Brain Tumor Detection And Classification With An Android App

Mrs.KPadmaja[1], ArpithaS[2], CharithaK[3], DiyaJain[4], SamanvithaG[5]
[1]AssistantProfessor, Dept of ISE
[2, 3, 4, 5]Dept of ISE
[1, 2] East West Institute of Technology, Bengaluru

Abstract- *Brain tumor detection is a critical task in medical diagnostics, requiring accurate and efficient models to classify tumor presence in MRI scans. This project employs a Convolutional Neural Network (CNN) to identify tumors in medical images. The system processes MRI images, classifies them as "tumor" or "no tumor," and predicts with high confidence. The model leverages Tensor Flow/Keras for implementation, and its performance is evaluated using metrics like accuracy and loss curves. The solution aims to assist medical professionals by providing a reliable pre-diagnostic tool for faster and more accurate detection.*

Keywords- Brain Tumor Detection, Convolutional Neural Network (CNN), TensorFlow/Keras, Image Classification, Medical Imaging, Machine Learning

I. INTRODUCTION

Brain tumors are one of the most severe health conditions, characterized by abnormal growth of cells in the brain. Accurate and early detection of brain tumors is crucial for effective treatment and better prognosis. Traditional diagnostic methods rely heavily on manual interpretation of MRI scans by radiologists, which can be time-intensive and prone to human error.

With advancements in machine learning, especially Convolutional Neural Networks (CNNs), it has become possible to automate image analysis tasks with high precision. CNNs are well-suited for image-based applications due to their ability to learn spatial hierarchies from raw pixel data.

This project leverages CNNs to classify MRI scans as either "tumor" or "no tumor." The proposed system preprocesses input images, trains a deep learning model on labeled MRI data, and evaluates its performance using metrics like accuracy and loss. By deploying such an automated detection system, the project aims to support medical professionals, reduce diagnostic time, and improve decision-making in clinic

II. LITERATURESURVEY

1. “Brain Tumor Detection Based on Deep Learning Approach”Authors:Ghosh,S.,Kaur,A.,andKoundal,D.
Published Year: 2020

This paper provides an extensive review of deep learning techniques for brain tumor detection. The authors discuss the advantages of CNNs and transfer learning in MRI image classification. They highlight the challenges in acquiring labeled datasets and propose solutions such as data augmentation and fine-tuning pre-trained models. The paper serves as a foundational study for identifying the suitability of CNN-based architectures in brain tumor analysis.

2. "Deep Convolutional Neural Networks for Brain Tumor Classification"
Authors: Rajasekaran,M.,Srinivasan,R. Published Year: 2021

This study uses a CNN-based architecture for classifying brain tumors into multiple classes. The authors explore different architectures, including AlexNet and VGG16, for their comparative performance on MRI datasets. They also emphasize using techniques like dropout and batch normalization to enhance model accuracy and reduce overfitting. This paper directly supports the use of CNNs in our project.

3. "MRI-Based Brain Tumor Detection Using Transfer Learning"
Authors:Patel,J.,and Sharma,V. Published Year: 2022

This research investigates transfer learning methods for brain tumor detection, employing pre-trained networks such as ResNet and Inception. The authors demonstrate how leveraging pre-trained models on large datasets significantly improves classification accuracy, especially for small datasets. Their findings emphasize that transfer learning is a time-efficient and highly accurate approach, applicable to resource-limited settings.

III. METHODOLOGY

1. Dataset Preparation

Sources: MRI images were collected from publicly available datasets categorized as tumor and non-tumor.
Preprocessing: Images were resized to 128x128 pixels for uniformity.RGB channels were retained to leverage color data.Data augmentation was performed to increase dataset size and variability.
Encoding: Labels were encoded using One-Hot Encoding, where tumor = [0] and non-tumor = [1].

2. Model Development

A Convolutional Neural Network (CNN) was used for feature extraction and classification due to its proven efficiency in image recognition tasks.
Input Layer: Accepts 128x128x3 images.
Convolutional Layers: Extract spatial features using a kernel size of 2x2 with ReLU activation.
MaxPooling: Reduces spatial dimensions to prevent overfitting and reduce computation.
Batch Normalization: Stabilizes learning by normalizing layer outputs.
Fully Connected Layers: Maps the extracted features to binary classification using dense layers.
Dropout Layers: Added to avoid overfitting by randomly deactivating neurons during training.
Output Layer: Sigmoid activation for binary classification (tumor or non-tumor).

3. Model Training

Loss Function: Categorical Cross-Entropy to measure classification error.
Optimizer: Adamax optimizer was used for adaptive learning rates and better convergence.
Metrics: Accuracy was used as the primary metric for evaluating model performance.
Training Parameters:
Epochs: 30
Batch Size: 40

4. Model Evaluation

Data was split into 80% training and 20% testing sets. Validation loss and accuracy metrics were tracked during training to assess overfitting. Plots of loss and accuracy trends were generated to visually analyze the model's performance.

5. Deployment
The trained model was saved and integrated into a FastAPI endpoint for real-time predictions. MRI images uploaded via the interface were processed and classified by the model, with confidence levels displayed to the user.

6. Prediction and Result Analysis

The model's prediction capability was tested on unseen MRI images.

Results included both the classification (tumor or non- tumor) and the confidence score, providing a robust and interpretable output for medical professionals

IV. SYSTEM ARCHITECTURE

1. User Interaction Layer

Purpose: Enables users to interact with the system. Functions: Upload MRI images for analysis. View the results, including tumor detection confidence.
Interface: A web-based UI created using Flet, ensuring ease of use and accessibility.

2. Preprocessing Layer

Purpose: Prepares the input MRI images for the model. Functions: Resize images to a fixed resolution (128x128 pixels) for consistency. Normalize pixel values to enhance model performance. Encode the image into a format acceptable by the classification model.

3. Model Inference Layer

Purpose: Performs tumor classification using a pre-trained Convolutional Neural Network (CNN).
Functions: Load the trained model. Classify the image into two categories: Tumor or No Tumor. Output the classification label along with confidence scores.

4. Backend API Layer

Purpose: Facilitates communication between the UI and the classification model.
Functions: Implemented using FastAPI, ensuring real-time predictions. Securely handle image uploads and retrieve predictions from the model.

5. Database Layer

Purpose: Manage user data and logs.

Functions: Store user registration details, such as email, password, and other credentials. Log analysis results for auditing and research purposes.
Technology: PostgreSQL is used for its scalability and robust data handling.

V. HADWARE AND SOFTWARE REQUIREMENTS

Processor (CPU):

Requirement: Intel Core i5/i7 (or equivalent) or higher.
Purpose: The CPU is crucial for running the image preprocessing, model inference, and server-side API. Higher clock speeds and multiple cores help in faster processing.

Graphics Processing Unit (GPU):

Requirement: NVIDIA GTX 1060 or higher (Recommended for training); for inference, GPU is not mandatory.
Purpose: Deep learning models, especially convolutional neural networks (CNNs), benefit from GPU acceleration. A dedicated GPU will speed up model training significantly, reducing time from hours to minutes.

RAM (Memory):

Requirement: Minimum of 8GB RAM (16GB or more recommended).
Purpose: Sufficient RAM is needed to load large datasets, perform image preprocessing, and handle multiple requests in real-time during deployment.

Storage:

Requirement: 500GB HDD or SSD (SSD recommended for faster data access).
Purpose: For storing images, model weights, training datasets, and results. SSDs provide faster read/write speeds, making th
Libraries & Frameworks:
TensorFlow/Keras: For building, training, and deploying deep learning models (CNN in this case). TensorFlow is optimized for handling large datasets and providing GPU support.
NumPy: A library for numerical computations, used for manipulating data, including image transformations.
Pandas: For data manipulation and processing.
Scikit-Learn: For handling machine learning tasks like data splitting, encoding, and performance evaluation.
Matplotlib: For visualizing model performance and generating plots for metrics like losystem more responsive during training and inference.

Network:

Requirement: High-speed internet connection (for cloud-based services or data transfer).
Purpose: Fast internet is needed for downloading pre- trained models, transferring large MRI images, and interacting with the server when deployed online.

Operating System:

Requirement: Windows 10/11 or Linux (Ubuntu recommended).
Purpose: A stable operating system is required for running development tools, managing system resources, and ensuring compatibility with machine learning libraries and frameworks.
Requirement: Windows 10/11 or Linux (Ubuntu recommended).
Purpose: A stable operating system is required for running development tools, managing system resources, and ensuring compatibility with machine learning libraries and frameworks.

VI. CONCLUSION

This project demonstrates the potential of CNNs in automating brain tumor detection from MRI scans. The model achieves high accuracy, offering a scalable solution for pre-diagnostic assistance in clinical settings. By integrating advanced techniques like data augmentation and deploying the model in real-world applications, this system can enhance diagnostic efficiency and accessibility.

By using convolutional neural networks (CNNs) trained on MRI images, the system achieves high performance in classifying images into tumor-positive and tumor-negative categories. The use of hardware accelerators such as GPUs ensures that the system is efficient and capable of handling large datasets, while the deployment on platforms like Flask or FastAPI enables real-time inference in practical scenarios.

The integration of advanced frameworks such as TensorFlow, OpenCV, and Keras has simplified the implementation process and enhanced the scalability of the system. Furthermore, the emphasis on user-friendly design ensures that medical professionals can use the tool without requiring advanced technical expertise.

VII. ACKNOWLEDGMENT

This project received support from the East West of Technology. We express our sincere appreciation guide, Mrs. K Padmaja . Assistant Professor Department of Information Science and Engineering for her valuable guidance and

significant contributions. We also acknowledge Mrs.Shruthi , Prof & Head of ISE Department, for her support and mentorship throughout the project. special thanks go to Principal Dr. Channakeshavalu for wavering encouragement. We are indebted to pesence for their constructive feedback on earlier versions . Any shortcomings in the manuscript remain our and should not reflect negatively for the proffesional mentioned above

REFERENCES

[1] X. Y. Jing, F. Wu, Z. Li, R. Hu and D. Zhang, "Deep learning Alogrithms," in IEEE Transactions on Image Processing, vol. 25, no. 6, pp. 2712- 2725, June 2020.

[2] T. Rehman, M. Khan, A. Zia, et al,"Deep Learning-Based Brain Tumor Detection and Classification Using MRI Scans",inIEEE,vol.25,no.6,pp.2720-2755,June 2022.