International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

BRAIN TUMOR DETECTOR USING DEEP LEARNING

MS.B.Sunitha[1], K. Prashanth[2], M.Mahender[3]
[1]Assistant Professor, VidyaJyothi Institute of Technology, Hyderabad
[2,3]UG Student,Dept of CSE(Data Science) , VidyaJyothi Institute of Technology, Hyderabad

Abstract: This project presents a deep learning-based approach for the automatic detection of brain tumors from MRI (Magnetic Resonance Imaging) images. Brain tumors are serious medical conditions that require early and accurate diagnosis to improve patient outcomes. Manual analysis of MRI scans is time-consuming and may lead to human errors, creating a need for an efficient automated system.
In this work, a Convolutional Neural Network (CNN) model based on the pre-trained VGG16 architecture is used to classify brain MRI images into two categories: Tumor and No Tumor. The dataset is preprocessed by resizing images to 224×224 pixels and normalizing pixel values. Transfer learning is applied by freezing the base layers of VGG16 and adding custom classification layers to improve performance with limited data.
The model is trained and evaluated using standard performance metrics such as accuracy, confusion matrix, precision, recall, and F1-score. The experimental results demonstrate that the model achieves approximately 90% accuracy and performs effectively in detecting brain tumors.
This system can assist medical professionals by providing faster and more reliable preliminary diagnosis. Future enhancements may include using larger datasets, applying advanced deep learning models, and extending the system for multi-class tumor classification.

1.Introduction:

Brain tumor detection is one of the most critical challenges in the medical field, as early and accurate diagnosis can significantly improve patient survival rates. A brain tumor is an abnormal growth of cells inside the brain, which can be life-threatening if not detected in time. Doctors typically use MRI (Magnetic Resonance Imaging) scans to identify tumors. However, manual analysis of MRI images is time-consuming and depends heavily on the expertise of medical professionals.
With the advancement of Artificial Intelligence (AI) and Deep Learning, it is possible to automate the detection process. In this project, a deep learning model based on the VGG16 architecture is used to classify brain MRI images into tumor and non-tumor categories. The images are preprocessed using resizing and normalization techniques, and transfer learning is applied to improve performance.

1.1 Problem Statement

Detecting brain tumors manually from MRI images is a complex and time-consuming process that requires expert knowledge. It may also lead to errors due to fatigue or human limitations. Therefore, there is a need for an automated system that can accurately and efficiently classify MRI images as tumor or non-tumor.

1.2 Objectives

The main objectives of this project are:
• To develop a deep learning model for brain tumor detection
• To use the VGG16 model with transfer learning for classification
• To preprocess MRI images using resizing and normalization
• To evaluate the model using accuracy, confusion matrix, and classification report

1.3 Importance of Study

• Early detection of brain tumors can save lives
• Reduces the workload of medical professionals

DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

2. Literature Review:

Brain tumor detection is a crucial area of research in the field of medical image analysis. Early detection of brain tumors plays a significant role in improving patient survival rates and treatment outcomes. Traditionally, MRI (Magnetic Resonance Imaging) scans are used by radiologists to identify abnormalities in the brain. While MRI provides detailed images of brain structures, manual analysis of these images is time-consuming, subjective, and dependent on the expertise of the medical professional. This has led researchers to explore automated methods for brain tumor detection using machine learning and deep learning techniques.In the early stages of research, traditional machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Trees, and Naive Bayes were widely used for classification tasks. These approaches required manual feature extraction, where features such as texture, intensity, shape, and edges were extracted from MRI images. Although these methods provided moderate accuracy, they had several limitations. The process of selecting relevant features was complex and required domain knowledge. Moreover, these models were not robust when dealing with variations in MRI images, such as noise, contrast differences, and varying tumor shapes.With the advancement of Artificial Intelligence, deep learning techniques have significantly improved the performance of image classification systems. Convolutional Neural Networks (CNNs) have become the most widely used models for medical image analysis. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers, which automatically learn hierarchical features from images. This eliminates the need for manual feature extraction and improves overall accuracy.Several CNN architectures have been proposed for brain tumor detection, including AlexNet, GoogLeNet, ResNet, and VGG16. Among these, VGG16 is one of the most popular models due to its simple and uniform architecture. It consists of 16 layers with small convolution filters (3×3), which help in capturing fine details from images. VGG16 has been successfully applied in various image classification tasks and has shown strong performance in medical imaging applications, including brain tumor detection.

Another important concept widely used in recent studies is transfer learning. Transfer learning involves using a model that has already been trained on a large dataset, such as imagenet, and adapting it to a specific problem. This approach is particularly useful when the available dataset is small, as it reduces training time and improves model performance. In brain tumor detection, transfer learning allows the model to utilize pre-learned features and focus on learning task-specific patterns.

In addition to model selection, preprocessing techniques play a vital role in improving model accuracy. Common preprocessing steps include resizing images to a fixed size, normalizing pixel values, and converting images into numerical arrays. These steps ensure consistency in input data and make it suitable for training deep learning models. Proper preprocessing helps in reducing noise and improving feature extraction.

Recent research has also focused on developing complete systems that integrate deep learning models with user-friendly interfaces. Web-based applications using frameworks such as Flask enable users to upload MRI images and obtain predictions instantly. These systems are designed to assist healthcare professionals by providing quick and reliable results, thereby improving efficiency in clinical settings.

Despite significant advancements, there are still challenges in brain tumor detection systems. One of the major challenges is the availability of large and high-quality datasets. MRI images may vary in resolution, brightness, and quality depending on the scanning equipment used. These variations can affect the performance of the model. Additionally, most systems are limited to binary classification (tumor vs. Non-tumor) and do not classify different types of tumors.

Overall, the literature suggests that deep learning techniques, particularly CNN-based models like VGG16 combined with transfer learning, provide more accurate and reliable results compared to traditional machine learning methods. These approaches have the potential to significantly improve the efficiency and accuracy of brain tumor detection systems and support medical professionals in diagnosis.

DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

3. Methodology:

The methodology describes the step-by-step process used to develop the brain tumor detection system. It includes system design, data preprocessing, model development using VGG16, training, and evaluation. The proposed system uses a deep learning approach with transfer learning to classify MRI images into tumor and non-tumor categories.

3.1 System Design Approach

The system is designed to automatically detect brain tumors from MRI images using a Convolutional Neural Network (CNN). The overall workflow of the system is as follows:

- Collection of MRI image dataset
- Preprocessing of images (resizing and normalization)
- Feature extraction using a pre-trained VGG16 model
- Classification using fully connected layers
- Evaluation of model performance
- Prediction of tumor or non-tumor for new images

The system uses transfer learning, where a pre-trained model is adapted for the specific task of brain tumor detection. This approach improves accuracy and reduces training time.

3.2 Data Collection and Preprocessing

The dataset consists of brain MRI images categorized into two classes:

- Tumor
- No Tumor

Data Collection:

The dataset is obtained from an online source and organized into separate folders based on class labels.

Preprocessing Steps:

Before training the model, the images are preprocessed as follows:

- Image Resizing: All images are resized to 224×224 pixels to match VGG16 input size
- Normalization: Pixel values are scaled from 0–255 to 0–1
- Label Encoding: Labels are converted into numerical format using LabelBinarizer
- Train-Test Split: The dataset is divided into training and testing sets (80% training, 20% testing)

These preprocessing steps ensure that the data is consistent and suitable for training the deep learning model.

3.3 Model Design using VGG16

In this project, the VGG16 pre-trained Convolutional Neural Network is used for feature extraction and classification.

Why VGG16?

- Deep architecture with 16 layers
- Pre-trained on ImageNet dataset
- Provides high accuracy in image classification tasks

Model Architecture:

- The original top layers of VGG16 are removed
- The base model is used for feature extraction
- Custom layers are added:
  - Average Pooling Layer
  - Flatten Layer
  - Dense Layer with ReLU activation
  - Dropout Layer (to reduce overfitting)
  - Output Layer with Softmax activation (2 classes)

Transfer Learning:

- All layers of the VGG16 base model are frozen
- Only the newly added layers are trainedThis design allows the model to use pre-learned features while adapting to the brain tumor classification task.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

3.4 Training Process

The model is trained using the preprocessed dataset.

Training Details:

- Optimizer: Adam
- Learning Rate: 0.001
- Loss Function: Categorical Crossentropy
- Batch Size: 32
- Epochs: 20

During training:

- The model learns patterns from MRI images
- Training and validation accuracy are monitored
- The model adjusts weights to improve performance

To avoid retraining during testing or demo, the trained model is saved as a .h5 file and can be loaded later for prediction.

3.5 Evaluation Metrics

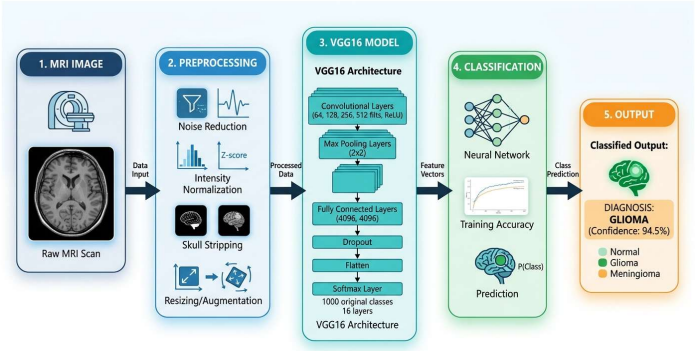

Figure 3.1: Methodology Workflow of Brain Tumor Detection System

After training, the model is evaluated using standard performance metrics:

1. Accuracy

Measures the overall correctness of predictions.

2. Confusion Matrix

Displays correct and incorrect predictions:

- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

3. Classification Report

Includes:

- Precision – accuracy of positive predictions
- Recall – ability to detect actual tumor cases
- F1-Score – balance between precision and recall

These metrics help in analyzing the effectiveness of the model in detecting brain tumors.

4. Implementation:

The implementation phase involves developing the brain tumor detection system using Python and deep learning libraries. The system is implemented using a Convolutional Neural Network (CNN) based on the VGG16 architecture. The implementation includes dataset handling, preprocessing, model building, training, evaluation, and prediction.

Tools and Technologies Used:

- Programming Language: Python
- Deep Learning Framework: TensorFlow and Keras

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

• Image Processing: OpenCV
• Libraries: NumPy, Matplotlib, scikit-learn
• IDE: Jupyter Notebook / VS Code

4.1 Importing Required Libraries

All necessary libraries are imported, including TensorFlow/Keras for model building, NumPy for numerical operations, OpenCV for image processing, and scikit-learn for data splitting and evaluation.

4.2 Loading the Dataset

• The dataset of MRI images is loaded from a directory
• Image file paths are collected using the imutils library
• Images are categorized into two classes: Tumor and No Tumor

4.3 Image Preprocessing

Each image is processed before being fed into the model:
• Images are resized to 224×224 pixels
• Pixel values are normalized to the range 0–1
• Images are converted into NumPy arrays
• Labels are extracted from folder names

4.4 Data Preparation

• Labels are converted into binary format using LabelBinarizer
• Labels are further converted into categorical format
• The dataset is split into training and testing sets (80% training, 20% testing)

4.5 Model Building

• The pre-trained VGG16 model is loaded without top layers
• All layers of VGG16 are frozen
• Custom layers are added:
  – Average Pooling layer
  – Flatten layer
  – Dense layer with ReLU activation
  – Dropout layer
  – Output layer with Softmax activation

4.6 Model Compilation

The model is compiled using:
• Optimizer: Adam
• Loss Function: Categorical Crossentropy
• Metrics: Accuracy

4.7 Model Training

• The model is trained using training data
• Validation is performed using test data
• Training is done for 20 epochs with a batch size of 32
• Accuracy and validation accuracy are monitored

Figure 4.1: Training and Validation Accuracy Graph

4.8 Model Evaluation

• Predictions are made on test data
• Confusion matrix is generated

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

• Classification report is printed
• Model performance is analyzed using accuracy, precision, and recall

4.9 Model Saving and Loading

• After training, the model is saved as brain_tumor_model.h5
• The saved model can be loaded later for prediction without retraining

4.10 Prediction on New Images

• A new MRI image is provided as input
• The image is preprocessed (resized and normalized)
• The trained model predicts whether the image contains a tumor or not
• The output is displayed as:
  – Tumor Detected
  – No Tumor

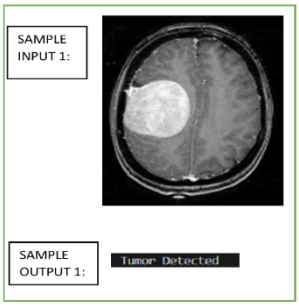

Figure 4.2: Sample Prediction Output

4.11 Result Visualization

• Accuracy graphs are generated during training
• These graphs help in understanding model performance and learning behavior

5.System Architecture:

The system architecture represents the overall structure and workflow of the brain tumor detection system. It shows how input data is processed step-by-step to generate the final prediction. The system is designed using a deep learning approach based on the VGG16 model.

The architecture consists of the following main components:

1. Input Layer (MRI Image)

The system takes a brain MRI image as input. The image can be provided by the user through the interface or selected from the dataset.

2. Preprocessing Module

The input image is preprocessed before being passed to the model. The preprocessing steps include:

• Resizing the image to 224×224 pixels
• Normalizing pixel values to the range 0–1
• Converting the image into an array format

These steps ensure that the image is compatible with the VGG16 model.

3. Feature Extraction (VGG16 Model)

The preprocessed image is passed through the pre-trained VGG16 model. This model extracts important features such as edges, textures, and patterns from the MRI image.

• The base layers of VGG16 are frozen
• Only feature extraction is performed at this stage

4. Classification Layer

After feature extraction, the output is passed to custom layers:

• Flatten layer
• Dense layer with ReLU activation
• Dropout layer
• Output layer with Softmax activation

These layers classify the image into two categories: Tumor or No Tumor.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

5. Output Layer (Prediction Result)

The final output is displayed as:

• Tumor Detected

• No Tumor

The system may also display the confidence score of the prediction.

Working Flow of the System

The complete workflow of the system is:

MRI Image → Preprocessing → VGG16 Feature Extraction → Classification → Output Result

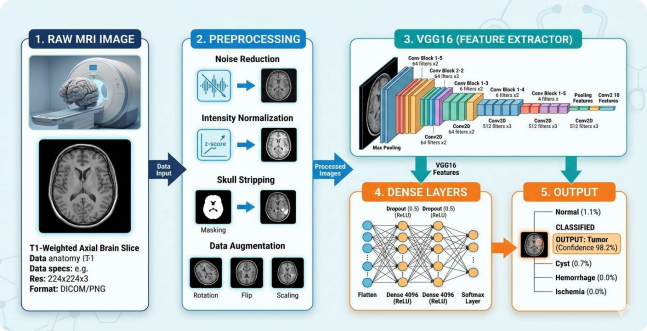

Figure 5.1: System Architecture of Brain Tumor Detection System

6.Model Implementation:

This section describes the implementation of the brain tumor detection system using a deep learning approach. The system is developed in Python using TensorFlow and Keras libraries. It focuses on training a Convolutional Neural Network (CNN) based on the VGG16 architecture and using the trained model for prediction.

6.1 Backend (Deep Learning Model)

The backend of the system is implemented using Python. The core model is built using the VGG16 pre-trained Convolutional Neural Network.

The implementation includes the following steps:

• Importing required libraries such as NumPy, OpenCV, TensorFlow, and scikit-learn

• Loading MRI images from the dataset directory

• Preprocessing images by resizing them to 224×224 pixels and normalizing pixel values

• Converting labels into numerical format using LabelBinarizer

• Splitting the dataset into training (80%) and testing (20%) sets

The VGG16 model is loaded without its top layers, and custom layers are added:

• Average Pooling layer

• Flatten layer

• Dense layer with ReLU activation

• Dropout layer

• Output layer with Softmax activation

Transfer learning is applied by freezing all layers of the VGG16 base model. The model is compiled using the Adam optimizer, categorical crossentropy loss function, and accuracy as the evaluation metric.

The model is trained for 20 epochs with a batch size of 32. After training, the model is saved as brain_tumor_model.h5 for future use.

The trained model is loaded from the saved file using the following code:

```
from tensorflow.keras.models import load_model
model = load_model("brain_tumor_model.h5")
```

6.2 Prediction Script (Inference Module)

A separate prediction script is implemented to test new MRI images without retraining the model. The script performs the following operations:

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

• Loads the trained model from the saved .h5 file
• Takes the path of a test MRI image as input
• Checks whether the image exists
• Preprocesses the image (resizing and normalization)
• Converts the image into the required input format
The model then predicts the class of the image:
• Output label 0 → No Tumor
• Output label 1 → Tumor
The prediction is displayed along with a confidence score, indicating the probability of the predicted class.

6.3 Working Flow of Prediction

The overall working flow of the system is as follows:
Input MRI Image → Preprocessing → Load Trained Model → Prediction → Output Result
The system allows efficient and fast prediction of brain tumors without requiring retraining. It can be used directly for demonstration and testing purposes.
The implementation successfully demonstrates how a trained deep learning model can be used for real-time prediction using a simple and efficient approach.

7. Testing and Output:

This section presents the testing procedure and results of the brain tumor detection system. The trained model is evaluated using unseen test data to measure its performance and reliability.

7.1 Testing Process

After training the model, testing is performed using a separate dataset that was not used during training. The dataset is split into 80% training and 20% testing.
The testing process includes:
• Loading the trained model (brain_tumor_model.h5)
• Passing test images (testX) to the model
• Generating predictions using the trained model
• Comparing predicted labels with actual labels
The predictions are generated using the model's predict() function, and class labels are obtained using argmax.

7.2 Confusion Matrix

The confusion matrix is used to evaluate the classification performance of the model. It shows the number of correct and incorrect predictions.
Confusion Matrix:

```
Confusion Matrix:
[[19  1]
 [ 4 27]]
```

Where:
• 19 → Correctly predicted No Tumor (True Negatives)
• 27 → Correctly predicted Tumor (True Positives)
• 1 → Incorrectly predicted Tumor (False Positive)
• 4 → Tumor cases missed by the model (False Negatives)
This indicates that the model performs well in identifying both classes, with a small number of errors.

7.3 Classification Report

The classification report provides detailed evaluation metrics such as precision, recall, and F1-score.
• Precision measures how accurate the predictions are
• Recall measures how well the model detects actual tumor cases
• F1-score balances precision and recall

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

From the results:
• Tumor detection precision is high (around 0.96)
• Recall for tumor is slightly lower (around 0.87), indicating a few missed cases
• Overall accuracy of the model is approximately 90%

```
Classification Report:
Class labels: ['no' 'yes']
              precision    recall  f1-score   support

          no       0.83      0.95      0.88        20
         yes       0.96      0.87      0.92        31

    accuracy                           0.90        51
   macro avg       0.90      0.91      0.90        51
weighted avg       0.91      0.90      0.90        51
```

7.4 Testing on New Image

The model is also tested on a new MRI image using a separate prediction script.
Steps involved:
• Load the trained model
• Provide input image path
• Preprocess the image (resize and normalize)
• Predict the class using the model
Output example:
• Tumor Detected with high confidence
• No Tumor detected for normal images
The system also displays the confidence score of the prediction.

7.5 Result Analysis

The model shows good performance in classifying brain MRI images. It achieves high accuracy and reliable predictions for both tumor and non-tumor classes. However,small number of tumor cases are missed (false negatives), which can be improved with more data or advanced models.
The testing results confirm that the system is effective and can be used as a supportive tool for brain tumor detection.

8.Case Study:

This section presents a detailed analysis of the brain tumor detection system, including the experimental setup, functional testing, performance evaluation, and final outcomes. The case study demonstrates how the system performs in real-world scenarios using MRI images.

8.1 Experimental Setup

The experiment is conducted using a dataset of brain MRI images categorized into two classes: Tumor and No Tumor. The system is implemented using Python with TensorFlow and Keras libraries.
The setup includes:
• Image preprocessing using resizing (224×224) and normalization
• Label encoding using LabelBinarizer
• Dataset split into training (80%) and testing (20%)
• VGG16 pre-trained model with transfer learning
• Training configuration: 20 epochs and batch size of 32
The trained model is saved as brain_tumor_model.h5 and reused for testing and prediction.

8.2 Functional Testing

Functional testing is performed to ensure that all components of the system are working correctly.
The following functionalities are tested:
• Loading of the trained model
• Image preprocessing steps

DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

• Prediction on test dataset
• Prediction on new MRI images

The system successfully processes input images and produces correct classification results as Tumor or No Tumor.

8.3 Performance Evaluation

The performance of the model is evaluated using accuracy, confusion matrix, and classification report, as discussed in Section 7.

The model achieves approximately 90% accuracy with high precision in detecting tumor cases. However, a few tumor cases are missed, indicating scope for further improvement.

8.4 Prediction Analysis

The system is tested on new MRI images using the prediction script.

The following observations are made:

• The system correctly classifies tumor and non-tumor images
• It provides fast prediction results
• It displays the output along with confidence score

This demonstrates that the model can effectively handle unseen data and provide reliable predictions.

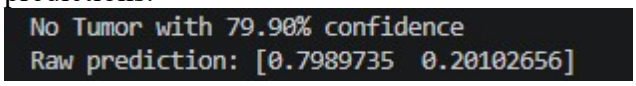

Figure 8.1: Prediction Output with Confidence Score

8.5 Outcome

The brain tumor detection system is successfully implemented using deep learning techniques. The model performs well in classifying MRI images and achieves good accuracy.

The system:

• Accurately detects tumor and non-tumor cases
• Reduces manual effort in diagnosis
• Provides quick and reliable predictions
• Can be further improved with more data and advanced models

Overall, the project demonstrates the effectiveness of using VGG16 with transfer learning for brain tumor detection.

9. Challenges and Limitations:

This section discusses the difficulties faced during the development of the brain tumor detection system and the limitations of the proposed approach.

9.1 Challenges

During the implementation of the project, several challenges were encountered:

• Limited Dataset Size:

The dataset used for training was relatively small, which can affect the model's ability to generalize well to new data.

• Image Variability:

MRI images may vary in terms of quality, brightness, and orientation, making it difficult for the model to learn consistent features.

• Preprocessing Complexity:

Ensuring all images are correctly resized and normalized is important. Any inconsistency in preprocessing can affect model performance.

• Model Training Time:

Training a deep learning model like VGG16 requires significant computational resources and time.

• Avoiding Overfitting:

With limited data, there is a risk that the model may memorize training data instead of learning general patterns.

9.2 Limitations

Despite achieving good performance, the system has certain limitations:

DOI:10.46647/ijetms.2026.v10i03.011 ISSN: 2581-4621

• Binary Classification Only:
The model can only classify images as Tumor or No Tumor. It does not identify different types of tumors.
• No Graphical User Interface:
The system is implemented using a script-based approach and does not include a user-friendly interface.
• Dependence on Dataset Quality:
The accuracy of the model depends heavily on the quality and diversity of the dataset used.
• False Negatives:
Some tumor cases are not detected by the model, which is critical in medical applications.
• Not a Medical Replacement:
The system is designed as a supportive tool and cannot replace professional medical diagnosis.
Overall, while the system demonstrates good performance, addressing these challenges and limitations can further improve its accuracy and usability.

10.Conclusion and Future Work:

This project presents a deep learning-based system for brain tumor detection using MRI images. A Convolutional Neural Network (CNN) based on the VGG16 architecture is used to classify images into tumor and non-tumor categories.
The system includes data preprocessing, model training, evaluation, and prediction using a saved model. The trained model achieves approximately 90% accuracy and performs well in identifying tumor cases. The use of transfer learning helps in improving performance while reducing training time.
The system successfully demonstrates how artificial intelligence can assist in medical image analysis. It provides fast and reliable predictions and can be used as a supportive tool for doctors.
However, there are some limitations, such as the use of a limited dataset and the possibility of false negative predictions. The system currently performs only binary classification and does not identify different types of tumors.
Future Work:
The project can be further improved in the following ways:
• Increasing the dataset size for better accuracy and generalization
• Using advanced models such as ResNet or EfficientNet
• Implementing data augmentation techniques
• Developing a graphical user interface for better usability
• Extending the system to classify different types of brain tumors
• Improving model performance to reduce false negatives
Overall, the project demonstrates the potential of deep learning in healthcare and provides a foundation for further research and development.

11.References:

- Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG16)*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*.
- TensorFlow Documentation. Available at: https://www.tensorflow.org
- Keras Documentation. Available at: https://keras.io
- Scikit-learn Documentation. Available at: https://scikit-learn.org
- Brain MRI Dataset (Kaggle or other source used in the project)