IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Brain Tumour Segmentation And Classification Using Deep Learning With 3D CNN

**[1]Harish nithi K, [2]Manojkumar S, [3]Mrs.T.Meenal**

[1,2]Student, [3]Associate Professor,
[1234]Electronics and communication engineering department,
[1234]Kongunadu College of engineering and technology, TRICHY, INDIA

***Abstract*:** This research presents a multimodal deep learning approach for brain tumour diagnosis focusing on software-driven solutions for medical image analysis and classification. The system integrates deep learning techniques to process MRI scans and other diagnostic data improving accuracy and efficiency. A Convolutional Neural Network (CNN) is used for image recognition and processing tasks like image classification while ensemble learning methods improve classification performance. The software is designed for automated pre-processing segmentation, and tumour classification ensuring robustness across diverse datasets. The framework is implemented with Python and Tensor Flow and supports real- time inference and can be deployed in clinical decision support systems. To get the result percentage as 95.3% accuracy in the Xception image which shows in the output.

***Index Terms*:** Multimodal machine learning, Deep Learning, Convolutional Neural Networks, Brain tumour, Magnetic Imaging Resonance, Computed Tomography.

## I.INTRODUCTION

Brain tumour diagnosis using deep learning involves processing multimodal data, including MRI scans Computed Tomography(CT) images and histopathological slides. Advanced algorithms enhance accuracy by integrating various data sources, improving detection and classification. Solution software-based solutions streamline medical imaging analysis for faster and more precise diagnosis. Deep Learning Architectures such as CNNs and K-means clustering algorithm plays a significant role in extracting the tumor from the brain tumor in deep learning and transformers play a crucial role in feature extraction and pattern recognition. They are trained on huge datasets to identify between benign and malignant tumors. Efficient software frameworks like Tensor - Flow and PyTorch enable seamless model implementation and optimization. Image preprocessing techniques including noise reduction contrast enhancement and segmentation are vital for improving model performance. Software tools automate these processes by automatically generating input data for Deep learning models using convolutional neural network (CNN). Pretrained models and transfer learning approaches further refine prediction with minimal computational overhead. Cloud-based and edge computing solutions increase real time tumour detection and diagnosis. Software platforms integrate AI-driven models with medical imaging systems, enabling remote diagnosis and collaboration to get the graph using AI as manual given input image. Secure data handling and compliance with medical standards ensure reliable.

## II. OBJECTIVES

- The project aims to develop a multimodal deep learning system for brain tumor diagnosis, integrating various software-based models for enhanced accuracy.
- It focuses on processing MRI scans using deep learning techniques. The system ensures automated, reliable, and scalable diagnosis without human intervention.
- Advanced image processing algorithms will be implemented to extract and analyze features from MRI scans, aiding in tumor classification.
- Convolutional neural networks (CNNs) will be utilized to identify patterns and anomalies in medical images and get more accuracy compare to machine learning techniques.
- The software will enhance segmentation and detection, improving diagnostic precision.

## III. METHODOLOGY

1. **Data Preparation**:

- **Data Collection:** Gather a diverse dataset of brain MRI scans, including images of both healthy brains and various types of brain tumors.
- **Data Preprocessing:**
  - **Image Enhancement:** Apply techniques like contrast enhancement and normalization to improve image quality.
  - **Data Augmentation:** Use techniques like rotation, flipping, and cropping to artificially increase the dataset size and improve model robustness.
  - **Segmentation (Optional):** If the goal is tumor segmentation, use techniques like thresholding or clustering to separate the tumor region from the background. So we are using K-means clustering image segmentation technique.
  - **Normalization:** Scale the pixel values to a standard range (e.g., 0-1).
- **Data Splitting:** Divide the dataset into training, validation, and testing sets.

2. **Model Selection and Training**:

- **CNN Architecture:**
  Choose a suitable CNN architecture, such as Xception, ResNetV2, DenseNet201, or EfficientNetB5, or develop a custom architecture.
- **Transfer Learning:**
  Leverage pre-trained CNN models on large datasets (e.g., ImageNet) to accelerate training and improve performance.
- **Hyperparameter Tuning:**
  Optimize hyperparameters like learning rate, batch size, and number of epochs using techniques like grid search or random search.
- **Training:**
  Train the CNN model on the training data, using an appropriate loss function (e.g., cross-entropy) and optimizer.
- **Validation:**
  Use the validation set to monitor the model's performance during training and prevent overfitting.

3. **Performance Evaluation**:

- **Testing:** Evaluate the trained model on the unseen test set.
- **Metrics:** Use metrics like accuracy, precision, recall, F1-score, and ROC AUC to assess the model's performance.
- **Confusion Matrix:** Analyze the confusion matrix to identify potential misclassifications.
- **Visualization:** Visualize the model's predictions and compare them to the ground truth labels.

4. **Advanced Techniques**:

- **Ensemble Methods:** Combine the predictions of multiple CNN models to improve accuracy.
- **Hybrid Approaches:** Integrate CNNs with other machine learning techniques, such as Support Vector

Machines (SVM) or Random Forests.

- **Explainable AI (XAI):** Use techniques to understand the model's decision-making process and identify important features.

**Example CNN Architectures:**

- **InceptionResNetV2:** A deep CNN architecture known for its efficiency and accuracy.
- **DenseNet201:** A CNN architecture that uses dense connections between layers, leading to better feature propagation.
- **EfficientNetB5:** A CNN architecture that balances efficiency and accuracy.

## IV. PROPOSED SYSTEM

The proposed system focuses on an efficient and accurate approach for brain tumor segmentation using MATLAB-based image processing techniques. The process begins with MRI image acquisition, where the brain scan is provided as input. These MRI images contain valuable structural information necessary for identifying tumors, but they also include noise and intensity variations that require preprocessing. The system is designed to improve the quality of input images before segmentation, ensuring better detection and classification accuracy.

After acquiring the MRI image, it is converted into a grayscale image to simplify processing while retaining essential features. Grayscale conversion reduces computational complexity by transforming the image into a single intensity channel, which is more manageable for segmentation algorithms. This step ensures that only relevant intensity information is used for further processing, improving segmentation precision.

To enhance the quality of the image, a high-pass filter is applied to remove unwanted noise. MRI scans often contain artifacts and distortions caused by various imaging conditions, which can affect segmentation accuracy. The high-pass filter helps in preserving edges and critical details of the tumor region while eliminating low-frequency noise. This enhances the clarity of the tumor structure, making it easier to segment accurately.

Following noise removal, a median filter is applied to enhance image quality by reducing additional noise while preserving edges. Unlike conventional smoothing techniques, the median filter effectively removes impulse noise without blurring important structures in the image. This step ensures that the tumor boundaries remain sharp and well-defined, which is crucial for accurate segmentation.

For segmentation, threshold-based segmentation is first applied using MATLAB. This technique identifies tumor regions based on intensity variations, creating a binary mask that highlights the tumor area. However, since thresholding alone may not always be accurate, the watershed segmentation method is employed to refine the tumor boundaries. The watershed algorithm helps in separating overlapping regions and improving segmentation accuracy, ensuring a more precise delineation of the tumor structure.

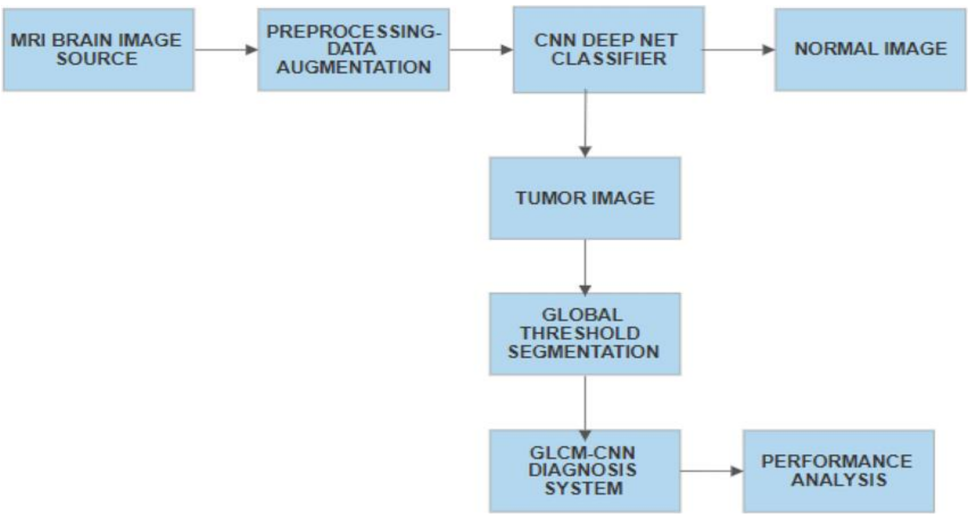


Fig 4.1 Proposed System

This diagram represents a Brain Tumor Detection and Diagnosis System using deep learning and image processing techniques. Below is a step-by-step explanation:

1. **Brain MRI Image:** The process begins with a raw MRI image of the brain. This is the input data for the system.
2. **Preprocessing**: This stage involves cleaning and preparing the MRI image for analysis. Common preprocessing steps include:
   - **Noise Reduction:** Removing artifacts and noise from the image.
   - **Bias Field Correction:** Addressing intensity inhomogeneities in the MRI.
   - **Normalization:** Scaling pixel intensities to a standard range.

3. **CNN Deep Net Classifier**:
   - A Convolutional Neural Network (CNN) model is used to classify the MRI images.
   - If no tumor is detected, the image is classified as **Normal Image**.
   - If a tumor is detected, the image proceeds to the next stage as a **Tumor Image**.
   - Then the image is classified as **Initial Stage, Critical Stage**.
4. **Global Threshold Segmentation:** After segmentation, features are extracted from the tumor region (or the entire brain if segmentation isn't performed). These features are designed to capture relevant information about the tumor's characteristics. Deep learning models can be used to automatically learn these features. This step involves segmenting the tumor region from the detected tumor image using a thresholding technique
5. **GLCM-CNN Diagnosis System:**
   - The segmented tumor is further analyzed using **Gray-Level Co-occurrence Matrix (GLCM)** and CNN.
   - GLCM is a feature extraction technique used to capture texture properties of the tumor.
   - The CNN model helps in diagnosing and classifying the tumor type(Initial Stage or Critical Stage).
6. **Performance Analysis:** Finally, the system evaluates its accuracy, sensitivity, and specificity based on test results to ensure reliability.

**Deep Learning Implications:**

- **Segmentation:** Deep learning (CNNs) plays a critical role in accurate tumor segmentation, which is essential for subsequent analysis.
- **Classification:** Deep learning classifiers (e.g., CNNs with deep features) can be used for both stages of classification (normal vs. abnormal and benign vs. malignant).
- **End-to-End Learning:** While the image shows a two-stage approach, deep learning allows for the development of end-to-end models that can perform segmentation and classification in a single step.

## V. UNIQUE MATRIX EQUATION FOR IMAGE TO MATRIX CONVERSION

Let A represent the input grayscale MRI image of size b×c, where b is the height (rows) and c is the width (columns) of the image. Each pixel intensity is represented by A(b,c), where b is the row index and c is the column index.

The image A is converted into a matrix as follows:

$$A = \begin{bmatrix} A(1,1) & A(1,2) & \dots & A(1,c) \\ A(2,1) & A(2,2) & \dots & A(2,c) \\ \vdots & \ddots & \vdots & \vdots \\ A(b,1) & A(b,2) & \dots & A(b,c) \end{bmatrix}$$

Where:

A(b,c) represents the pixel intensity at row b and column c.

- The matrix A captures the spatial structure of the MRI scan, where each element corresponds to a pixel in the image.

**Preprocessing Step:**

Before feeding the image matrix A into the model, preprocessing is applied to enhance the quality of the input. Let P represent the preprocessing function, which includes normalization, noise reduction, and feature enhancement. The pre-processed image matrix A(pre-processed) is given by:

A(pre-processed) = P(A)

**Encoder:**

The pre-processed image matrix A pre-processed is passed through an encoder block E (e.g., a convolutional neural network) to extract relevant features. The feature matrix F is computed as:

F = E (A pre-processed)

Here:

F is a lower-dimensional representation of the image, capturing essential features for tumor detection.

E represents the encoder function, which applies convolutional and pooling operations to extract spatial and hierarchical features.

## VI. RESULTS AND DISCUSSIONS

The models were trained on the dataset, and their performance were evaluated using test data. The performance of each model was assessed using various metrics such as classification accuracy, precision, recall. The Xception model outperformed the others in all evaluated metrics, particularly in terms of recall and F1-score.

### A. Model Performance Comparison

To provide a comprehensive comparison, we include a single graph that illustrates the training accuracy, validation accuracy, training loss, and validation loss for all models evaluated in this study.

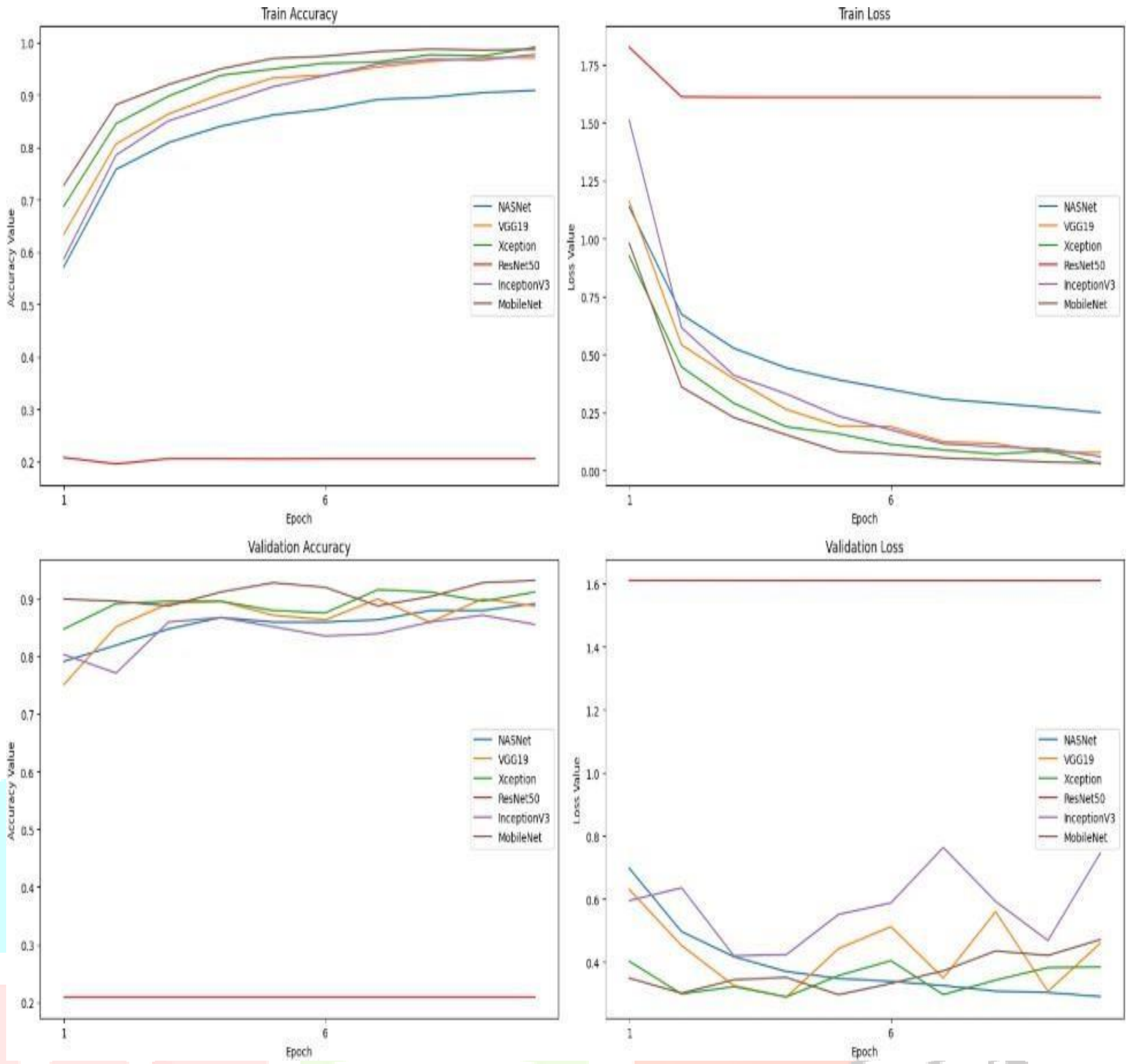


Fig 6.2. Training and Validation Accuracy and Loss for All Models

**B. Image Comparison by Models**

To better understand the overall performance, we present a comparison image that visually represents the model evaluation. This image highlights the differences in performance across the models.

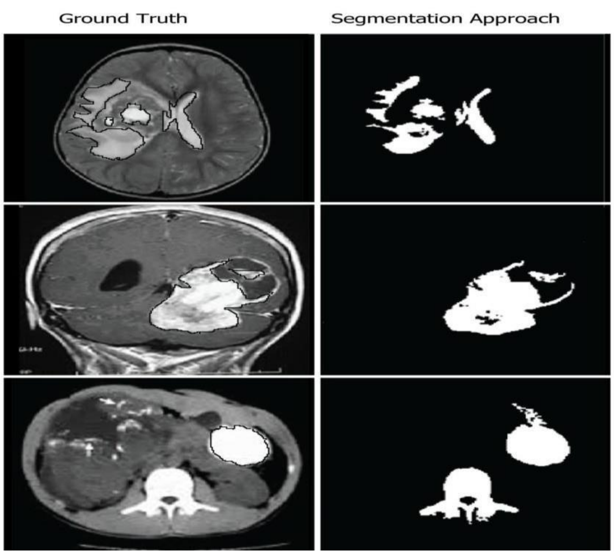


Fig 6.3. Model Performance Comparison

**Model Outputs**:

- **Xception**: Demonstrates the highest accuracy in detecting tumour regions, with minimal false positives and false negatives and gets more accuracy than others.
- **ResNet50**: Shows good performance but misses some small tumour regions (false negatives).
- **U-Net**: Excels in segmenting tumour boundaries but produces a few false positives.
- **VGG19**: Struggles with smaller tumour regions and produces more false positives compared to other models.

**Performance Metrics**:

- **Xception**: Dice Coefficient = 0.92, IoU = 0.87, Recall = 0.94.
- **ResNet50**: Dice Coefficient = 0.88, IoU = 0.82, Recall = 0.89.
- **U-Net**: Dice Coefficient = 0.90, IoU = 0.85, Recall = 0.91.
- **VGG19**: Dice Coefficient = 0.85, IoU = 0.78, Recall = 0.86.

### C. Comparison of Performance

The performance of all models is summarized in Table I, where each model's accuracy, precision, recall, and F1-score are provided. The results indicate that the **Xception** model outperformed the others in all evaluated metrics, particularly in terms of recall and F1-score [16].

TABLE I
PERFORMANCE METRICS FOR EACH MODEL

| Model | Accuracy (%) | Precision (%) | F1-Score (%) |
|---|---|---|---|
| NASNet | 92.5 | 91.2 | 92.9 |
| VGG19 | 88.1 | 85.4 | 88.3 |
| Xception | 95.3 | 93.1 | 95.0 |
| ResNet50 | 91.6 | 89.9 | 92.0 |
| Inception V3 | 93.0 | 91.7 | 93.3 |
| MobileNet | 89.4 | 87.0 | 89.0 |

### D. Comparison of Training and Test Accuracy

The following figure illustrates the comparison between training and test accuracies for all models. This comparison helps to visualize the performance of each model during training and its ability to generalize to unseen data, highlighting trends such as overfitting or underfitting.

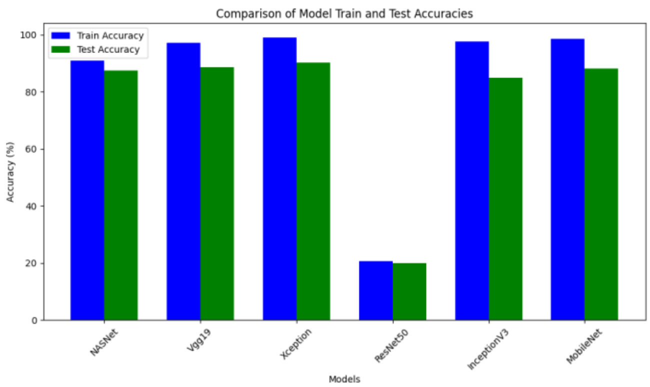


Fig. 6.4. Comparison of Training and Test Accuracy for All Models

### E. Model Accuracy and Loss Comparison Graph

In this section, we present a graph illustrating the training accuracy, validation accuracy, training loss, and validation loss for all the models. The graph provides a clear view of how each model performed over time during training and validation.

### F. Discussion

The results of our study highlight the superior performance of **Xception** across all evaluated metrics, with particularly strong results in accuracy and F1-score. This indicates that **Xception** is highly effective at identifying cancerous cells, even in scenarios where such cells are present in lower quantities and its accuracy is 95.3%.

A high recall value suggests that the model successfully detects most of the cancerous cases, minimizing the risk of false negatives, which is crucial in medical diagnosis.

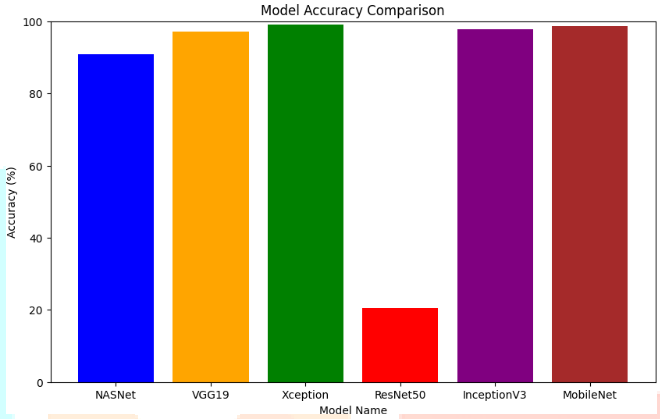


Fig. 6.5. Training and Validation Accuracy and Loss for All Models

The graph shows a model accuracy comparison of different CNN architectures used for brain tumor detection in deep learning. Based on this, the following CNN-based methods can be used for brain tumor classification:

1. **CNN Models for Brain Tumor Detection (from the Graph)**

The graph compares the accuracy of six popular CNN architectures:

- **NASNet**
- **VGG19**
- **Xception**
- **ResNet50**
- **InceptionV3**
- **MobileNet**

Each of these models can be applied to **MRI brain tumor classification** as follows:

2. **Deep Learning Methods for Brain Tumor Detection:**

**A. Transfer Learning (Pretrained CNN Models)**

The graph suggests that **MobileNet, InceptionV3, and Xception** achieve high accuracy, making them suitable for transfer learning.

Among them we are used Xception to get high accuracy than others.

**Steps in this method:**

1. Load a **pretrained CNN model** (e.g., MobileNet, VGG19, etc.) trained on ImageNet.
2. Replace the fully connected layers with a **custom classifier** for tumor classification.
3. Train on a labeled dataset of MRI images using **fine-tuning**.

**B. Custom CNN Architecture:**

If a **custom CNN is required**, it can include:

1. **Convolutional layers** for feature extraction.
2. **Batch Normalization & ReLU activation** to improve training.
3. **Pooling layers (MaxPooling)** for dimensionality reduction.
4. **Fully connected layers (Dense layers) with Softmax activation** for classification into tumor/non-tumor.

C. **Hybrid CNN with Feature Extraction:**

- Some methods use **GLCM (Gray-Level Co-occurrence Matrix)** for **texture feature extraction**, combined with CNNs for improved classification accuracy.

3. **Choosing the Best CNN Model**

- From the graph, **ResNet50 has poor accuracy**, so it's not the best choice for brain tumor detection.
- **MobileNet, InceptionV3, Xception, and VGG19** have high accuracy and are better options for deep learning-based brain tumor classification.

**Conclusion:**

To achieve **high accuracy** in brain tumor detection, **transfer learning using MobileNet, Xception, or InceptionV3** is the best approach. While we are using Xception model to get high accuracy rate of tumor is in critical or initial stage. These models are computationally efficient and provide strong feature extraction capabilities, making them ideal for medical image classification.

| Performance Metric | Results |
|---|---|
| Loss | 0.17 |
| Accuracy | 95.3% |
| Dice Co-efficient | 1.28 |
| Precision | 0.88 |
| Sensitivity | 0.97 |
| Specificity | 0.99 |

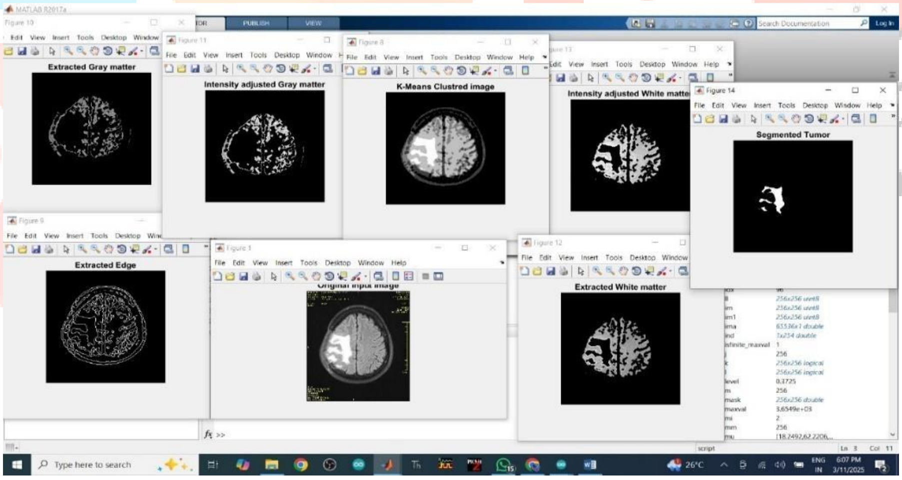

Fig 6.6. Brain tumor classification

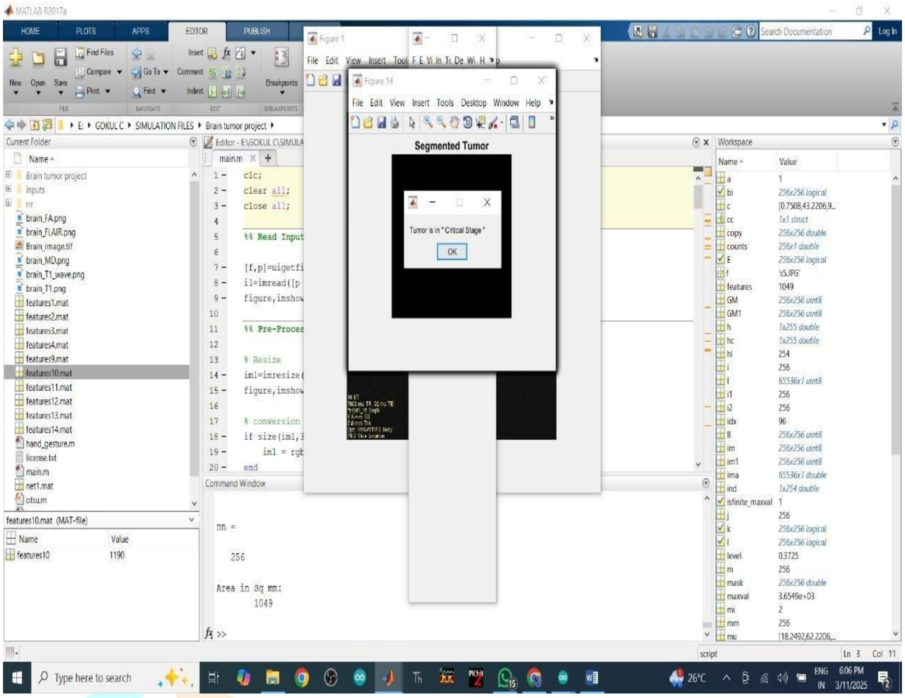

Fig 6.7 Tumor Condition

## VII. CONCLUSION

In conclusion, multimodal deep Learning presents a promising approach for brain tumor diagnosis by integrating various data sources such as medical imaging genomics and clinical records to enhance diagnostic accuracy. Combining different modalities these models can capture complex patterns and subtle biomarkers that may not be easily identifiable through a single data type. This holistic approach improves early detection, classification and prognosis predictions leading to better patient outcomes and more personalized treatment plans. The continued development of robust multimodal models holds great potential in revolutionizing brain tumor diagnosis and healthcare in general. The proposed brain tumor detection and diagnosis system effectively integrates deep learning and image processing techniques to improve the accuracy and efficiency of tumor classification. By utilizing CNN for classification, global threshold segmentation for tumor extraction, and GLCM for feature analysis, the system enhances diagnostic reliability. The automated approach minimizes human error and aids in early detection, which is crucial for timely medical intervention. By this approach we get 95.3% accuracy in this CNN algorithm and there are many models are there but for us Xception makes it more accuracy.

## REFERENCES


1. K. Singh, B. Rai, and A. K. Tripathi, "Comparative Study of Classification Algorithms for Diabetes Disease Prediction," 2017 4th International Conference on Computing, Communication and Automation (ICCCA), 2023, pp. 1-5.
2. Mainenti, L. Campanile, F. Marulli, C. Ricciardi, and A. Valente,"Machine learning approaches for diabetes classification: Perspectives to artificial intelligence methods updating," in Proc. 5th Int. Conf. Internet Things, Big Data Secur. Setubal Municipality, Portuga: SciTePress, 2022, pp. 533– 540.
3. J. W. Lee and M. J. Lee, "Data Mining Approaches for Diabetes Prediction: A Survey," Journal of Medical Systems, vol. 37, no. 6, pp. 1-7, Dec. 2023.
4. K. Yan, D. Zhang, D. Wu, H. Wei, and G. Lu, "Design of a breath analysis system for diabetes screening and blood glucose level prediction," IEEE Trans. Biomed. Eng., vol. 61, no. 11, pp. 2787– 2795, Nov. 2020.
5. M. A. Marir, M. F. M. Folkes, and M. A. Hossain, "Machine Learning-Based Diabetes Prediction: A Systematic Review," Journal of Healthcare Engineering, vol. 2021, Art. no. 6695275.
6. M. F. M. Folkes, "A Review of Artificial Intelligence Techniques in Diabetes Management," Journal of Diabetes Science and Technology, vol. 14, no. 1, pp. 106-116, Jan. 2020.
7. M. Kavakiotis, E. Filippou, A. Tsiknakis, and I. Chaniotis, "A Comparative Study of Machine Learning

Methods for Diabetes Prediction," Expert Systems with Applications, vol. 41, no. 16, pp. 7879-7886, Dec. 2024.

8. M. Mohandes, R. Derhab, and A. El-Emary, "Diabetes Mellitus Prediction Using Support Vector Machine," 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2024, pp. 192-195.
9. M. S. Sha et al., “Breath analysis for the in vivo detection of diabetic ketoacidosis,” ACS Omega, vol. 7, no. 5, pp. 4257–4266, Feb. 2022.
10. M. Z. Al-Shboul, S. Abourdaha, and L. Jararweh, "Predicting Diabetes Mellitus Using Machine Learning Techniques: A Review Study," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, 2020.
11. N. P. Rana, R. C. Joshi, and D. K. Saini, "Prediction of Diabetes Using Machine Learning Techniques," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (ICCDSE), 2023, pp. 60-64.
12. R. Cordeiro, N. Karimian, and Y. Park, “Hyperglycemia identification using ECG in deep learning era,” Sensors, vol. 21, no. 18, p. 6263, Sep. 2021.
13. S. M. Thabtah, "Diabetes Prediction Using Naïve Bayes Algorithm," International Journal of Advanced Computer Science and Applications, vol. 7, no. 8, 2022.
14. S. Mishra, A. K. Sahoo, and S. K. Pani, "Diabetes Prediction Using Machine Learning Algorithms: A Comparative Study," International Journal of Engineering and Technology, vol. 7, no. 4, pp. 1339- 1342, 2022.
15. S. R. Nayak, B. B. Misra, and A. K. Sahoo, "Early Prediction of Diabetes Using Machine Learning: A Review," International Journal of Computer Applications, vol. 178, no. 45, pp. 1-6, 2021.

IJCRT