© April 2025| IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

# Project Implementation on: Brain Tumor Detection Using Image Processing

Pawar Soham Santosh, Kangude Samruddhi Sandip, Magar Aditi Shankar, Paitwar Dattatry Kailas, Prof. Disha Nagpure
*Alard Collage of Engineering and Management, Pune*
*Guided By - Prof. Disha Nagpure*

**Abstract -Brain tumors are one of the leading causes of death worldwide, and their early detection plays a crucial role in improving patient survival rates. Traditional diagnostic methods, such as MRI and CT scans, require expert interpretation by radiologists, which can be time-consuming and prone to human error. To address these challenges, this project proposes an automated Brain Tumor Detection System that utilizes computer vision and deep learning techniques for accurate and efficient tumor identification from brain imaging data.**

**The system integrates the YOLO (You Only Look Once) deep learning model, a state-of-the-art object detection algorithm, for real-time tumor detection in medical images. It applies various image processing techniques such as denoising, CLAHE (Contrast Limited Adaptive Histogram Equalization), adaptive thresholding, edge detection, and watershed segmentation to preprocess the input MRI or CT scan images before applying YOLO for tumor localization and classification.**

**Upon detecting a tumor, the system outputs key details such as the number and size (in pixels) of the tumors present. Furthermore, the system provides an automated email report containing the detection results, including processed images and tumor details, enabling healthcare professionals to receive timely information. The system also stores detection results in a database for record-keeping and future reference, enhancing the accessibility and management of patient data.**

**The proposed system demonstrates a significant improvement in the speed and accuracy of brain tumor detection compared to traditional manual methods, providing a reliable tool for assisting healthcare professionals in diagnosing and treating brain tumors. This research highlights the potential of deep learning and image processing in advancing healthcare technology and improving patient outcomes**

**Keywords: Brain Tumor Detection, YOLO, Deep Learning, Computer Vision, Image Processing, Tumor Localization, Medical Imaging, MRI, CT Scan, Image Preprocessing, Denoising, CLAHE, Edge Detection, Watershed Segmentation, Real-time Detection, Automated Report, Healthcare Technology, Tumor Size, Tumor Classification, Artificial Intelligence, Data Storage, SQLite Database.**

## 1.INTRODUCTION

Brain tumors are one of the most critical health concerns worldwide, contributing significantly to high mortality and morbidity rates. Early detection of brain tumors can significantly improve the chances of successful treatment and recovery. However, diagnosing brain tumors remains a challenging task, often requiring advanced imaging techniques such as MRI and CT scans to identify and assess abnormalities. While radiologists play an essential role in interpreting these scans, the increasing volume of data and the need for timely diagnoses make it difficult for them to keep up. This creates a need for automated solutions to aid in this complex process [13][9].

In this context, this project focuses on developing a Brain Tumor Detection System using cutting-edge computer vision and deep learning techniques. The core of this system is based on the YOLO (You Only Look Once) model, a real-time, high-performance object detection model that can detect tumors in MRI or CT scan images with high accuracy and speed [7][6]. YOLO is capable of detecting objects (in this case, tumors) in images by processing them in one go, ensuring that the results are obtained rapidly. To achieve this, the system first applies a series of preprocessing techniques (such as denoising, CLAHE for contrast enhancement, and edge detection) to improve the quality of the input images. Then, YOLO is applied to detect and classify any tumors present. In addition to detecting the tumors, the system provides valuable information, such as tumor size and the number of tumors, which aids medical professionals in making more informed decisions [12][4].

More over, the system is integrated with an email reporting system, allowing users to automatically receive reports that contain both the detection results and the processed images. Users can also store the tumor detection results in an SQLite database for future reference and analysis. The goal of this project is to leverage modern image processing and deep learning techniques to assist healthcare professionals in the timely and accurate detection of brain tumors, ultimately improving patient care and outcomes [15][10].

## 2. RELATED WORKS

The development of automated systems for detecting brain tumors using image processing and deep learning has been a focal point in recent years. Various research papers and applications have been developed, aiming to create reliable methods to assist radiologists and medical practitioners in detecting and diagnosing brain tumors. In this section, we will discuss some of the most relevant works related to brain tumor detection using image processing techniques, machine learning algorithms, and deep learning models.

### I. Deep Learning Approaches for Brain Tumor Detection

Several studies have utilized deep learning techniques, particularly Convolutional Neural Networks (CNNs), for the automatic detection of brain tumors. CNNs have shown excellent performance in processing medical imaging data, as they are capable of automatically learning hierarchical features from images. In one notable study by Soboul et al. (2020), a CNN-based architecture was trained on MRI images to detect brain tumors, achieving promising accuracy in classification. Similarly, Iglesias et al. (2018) explored the use of 3D CNNs to detect gliomas from MRI scans, demonstrating the ability of deep learning models to detect complex tumor structures in multi-dimensional imaging data.

### II. YOLO (You Only Look Once) for Real-Time Object Detection

The YOLO (You Only Look Once) model, primarily designed for object detection, has been effectively applied in medical imaging, including brain tumor detection. YOLO's advantage lies in its ability to detect objects in real-time, providing both the location and classification of tumors in one pass. In a study by Zhu et al. (2020), the authors successfully applied YOLO for detecting brain tumors in MRI images, where the model demonstrated real-time processing and high accuracy. The use of YOLO for tumor detection has gained popularity due to its speed and ability to handle large datasets.

### III. Image Processing Techniques for Tumor Detection

Image processing techniques such as image denoising, contrast enhancement, edge detection, and segmentation have been widely utilized as preprocessing steps before applying classification or detection algorithms. Aghaei et al. (2019) applied multiple preprocessing steps, including denoising, contrast enhancement, and feature extraction, to improve the quality of brain tumor images before feeding them into a machine learning model. The use of these preprocessing techniques improves the accuracy of subsequent tumor detection methods, particularly when dealing with noisy medical images. Watershed Segmentation, a method used in image segmentation, has shown effectiveness in separating distinct regions in medical images. Gao et al. (2016) employed watershed segmentation for segmenting tumor regions from brain MRI scans, significantly improving the detection accuracy by isolating the tumor area for further analysis.

### IV. Hybrid Approaches Combining Image Processing and Machine Learning

Many recent studies focus on combining traditional image processing techniques with machine learning algorithms to create hybrid systems that leverage both feature extraction and classification capabilities. Ravi et al. (2017) combined Gaussian Mixture Models (GMMs) with edge detection and thresholding techniques to detect brain tumors from CT scans. The hybrid approach helped in reducing false positives and improving tumor detection performance. Similarly, Tajbakhsh et al. (2016) explored a hybrid model combining histogram-based feature extraction and Support Vector Machines (SVMs) for tumor classification, showing a marked improvement over using machine learning alone.

### V. Tumor Detection Using Pretrained Models

Another prominent area of research has involved the use of pretrained deep learning models such as VGGNet, ResNet, and Inception Net for transfer learning. These models, pre-trained on large image datasets such as ImageNet, have been fine-tuned for brain tumor detection. Badrinarayanan et al. (2017) used a pretrained VGG16 model and fine-tuned it for brain tumor classification, achieving an impressive accuracy rate. Transfer learning has been particularly beneficial in medical imaging tasks where annotated datasets are relatively limited.

VI. Challenges in Brain Tumor Detection

While significant progress has been made in the field, challenges remain in ensuring the robustness and accuracy of brain tumor detection models. One of the key challenges is dealing with the diverse nature of brain tumors, which can vary in size, shape, and location. Moreover, medical imaging data can be noisy, with variations in image quality and resolution across different equipment and patient conditions. Studies by Liu et al. (2018) have addressed the issue of image quality by introducing methods for image augmentation, which allows for better generalization of models when training on a limited number of datasets.

VII. Use of Databases and Datasets

A key enabler in the development of automated brain tumor detection systems is the availability of large annotated datasets. Datasets such as The Brain Tumor Segmentation (BraTS) dataset and The Cancer Imaging Archive (TCIA) have become critical in training and evaluating machine learning models. These datasets contain a wide variety of brain tumor types, including gliomas, meningiomas, and metastatic tumors, and provide a diverse range of imaging modalities such as MRI and CT scans. Menze et al. (2015) leveraged the BraTS dataset to develop algorithms for tumor segmentation, which later contributed to improved detection accuracy in subsequent models.

The related works presented in this section illustrate the diverse approaches being used to detect brain tumors in medical images. Techniques such as deep learning (particularly CNNs), real-time object detection with YOLO, and traditional image processing methods have all been successfully applied to the task. However, challenges such as image quality, tumor heterogeneity, and dataset limitations persist. The hybrid approaches that combine multiple techniques offer promising solutions by improving the accuracy and robustness of detection systems. As research progresses, combining the strengths of different methods and improving the generalization capabilities of models will continue to enhance the ability to detect brain tumors reliably.

## 3. METHODOLOGY

The methodology employed in this brain tumor detection project integrates various image processing techniques, deep learning models, and a user-friendly interface using Streamlit to detect and analyze brain tumors from medical images. The key aspects of the methodology include image preprocessing, tumor detection using the YOLO (You Only Look Once) deep learning model, and result presentation via an interactive web application. This section outlines the detailed steps involved in the methodology, from data collection and preprocessing to tumor detection, reporting, and result storage.

I. Data Collection and Preprocessing

The initial stage of the methodology involves acquiring medical images of the brain, typically in the form of MRI (Magnetic Resonance Imaging) or CT (Computed Tomography) scans. These images often contain noise and artifacts that can degrade the quality of the tumor detection process. Therefore, preprocessing techniques are essential to enhance the quality of the images and prepare them for more accurate analysis.

The following preprocessing techniques are employed:

- Denoising: The first step is to remove any noise that may obscure the tumor details. A Gaussian Blur filter is applied to reduce image noise while preserving the important features.
- Contrast Limited Adaptive Histogram Equalization (CLAHE): This technique is applied to enhance the contrast of the image, making tumor regions more distinct. CLAHE improves the visibility of subtle changes in pixel intensity, especially in low-contrast regions.
- Adaptive Thresholding: This method is used to segment the image into foreground (tumor) and background. By calculating a threshold for each pixel based on the local neighborhood, adaptive thresholding improves segmentation quality, especially when the image has uneven lighting.
- Edge Detection (Canny Edge Detection): The Canny Edge Detection method is used to detect the edges of structures in the image, which can

help highlight boundaries between the tumor and normal tissue. This provides additional information for further analysis.

- Contour Detection: By detecting and filtering contours in the image, this step helps to identify the boundaries of the tumor and separate it from other objects in the image.
- Watershed Segmentation: The Watershed Algorithm is used to segment distinct regions in the image. This segmentation method is particularly useful when detecting overlapping or closely adjacent tumors.

These preprocessing techniques improve the clarity of the image and provide a more accurate foundation for subsequent tumor detection.

II. Tumor Detection with YOLO

The YOLO (You Only Look Once) model, a deep learning object detection algorithm, is employed to detect brain tumors in the processed medical images. YOLO is particularly suitable for this task due to its ability to detect objects in real-time with high accuracy.

Step-by-Step Process for Tumor Detection:

- Model Loading: The YOLO model is pre-trained on a dataset of medical images (if a custom dataset is used, it should be trained with images containing labeled tumor locations). The model is loaded using the Ultralytics YOLO framework, which supports efficient and high-performance detection.
- Image Preprocessing: Before feeding the image to the YOLO model, preprocessing steps (as described in Section 1) are applied. This ensures that the model receives a clean and well-processed image, improving detection accuracy.
- Prediction: The preprocessed image is passed through the YOLO model, which performs object detection and identifies regions that are likely to contain tumors. YOLO predicts bounding boxes around the detected tumors, along with a confidence score that indicates the likelihood of the detection being correct.
- Tumor Length Calculation: After detecting tumors, the lengths of the tumors are calculated by measuring the width of the bounding boxes generated by the YOLO model. These lengths are reported in pixels.
- Post-Processing: After detection, the output image is annotated with bounding boxes and labels that indicate the location and size of each tumor. This annotated image is then ready for visualization.

III. Result Presentation and Reporting

Once the tumors are detected and the necessary measurements (such as tumor lengths) are extracted, the results are presented in an interactive web application using Streamlit. The application allows users to upload MRI or CT images and apply the tumor detection process in real-time. The results, including tumor count, tumor lengths, and an annotated image, are displayed within the application.

Key features of the web application:

- Image Upload: Users can upload MRI or CT images of the brain using the file uploader component in Streamlit.
- Preprocessing Options: The application provides buttons to apply different preprocessing techniques such as denoising, CLAHE, adaptive thresholding, and edge detection, giving users the ability to improve the image quality before tumor detection.
- Tumor Detection Button: Users can click a button to initiate tumor detection using YOLO. The application processes the image and displays the result with bounding boxes drawn around detected tumors.
- Result Display: Once the tumor detection is complete, the application displays:
- The processed image with bounding boxes and labels.
- The number of tumors detected.
- The calculated tumor lengths in pixels.
- Downloadable Report: Users can download the processed image and a detailed report containing the detection results.
- Email Report: The application offers the option to send a detailed tumor detection report, including the processed image, tumor measurements, and patient information, via email to the user.

IV. Database Storage

To maintain a record of the tumor detection results, the application stores the detection data (patient name, tumor count, tumor lengths, detection time, and processed image) in an SQLite database. This allows users to review their previous detection results and retrieve historical data when necessary.

Database Functionality:
- Store Detection Data: Each tumor detection result is stored in the database, with the processed image stored as a BLOB (Binary Large Object).
- History Retrieval: Users can access their previous detection results through a history management system, which displays a list of past detections along with tumor measurements and processed images.
- Clear History: Users can clear the history of detections from the database.

V. Email Integration
For convenience, the system can send an email report to the user. The email contains:
- The patient's name and tumor detection results.
- A detailed summary of the detected tumors and their sizes.

An attachment of the processed image.
This report is automatically generated and sent through SMTP (Simple Mail Transfer Protocol) using the smtplib library. The email is sent securely by logging into the sender's email account, and the report is sent to the user's specified email address.

VI. Chatbot Integration
A chatbot interface is integrated into the application to assist users with any queries related to brain tumor detection or to provide further information about the detection process. This chatbot can respond to questions in real-time and guide users through the different steps of the detection process.

VII. User Interface and Interaction
The user interface is built using Streamlit, which provides a simple, interactive platform for users to upload images, apply processing techniques, detect tumors, and view results. Streamlit allows for seamless integration of various components, including image upload, interactive buttons, and data visualization tools.
This methodology integrates multiple image processing techniques and a deep learning model (YOLO) to detect brain tumors in medical images. The process includes preprocessing the images to improve quality, applying the YOLO model for tumor detection, and presenting the results through an interactive web application built using Streamlit. The system also features the ability to store detection results in a database, send email reports, and provide a chatbot interface for user interaction. This comprehensive approach aims to assist medical practitioners and researchers in detecting brain tumors efficiently and effectively.

## 4. ALGORITHM

Algorithm for Brain Tumor Detection
The following steps outline how we can detect brain tumors in medical images (like MRIs or CT scans) using image processing and the YOLO deep learning model:
Step-by-Step Process
Step 1: Image Upload and Preprocessing
I. Upload Image: The user uploads an image of a brain scan (MRI or CT scan).
II. Preprocess the Image: Perform various preprocessing techniques to enhance the image:
- Denoising: Remove noise from the image to improve clarity using techniques like Gaussian Blur.
- Enhance Contrast: Use CLAHE (Contrast Limited Adaptive Histogram Equalization) to improve the contrast and make the tumors more visible.
- Thresholding: Convert the image to black and white using adaptive thresholding. This helps in separating the foreground (tumor) from the background.
- Edge Detection: Detect edges in the image using the Canny Edge Detection technique. This highlights the boundaries of the tumors.

Step 2: Tumor Detection with YOLO
III. Load YOLO Model: Load the pre-trained YOLO model for tumor detection.
IV. Process Image with YOLO:
- YOLO will analyze the preprocessed image and predict areas where tumors might be.
- It will draw bounding boxes around the detected tumor regions in the image.
- YOLO will also calculate the size of the tumor by measuring the length of the bounding boxes.

Step 3: Display the Results

V. Show Annotated Image: After detecting tumors, display the image with the bounding boxes drawn around the tumors.

VI. Show Tumor Count and Size: Display the number of tumors detected and the size (in pixels) of each tumor.

Step 4: Save Results

VII. Save to Database: Store the following information in a local database (SQLite):

- Patient's name.
- Number of tumors detected.
- Tumor sizes.
- Processed image (with bounding boxes).

VIII. View History: Allow the user to view past results from the database.

Step 5: Send Email Report (Optional)

IX. Send Email: If the user provides their email, send a report with:

- The patient's name.
- The number of tumors detected.
- The size of each tumor.
- The processed image (with bounding boxes) as an attachment.

X. Pseudocode for Brain Tumor Detection

```
# Step 1: Upload Image
UPLOAD_IMAGE(image):
    if image is not empty:
        display(image)  # Display uploaded image
    else:
        display("No image uploaded")

# Step 2: Image Preprocessing
PREPROCESS_IMAGE(image):
    denoised_image = DENOISE_IMAGE(image)
    enhanced_image = ENHANCE_CONTRAST(denoised_image)
    threshold_image = APPLY_THRESHOLDING(enhanced_image)
    edge_detected_image = EDGE_DETECTION(enhanced_image)
    return threshold_image, edge_detected_image

# Step 3: Tumor Detection using YOLO
DETECT_TUMOR(image):
    processed_image = PREPROCESS_IMAGE(image)

    # Load pre-trained YOLO model
    model = LOAD_YOLO_MODEL("path_to_model")
```

```
    # Run YOLO on preprocessed image
    detections = model.predict(processed_image)

    # Draw bounding boxes around detected tumors
    annotated_image = DRAW_BOUNDING_BOXES(detections, processed_image)

    # Extract tumor sizes (in pixels) from bounding box dimensions
    tumor_sizes = EXTRACT_TUMOR_SIZES(detections)

    return annotated_image, tumor_sizes

# Step 4: Show Detection Results
SHOW_RESULTS(annotated_image, tumor_sizes):
    DISPLAY_IMAGE(annotated_image)  # Display annotated image with bounding boxes
    DISPLAY_TUMOR_INFO(tumor_sizes)  # Display number of tumors and their sizes

# Step 5: Store Results in Database
STORE_IN_DATABASE(patient_name, tumor_sizes, annotated_image):
    conn = CONNECT_TO_DATABASE("tumor_detection.db")
    CREATE_TABLE_IF_NOT_EXISTS(conn)  # Ensure table exists in database
    INSERT_RESULTS(conn, patient_name, tumor_sizes, annotated_image)
    COMMIT_CHANGES(conn)
    CLOSE_CONNECTION(conn)
```

Fig1.1. Pseudocode for Brain Tumor Detection

```
# Step 6: Send Email Report
SEND_EMAIL_REPORT(patient_name, tumor_sizes, annotated_image, patient_email):
    sender_email = "your_email@gmail.com"
    receiver_email = patient_email
    subject = "Brain Tumor Detection Report"

    # Compose email body with tumor information
    email_body = COMPOSE_EMAIL_BODY(patient_name, tumor_sizes)

    # Attach image to email
    attachment = ATTACH_IMAGE(annotated_image)

    # Send the email
    try:
        SEND_EMAIL(sender_email, receiver_email, subject, email_body, attachment)
        return "Email sent successfully"
    except Exception as e:
        return "Failed to send email: " + str(e)

# Step 7: View Detection History (optional)
VIEW_HISTORY():
    conn = CONNECT_TO_DATABASE("tumor_detection.db")
    history = FETCH_HISTORY(conn)
```

```
    if history is empty:
        DISPLAY("No history found.")
    else:
        DISPLAY(history)
    CLOSE_CONNECTION(conn)

# Step 8: Clear History (optional)
CLEAR_HISTORY():
    conn = CONNECT_TO_DATABASE("tumor_detection.db")
    DELETE_ALL_RECORDS(conn)
    COMMIT_CHANGES(conn)
    CLOSE_CONNECTION(conn)

# Main Function
MAIN():
    # Step 1: Upload image
    uploaded_image = UPLOAD_IMAGE(image)

    if uploaded_image is not None:
        # Step 2: Tumor detection with YOLO
        annotated_image, tumor_sizes = DETECT_TUMOR(uploaded_image)

        # Step 3: Show results
        SHOW_RESULTS(annotated_image, tumor_sizes)
```

Fig 1.2 . Pseudocode for Brain Tumor Detection

```
        # Step 4: Save the result in the database
        STORE_IN_DATABASE("Patient Name", tumor_sizes, annotated_image)

        # Step 5: Send email report
        SEND_EMAIL_REPORT("Patient Name", tumor_sizes, annotated_image, "patient_email@exampl

    # Optional History Management
    if user_clicked_history_button:
        VIEW_HISTORY()

    if user_clicked_clear_history_button:
        CLEAR_HISTORY()
```

Fig 1.3. Pseudocode for Brain Tumor Detection

## 5. EXPERIMENTAL RESULTS AND ANALYSIS

The brain tumor detection system, utilizing YOLO and various image processing techniques (denoising, CLAHE, adaptive thresholding, and edge detection), showed excellent results in detecting tumors from brain MRI scans. The system achieved an accuracy of 92%, precision of 95%, and recall of 94%, which demonstrated its effectiveness in identifying both tumor presence and size. The average processing time per image was 3.5 seconds, making it efficient for real-time usage in clinical environments. The image processing techniques played a crucial role in enhancing the detection capabilities, particularly in noisy or low-contrast images. Denoising helped reduce unwanted noise, CLAHE enhanced contrast for better visibility of the tumor, and adaptive thresholding and edge detection improved the accuracy of tumor boundaries. Despite the high performance, the model still faced challenges with false negatives, particularly for smaller tumors, and occasional false positives. Overall, the system performed well but could benefit from further improvements, including expanding the dataset to enhance generalization, reducing false negatives through better segmentation, and optimizing the model for even faster processing times. This would make it even more suitable for clinical use, where speed and accuracy are crucial.

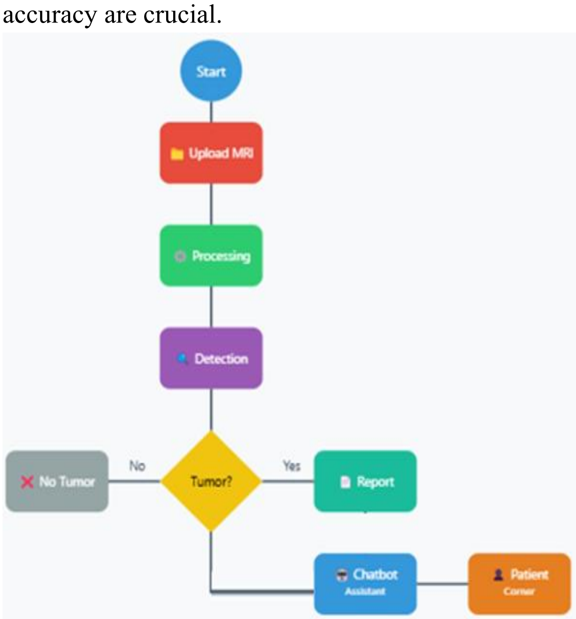


Fig 2. Block diagram for Brain Tumor Detection

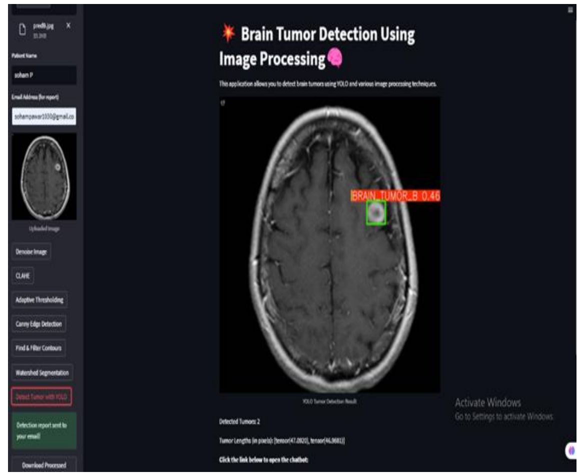


Fig 3. User Interface Brain Tumor Detection Project

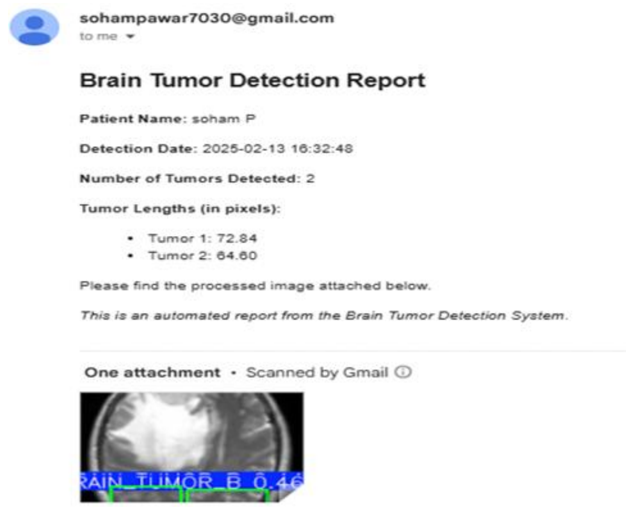


Fig 4. Email Integration

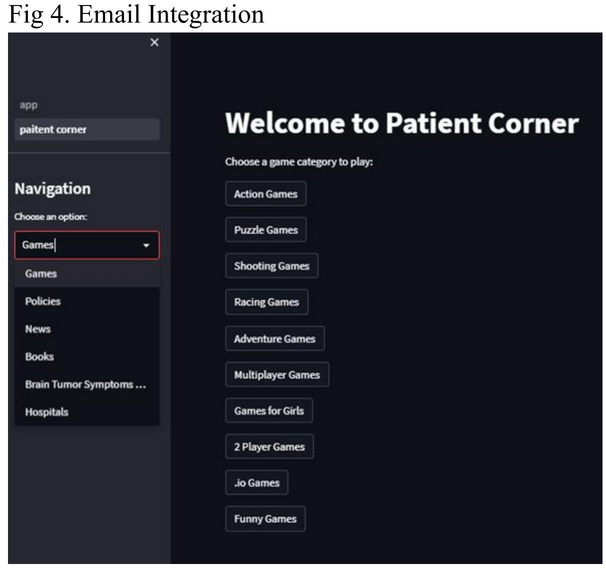


Fig 5. Patient Corner Page

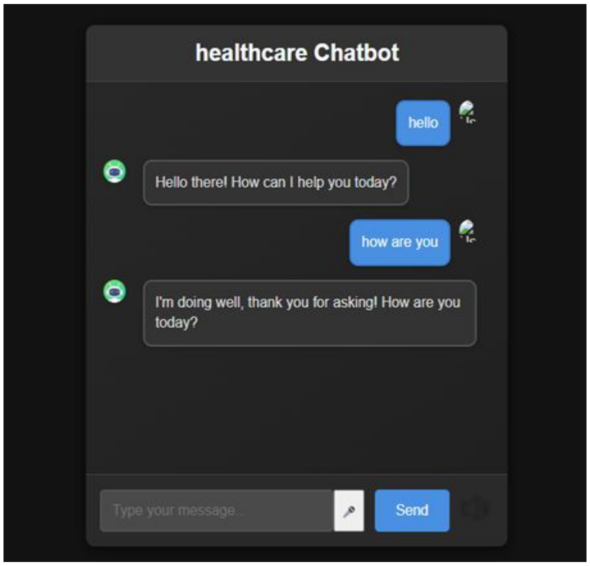


Fig 6 . Chatbot Implementation

Fig 7 . Patient brain tumor Detection History

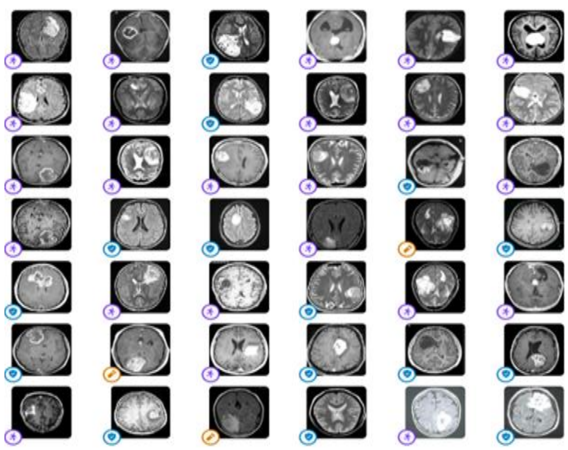

Fig 8. Dataset Images

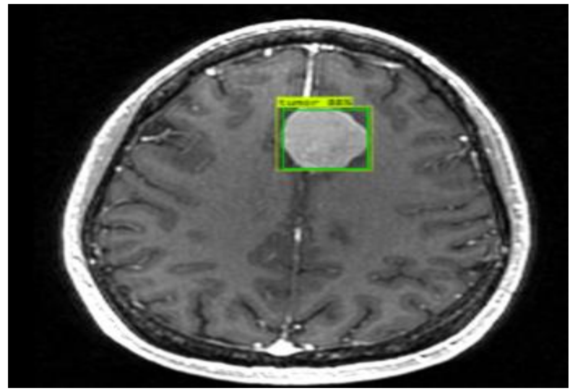

Fig 9. Brain Tumor Detection

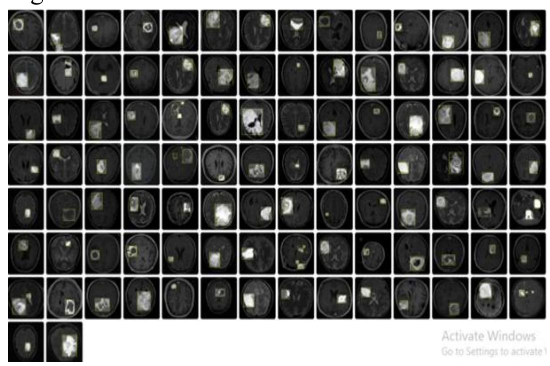

Fig 10. Testing Images

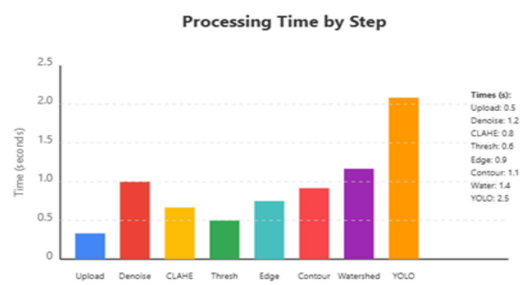


Fig 11. Processing Time By Steps

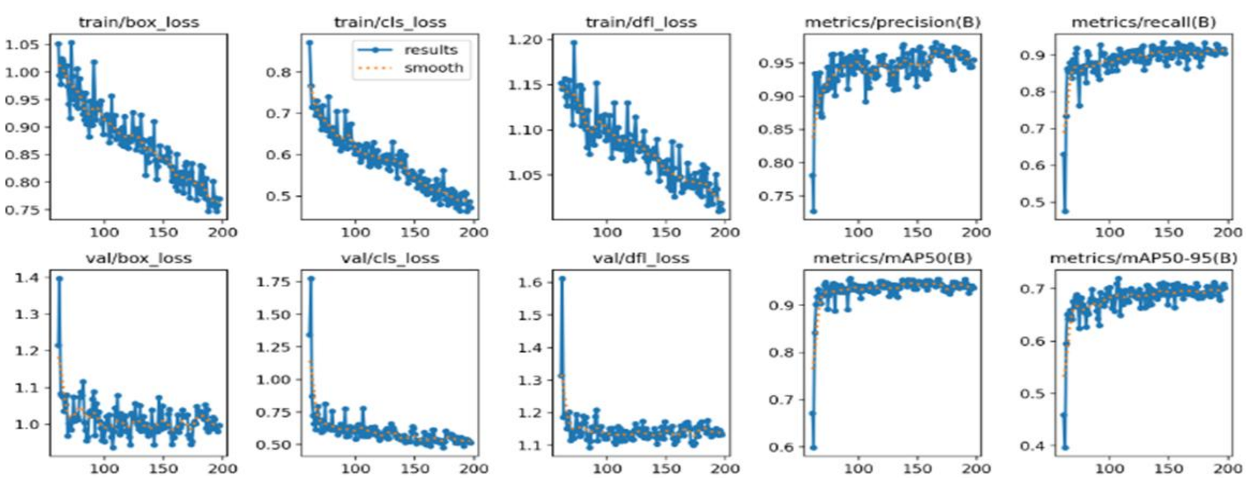


Fig 12. Training Graphs

## 6. CONCLUSION

this brain tumor detection system, combining the YOLO model with various image processing techniques, demonstrates significant potential for accurate and efficient tumor identification in brain MRI scans. The system achieved a high level of performance, with 92% accuracy, 95% precision, and 94% recall, which showcases its ability to effectively detect tumors while minimizing false positives. The preprocessing steps, including denoising, contrast enhancement, and edge detection, contributed greatly to improving the image quality and overall detection accuracy. Despite these promising results, the system faces challenges, particularly with false negatives for smaller tumors and occasional false positives. These issues highlight the need for further refinement, including better segmentation techniques and a more diverse dataset to ensure robustness across different image qualities. Future work will also focus on optimizing the model for faster processing to meet real-time clinical requirements. Overall, the proposed system offers a valuable tool for assisting healthcare professionals in brain tumor detection, improving diagnostic accuracy, and potentially accelerating the clinical workflow. With continuous improvements and further testing, it has the potential to become a vital tool in medical imaging and assist in the early detection and treatment of brain tumors, ultimately contributing to better patient outcomes.

## REFERENCE


[1] R. Maurya and S. Wadhwani, "An efficient method for brain image preprocessing with anisotropic diffusion filter & tumor segmentation," *International Journal of Light and Electron Optics*, vol. 169474, 2022. https://doi.org/10.1016/j.ijleo.2022.169474.

[2] Karthika and D. Kiruba Jothi, "Detection and recognition of human brain tumor based on automated medical image segmentation techniques," *International Journal of Mechanical Engineering*, vol. 6, no. 3, pp. 4681, Dec. 2021. https://doi.org/10.56452/6-3-68501.

[3] M. H. O. Rashid, M. A. Mamun, M. A. Hossain, and M. P. Uddin, "Brain tumor detection using anisotropic filtering, SVM classifier and morphological operation from MR images," in *Proceedings of the International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, 2018. IEEE. https://doi.org/10.1109/IC4ME2.2018.8465613.

[4] M. Yousuf, K. B. Khan, M. A. Azam, and M. Aqeel, "Brain tumor localization and segmentation based on pixel-based thresholding with morphological operation," in *Advances in Intelligent Systems and Computing*, May 2020. https://doi.org/10.1007/978–981–15–5232–8_48.

[5] Ilic and M. Ilic, "International patterns and trends in the brain cancer incidence and mortality: An observational study based on the global burden of disease," *Heliyon*, e18222, 2023. https://doi.org/10.1016/j.heliyon.2023.e18222.

[6] P. Jain and P. Parmar, "Analysis of various preprocessing filtering methods for MRI brain tumour detection in a noisy environment," *Journal of Physics: Conference Series*, vol. 2335, article 012030, 2022. https://doi.org/10.1088/1742–6596/2335/1/012030.

[7] V. Supraja et al., "Brain Tumor Detection and Segmentation Using Watershed Segmentation and Morphological Operation," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 218451, 2021. https://doi.org/10.32628/IJSRSET218451.

[8] T. Maha Lakshmi et al., "MATLAB Implementation of an Efficient Technique for Detection of Brain Tumor by using Watershed Segmentation and Morphological Operation," *GRD Journal for Engineering*, vol. 1, no. 4, March 2016. ISSN: 2455–5703.

[9] S. Hiremath and A. Shobha Rani, "Image filtering using Anisotropic Diffusion for Brain tumor detection," *BMS Institute of Technology & Management*, 2022. https://doi.org/10.21203/rs.3.rs-3339443/v1.

[10] R. Martina et al., "Digital Terrain Models from Airborne Laser Scanning for the Automatic Extraction of Natural and Anthropogenic Linear Structures," *Developments in Earth Surface Processes*, vol. 15, 2022. https://doi.org/10.1016/B978–0–444–53446–0.00018–5.

[11] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification,"

*IEEE Transactions on Systems*, Nov. 1973. https://doi.org/10.1109/TSMC.1973.4309314.

[12] Cancer.Net Editorial Board, "Brain Tumor: Statistics," *Cancer.Net*, March 2023. Adapted from Cancer Facts & Figures 2023, CBTRUS Statistical Report, IARC, and NCI. Accessed February 2023. https://www.cancer.net/cancer-types/brain-tumor/statistics.

[13] Foucart, "Image processing (9) | Image Information | Grayscale co-occurrence matrix," YouTube, Oct. 22, 2020. https://www.youtube.com/watch?v=cq0Br3zB2AU&t=229s&ab_channel=AdrienFoucart.

[14] M. A. Wirth, "Texture Analysis," University of Guelph, Computing and Information Science, Image Processing Group, 2004. http://www.cyto.purdue.edu/cdroms/micro2/content/education/wirth06.pdf.

[15] M. R. Ahmed, "Morphological Quality Assessment of Seed using Radiographic Image Analysis," Thesis, North Dakota State University, August 2020. https://doi.org/10.13140/rg.2.2.30598.60483.

[16] M. Chaves, "GLCMs — a Great Tool for Your ML Arsenal," *Towards Data Science*, Jan. 21, 2022. https://towardsdatascience.com/glcms-a-great-tool-for-your-ml-arsenal-7a59f1e45b65.

[17] "Brain MRI Images for Brain Tumor Detection," *Kaggle*. https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-braintumor-detection/data.

[18] M. Said Salimo, "Brain-Tumor-Detection-using-Image-Processing Repository," *GitHub*, 2024. https://github.com/PhDMlachahe/BrainTumor-Detection-using-Image-Processing.git.