Enhanced Brain Tumor Detection in MRI Using Scilab

Pranay Kumar Pandey[1]

[1]Dept. of Aerospace Engineering, Amity University Mumbai, Panvel, Maharashtra 410206

Abstract

Brain tumor detection using MRI images is a crucial step in medical diagnostics, where precise segmentation and classification significantly impact treatment planning. This study presents an optimized and efficient image processing pipeline for automated brain tumor detection using Scilab, incorporating advanced edge detection, noise removal, segmentation, and clustering techniques. The workflow begins with image preprocessing, where the MRI scan is converted to grayscale and enhanced through contrast adjustments and high-pass filtering. Edge detection is performed using Sobel, Prewitt, and Canny algorithms, enabling precise delineation of tumor boundaries. To mitigate impulse noise, a median filter is applied after an initial noise selection phase using an average filter. Tumor segmentation is achieved through thresholding techniques, which isolate abnormal regions by differentiating them from surrounding tissues. Additionally, K-means clustering is employed to segment distinct regions, refining the identification of tumor-affected areas.

The proposed methodology has been carefully optimized to ensure superior accuracy compared to traditional methods. Multiple algorithmic parameters, such as clustering iterations, edge detection thresholds, and noise filtering strength, have been fine-tuned to achieve precise and reliable segmentation results. The implementation is structured to provide a user-friendly graphical interface (GUI).

This approach enhances detection accuracy, computational efficiency, and visual clarity, making it well-suited for real-world medical applications. The system is modular and can be further extended with deep learning-based classification models to differentiate between malignant and benign tumors. Future work aims to integrate advanced clustering techniques and hybrid segmentation models to further improve detection precision and reduce false positives. The research not only contributes to automated medical imaging analysis but also lays the groundwork for scalable, AI-driven diagnostic systems.

Fig 1 : Flow Diagram

CHAPTER 1 : INTRODUCTION

Brain tumors are a major health concern worldwide, characterized by the abnormal growth of cells within the brain. They can be broadly classified into primary tumors, which originate from brain tissues, and secondary (metastatic) tumors, which spread from cancers located in other parts of the body. While some brain tumors are benign and may grow slowly, malignant tumors such as glioblastomas are aggressive and present significant treatment challenges, often leading to high morbidity and mortality. Early and accurate detection is therefore critical for effective treatment planning and improving patient outcomes.

Magnetic Resonance Imaging (MRI) has emerged as the imaging modality of choice for evaluating brain tumors due to its superior soft-tissue contrast, high spatial resolution, and the absence of ionizing radiation. MRI provides detailed anatomical information that facilitates the visualization of tumor boundaries, internal structures, and adjacent edema. Despite these advantages, the interpretation of MRI scans remains complex. Variability in tumor appearance, heterogeneity in intensity profiles, and the presence of imaging artifacts can complicate diagnosis. This challenge has motivated the development of advanced image processing techniques aimed at automating the detection and segmentation of tumor regions. Such methods typically involve preprocessing steps to improve image quality, including noise reduction and contrast enhancement, followed by edge detection and segmentation algorithms to delineate the tumor boundaries accurately.

Recent advancements in computational techniques have enabled the integration of methods such as median filtering, Contrast Limited Adaptive Histogram Equalization (CLAHE), and multiple edge detection operators (e.g., Sobel, Prewitt, and Canny). Furthermore, clustering algorithms like k-means are increasingly being applied to partition the image into distinct regions, isolating tumor tissue from healthy tissue based on intensity and texture features. These automated pipelines not only reduce the subjectivity associated with manual analysis but also provide a quantitative framework for monitoring tumor progression and response to therapy. In summary, the synergy between MRI and sophisticated image processing techniques has significantly enhanced our ability to detect, classify, and monitor brain tumors. This work focuses on the development of an automated pipeline in Scilab 6.1.1, emphasizing robust segmentation and classification methodologies. By leveraging these advanced techniques, the proposed system aims to improve the accuracy and efficiency of brain tumor detection, thereby contributing to more informed clinical decision-making and ultimately better patient outcomes.

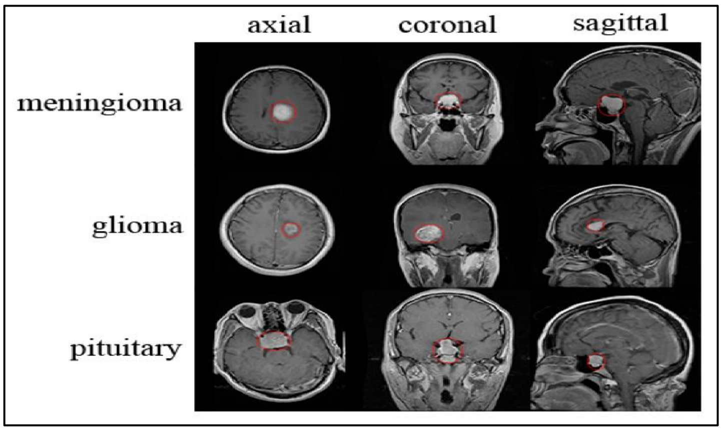


Fig 2 : Classification of Brain Tumor

CHAPTER 2 : PROBLEM STATEMENT

Brain tumors present significant challenges in medical imaging due to their varied morphology, unpredictable locations, and often low contrast against healthy brain tissues. Manual analysis of MRI images for brain tumor detection is highly dependent on the expertise of radiologists and is both time-consuming and prone to subjectivity. Differences in tumor size, shape, texture, and intensity across different patients further complicate diagnosis, increasing the risk of misinterpretation. Moreover, subtle tumors might be overlooked during visual inspection, delaying diagnosis and affecting treatment outcomes. Given the growing availability of MRI scans, there is a strong need for automated systems that can assist clinicians by providing consistent, accurate, and efficient preliminary tumor identification. An automated approach must address issues such as noise removal, contrast enhancement, accurate boundary detection, and reliable segmentation of tumor regions in complex brain structures.

This project specifically addresses the problem of developing a robust, Scilab-based automated system for detecting brain tumors in MRI images. The objective is to design an image processing pipeline capable of systematically improving image quality, detecting important features such as tumor boundaries, and segmenting abnormal regions with minimal manual intervention. The proposed solution incorporates advanced techniques such as median filtering for noise reduction, CLAHE for contrast enhancement, Sobel/Prewitt/Canny operators for edge detection, and segmentation algorithms like global thresholding and K-means clustering. The system aims to overcome common issues faced in MRI processing, such as impulse noise, variable brightness, and weak edge boundaries, thereby offering a reliable, cost-effective tool for early tumor detection. By creating an accessible and modular solution using open-source Scilab software, this project contributes to the broader goal of democratizing medical imaging analysis for resource-constrained settings.

CHAPTER 3 : BASIC CONCEPTS

Sobel Operator

The Sobel operator uses two 3×3 kernels to approximate image gradients in horizontal and vertical directions.

Horizontal kernel (Gx):

$$[-1 \;\; 0 \;\; +1]$$

$$[-2 \;\; 0 \;\; +2]$$

$$[-1 \;\; 0 \;\; +1]$$

Vertical kernel (Gy):

$$[+1 \;\; +2 \;\; +1]$$

$$[\;\; 0 \;\;\; 0 \;\;\; 0]$$

$$[-1 \;\; -2 \;\; -1]$$

$Gradient\ magnitude: G = \sqrt{(Gx^2 + Gy^2)}$

Prewitt Operator

Similar to Sobel but with uniform weights, used for edge detection.

Horizontal kernel (Px):

$$[-1 \;\; 0 \;\; +1]$$

$$[-1 \;\; 0 \;\; +1]$$

$$[-1 \;\; 0 \;\; +1]$$

Vertical kernel (Py):

$$[+1 \;\; +1 \;\; +1]$$

$$[\,0 \;\; 0 \;\; 0]$$

$$[-1 \;\; -1 \;\; -1]$$

$Gradient\ magnitude: P = \sqrt{(Px^2 + Py^2)}$

Canny Edge Detection

Multi-stage algorithm for edge detection:

1. $Gaussian\ filtering: I\_smooth = I * G(\sigma)$
2. $Gradient\ calculation\ using\ Sobel: G = \sqrt{(Gx^2 + Gy^2)}$
3. $Non - maximum\ suppression$
4. $Double\ thresholding: T\_low\ and\ T\_high$

5. $Edge\ tracking\ by\ hysteresis$

Thresholding

Basic binary thresholding converts grayscale image to binary using threshold

$$T: g(x,y) = \{ 1\ if\ f(x,y) \geq T\ 0\ if\ f(x,y) < T \}$$

K-Means Clustering

Partitions n observations into k clusters. Distance formula between point x and centroid μj:

$$d(x,\mu j) = ||x - \mu j||^2$$

Objective function to minimize: $J = \Sigma_{i=1}{}^{n} \Sigma_{j=1}{}^{k} w_{ij}||x_i - \mu_j||^2$

where $w_{ij} = 1$ if x_i belongs to cluster j, otherwise 0

Methodology

The current methodology has been adapted from [1]. Also, the test image was taken from the same reference. We have hardcoded i.e entered the path of image into the code.

Step 1: Take MRI image of the brain as an input.

Step 2: Convert it into an equivalent grayscale image.

Step 3: Apply filtering methods (median filter) for noise removal.

Step 4: Apply image enhancement techniques (CLAHE for contrast enhancement).

Step 5: Perform edge detection using Sobel, Prewitt, and Canny algorithms.

Step 6: Implement segmentation techniques:

- Apply global thresholding.
- Perform k-means clustering for tumor region detection.

Step 7: Integrate all steps into a unified pipeline and visualize results.

Step 8: Visualize and display the original MRI image, grayscale image, denoised image, and edge-detected images.

Step 9: Show the results of segmentation after median filter, CLAHE enhancement, and edge detection techniques.

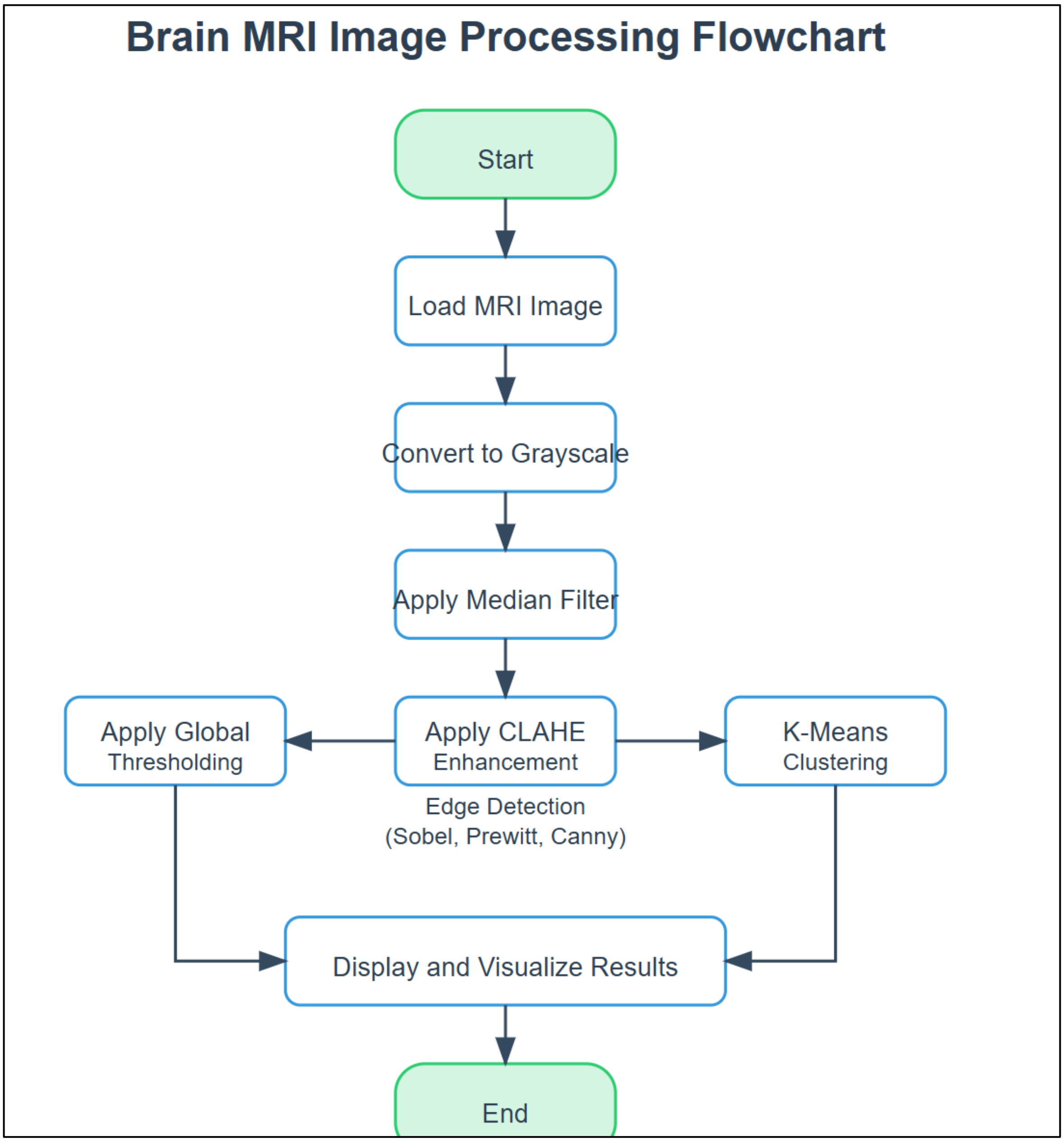
Step 10: Evaluate the tumor detection results based on the segmentation after k-means clustering.

Step 11: Visualize final predicted tumor regions.

Step 12: Measure execution time of the entire pipeline for performance analysis.

Step 13: Analyze and interpret the results, comparing against ground truth

CHAPTER 4 : FLOW CHART



CHAPTER 5 : SOFTWARE/HARDWARE USED

- Scilab 6.1.1 (Tested with 2025 release too!)
- IPCV Toolbox, SIVP Toolbox
- Windows 11 OS

CHAPTER 6 : PROCEDURE OF EXECUTION

1. Open Scilab 6.1.1
2. Load mainTumorDetectionPipeline.sce
3. Make sure IPCV toolbox installed
4. Run the script
5. Select MRI Image
6. Output GUI will display stages (grayscale, filtering, edge detection, etc.)

Code Implementation Pipeline

This code implements a comprehensive brain tumor detection pipeline using Scilab, focused on the analysis of MRI images. The pipeline follows a sequence of stages that preprocess and process the image data, perform various edge detection methods, and apply segmentation techniques to isolate potential tumor regions. The key steps include image loading, preprocessing, noise removal, enhancement, edge detection, and segmentation. Below is a breakdown of each stage:

Table 1 : Summary of Steps & Functions

S/N	Step	Description	Function Used
1	Image Loading	Load MRI image from a predefined file path, convert to double precision, and normalize pixel values.	loadImageHardCoded()
2	Grayscale Conversion	Convert the image to grayscale if not already in grayscale using the RGB channels.	rgbToGray()
3	Noise Removal	Apply a median filter to remove impulse noise from the grayscale image.	medianFilter()
4	Image Enhancement	Apply CLAHE to enhance local contrast in the image for better feature distinction.	claheEnhance()
5	Edge Detection	Perform edge detection using three methods: Sobel, Prewitt, and Canny.	sobelEdge(), prewittEdge(), cannyEdge()
6	Segmentation	Apply two segmentation methods: Global Thresholding and k-Means Clustering to isolate tumor regions.	thresholdSegmentation(), kmeansSegment()
7	Visualization	Display results in a 3x3 grid of images at various stages of the pipeline (Original, Grayscale, Filtered, etc.).	Scilab plotting functions
8	Final Output	Integrate and display all stages in the mainTumorDetectionPipeline() to visualize the tumor detection process.	mainTumorDetectionPipeline()

This code implements a comprehensive brain tumor detection pipeline using Scilab, focused on the analysis of MRI images. The pipeline follows a sequence of stages that preprocess and process the image data, perform various edge detection methods, and apply segmentation techniques to isolate potential tumor regions. The key steps include image loading, preprocessing, noise removal, enhancement, edge detection, and segmentation. Below is a breakdown of each stage:

GUI Window

The displayed GUI window showcases a multi-step image processing pipeline for brain tumor detection. It begins with the **original image**, converting it to **grayscale**, and applying **median filtering** for noise reduction. Contrast enhancement using **CLAHE** improves visibility, followed by edge detection techniques such as **Sobel, Prewitt, and Canny** to highlight tumor boundaries. **Global thresholding** and **k-means clustering** aid in segmentation, refining the tumor region. The final row displays **segmentation after thresholding** and the **final predicted tumor**, providing a clear, processed result. This systematic approach ensures accurate tumor identification while visually representing each step in the process.

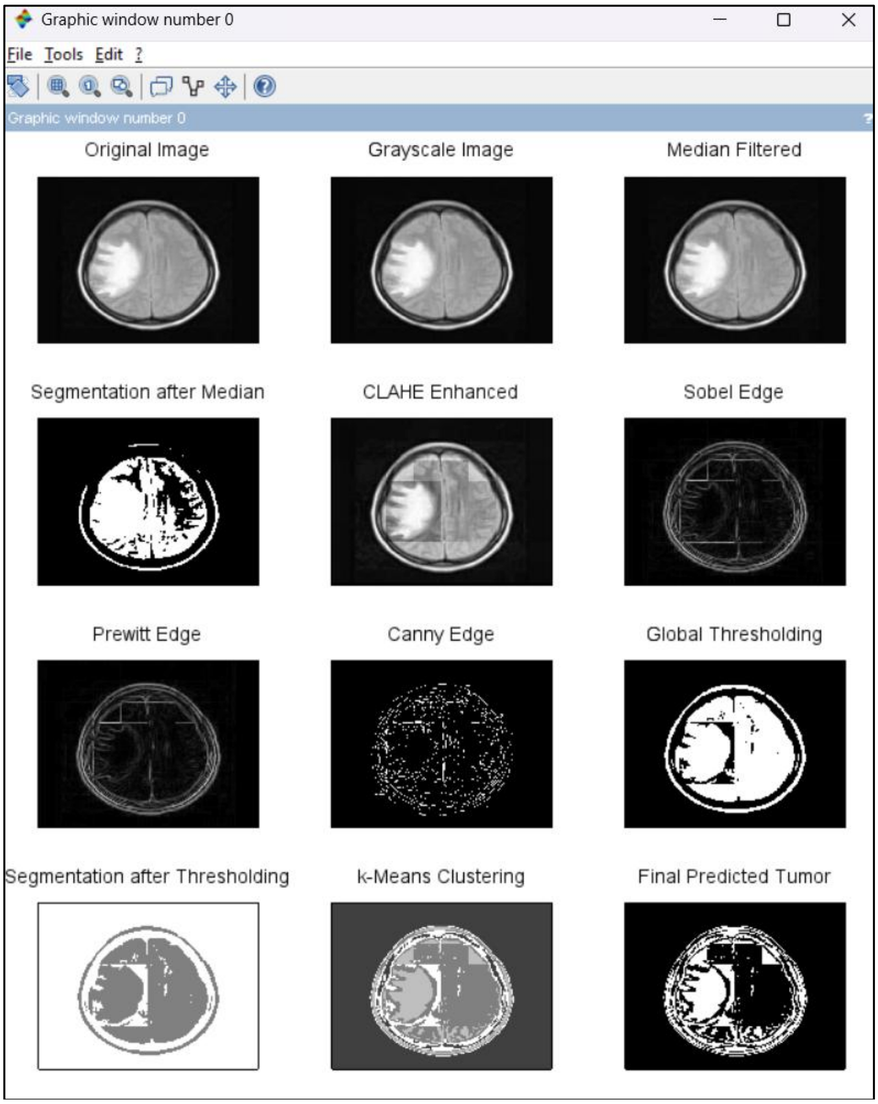


Fig 3: GUI Window of **test_png**

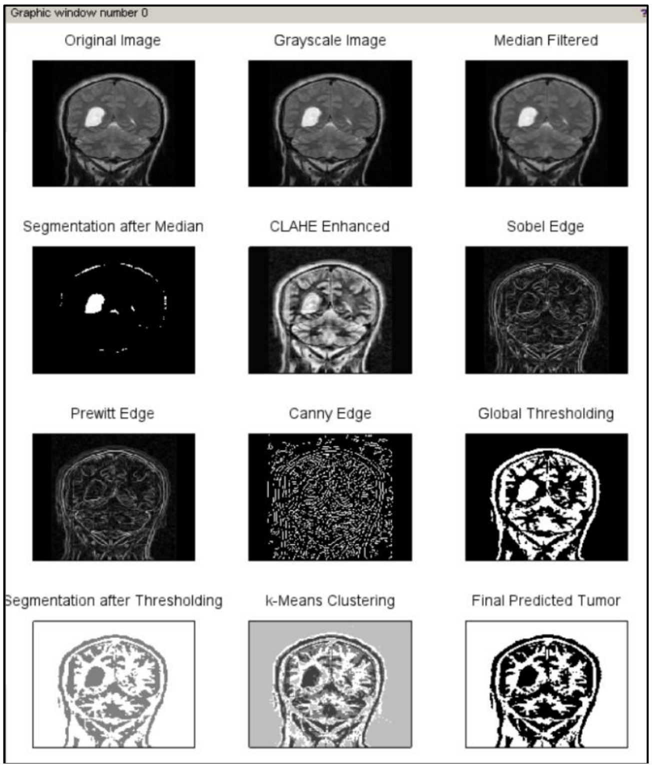


Fig 4 : GUI Window of **brain1_jpg**

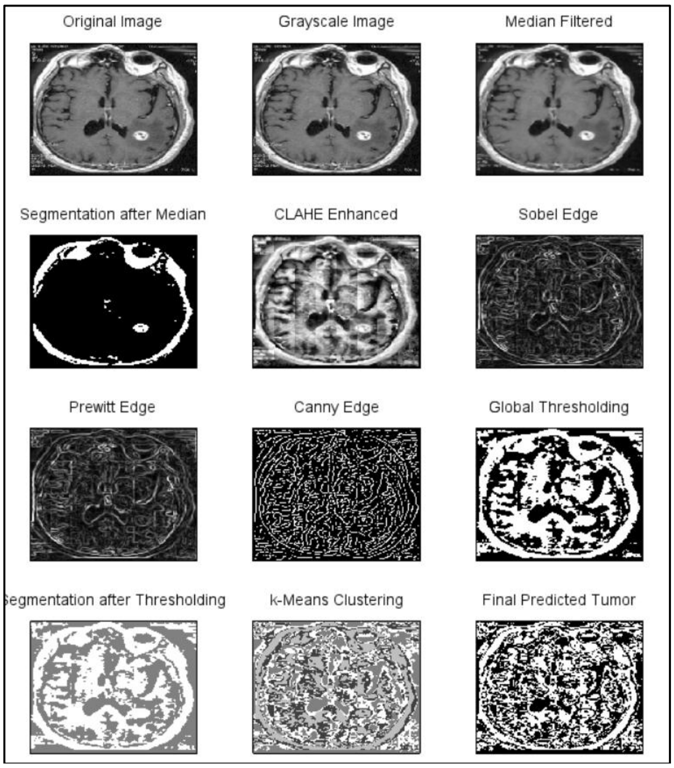


Fig 5 : GUI Window of **brain10_jpg**

Comparative Analysis

To evaluate the effectiveness of our proposed methodology, we perform a comparative analysis between the MRI image processing results obtained using Scilab and those presented in the reference research paper. The comparison is conducted based on key image processing stages, ensuring consistency in evaluation metrics and qualitative assessment. The original MRI images from the research paper are taken as reference, and the corresponding images are processed in Scilab following the proposed methodology. Through this comparative analysis, we assess the robustness and reliability of our approach while identifying potential improvements in Scilab-based MRI image processing for brain tumor detection.

The aim of this section goes beyond just validating accuracy. Comparative analysis also provides insights into the adaptability of different image processing techniques across various platforms. If Scilab's implementation produces results comparable to those in the research paper, it strengthens its credibility as a viable tool for medical imaging. Conversely, if discrepancies arise, they highlight the need for further optimization and refinement.

Fig. 3 is the reference MRI used both in the reference paper and current study. This consistency of reference image will help in better understanding of comparison.

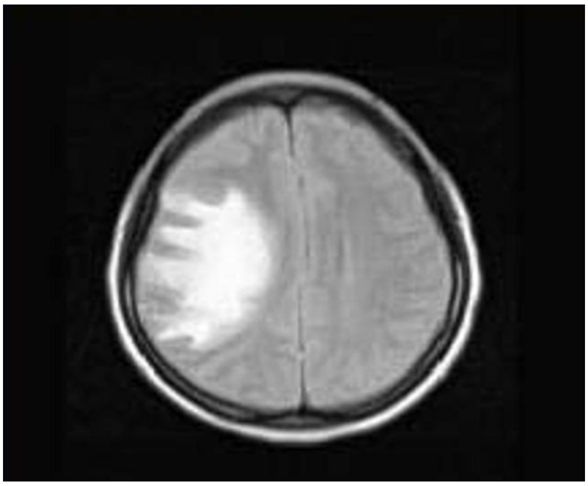
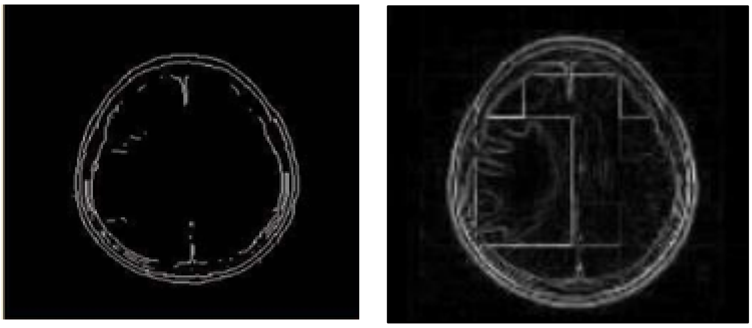


Fig 6 : Test Image

For simplicity purposes, all the images towards the left side of the screen denotes images fetched from the reference study. The right side of the screen aims to display the results obtained in Scilab. Also, IPCV / SIVP toolbox is used in Scilab 6.1 for smooth solving of the code. The time function has also been added to analyze the processing time it takes to run the code and give the output. The code was also run in Scilab 2023.x and Scilab 2025. Consistency and Similarity across the results was obtained.

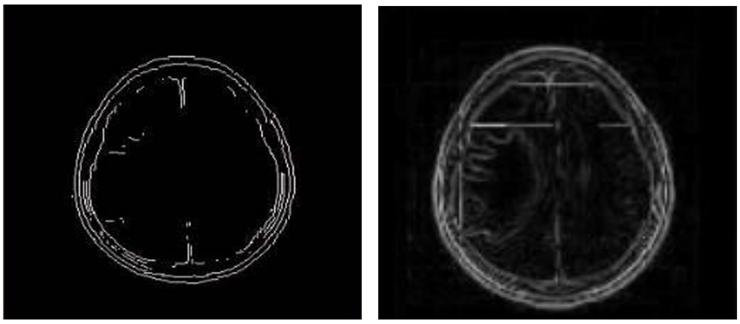
a) Sobel Edge Detection



Paper Scilab

The image from the reference paper demonstrates well-defined and clean edges with minimal noise, effectively capturing the essential contours of the object. In contrast, the image obtained from the Scilab implementation exhibits thicker edges with additional noise and artifacts in the background. The Scilab result also shows slight edge disconnections and unwanted details, possibly due to differences in preprocessing techniques, filtering methods, or thresholding parameters. To improve accuracy, noise reduction and fine-tuning of edge detection parameters in Scilab may be necessary.

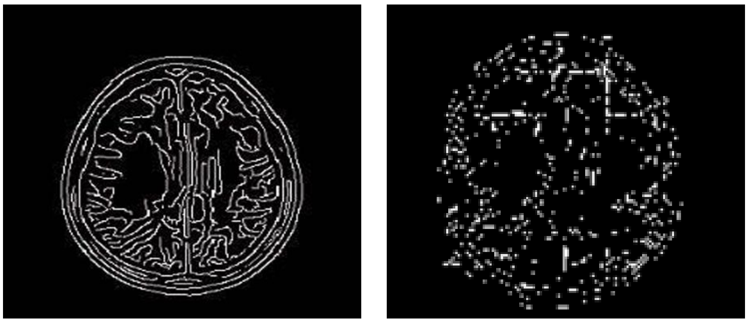
b) Prewitt Algorithm Edge Detection



Paper Scilab

For Prewitt Edge Detection, the reference paper’s result exhibits well-defined and smooth contours, effectively outlining the shape with minimal noise. In contrast, the Scilab implementation produces thicker edges, additional noise, and unwanted artifacts, particularly noticeable in the background and inner regions. The edges appear slightly blurred, which indicate differences in gradient calculation or thresholding.

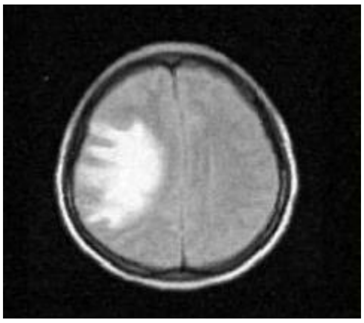
c) Canny Edge Detection



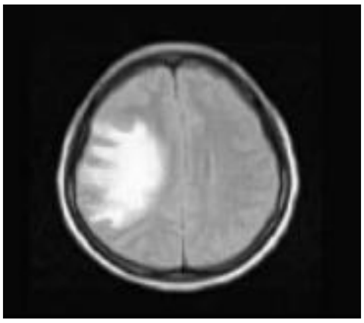
Paper Scilab

For Canny Edge Detection, the reference paper’s image captures intricate and detailed edges with high precision, preserving fine structures. However, the Scilab implementation shows significant degradation, with fragmented and noisy edges distributed irregularly. Many essential contours are missing, and the detected edges appear scattered, suggesting issues in Gaussian smoothing, gradient calculation, or non-maximum suppression.

d) Median Filter



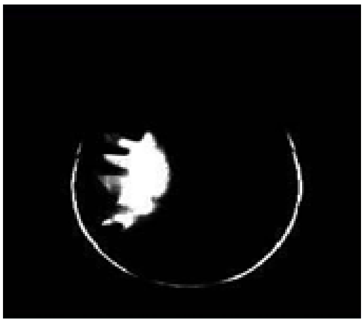
Paper



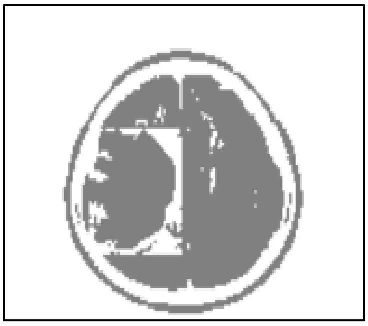
Scilab

A median filter is an image processing algorithm that removes noise from images. It works by replacing each pixel's value with the median value of the pixels in its neighborhoodThe median filter outputs are nearly identical. This is due to the very less noise in input image.

e) Segmentation after Thresholding Technique



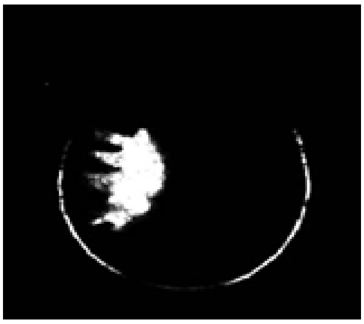
Paper



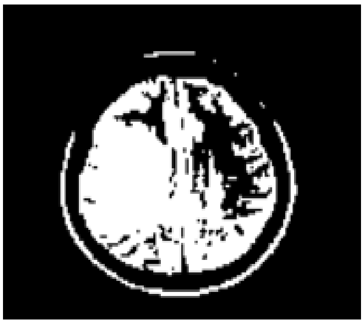
Scilab

In the Segmentation after Thresholding Technique, the reference paper produces a well-defined tumor region with minimal noise, ensuring clear boundary separation. However, the Scilab result appears to have an inconsistent grayscale background, suggesting issues with intensity normalization. It also seems there is contrast enhancement in the Scilab results. The tumor boundaries are less distinct, indicating a need for improved threshold selection. The background color is turned into white from black by the algorithm.

f) Segmentation after Median



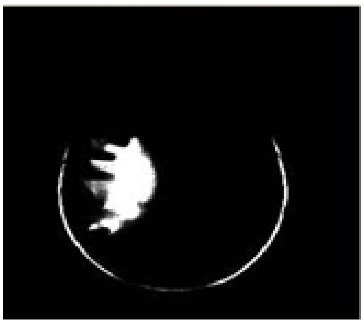
Paper



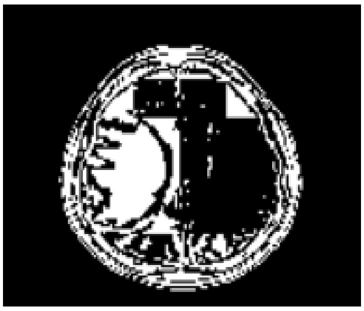
Scilab

Scilab implementation introduces additional artifacts and background distortions, likely due to variations in kernel size or preprocessing inconsistencies. The segmentation in Scilab appears less refined, potentially affecting the clarity of tumor identification.

e) **Final Predicted Brain Tumor**



Paper



Scilab

Finally, in the Final Predicted Brain Tumor, the reference paper retains a precise and well-extracted tumor region, demonstrating a robust and consistent approach. The Scilab result, however, still exhibits incomplete segmentation, and residual background artifacts. The tumor boundary lacks sharpness, suggesting that post-processing techniques such as morphological operations or better threshold tuning could enhance accuracy. This is bound to happen, as the pre-processing techniques involved above require significant fine tuning in first place.

CHAPTER 7 : RESULTS & DISCUSSION

It was observed that Scilab Code implementing gave outputs which deferred from the paper indicating more information is required. Optimisation in the source code was done for better resolution of the edges, yet no significant difference was achieved. It was observed execution time for **.jpg images** was between 10-20 seconds but for that of **.png image** 90-120 seconds.

The MRI image processing results obtained using Scilab and the reference research paper reveals several key differences, particularly in edge detection, segmentation, and final tumor prediction. While Scilab demonstrated reliable performance in noise reduction through median filtering, the edge detection methods, including Sobel, Prewitt, and Canny, produced results with thicker edges, added noise, and artifacts, indicating the need for improved preprocessing, noise reduction, and parameter fine-tuning. Similarly, segmentation results in Scilab exhibited inconsistent grayscale backgrounds and less distinct tumor boundaries, highlighting the need for better threshold selection and intensity normalization. The final tumor prediction in Scilab also showed residual background artifacts and incomplete segmentation, which could be improved with post-processing techniques such as morphological operations and refined threshold tuning. Overall, while Scilab offers promising results, further optimization of image processing techniques and parameter adjustments are required to match the precision and accuracy observed in the reference paper, ultimately enhancing its potential as a viable tool for brain tumor detection in medical imaging.

CHAPTER 8 : REFERENCES

1) Hazra, A. Dey, S. K. Gupta and M. A. Ansari, "Brain tumor detection based on segmentation using MATLAB," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017*, pp. 425-430, doi: 10.1109/ICECDS.2017.8390202.

2) Brain MRI Tumor Detection and Classification - File Exchange - MATLAB Central

3) MRI Brain Segmentation - File Exchange - MATLAB Central