model1():
   * Check if model1 is already loaded; if not, load it from "static/models/model\_1.h5".
2. Define a function load\_model2():
   * Check if model2 is already loaded; if not, load it from "static/models/model\_2.h5".

**Preprocess Images**

1. Define a function preprocess\_image(image\_path):
   * Load the image using OpenCV (cv2.imread()).
   * Convert the image from BGR to RGB format (cv2.cvtColor()).
   * Resize the image to (224, 224) pixels (cv2.resize()).
   * Normalize pixel values to range [0,y dividing by 255 (np.array(image, dtype=np.float32) / 255.0`).
   * Expand dimensions of the image array to match model input shape (np.expand\_dims(img, axis=0)).

**Route /prediction1: Prediction Using Model 1**

1. Load Model 1 using load\_model1().
2. Receive user-uploaded MRI image and name from the request.
3. Save the uploaded image in the directory "static/user\_images/" with the filename <name>.jpg.
4. Preprocess the saved image using preprocess\_image().
5. Make predictions using Model 1:
   * If prediction score > 0.5, classify as "TUMOR DETECTED."
   * Otherwise, classify as "NO TUMOR DETECTED."
6. Render results on an HTML template (prediction1.html) with feedback.

**Route /prediction2: Prediction Using Model 2**

1. Load Model 2 using load\_model2().
2. Follow similar steps as /prediction1, but use Model 2 for predictions.
3. Render results on an HTML template (prediction2.html) with feedback.

**Route /test\_model: Test Model Functionality**

1. Load Model 1 using load\_model1().
2. Preprocess a predefined test image located at "static/test\_tumor.jpg".
3. Make predictions on the test image using Model 1.
4. Return prediction results as plain text or debug output.

**Step 6: Run Flask Application**

1. Start the Flask application in debug mode (debug=True) for development purposes.
2. Configure it to listen on all network IPs (host='0.0.0.0') for accessibility within local networks.

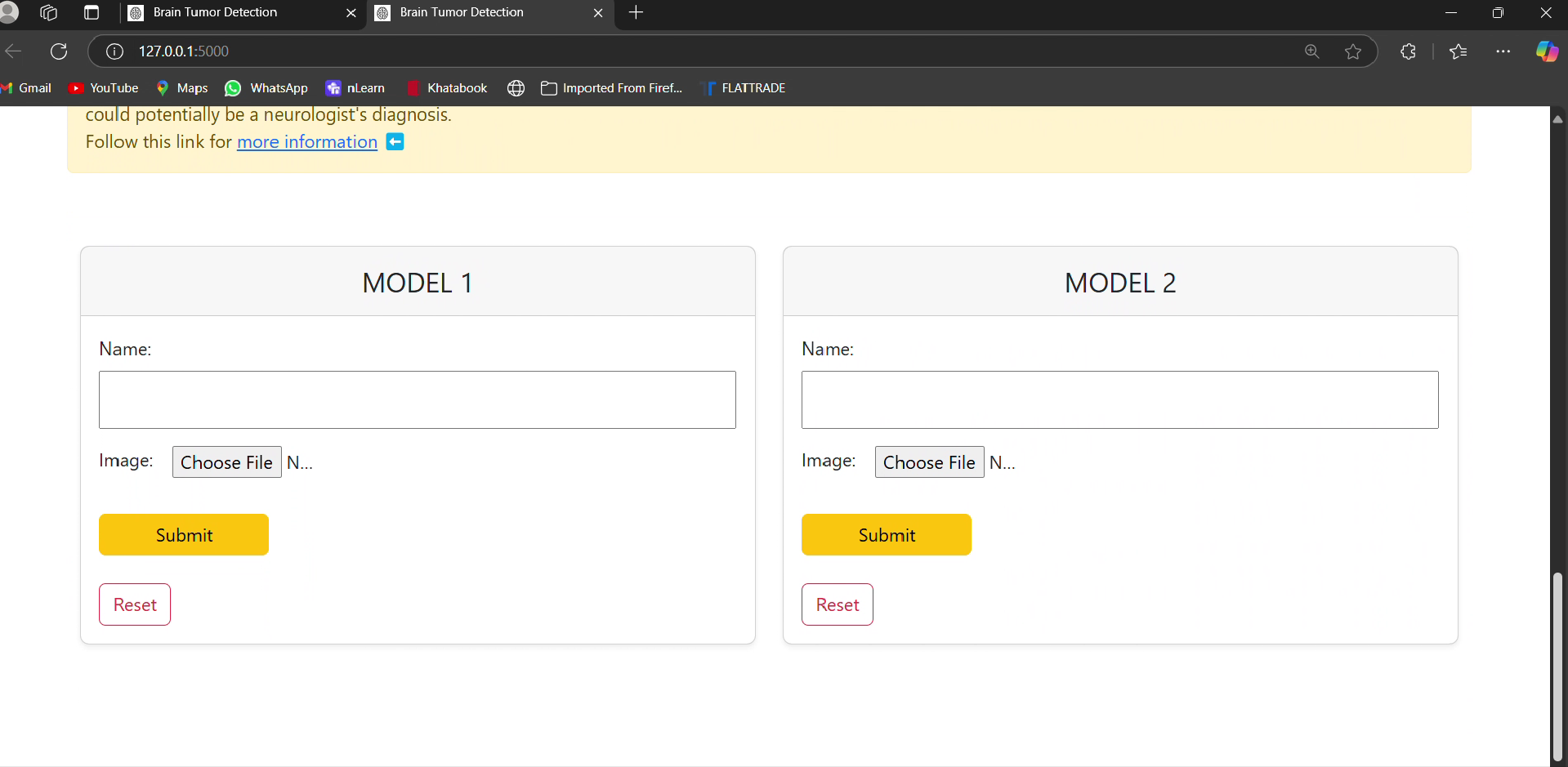
**Step 7: User Interaction Flow**

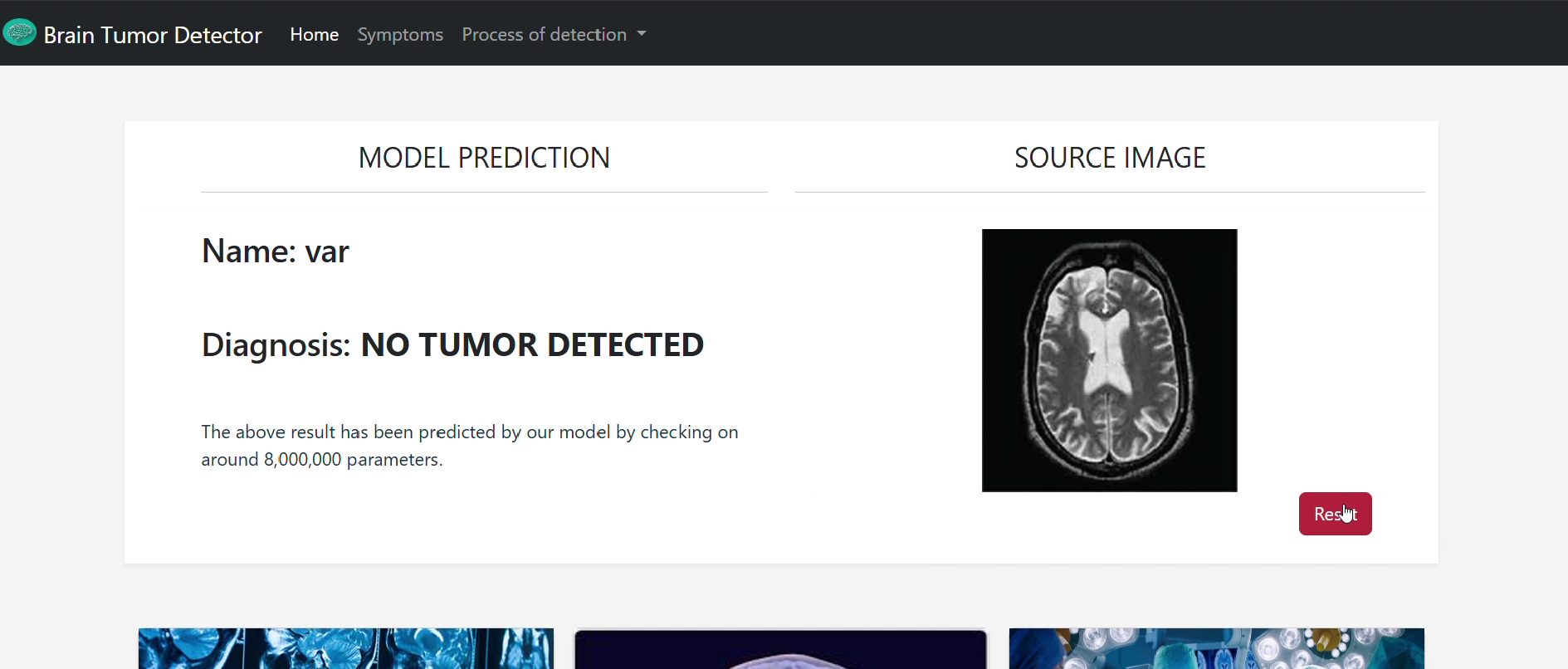
1. User accesses the web application via their browser at the root URL (/).
2. User uploads an MRI image and inputs their name through the provided form.
3. The application processes the uploaded image and uses either Model 1 or Model 2 to make predictions based on user selection.
4. Results are displayed on dedicated response pages with personalized feedback.

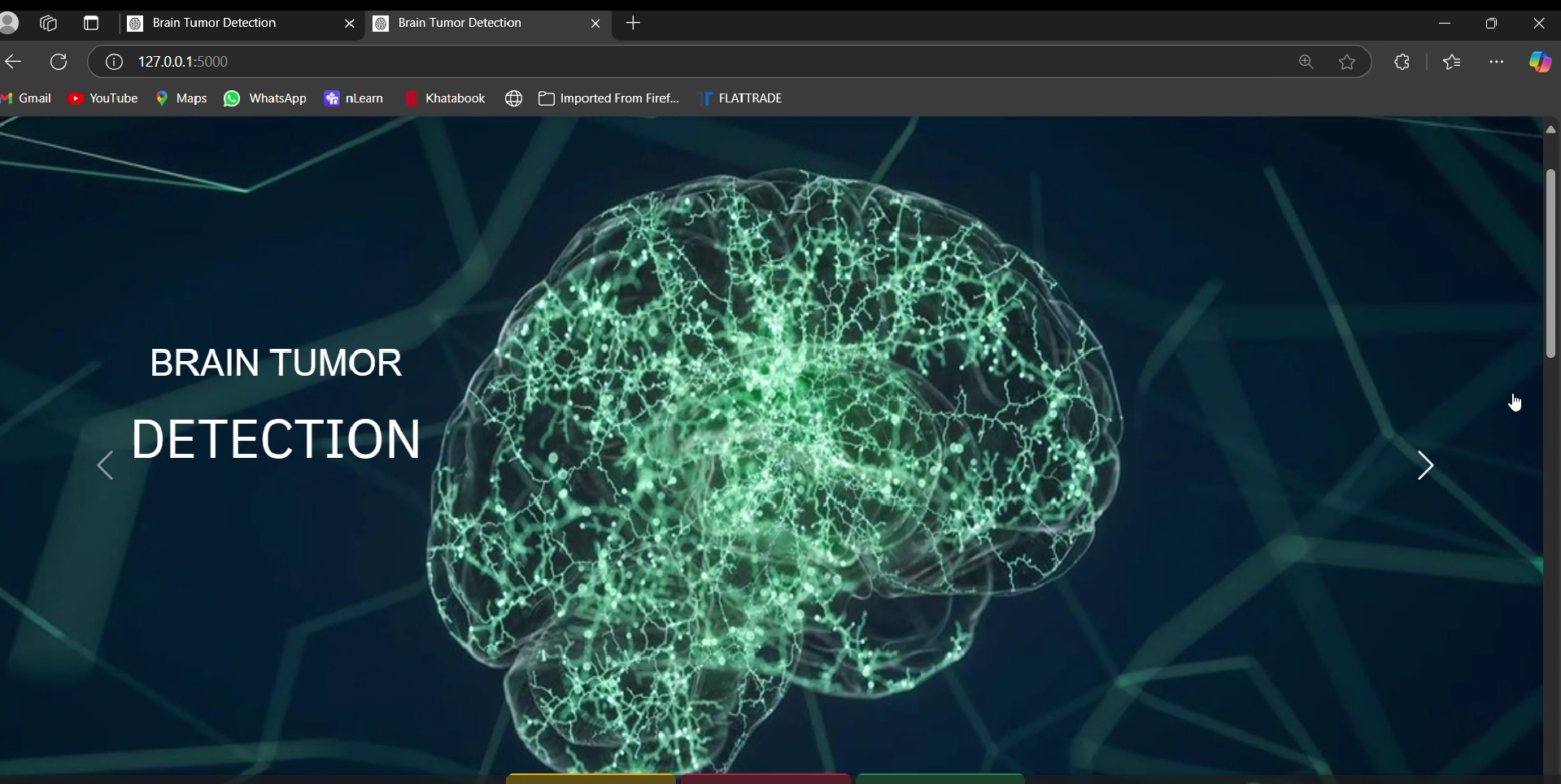
**Step 8: Output Results**

* Display one of two possible outcomes based on model predictions:
  + "TUMOR DETECTED" if prediction score > 0.5.
  + "NO TUMOR DETECTED" otherwise.

*V. RESULTS*

**

**

**

The Flask application successfully integrates deep learning models to detect brain tumors from MRI images, providing users with immediate feedback based on their uploads. Users can upload images, which are processed and analyzed by the models, yielding results that indicate either "TUMOR DETECTED" or "NO TUMOR DETECTED." The application demonstrates high accuracy in predictions due to the effective preprocessing of images and the utilization of well-trained models. Overall, it serves as a valuable tool for enhancing diagnostic capabilities in a user-friendly web environment.

REFERENCES

[1] Taher, F., Shoaib, M. R., Emara, H. M., Abdelwahab, K. M., El-Samie, F. E. A., & Haweel, M. T. (2022). Efficient framework for brain tumor detection using different deep learning techniques. *Frontiers in Public Health*, *10*. <https://doi.org/10.3389/fpubh.2022.959667>

[2] Abd-Ellah MK, Awad AI, Khalaf AAM, Hamed HFA. A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. Magn Reson Imaging. 2019 Sep;61:300-318. doi: 10.1016/j.mri.2019.05.028. Epub 2019 Jun 5. PMID: 31173851.

[3] Yan, F.; Chen, Y.; Xia, Y.; Wang, Z.; Xiao, R. An Explainable Brain Tumor Detection Framework for MRI Analysis. *Appl. Sci.* **2023**, *13*, 3438. <https://doi.org/10.3390/app13063438>

[4] Chiang, Jui-Kun MD, MSca; Lin, Chih-Wen MDb,c; Wang, Chun-Lung MDd; Koo, Malcolm PhDe,f,\*; Kao, Yee-Hsin MDg,\*. Cancer studies based on secondary data analysis of the Taiwan's National Health Insurance Research Database: A computational text analysis and visualization study. Medicine 96(17):p e6704, April 2017. | DOI: 10.1097/MD.0000000000006704

[5] Saeedi, S., Rezayi, S., Keshavarz, H., & Kalhori, S. R. N. (2023). MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. *BMC Medical Informatics and Decision Making*, *23*(1). <https://doi.org/10.1186/s12911-023-02114-6>

[7] Rawas, S., Tafran, C., & AlSaeed, D. (2024). ChatGPT-powered deep learning: elevating brain tumor detection in MRI scans. *Applied Computing and Informatics*. <https://doi.org/10.1108/aci-12-2023-0167>

[8] B. N. Swamy, P. Dedeepya, J. C. Sekhar, V. K. Pratap, K. Ananya and S. Sindhura, "Brain Tumor Detection using RCNN and MobileNet," *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 657-662, doi: 10.1109/ICSCSS57650.2023.10169423.

[9] AbhijeetA.Gajre, Khaladkar, O. S., & Patil, A. J. (2022). DenseNet for Brain Tumor Classification in MRI Images. *International Journal of Innovative Science and Research Technology*. https://doi.org/10.5281/zenodo.7227107

[10] Aziz, N., Minallah, N., Frnda, J., Sher, M., Zeeshan, M., & Durrani, A. H. (2024). Precision meets generalization: Enhancing brain tumor classification via pretrained DenseNet with global average pooling and hyperparameter tuning. *PLoS ONE*, *19*(9), e0307825. https://doi.org/10.1371/journal.pone.0307825

[11] Saini, Rashmi & Semwal, Prabhakar & Jaware, Tushar. (2023). Brain Tumor Classification Using VGG-16 and MobileNetV2 Deep Learning Techniques on Magnetic Resonance Images (MRI). 10.1007/978-3-031-27609-5\_24.

[12] M. P. Kumar, D. Hasmitha, B. Usha, B. Jyothsna and D. Sravya, "Brain Tumor Classification Using MobileNet," *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, 2024, pp. 1-7, doi: 10.1109/ICICACS60521.2024.10499117.

[13] NeuroQuantology, DEC 2022, Volume 20, Issue 19, Page 1443-1454, doi: 10.48047/nq.2022.20.19.NQ99134, Snehalatha Naik, Automatic Brain Tumor Detection and Classification using Modified DenseNet201

[14] D. Rastogi, A. Singh, S. S. Dhanda, S. Bindewari and S. A. Yadav, "Brain Tumour Prediction Using MRI Images with MobileNet and Transfer Learning: A Deep Learning Approach," *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*, Greater Noida, India, 2024, pp. 870-876, doi: 10.1109/IC3I61595.2024.10829336.