Brain Tumor Detection

LAHROURI Nadia
Ibn Zohr University-Faculty of Science
Center of Excellence IT – Data Analytics and Artificial Intelligence
Agadir,Morocco
nadialahrouri12@gmail.com

SAKOUTY Maryam
Ibn Zohr University-Faculty of Science
Center of Excellence IT – Data Analytics and Artificial Intelligence
Agadir,Morocco
marysakouti@gmail.com

supervised by: Dr.MAAROUF Otman
Ibn Zohr University-Faculty of Science
Center of Excellence IT
Agadir,Morocco

Brain tumor detection is a critical task in medical imaging that can significantly aid early diagnosis and treatment planning. In this project, we used a sizable collection of brain tumor images from Kaggle to tackle the challenging task of brain tumor detection in MRI scans. Classification and segmentation are the two primary phases of the solution. First, MRI brain images are classified using MobileNet to determine whether a tumor is present or not. Second, U-Net, based on MobileNetV2, is applied to segment the tumor region from the detected abnormal scans. The preprocessing pipeline includes resizing, normalization, and data augmentation. Evaluation metrics such as accuracy. The results demonstrate that lightweight models can effectively handle both classification and segmentation tasks with high accuracy and low computational cost, making them suitable for real-world medical applications.

Keywords: brain tumor detection, MRI, deep learning, maching learning, atificial intelligence, medical images, MobileNet, U_Net-MobileNetV2, convolution neural networks (CNN).

I. Introduction

The healthcare industry has been rapidly transformed by technological advances in recent years, and an important component of this transformation is artificial intelligence (AI) technology. AI is a computer system that simulates human-like intelligence and has many applications in medicine. One such area is the fight against brain tumors. Brain tumors are a major public health problem in the healthcare sector, and accurate diagnosis, treatment, and follow-up processes are critical. AI has become an important tool for improving these processes and has great potential for early diagnosis and treatment of brain tumors.

Brain tumors affect human health due to their location1. AI is designed to help diagnose and treat complex diseases such as brain tumors by combining technologies such as big data analytics, machine learning, and deep learning. AI has the ability to detect and classify tumors by analyzing brain imaging techniques, such as Magnetic Resonance Imaging (MRI). AI algorithms can help determine the size, location, class of tumors. This helps physicians make a more accurate diagnosis and treatment plan, and helps patients better understand their health.

In this context, artificial intelligence (AI), and more specifically deep learning, offers automated, accurate, and fast solutions to assist radiologists in the diagnostic process.

This project aims to develop an intelligent system for brain tumor detection and segmentation using MRI images. It is divided into two main tasks:

Image Classification : To detect the presence or absence of a brain tumor using a lightweight MobileNet model.

Tumor Segmentation : To accurately segment the tumor region using a U-Net architecture with a MobileNetV2 encoder.

The project was implemented using tools such as Python, TensorFlow/Keras, OpenCV, and Google Colab. For deployment, Flask was used to create a functional and accessible web interface, providing clear visual results and demonstrating the practical application of AI in medical imaging.

II. Materials and Methods

1. Overall Architecture of Brain Tumor Detection

Image analysis of brain tumors is challenging, since these tumors can vary widely in size, shape, and location. Researchers have proposed several different methods for detecting anomalies in data that cannot be directly observed, each with their own set of advantages and disadvantages. The availability of a benchmark dataset capable of assessing the efficacy of state-of-the-art procedures is vital for the objective assessment of the performance of these methods. Different devices can produce brain tumor images with varying degrees of sharpness, contrast, number of slices, and pixel spacing. Here, we describe the architecture and technological details of the proposed system that make it possible to detect brain tumors in photos quickly and accurately. Brain tumor picture preprocessing, enhancements, training. Several potential methods for detecting and describing brain tumors have been proposed, and these have been covered in prior research. Unfortunately, these methods have only been successfully implemented in a select few studies, with mixed outcomes at best. The suggested method's primary focus is on providing accurate brain tumor detection in MRI scans.

2. Data source

The dataset consists of a total of 3 929 MRI Images for Brain Tumor Detection dataset, publicly available on Kaggl. The dataset consists of T1-weighted contrast-enhanced MRI scans categorized into two classes : 'yes' (tumor present) and 'no' (tumor absent). Each image is in JPEG format and varies in resolution.

For the segmentation task, an additional custom annotation or a dataset with labeled masks was used, aligned with the classification dataset, enabling supervised training of a U-Net-based model.

3. Deep learning

Deep learning is a subset of machine learning that focuses on training artificial neural networks to perform complex tasks by learning patterns and representations directly from data. Unlike traditional machine learning approaches that require manual feature engineering, deep learning algorithms autonomously extract hierarchical features from data, leading to the creation of powerful and highly accurate models. In this study, a CNN architecture is employed.

4. Machine learning

Machine learning (ML) is a branch of artificial intelligence (AI) focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data.

5. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network that excels at analyzing and learning visual features from data, particularly images. It's a feedforward neural network that uses convolutional layers to filter inputs for useful information, effectively learning features via filter optimization. CNNs are widely used in computer vision tasks like image recognition, object detection, and classification.

6. U-Net

The U-Net model is a fully convolutional neural network architecture primarily used for image segmentation. It's characterized by its encoder-decoder structure, where the encoder (contracting path) downsamples the input image to extract features, and the decoder (expansive path) upsamples the feature maps to produce a segmented output.

7. Classification MobileNet

For binary classification (tumor vs. no tumor), we used the lightweight and efficient MobileNet architecture. The model was fine-tuned using transfer learning, with the top classification layers replaced by a dense output layer with sigmoid activation. The model was compiled using the binary cross-entropy loss function and optimized with the Adam optimizer.

8. Segmentation U_Net with MobileNetV2 Encoder

For pixel-wise segmentation of tumor regions, a modified U-Net architecture was employed. The encoder part was based on MobileNetV2, a pretrained model on ImageNet, while the decoder used upsampling and concatenation layers to reconstruct segmentation masks. The model output was a binary mask, trained with binary cross-entropy and Dice loss to handle class imbalance.

9. Dice Coefficient (Dice Similarity Coefficient - DSC)

The **Dice Coefficient** is a statistical metric used to measure the similarity between two sets, commonly used in image segmentation to evaluate the overlap between the predicted segmentation and the ground truth.

10. Intersection over Union (IoU or Jaccard Index)

The IoU Score measures the ratio between the intersection and the union of the predicted and ground truth masks.

11. Performance metric

Performance evaluation methods such as Accuracy, Precision, Recall, and F-score are used to evaluate models created for classification problems such as image processing. These methods are obtained from the confusion matrix.

12. Equations

TP, TN, FP, FN: correspond to the true negative, true positive, false positive, and false negative values.

Equation 1 : Accuracy

$$Accuracy = \frac{T_P + T_N}{T_P + F_P + F_N + T_N} \tag{1}$$

Equation 2: Precision

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Equation 3: Recall

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Equation 4: F1-Score

$$F\text{-}score = \frac{2 * Precision * Recall}{Precision + Recell} \tag{4}$$

Equation 5: 1. ReLU

$$f(x) = max\ (0, x) \tag{5}$$

Equation 6: Dice Coefficient (Dice Similarity Coefficient - DSC)

$$\text{Dice} = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{6}$$

Where:

1. A is the set of predicted pixels (segmentation mask)
2. B is the set of ground truth pixels.

Equation 7: 1. Intersection over Union (IoU or Jaccard Index)

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{7}$$

Where:

- A is the set of predicted pixels,
- B is the set of ground truth pixels.

III. EXPERIMENTAL SETTINGS AND RESULTS

1. Experimental Setup

This study focuses on the problem of image classification using deep learning methods, specifically in the context of medical imaging. Brain tumor classification is one of the most critical tasks in medical image analysis, and deep neural networks have demonstrated great promise in this domain. In this work, a pre-trained MobileNet model

was used and fine-tuned to classify brain MRI images into two categories: tumor and no tumor.
All experiments were carried out using Google Colab with GPU support. The dataset was divided into 75% for training, 15% for validation, and 10% for testing. Input images were resized to 224×224 pixels and normalized before training. Data augmentation was also applied to improve generalization and reduce overfitting.

2. MobileNet-Based CNN Architecture

The MobileNet model used in this study was initialized with ImageNet weights, and the top classification layers were removed. A custom classification head was added to adapt the model to the specific task of binary classification.

3. Model Architecture

First, we need to determine the architecture of our model. The input shape of our data is 224×224 pixels with 3 channels (RGB), and since we are dealing with a binary classification problem (tumor vs. no tumor), the number of output classes is set to 2. The MobileNet model is used as the backbone and initialized with weights pre-trained on the ImageNet dataset. The top layers of MobileNet are removed to allow the integration of a custom classification head tailored to the binary classification task. The output of the base model is passed through an AveragePooling2D layer with a pool size of 4×4, which reduces the spatial dimensions and preserves the learned features. This is followed by a Flatten layer to convert the feature maps into a 1D vector, then two Dense layers with 256 neurons each and ReLU activation functions. To prevent overfitting, Dropout layers with a rate of 0.3 are added after each Dense layer. Finally, a Dense output layer with 2 neurons and a softmax activation function is used to calculate class probabilities. The model is compiled using the Adam optimizer and the sparse categorical cross-entropy loss function, which is suitable for integer-labeled binary classification problems. The key metric monitored during training is accuracy. The training is conducted over 10 epochs with a batch size of 32, and the entire model is trained using mini-batch gradient descent with validation monitoring. All layers in the base MobileNet are frozen to retain the general visual features learned during pretraining, allowing the newly added layers to specialize on the brain tumor classification task.

4. MobileNetV2-Based U-Net Architecture

In this study, a U-Net architecture based on MobileNetV2 was implemented for the task of brain tumor segmentation. The model was designed to leverage the lightweight and efficient feature extraction capabilities of MobileNetV2, pretrained on ImageNet, while integrating a U-Net style decoder to generate precise segmentation masks. The input images have a resolution of 224×224 with three color channels. The encoder part of the model uses specific intermediate outputs from MobileNetV2—namely from the layers “block_1_expand_relu”, “block_3_expand_relu”, “block_6_expand_relu”, “block_13_expand_relu”, and “block_16_project”—which capture multiscale semantic and spatial information at different levels of abstraction. These layers are used as skip connections to bridge the encoder and decoder. All encoder weights are frozen to preserve pretrained features and accelerate convergence. The decoder is composed of five consecutive upsampling blocks using the UpSampling2D layer, followed by concatenation with the corresponding encoder features and two Conv2D layers with ReLU activation. The number of filters decreases progressively from 256 to 16 across the decoder path. The final output layer is a Conv2D with a single filter and a sigmoid activation function, producing a binary segmentation mask. The model is compiled using the Adam optimizer, with binary cross-entropy as the loss function and accuracy as the primary evaluation metric. This configuration provides an efficient yet powerful framework for accurate tumor localization with minimal computational overhead.

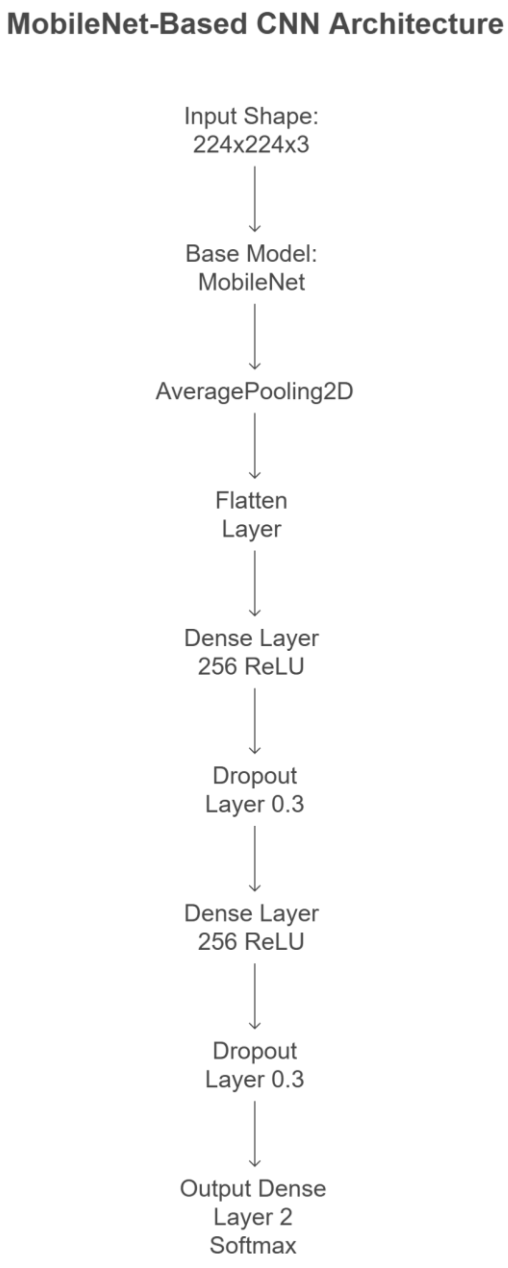


Figure 1 : MobileNet Based CNN Architecture

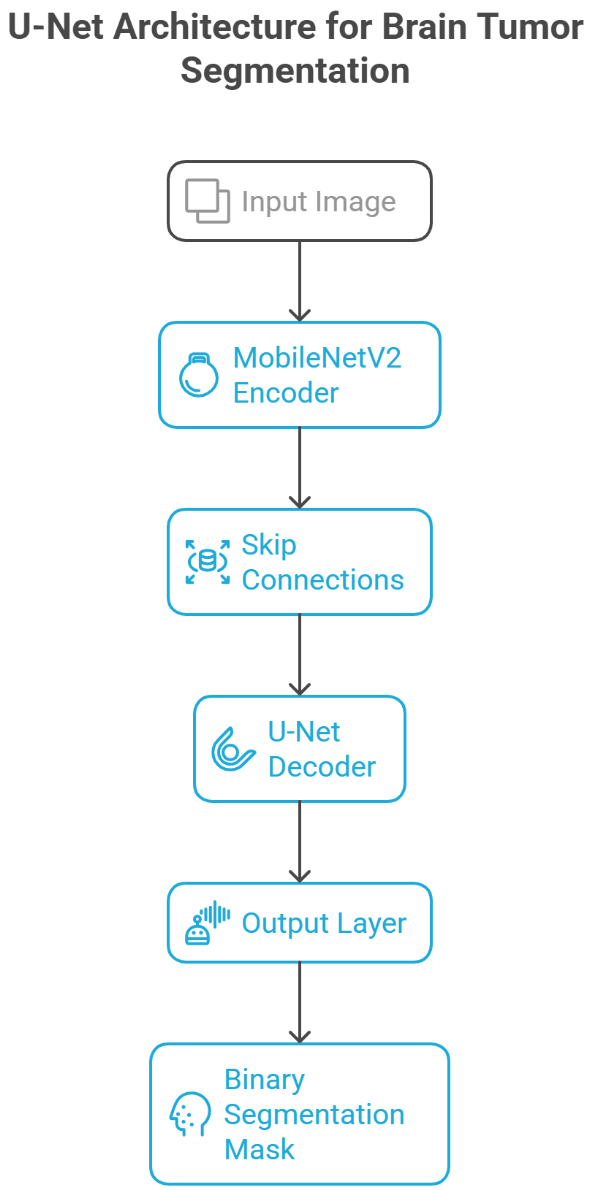


Figure 2: U-Net Architecture

5. Training Settings and Evaluation Metrics

The MobileNetV2-based U-Net model was trained to perform binary segmentation of brain tumors. The training was conducted using the Adam optimizer with a learning rate of 0.001, and the loss function used was binary cross-entropy, as the segmentation task was defined as a pixel-wise binary classification problem (tumor vs. non-tumor). The model was trained for 20 epochs with a batch size of 8, which provided a balance between training speed and convergence stability. During training, accuracy was monitored as the primary evaluation metric. Additionally, training and validation loss curves were tracked to assess the model's generalization capability.

To enhance training stability and avoid overfitting, several strategies were applied. First, the encoder (MobileNetV2) was frozen to preserve the pretrained weights and reduce the number of trainable parameters. Second, early stopping and learning rate scheduling techniques were integrated. The ReduceLROnPlateau callback was used to monitor the validation loss and automatically reduce the learning rate by a factor of 0.5 if no improvement was observed for 3 consecutive epochs, with a minimum learning rate threshold set to $1e{-6}$. This dynamic adjustment of the learning rate helped the model converge more smoothly and avoid getting stuck in local minima.

After training, the model was evaluated on the test set using accuracy and qualitative visual comparison between predicted and ground truth masks. The results demonstrated that the MobileNetV2-U-Net model effectively captured tumor regions with high precision while maintaining low computational complexity.

VI. Results and discussion

The confusion matrix of the study on the classification of tumor, non-tumor.

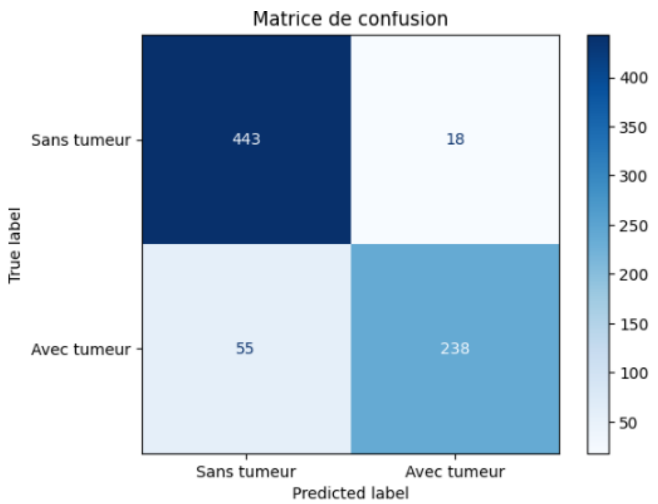


Figure 3: Confusion matrix

The training and validation accuracy loss graphs of the models created with U_Net MobileNet Encoder

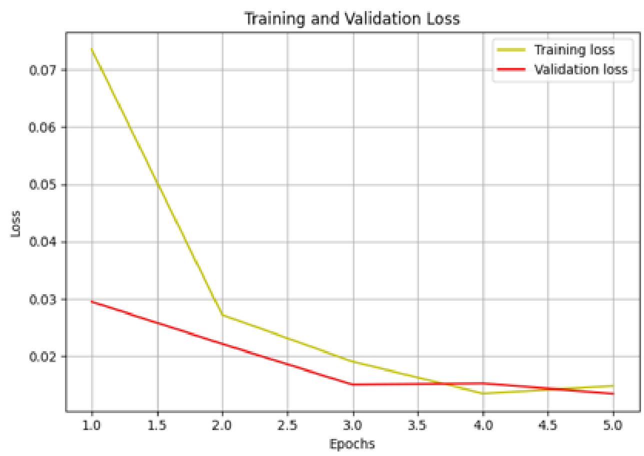


Figure 4: Training and validation Loss

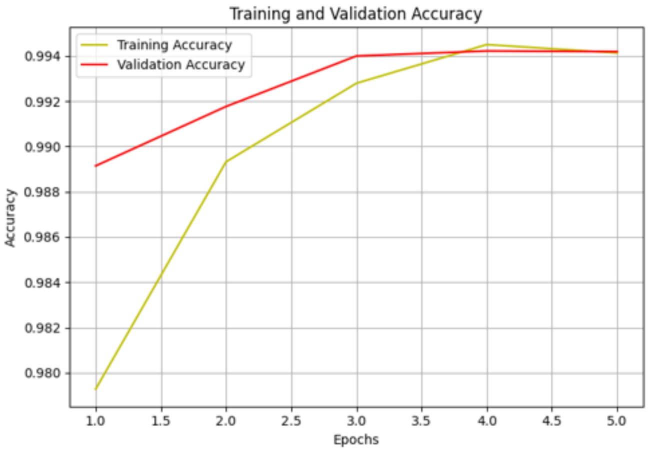


Figure 5: Training and validation Accuracy

The Dice score ranges from 0 to 1, where 1 indicates perfect overlap and 0 indicates no overlap. It is particularly useful for imbalanced classes, such as small tumor regions in medical images.

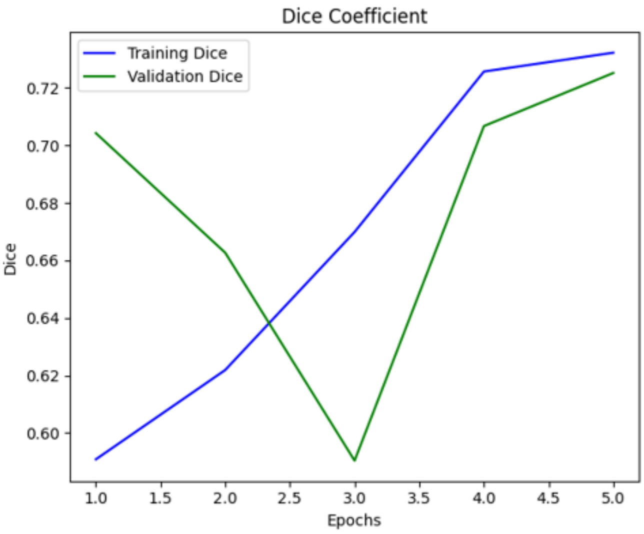


Figure 6: Dice Coefficient

IoU ranges from 0 to 1, with higher values indicating better overlap. It is a stricter metric than Dice, especially in cases of partial overlap.

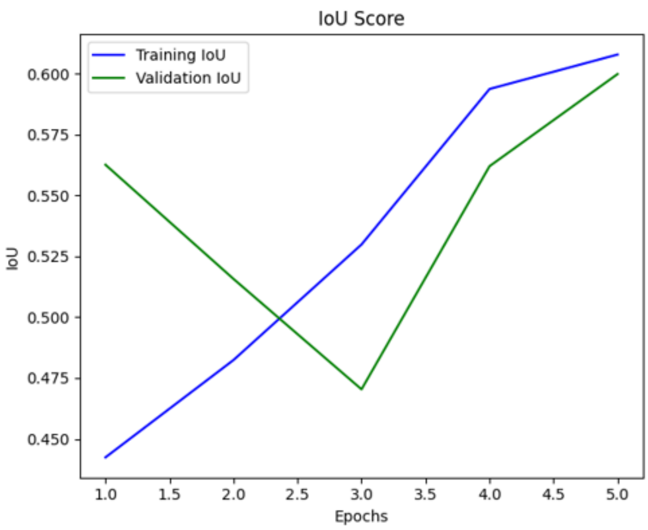


Figure 7: IoU Score

Table 1 Classification (MobileNet)Results

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.7240	0.7904	0.5980	0.3983
3	0.8232	0.8621	0.3668	0.2973
5	0.8398	0.8382	0.3223	0.3217
10	0.9132	0.9218	0.2158	0.2132

Table 2: Segmentation Results (U-Net with MobileNetV2)

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.9435	0.9890	0.1333	0.0338
3	0.9926	0.9922	0.0191	0.0200
5	0.9925	0.9937	0.0252	0.0150

VII. User Interface and Deployement

1. Interface Design

In order to interact with our system in real-time for the detection of brain tumors, we designed a user-friendly web interface via the use of Gradio ,which is a Python library developed to facilitate the deployment of machine learning models. With this interface, users can upload the brain please do a spell check MRI images, execute the classification and segmentation model and visualize the results.

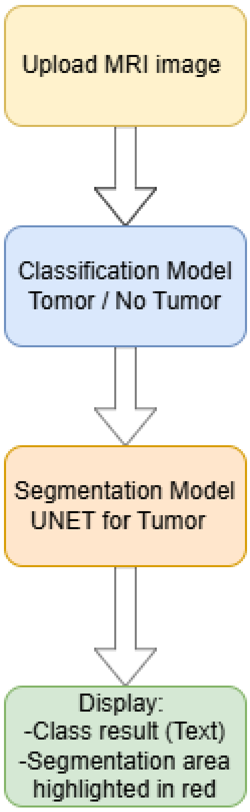


Figure 8 : Overview of the Gradio-based User Interface workflow.

Interface generalities some of the key interface features include:

- **Image uploading:** Users are able to upload MRI images in various formats. The models require RGB input and the system changes images from gray or alpha channel (RGBA) to RGB if necessary.

- **Preprocessing:** The size of the uploaded images is resized to 224×224 pixels and normalized by dividing by 255 to have pixel values in the range of [0,1] to fit the input of the models.

- **A classification:** a pre-trained convolutional neural network predicts whether there is a tumor or not.

- **Segmentation:** If a tumor is detected, the segmentation model (U-Net) generates a mask that highlights the tumor region in red on the original image.

- **Display of result:** The classification result is displayed as text, the segmentation mask as a visual image.

- **Error Handling:** It returns human-readable messages for incorrect input image or for any error during prediction.

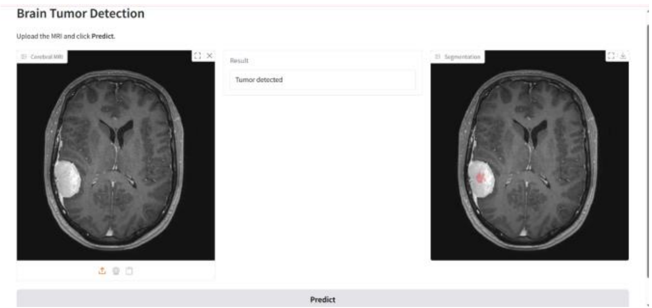
2. Implementation Details

The interface was designed using Gradio's Blocks API for a clean, modular presentation. It is running in a dedicated server port, fully accessible remotely via public URL with ngrok tunneling through development.

Both the classification and segmentation models require 224×224×3 input sizes; so, images are preprocessed with this in mind. The system invokes the models asynchronously to keep the UI responsive.

3. Interface screenshots

a. MRI image with tumor detected, segmented tumor area highlighted in red.



b. Interface indicating no tumor is found for a specific MRI input.

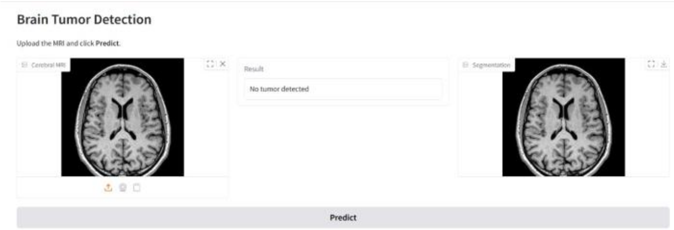


Figure 9: Screenshots of the Gradio interface for the time-of-use brain tumor detection.

These cases indicate the effectiveness of the system in classifying and segmenting the brain tumors.

4. Future Work

Gradio is great for quick prototyping and demonstrations, production deployment would benefit from wrapping the models behind a REST API using frameworks like FastAPI or Flask. Docker containerization and cloud-native orchestration like Kubernetes would scale and reliability Further enhancements may include tumor heatmap visualization, batch processing of images, and multi-mode imaging support.

References

[1] The dataset is shared open source. Availability Link : https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

[2] Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging

[3] Akmalbek Bobomirzaevich Abdusalomov 1,*, Mukhriddin Mukhiddinov 1, Taeg Keun Whangbo 1,* Editor: Hirotaka Hasegawa1 : https://www.nature.com/articles/s41598-024-52823-9

[4] Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN, Mohammad Zafer Khaliki & Muhammet Sinan Başarslan Scientific Reports volume 14, Article number: 2664 (2024) Cite this article https://pmc.ncbi.nlm.nih.gov/articles/PMC10453020/

[5] https://www.ibm.com/think/topics