Brain Tumor Detection

Submitted for

Programming for AI ABAI5003L

Submitted by:

Sourabh

A25MTAI0007

Submitted to

DR. SHAKSHI SHARMA

July-Dec 2025

SCHOOL OF ARTIFICIAL INTELLIGENCE

BENNETT UNIVERSITY
THE TIMES GROUP

Table Of Contents

Sr.No	Content	Page No

Abstract

Brain Tumor Detection is very critical and most challenging task in medical due lack of precise, accurate, fast, clear and consistent diagnosis. MRI (Magnetic Resonance Imaging) provides good visibility of soft tissues, and ideal for locating abnormal growths of tissues or cells. This manual approach by radiologists can include a lot of time consumption and chances to human error increase. To address this problem, the proposed end-to-end AI-based solution utilizes deep learning techniques to automatically classify Tumor and categories of tumor with the help of MRI images analysis.

In this work, we use Convolutional Neural Network (CNN) based on the VGG16 architecture is fine-tuned used to learn discriminative features from MRI datasets. The dataset is preprocessed by following steps resizing, normalization, noise reduction, and data augmentation to improve model robustness and generalization.

The developed system can able to assist radiologists by providing fast and reliable preliminary screening, results reducing diagnosis time and improving treatment plans. This approach holds potential for integration into clinical workflows, mobile health applications, and automated radiology systems. Future improvements may include multi-class tumor classification, tumor segmentation, and 3D MRI analysis for more comprehensive diagnostic support using new approaches, models and leanrings.

Introduction

Brain tumors represent one of the critical and life-threatening pathologies within the central nervous system. Early diagnosis and characterization of brain tumors are of paramount importance to determine the treatment plan for a patient, which improves survival rates and reduces long-term neurological damage. MRI is widely regarded as the most effective imaging modality for detecting abnormalities within the brain because of its superior contrast resolution and non-invasive nature. However, manual interpretation of MRI images is extremely specialized and requires experienced neuroradiologists. Human examination often suffers from visual fatigue, subtle boundaries of tumors, overlapping tissue structures, and differences in imaging conditions. These factors may cause difficulties in early diagnosis or misinterpretation, especially in resource-limited medical settings where expertise by experienced radiologists may not be available.

The rapid growth in AI and deep learning opened a new perspective on medical image analysis. Among these deep learning methods, CNNs have been outstanding in recognizing complex patterns from images, which can hardly or could not be identified by the human eye. Their efficiency in automatically extracting features hierarchically from raw image data for classification, detection, and segmentation motivated their application for medical image classification. This project incorporates deep learning to develop an automated brain tumor detection system that can classify MRI images as tumor and non-tumor with a high degree of accuracy.

One of the major advantages of using deep learning is that it can learn from the raw data itself in an end-to-end manner, without explicitly requiring feature engineering. Classic machine learning methods involve features handcrafted by medical experts. However, these kinds of features generally do not generalize well across datasets, scanners, and different conditions of imaging. On the other hand, CNN-based models learn features meaningfully, hence are more adaptable and accurate. In CNN architectures, VGG16 has emerged as one of the powerful, therefore widely used models, because of its simplicity, depth, and proven capability for extracting rich spatial features. VGG16 model involves multiple convolutional layers where low-level to high-level features are progressively learned, and thus is highly effective for any medical image classification tasks.

The goal of the present project is to design and develop a deep learning-based model using the VGG16 architecture for the classification of brain MRI images as tumor and further classification into its types. The presence of transfer learning enables the model to leverage pre-trained weights on the ImageNet dataset and allows it to learn faster and make optimum performance in conditions where medical imaging data is limited. It is fine-tuned by adding customized layers consisting of Global Average Pooling, Dense layers, and a Softmax output layer designed for binary classification. Data preprocessing forms an important basis for good model performance. Resizing, normalization, noise reduction, histogram equalization, and data augmentation are performed to improve image quality, increase diversity in datasets, and reduce overfitting.

Brain tumor detection using an automated system is of increasing importance in today's healthcare scenario. Large-population countries like India are suffering from an ever-growing burden of neurological disorders, and accessibility to specialized radiologists remains poor in many parts of the country. Such an AI-assisted diagnosis system will help doctors with fast, preliminary reports, thereby easing the workload and enhancing diagnostic reliability. In cases where decisions need to be made urgently, such a system will prove very useful. Again, with increasing integration of AI in medical imaging gadgets and telemedicine platforms, such systems can be extended to rural and semi-urban healthcare set-ups where MRI scans may be available but expert interpretation is not possible.

In the last few years, deep learning has shown promising performance regarding the classification and segmentation of brain tumors. However, most of these works suffer from major issues like limited dataset size, poor generalization, lack of a standard preprocessing technique, and deployment challenges. This work identifies the gaps in existing works by proposing a robust preprocessing pipeline, augmenting the datasets, fine-tuning a very powerful CNN architecture, and performing thorough model performance evaluation using various evaluation metrics. Accuracy is not sufficient in medical applications; hence additional metrics like precision, recall, F1-score, and confusion matrix are utilized to ensure that the proposed model's performance is reliable and balanced for both classes.

Another important part of this work is model deployment and usability. A system cannot be said to be complete if it is not able to function in a natural setting. Therefore, the work also covers the development of an easy-to-use Python-based interface driven by either a simple script or a web framework. In this way, doctors, students, or researchers will be able to upload a magnetic resonance image and immediately get a prediction specifying whether there is a tumor or not. Such real-time prediction capability demonstrates the practical applicability of the model and its potential future integration into hospital systems.

The long-term implications of this project go beyond binary tumor classification. This workflow can be easily extended for multiclass tumor classification (glioma, meningioma, pituitary, no-tumor), tumor segmentation, and even 3D MRI analysis. More advanced architectures such as U-Net, ResNet, and EfficientNet may even further improve performance. It is also possible to integrate techniques from the field of explainable AI (XAI) that can enable radiologists to understand why the model is predicting a tumor, hence increasing trust and reliability. Deploying it on the cloud, developing mobile apps, and integrating with PACS will, therefore, turn this into a clinically usable tool.

Methodology

1. Dataset Acquisition and Organization

The first step in the methodology is collecting a suitable dataset of brain MRI images. For this project, publicly available MRI datasets were used, containing images categorized into two folder: *Training and Testing* which contain 4 Classess: *glioma, meningioma, pituitary, no-tumor*. The dataset has been taken from Kaggle to ensure sufficient data diversity. The dataset includes variations in contrast, brightness, noise, imaging angle, and tumor types, making the model more resilient to real-world scenarios.

The dataset is organized into separate folders based on the class labels. The structure is as follows:

```
Dataset/
  ├── Training/
  └── Testing /
```

This folder structure simplifies for training and testing. Each MRI image inside the folder inherits the label of the folder it belongs to. The dataset is further split into *training* and *testing* sets, ensuring that the model is evaluated on completely unseen data. Typically, 80% of the images are used for training, and 20% are reserved for testing.

2. Data Preprocessing

MRI images come in various sizes, contrast levels, and noise conditions. To make the dataset uniform and compatible with the VGG16 input standards, a comprehensive preprocessing pipeline is applied.

2.1 Resizing and Normalization

All input images are resized to **128 × 128 × 3** pixels to match the input shape of the model (as trained). Normalization is applied to scale pixel intensity values to the range **0 to 1**, improving the stability of the training process.

2.2 Data Augmentation

Since medical datasets are usually limited, augmentation techniques are applied to artificially increase the dataset size. Augmentation improves generalization and reduces overfitting by introducing variations such as:

- Contrast
- Random zoom in/out
- Brightness adjustments

This step significantly enhances the robustness of the model.

3. Model Architecture Design

The core of this work is a Convolutional Neural Network (CNN)-based architecture built using transfer learning. Transfer learning is chosen because pre-trained models like VGG16 already possess strong feature extraction capabilities from large datasets like ImageNet.

3.1 Transfer Learning with VGG16

VGG16 is a deep convolutional architecture consisting of 16 layers, known for its simplicity and strong spatial feature extraction. The convolutional base of VGG16 is imported **without the top fully connected layers**, allowing customization for the tumor detection task.

Key aspects:

- include_top=False (removing original classifier)
- Pre-trained weights from ImageNet
- Freeze initial layers to retain learned features
- Fine-tune upper layers for MRI-based features

3.2 Custom Classification Head

After the VGG16 feature extractor, a custom classification head is added:

- **Global Average Pooling**: Reduces spatial dimensions while retaining essential features.
- **Dense Layer (ReLU)**: Learns complex relationships and features.
- **Dropout (0.2)**: Reduces overfitting by randomly deactivating neurons during training.
- **Output Layer (Softmax)**: Produces probability distribution for the tumor

The final architecture efficiently combines deep feature extraction with a well-regularized classifier.

4. Model Training

4.1 Compilation

The model is compiled with the following parameters:

- **Optimizer**: Adam, with learning rate 0.0001
- **Loss Function**: Categorical Cross-Entropy (for two classes)

These settings ensure smooth convergence and prevent instability during training.

4.2 Batch Processing

The dataset is trained in mini-batches (batch size = 20) for efficient GPU utilization and stable gradient updates.

4.3 Early Stopping

Early stopping is used to monitor validation loss and halt training when no improvement is observed. This prevents overfitting and saves computational resources.

4.4 Epochs

The model is trained for approximately 20 epochs, depending on convergence trends. The training process is evaluated using training loss, validation loss, training accuracy, and validation accuracy curves.

5. Model Evaluation

After training, the model is tested on an unseen dataset to ensure that it generalizes well beyond the images it has learned.

5.1 Classification Report

Additional metrics are used to judge clinical usefulness:

- Precision
- Recall
- F1-score
- Support

Recall is essential in medical applications because missing a tumor (false negative) can be life-threatening.

Classification Report of Model:

```
41/41 ━━━━━━━━━━━━━━━━━━━━ 313s 8s/step
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.96      0.92       300
           1       0.91      0.98      0.95       405
           2       0.75      0.83      0.79       306
           3       0.97      0.70      0.81       300

    accuracy                           0.88      1311
   macro avg       0.88      0.87      0.87      1311
weighted avg       0.88      0.88      0.87      1311
```

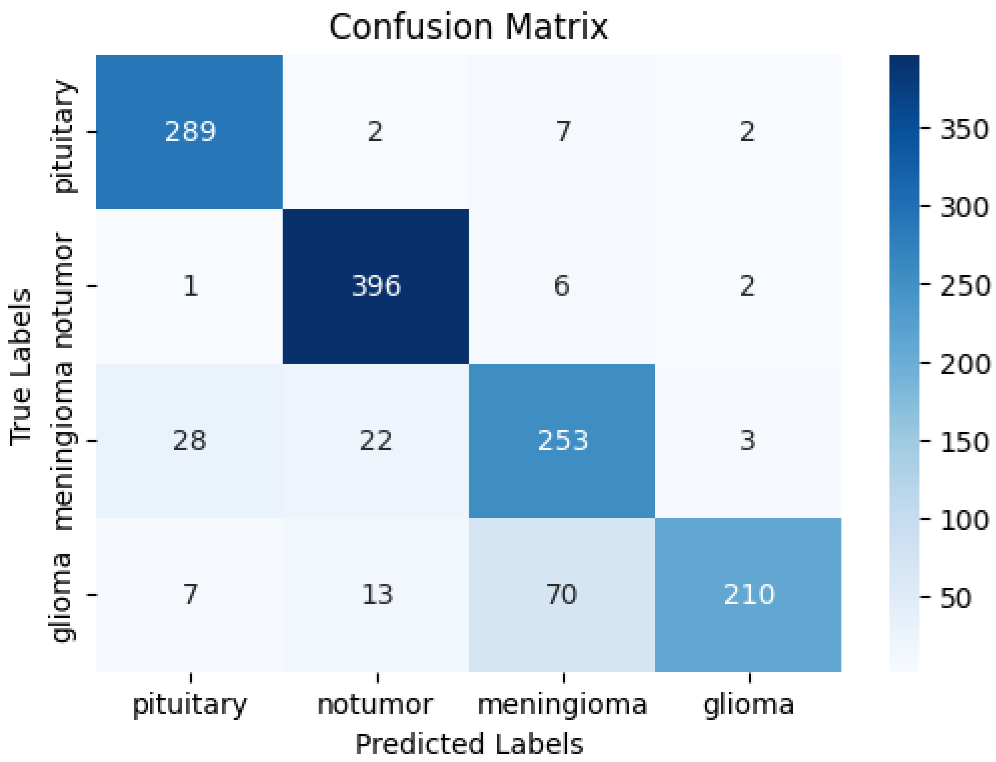
5.2 Confusion Matrix

A confusion matrix helps in understanding:

- How many tumor images were correctly classified
- How many non-tumor images were misclassified

Confusion Matrix with results:

```
Confusion Matrix:
[[289   2   7   2]
 [  1 396   6   2]
 [ 28  22 253   3]
 [  7  13  70 210]]
```

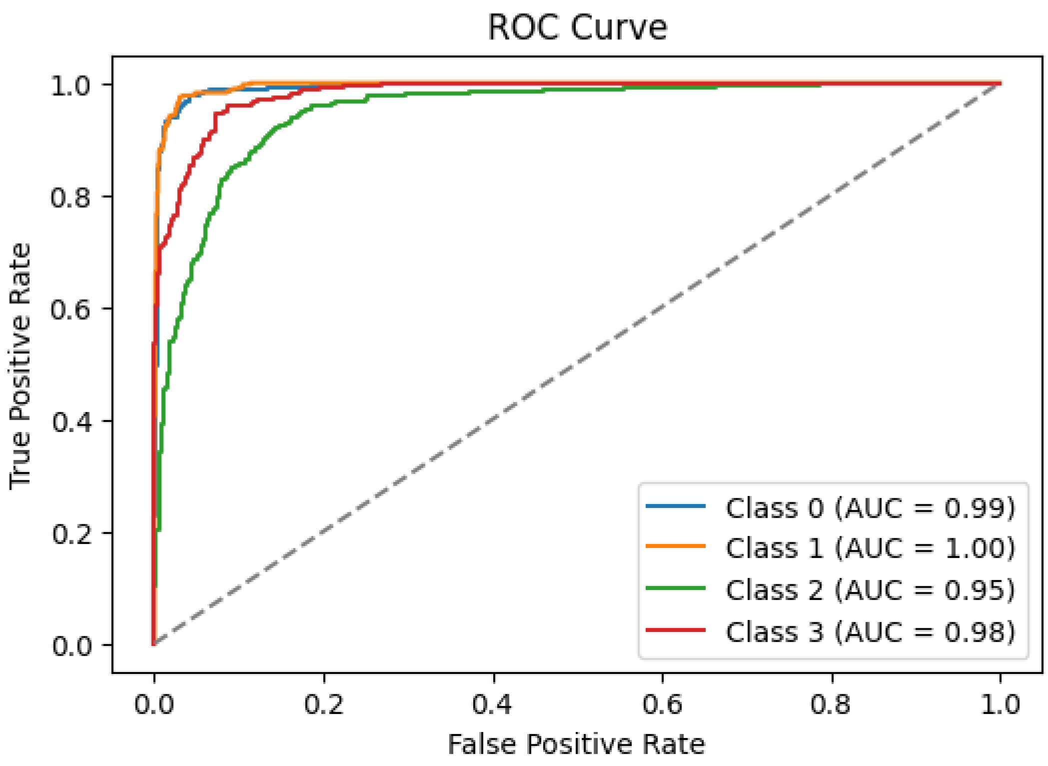


5.2 ROC Curve

The Receiver Operating Characteristic (ROC) Curve is a graphical performance evaluation tool used to measure how well a classification model distinguishes between two classes. In binary classification problems such as tumor vs non-tumor, the ROC curve helps analyze the model's ability to correctly classify positive and negative cases at different probability thresholds.

A model usually outputs a probability between 0 and 1. Depending on the chosen threshold (commonly 0.5), this probability is converted into a predicted class. Changing this threshold affects the number of correct and incorrect predictions. The ROC curve plots these changes across all possible thresholds, giving a complete picture of model performance.

Garph:



6. Deployment and Model Integration

A deep learning model is only useful when it can be applied in real-world situations. Therefore, deployment plays a crucial role in this project.

6.1 Python-Based Single-Script Deployment

A unified script (main.py) is designed to:

- Load trained model
- Accept an input MRI image
- Preprocess it
- Run classification
- Display the prediction

This makes testing and experimentation simple for students, doctors, or developers.

6.2 Web Interface / User Input

The model is integrated into a basic interface using gradio and Python such that users can upload an MRI image through a webpage.

6.3 Real-Time Predictions

The deployed system produces predictions in real-time, typically under one second per image. This simulates how a real hospital deployment may function.

7. System Workflow Summary

To summarize the entire methodology, the workflow consists of the following interconnected stages:

1. **Dataset Collection** → MRI images (tumor & non-tumor)
2. **Data Preprocessing** → resizing, normalization, augmentation
3. **Model Development** → VGG16 + custom dense layers
4. **Training** → Adam optimizer, early stopping
5. **Evaluation** → accuracy, confusion matrix
6. **Deployment** → unified script or web interface
7. **Prediction** → classify new MRI images

This pipeline ensures that the system is complete end-to-end, from raw data to a deployable intelligent model.

8. Significance of the Methodology

This methodology ensures:

- High performance using proven architectures
- Generalization due to augmentation
- Stability through careful preprocessing
- Practical usage through deployment integration

The holistic approach also ensures that the model is not only academically valuable but also practically applicable in real-world medical environments.

Hardware and Software Requirements

Hardware:

- GPU-enabled system (NVIDIA T4 / P100 / RTX preferred)

- 8–16 GB RAM

- Minimum 2 GB storage for the dataset

Software:

- TensorFlow 2.x

- Keras

- Python 3.10

- NumPy, Pandas, OpenCV

Matplotlib, Seaborn

- Google Colab / Jupyter Notebook / VS Code

6. Conclusions

The proposed VGG16-based brain tumor detection model shows high accuracy and is reliable. Deep learning proves to be very powerful for medical image analysis, especially for the classification of brain tumors. Our approach enhances the reliability of diagnosis and provides a quick automatic second opinion to the radiologist.

This project shows the entire development pipeline, starting from data preparation and augmentation to CNN feature extraction, model training, testing, and deployment. These techniques can easily be extended to multi-class tumor classification and segmentation models.

Future Scope

Future work involves the following:

Multi-class Tumor Detection

1.Classifying glioma, meningioma, and pituitary tumors separately.

2. Tumor Segmentation Accurately locating tumor boundaries using U-Net.

3. Explainable AI (XAI) Implementing Grad-CAM heatmaps to visualize tumor regions.

4. Larger Multi-center Datasets Training on MRI scans from multiple hospitals

5. Mobile Application Development Real-time image upload and prediction interface.

6. Integration with Hospital Systems Deploy in PACS systems for automatic reporting.

7. GAN-based MRI Generation: New Innovation Training GAN models to generate synthetic MRI scans to solve problems arising from small datasets.

8. Reinforcement Learning for Radiology Automation Dynamic adjustment of preprocessing and contrast level by RL agent can increase model's accuracy.

GitHub Link of Your Complete Project

https://github.com/SourabhHooda/Brain-Tumor-Detection