UNIVERSITI TEKNOLOGI MARA

SEMESTER	**:**	**MARCH 2024 – AUGUST 2024**
COURSE	**:**	**IMAGE PROCESSING**
COURSE CODE	**:**	**CSC566**
PROJECT TITLE	**:**	**BRAIN TUMOR DETECTION**

Group No:	GROUP 7
Name, Student ID:	1.TUAN AHMAD WAFIQ BIN TUAN MAHMUD (2023388809)
	2.WALID BURHANI BIN MOHD ZAMANI (2023189445)
	3.ABDUL QHADIR JAILANI BIN AZHARI (2023395763)
Lecturer Name:	TS DR RAJESWARI RAJU

TEMPLATE FOR PROJECT GRADING

Each group should complete and attach this sheet. Submit along during the presentation, together with the report and softcopy.

ITEMS	FULL MARKS	MARKS
PRESENTATION: 1. System Prototype in GUI 2. Overall project presentation 3. Delivery skills	**30**	
REPORT: 1. Introduction 2. Objectives 3. Data collection 4. Flowchart 5. Results of Prototype 6. System Prototype in GUI 7. Sample Input Output (Coding) 8. Conclusion 9. References	**60**	
BONUS MARKS	**10**	
TOTAL MARKS	**100**	

Table of Contents

1. Introduction

Brain tumors, which are also known as intracranial neoplasms, occur abnormal cell growths within the brain. The brain is the most powerful element of our body. Tumors symptoms may include recurrent headaches and migraines. As time goes on, it may potentially cause visual loss. Currently, there is less research on the causes of this unusual development pattern. Tumors are classed based on whether they are malignant and where they originated. The non-cancerous type of the tumor is called benign. These are easily recognized and have limited development rates. Cancerous tumors are referred to be malignant. These are aggressive and difficult to identify, becoming a risk to life(Bansal et al., 2017).

Modern technology assists a lot of people in the medical profession. For example, image processing is critical in the identification and analysis of brain tumors, resulting in major improvements to the field of medical imaging. Brain tumors, which are abnormal cell growths in the brain or surrounding tissues, must be detected early and precisely identified in order to be treated effectively. Traditional diagnostic approaches, such as neurological tests and biopsies, can be intrusive and limiting, but medical imaging is a non-invasive option that gives extensive information about tumor features. Magnetic Resonance Imaging (MRI), and Computed Tomography (CT) are the primary imaging modalities applied to diagnose brain tumors(Ghorpade & Bhapkar, 2021).

This research aims to create a prototype for identifying brain tumors using modern image processing techniques. The technology will assist in early detection and precise diagnosis of brain tumors through improvements in picture quality, segmenting appropriate regions and extracting critical characteristics. Despite challenges regarding the complexity of the brain and incorporating new techniques into clinical practice, the initiative aims to improve medical imaging, resulting in improved patient outcomes and more effective management of brain tumors.

2. Objectives

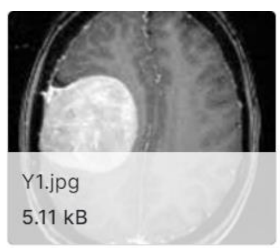
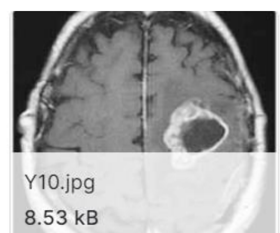
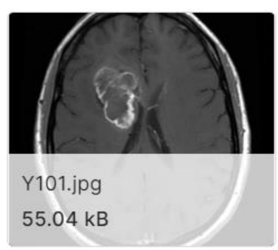
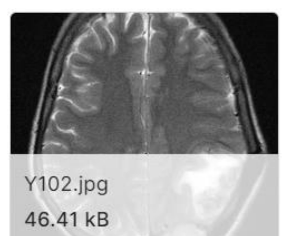
There are three main objectives in this project:

- To study and analyze the image processing techniques for brain tumor detection.
- To design and develop a prototype that can accurately detecting brain tumors as malignant or benign features.
- To evaluate the performance and accuracy of image processing techniques that were used in this prototype in detecting brain tumors.

3. Data collection

The data collection for this project involved collecting 30 images from the dataset that is available in Kaagle. This dataset includes two folders of images that contain brains that have a tumor and don't have a tumor. The names of the folders are yes and no. The total image combined in two folders is 30 .

This dataset relates to this study and really suitable, because the image of MRI brain that they collect were really clean and clear. More information for this dataset is depicted in the table below:

Category	Total images	Sample Data
Yes	15	   

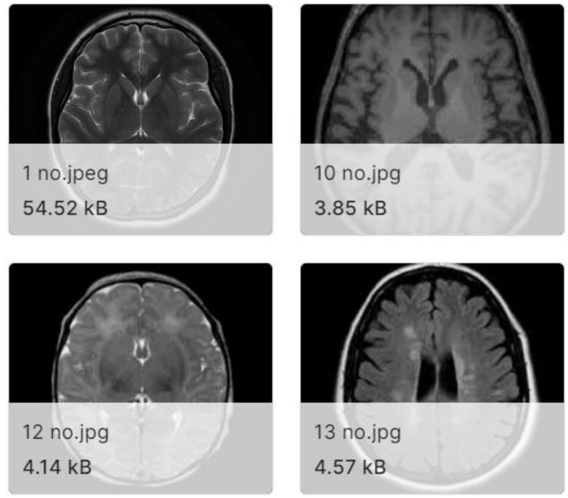
No	15	

Table 1: Data Collection

4. Flowchart

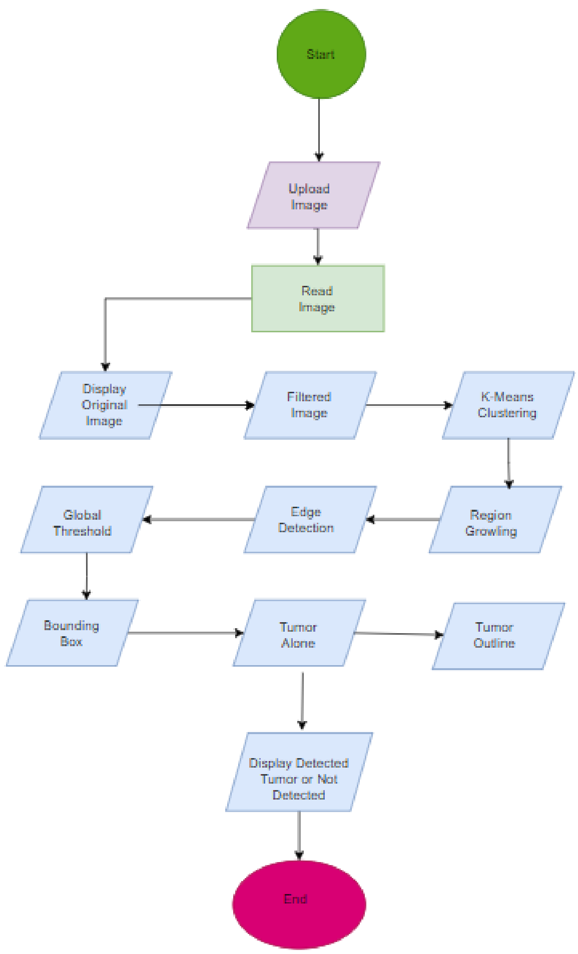
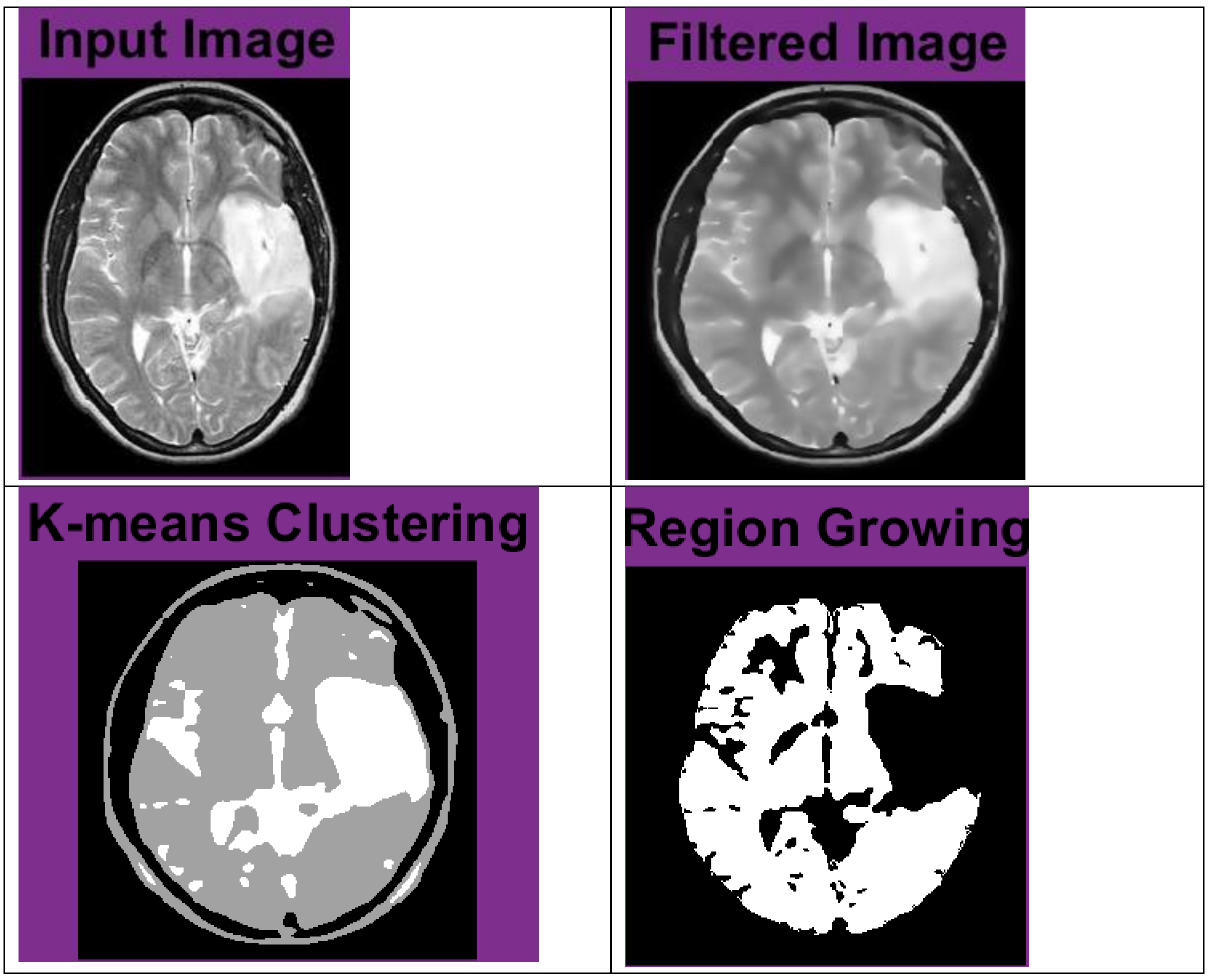


Figure 1: Flowchart of Brain Tumor Detectio

5. Results of Prototype

An encouraging findings have been observed in the brain tumor detection prototype, which demonstrates great accuracy in differentiating between cases of tumor and non-tumor. The prototype detects abnormal cell masses with great precision by combining machine learning algorithms learned on large datasets with advanced imaging methods like MRI. Preliminary experiments show that the system can interpret pictures quickly and give trustworthy diagnostic data, which might save the time needed for clinical decision making. With fast and accurate identification, this prototype's successful deployment represents a major advancement in early tumor detection and offers promise for better diagnostic accuracy and more efficient treatment planning. Figures 1 and 2 show the results of the tumor detection.

I. Brain with Tumor



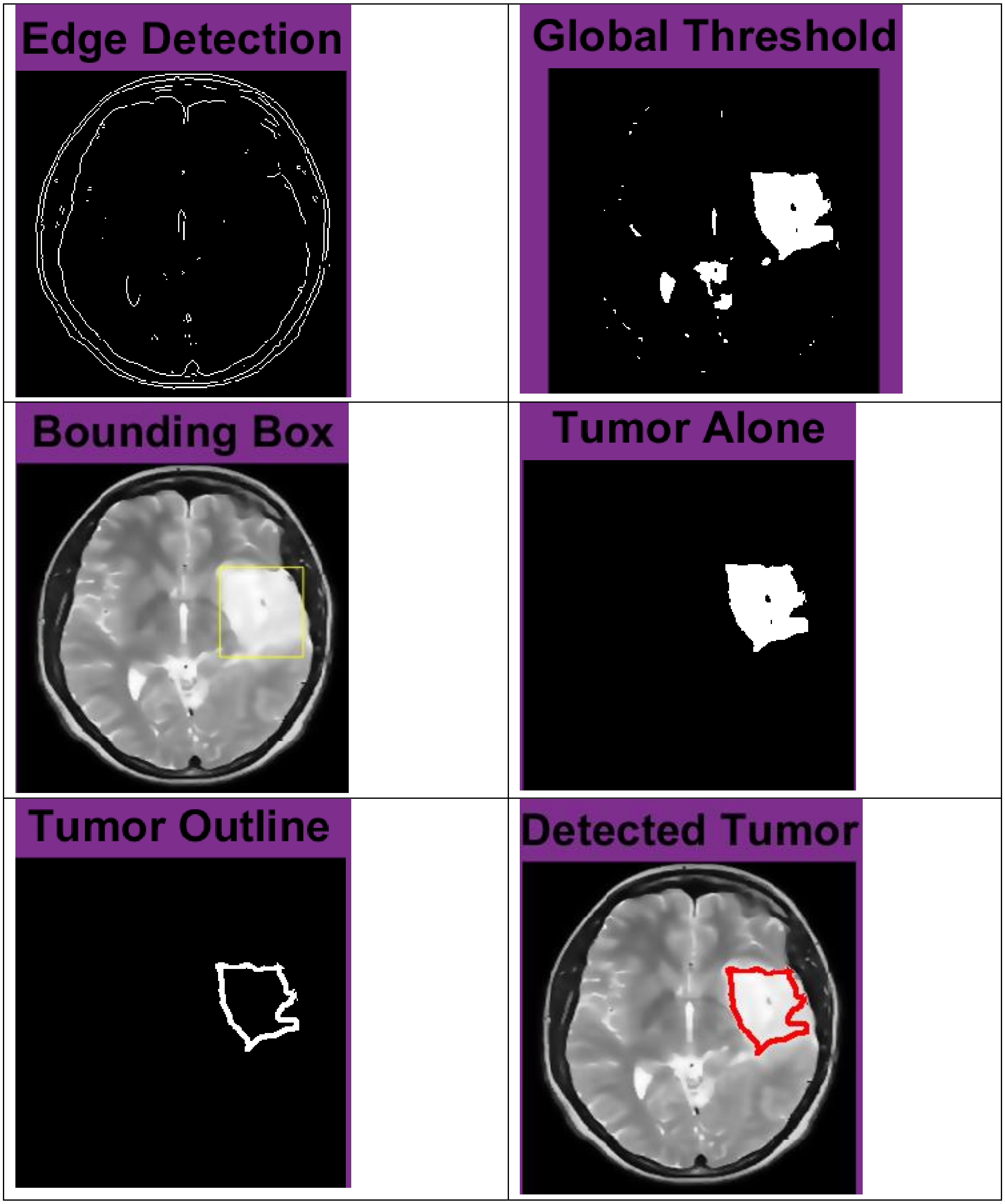
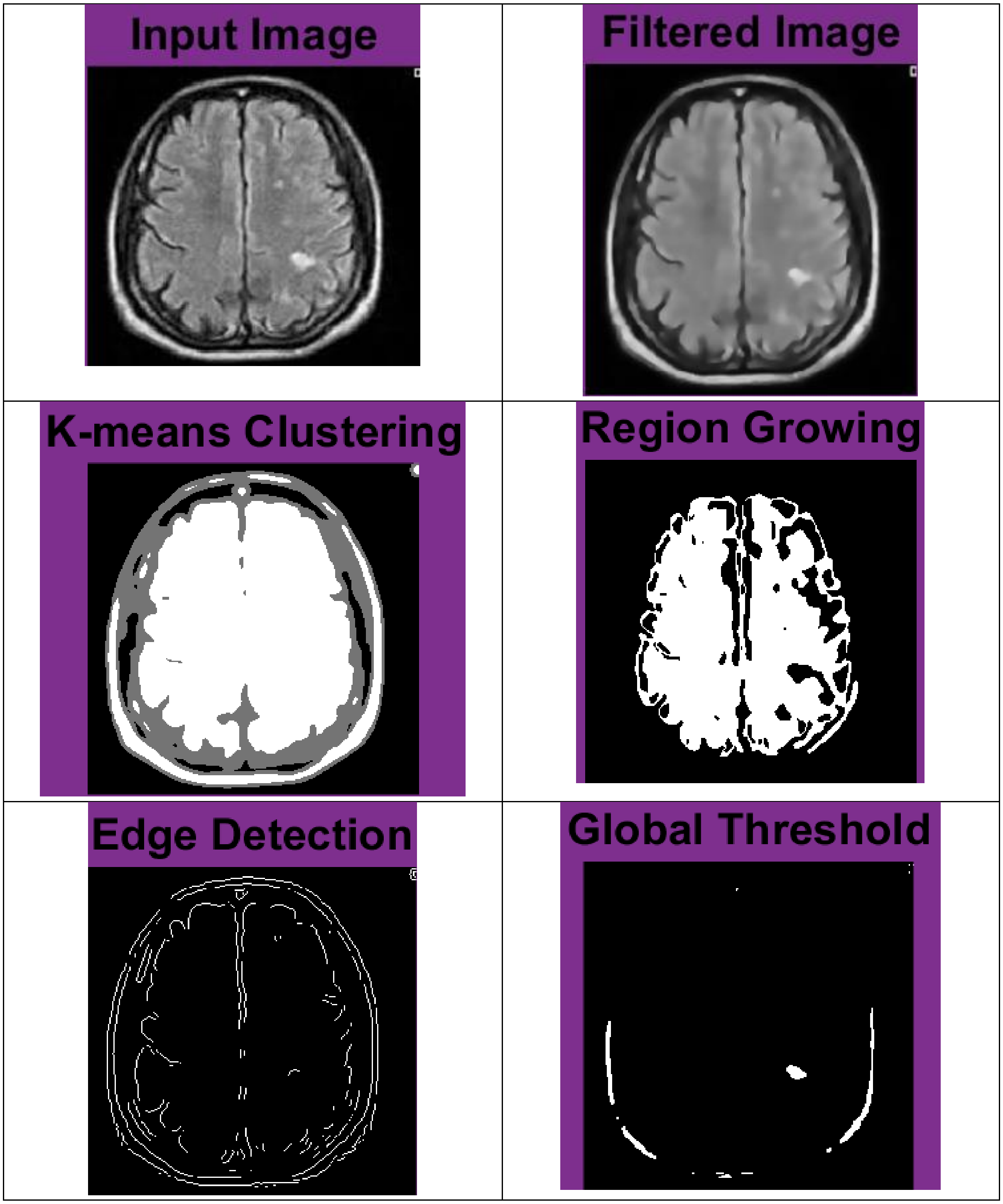


Figure 2: Brain with Tumor

ii. Brain without Tumor



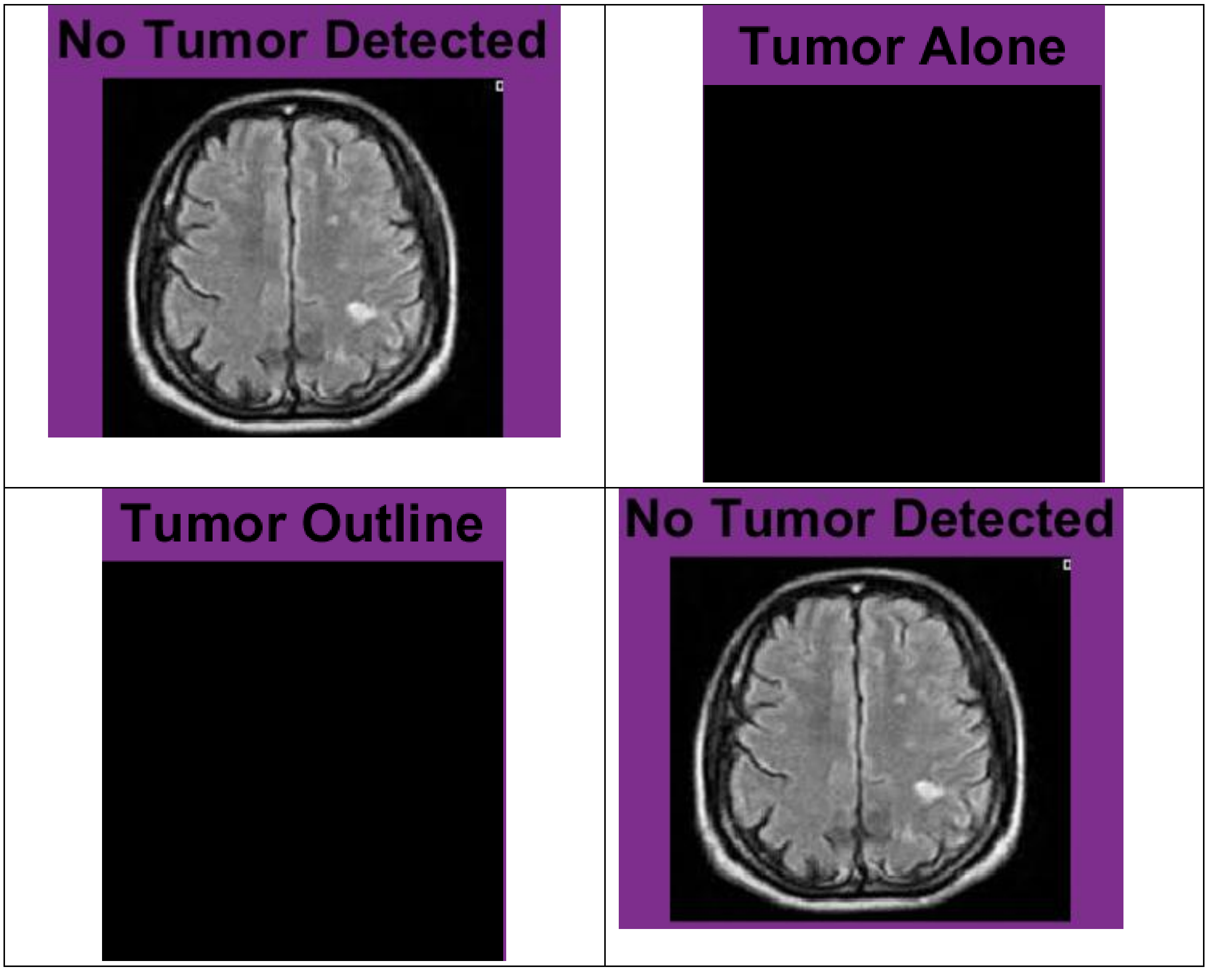


Figure 3:Brain without Tumor

6. System Prototype in GUI

A GUI (Graphical User Interface) system prototype is a preliminary model that shows the basic layout and operation of the user interface of a software program. It contains essential elements like reset buttons, and input fields that provide a visual of the design, typography, and fundamental interaction flow.

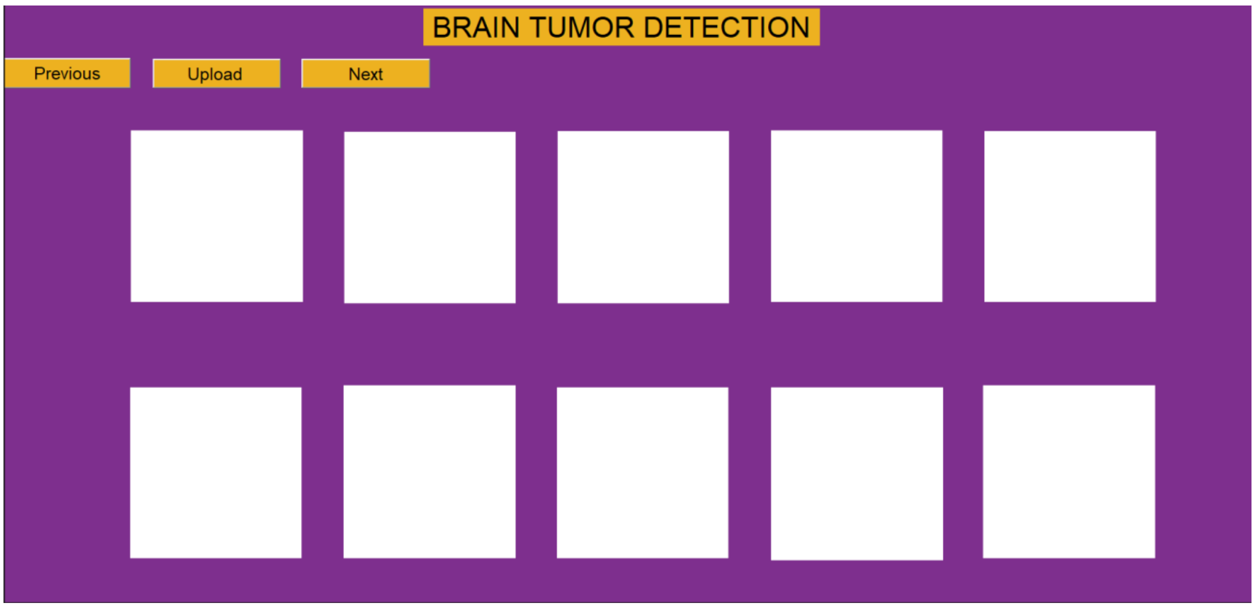


Figure 4:GUI 1

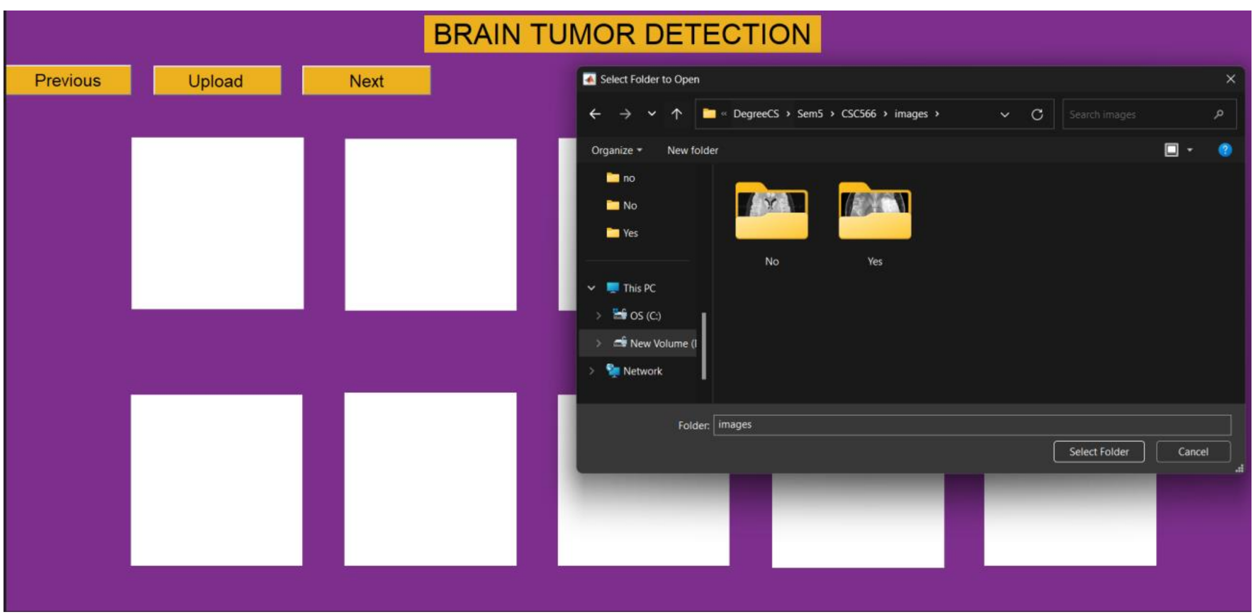


Figure 5:GUI 2

7. Sample Input Output of GUI

i. Brain with tumor

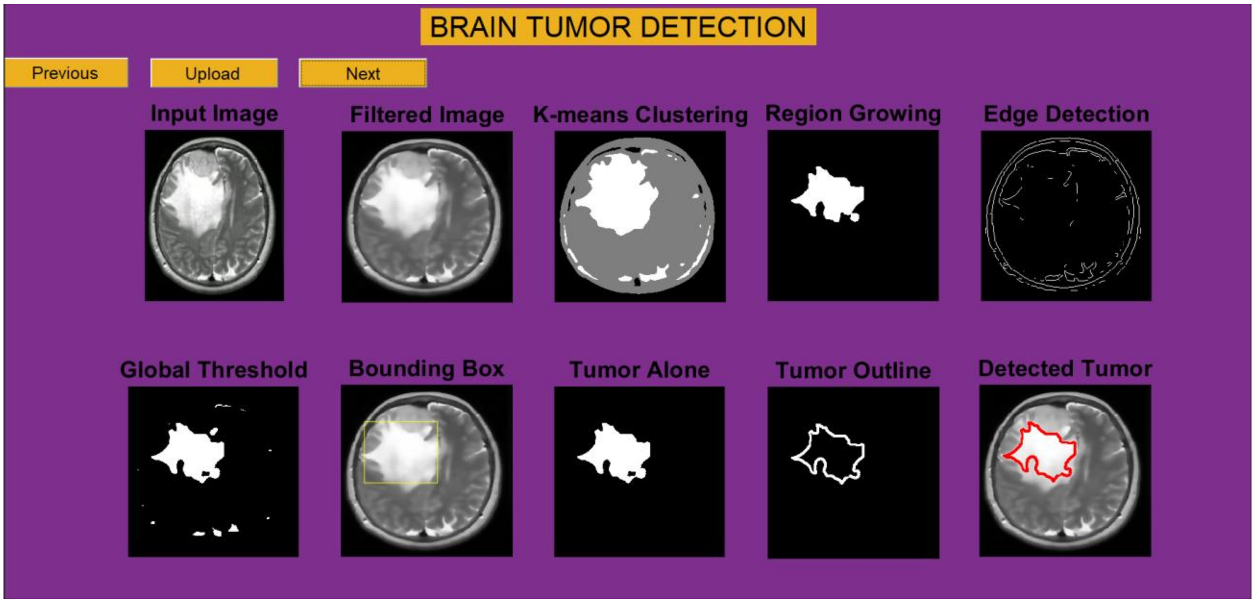


Figure 6: : Sample input and Output Brain with Tumor 1

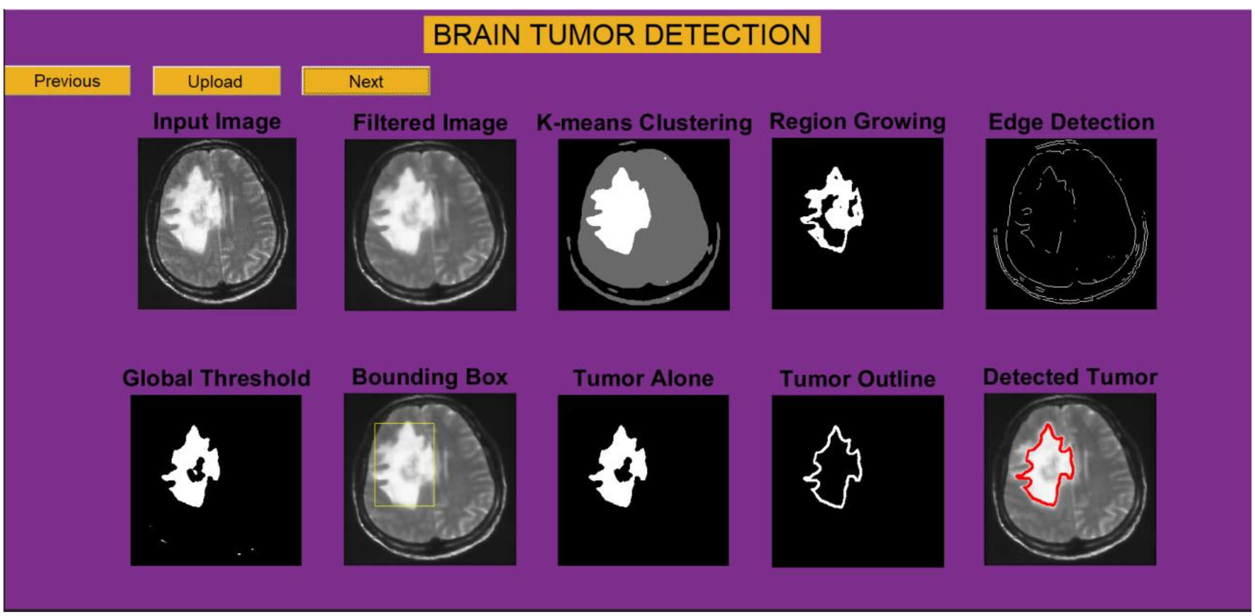


Figure 7: Sample input and Output Brain with Tumor 2

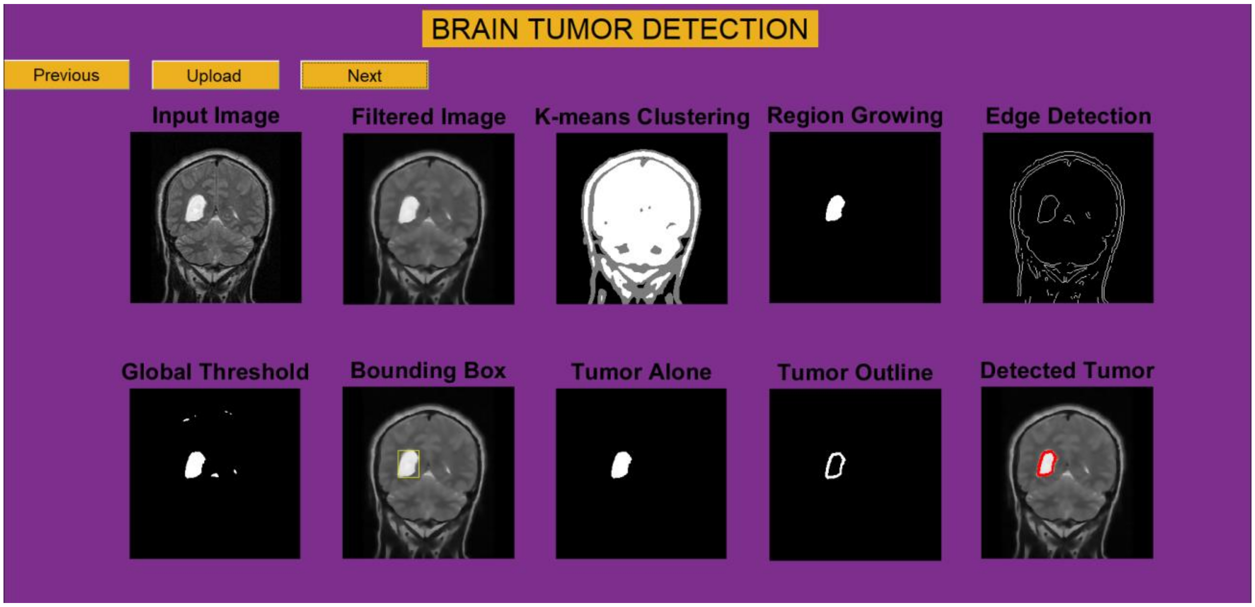


Figure 8: Sample input and Output Brain with Tumor 3

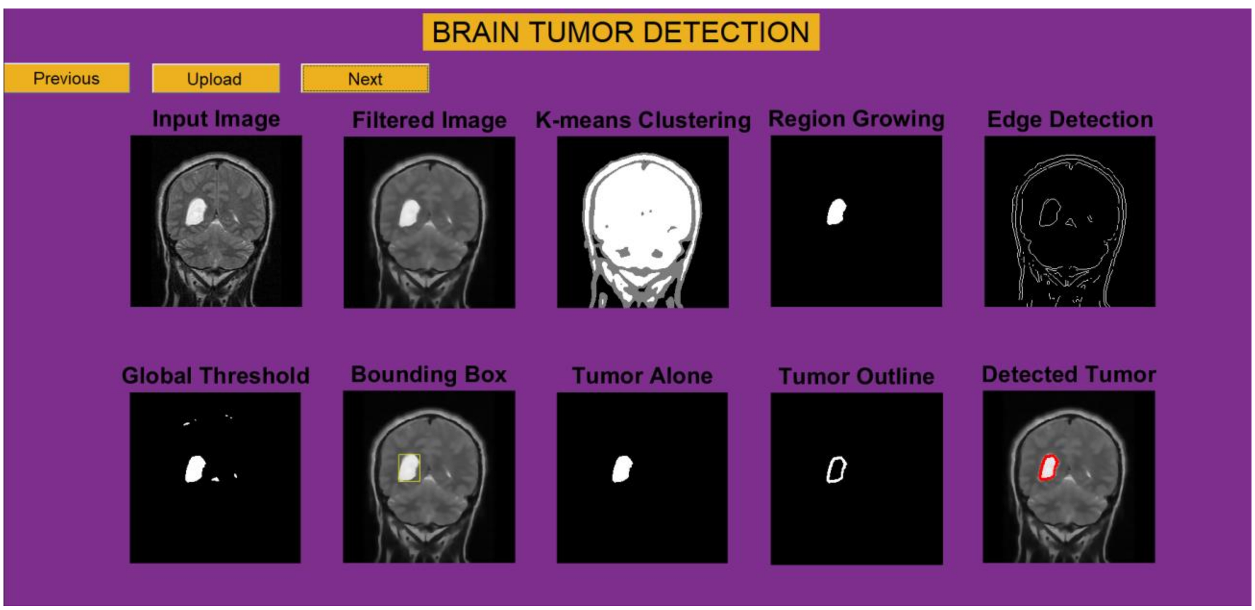


Figure 9: Sample input and Output Brain with Tumor 4

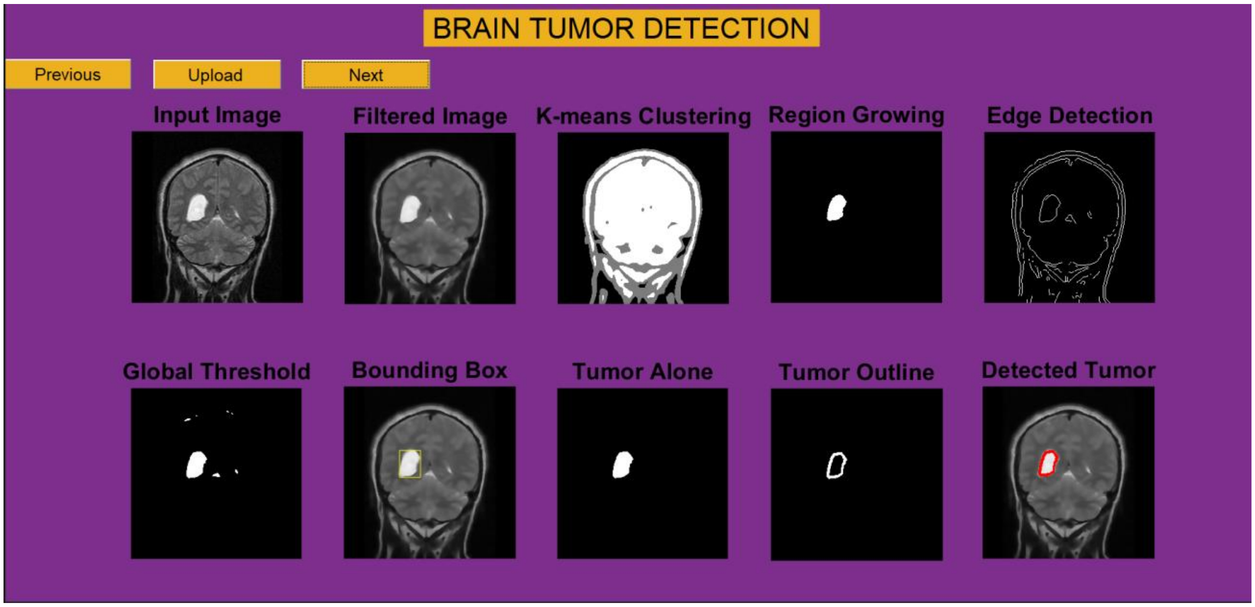


Figure 10: Sample input and Output Brain with Tumor 5

ii. Brain without Tumor

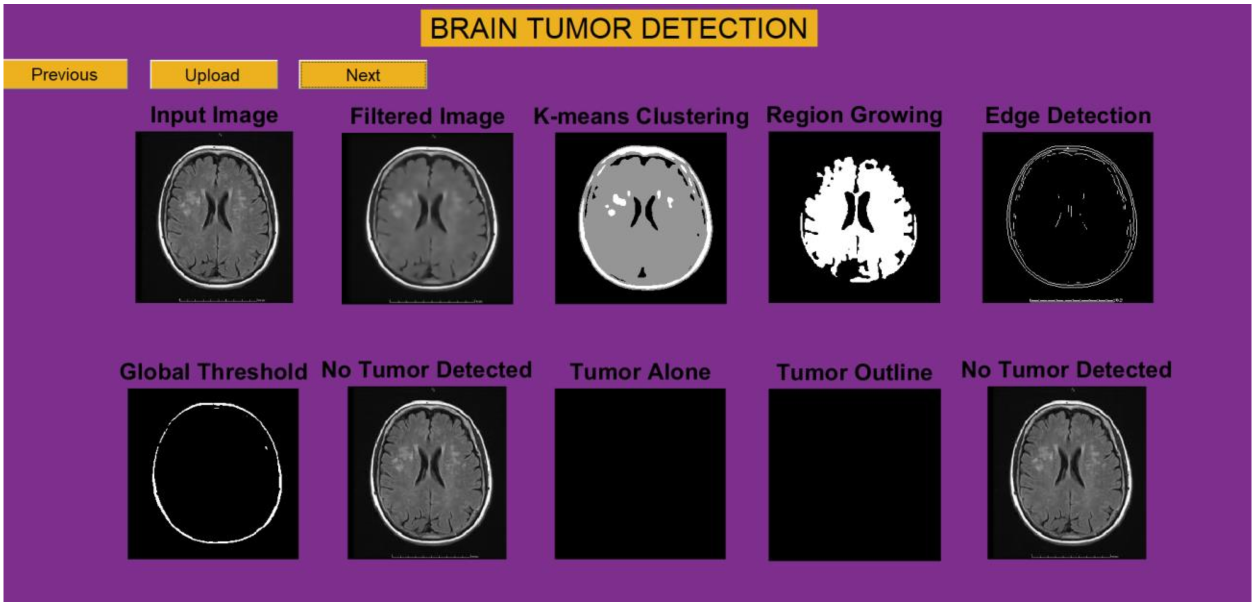


Figure 11 : Sample input and Output Brain without Tumor 1

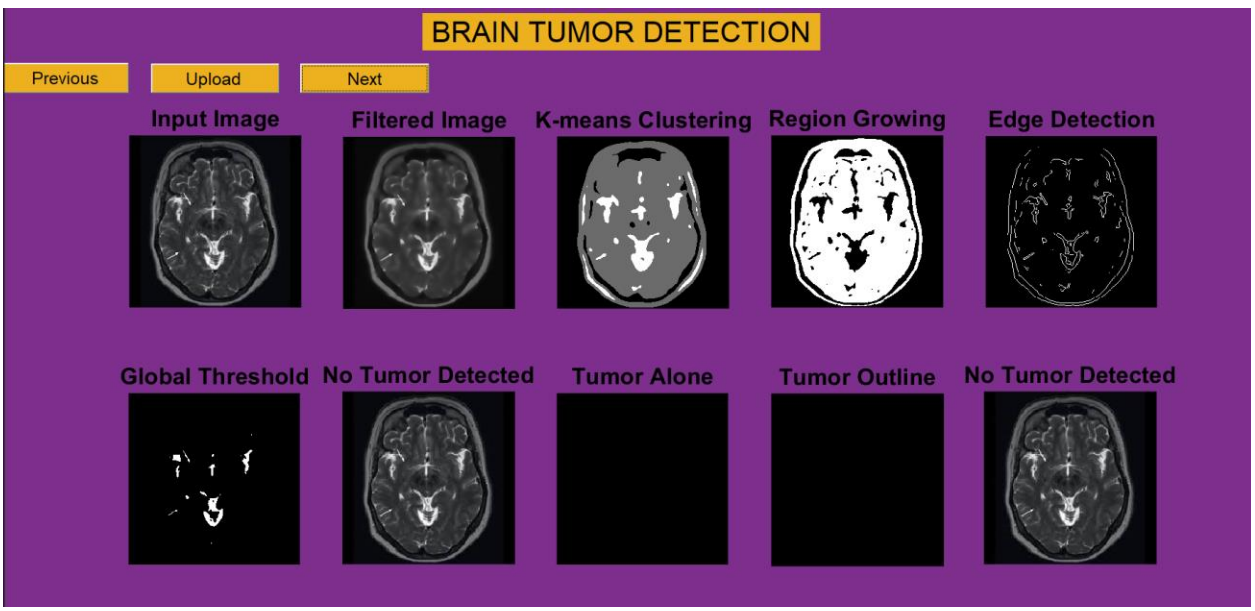


Figure 12: Sample input and Output Brain without Tumor 2

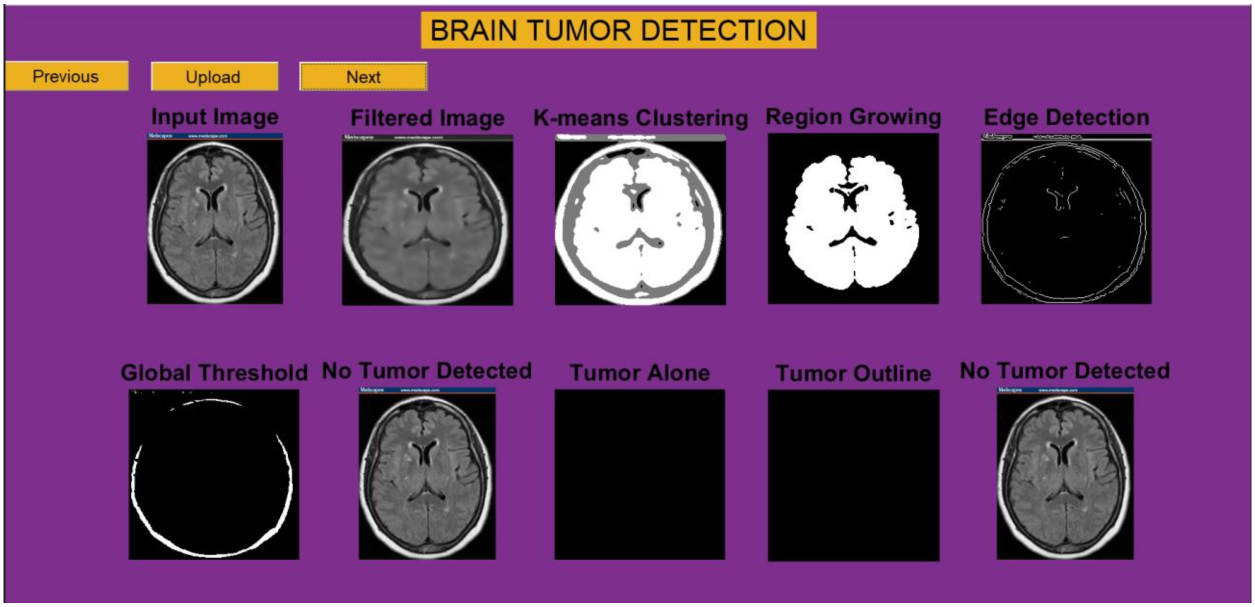


Figure 13: Sample input and Output Brain without Tumor 3

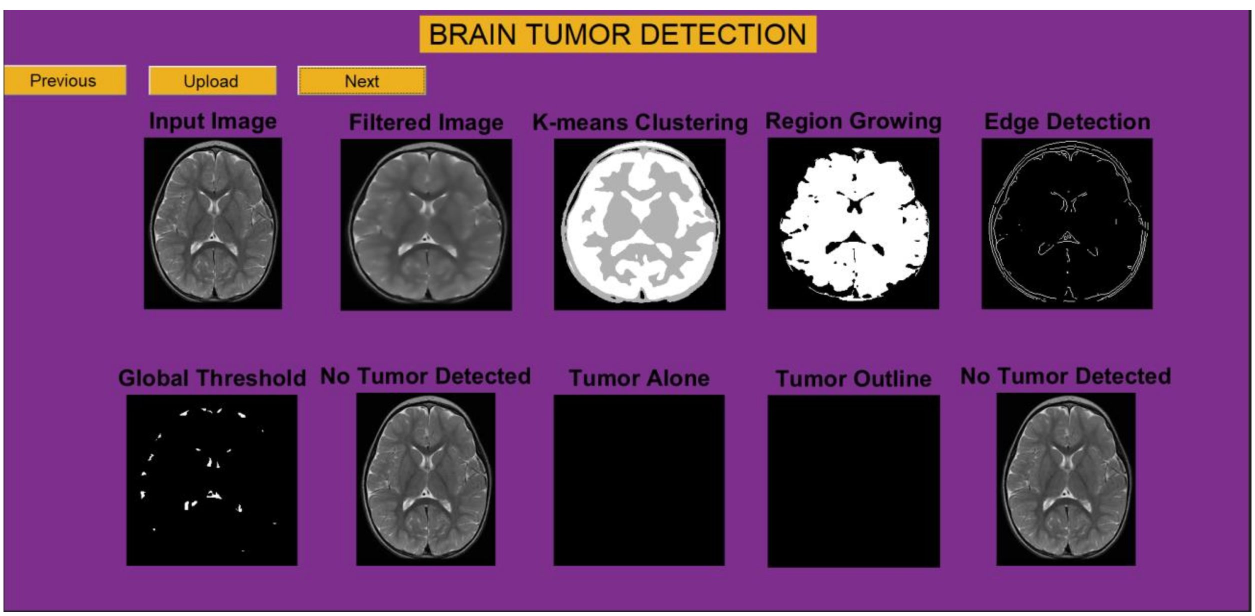


Figure 14: Sample input and Output Brain without Tumor 4

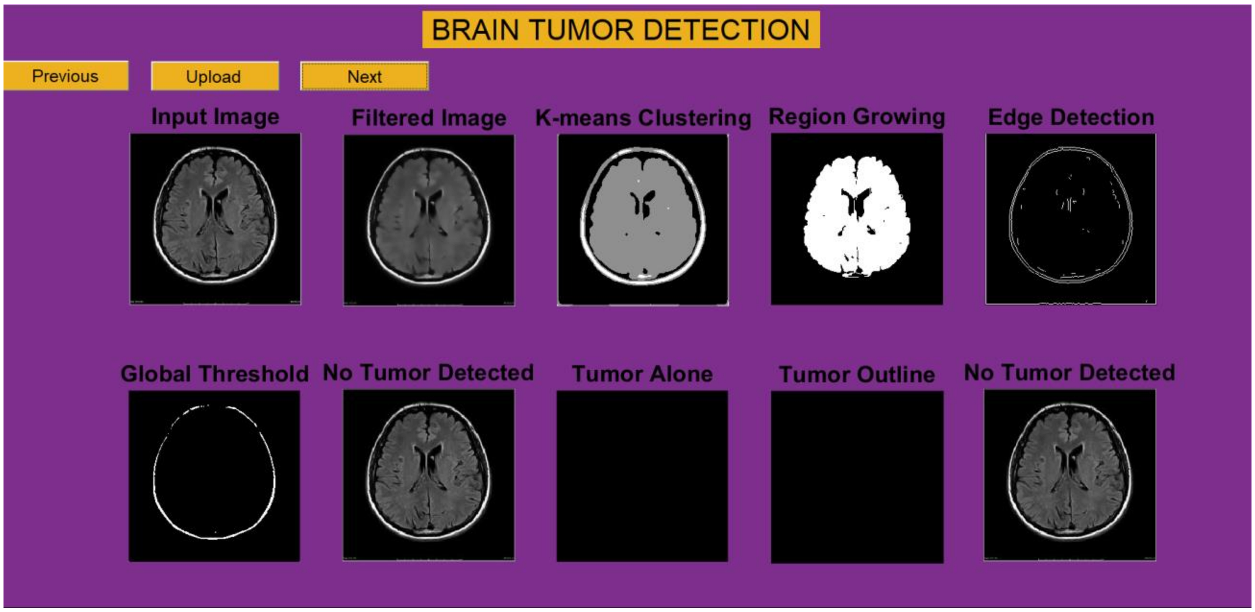


Figure 15: Sample input and Output Brain without Tumor 5

8. Source Code with Proper Comments

```
function varargout = BrainTumor_Detection(varargin)
% BRAINTUMOR_DETECTION MATLAB code for BrainTumor_Detection.fig

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                   'gui_Singleton', gui_Singleton, ...
                   'gui_OpeningFcn', @BrainTumor_Detection_OpeningFcn, ...
                   'gui_OutputFcn', @BrainTumor_Detection_OutputFcn, ...
                   'gui_LayoutFcn', [] , ...
                   'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

%% Opening Function
function BrainTumor_Detection_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
```

```
% Initialize handles for image processing
handles.imagePaths = {};
handles.currentImageIndex = 0;
handles.results = {};

% Update handles structure
guidata(hObject, handles);

%% Output Function
function varargout = BrainTumor_Detection_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

%% Upload Callback
function upload_Callback(hObject, eventdata, handles)
folder_name = uigetdir;
if folder_name
    image_files = dir(fullfile(folder_name, '*.jpg'));
    [~, sortedIdx] = sort({image_files.name});
    image_files = image_files(sortedIdx);
    handles.imagePaths = fullfile(folder_name, {image_files.name});
    handles.currentImageIndex = 1;
    handles.results = processImages(handles.imagePaths);
    guidata(hObject, handles);
    displayResults(handles);
end

%% Previous Button Callback
function previous_Callback(hObject, eventdata, handles)
if handles.currentImageIndex > 1
    handles.currentImageIndex = handles.currentImageIndex - 1;
    guidata(hObject, handles);
    displayResults(handles);
end

%% Next Button Callback
function next_Callback(hObject, eventdata, handles)
if handles.currentImageIndex < length(handles.results)
    handles.currentImageIndex = handles.currentImageIndex + 1;
    guidata(hObject, handles);
    displayResults(handles);
end

%% Image Processing Function
function results = processImages(imagePaths)
numImages = length(imagePaths);
results = cell(1, numImages);  % Initialize the results cell array
for i = 1:numImages
    imgPath = imagePaths{i};
    img = imread(imgPath);

    % --- Step 1: Convert to grayscale if necessary
    if size(img, 3) > 1
        inp = rgb2gray(img);
    else
        inp = img;
    end
```

```
    % --- Step 2: Anisotropic Diffusion
    num_iter = 10; delta_t = 1/7; kappa = 15; option = 2;
    inp = anisodiff_function(inp, num_iter, delta_t, kappa, option);  %
Call the function from the separate file
    inp = uint8(inp);
    inp = imresize(inp, [256, 256]);

    % --- Step 3: K-means Clustering
    preprocessedImg = imgaussfilt(inp, 2);
    num_clusters = 3;
    [clusteredImg, clusterCenters] = kmeans(double(preprocessedImg(:)),
num_clusters);
    clusteredImg = reshape(clusteredImg, size(preprocessedImg));
    segmentedImg = zeros(size(preprocessedImg));
    for k = 1:num_clusters
        segmentedImg(clusteredImg == k) = clusterCenters(k);
    end

    % --- Step 4: Region Growing
    seedPoint = [100, 100];
    regionImg = grayconnected(inp, seedPoint(1), seedPoint(2));

    % --- Step 5: Edge Detection
    edgesImg = edge(inp, 'sobel');

    % --- Step 6: Thresholding and Tumor Detection
    sout = imresize(inp, [256, 256]);
    t0 = mean(inp(:));
    th = t0 + ((max(inp(:)) + min(inp(:))) / 2);
    for j = 1:numel(inp)
        if inp(j) > th
            sout(j) = 1;
        else
            sout(j) = 0;
        end
    end
    label = bwlabel(sout);
    stats = regionprops(logical(sout), 'Solidity', 'Area', 'BoundingBox');
    density = [stats.Solidity];
    area = [stats.Area];
    high_dense_area = density > 0.7;
    max_area = max(area(high_dense_area));
    tumor_label = find(area == max_area);
    tumor = ismember(label, tumor_label);

    if max_area > 200
        tumorAlone = tumor;
        wantedBox = stats(tumor_label).BoundingBox;
    else
        tumorAlone = zeros(size(tumor));
        wantedBox = [];
    end

    % --- Step 7: Tumor Outline
    if max_area > 200
        dilationAmount = 5;
```

```
        rad = floor(dilationAmount);
        [r, c] = size(tumorAlone);
        filledImage = imfill(tumorAlone, 'holes');
        erodedImage = imerode(filledImage, strel('disk', rad));
        tumorOutline = tumorAlone;
        tumorOutline(erodedImage > 0) = 0;
        rgb = cat(3, inp, inp, inp);
        red = rgb(:, :, 1); red(tumorOutline) = 255;
        green = rgb(:, :, 2); green(tumorOutline) = 0;
        blue = rgb(:, :, 3); blue(tumorOutline) = 0;
        tumorOutlineInserted = cat(3, red, green, blue);
    else
        tumorOutline = zeros(size(inp));
        tumorOutlineInserted = img;  % Show original image with no tumor
detected caption
    end

    % Save the results
    results{i} = struct('img', img, 'inp', inp, 'segmentedImg',
segmentedImg, ...
        'regionImg', regionImg, 'edgesImg', edgesImg, 'sout', sout, ...
        'tumorAlone', tumorAlone, 'wantedBox', wantedBox, 'tumorOutline',
tumorOutline, ...
        'tumorOutlineInserted', tumorOutlineInserted);
end

%% Display Results Function
function displayResults(handles)
if handles.currentImageIndex > 0 && handles.currentImageIndex <=
length(handles.results)
    result = handles.results{handles.currentImageIndex};
    axes(handles.axes1); imshow(result.img); title('Input Image',
'FontSize', 20);
    axes(handles.axes2); imshow(result.inp); title('Filtered Image',
'FontSize', 20);
    axes(handles.axes3); imshow(result.segmentedImg, []); title('K-means
Clustering', 'FontSize', 20);
    axes(handles.axes4); imshow(result.regionImg); title('Region Growing',
'FontSize', 20);
    axes(handles.axes5); imshow(result.edgesImg); title('Edge Detection',
'FontSize', 20);

    % New axes for Threshold Techniques
    axes(handles.axes6); imshow(result.sout, []); title('Global
Threshold', 'FontSize', 20);

    if isempty(result.wantedBox)
        axes(handles.axes7); imshow(result.img); title('No Tumor
Detected', 'FontSize', 20);
    else
        axes(handles.axes7); imshow(result.inp); title('Bounding Box',
'FontSize', 20);
        hold on; rectangle('Position', result.wantedBox, 'EdgeColor',
'y'); hold off;
    end
    axes(handles.axes8); imshow(result.tumorAlone); title('Tumor Alone',
'FontSize', 20);
```

```
    axes(handles.axes9); imshow(result.tumorOutline); title('Tumor
Outline', 'FontSize', 20);
    if isfield(result, 'tumorOutlineInserted')
        axes(handles.axes10); imshow(result.tumorOutlineInserted);
        if isempty(result.wantedBox)
            title('No Tumor Detected', 'FontSize', 20);
        else
            title('Detected Tumor', 'FontSize', 20);
        end
    else
        axes(handles.axes10); imshow(result.img); title('No Tumor
Detected', 'FontSize', 20);
    end
end
```

9. Conclusion

Early brain tumor discovery is essential for better outcomes because it enables prompt intervention and therapy, which may avert serious consequences and increase survival rates. Improvements in genetic profiling and biomarker studies, together with the use of advanced imaging technologies like MRIs, have greatly increased diagnostic accuracy and speed. Medical personnel can better control and treat benign and malignant growths by detecting tumors early on. This lowers the danger of brain injury and boosts the efficacy of therapeutic approaches. In order to guarantee that brain tumors are identified as soon as feasible and that patients have the best prognosis possible, comprehensive screening techniques and raised awareness are crucial.

Healthcare providers may better control and cure benign and malignant growths by detecting tumors in the brain early on. This lowers the risk of brain damage and boosts the efficacy of therapeutic approaches. To ensure early detection, thorough screening procedures, public and healthcare professional awareness campaigns, and education are essential. Early management improves the patient's overall prognosis in addition to reducing the pressure that tumor development is putting on the skull. Hence, the battle against brain tumors ultimately aims to preserve lives and improve the quality of life for affected individuals. Progress in this regard has been made possible by enhanced diagnostic technologies, proactive healthcare interventions, and continuous research.

10. References

Bansal, A., Singhal, A., Amity University. School of Engineering and Technology. Department of Computer Science and Engineering, Amity University, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, & Institute of Electrical and Electronics Engineers. (2017). *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering : 12th-13th January 2017, Amity University, Noida, Uttar Pradesh, India*.

Ghorpade, N., & Bhapkar, H. R. (2021). Brain MRI segmentation and tumor detection: Challenges, techniques and applications. *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, 1657–1664. https://doi.org/10.1109/ICICCS51141.2021.9432346