National University of Sciences and Technology (NUST),
Balochistan Campus (NBC)

Department of Computer Science

Spring Semester FEB 2024 – JUN 2024

End Semester Project

Porject Title: Brain Tumor Detection

Digital Image Processing

Instructor: Dr. Muhammad Tahir

Team Members	CMS ID
Muhammad	391855
Hajra Mehmood	396466

Course Code: EE-433

Date of Submission: 4th June, 2024

Detailed Report on Brain Tumor Detection Project

1. Introduction

Brain tumors are abnormal growths of cells in the brain that can be malignant or benign. Early and accurate detection of brain tumors is crucial for effective treatment and improved patient outcomes. This project aims to develop a MATLAB-based graphical user interface (GUI) for the detection of brain tumors using digital image processing techniques. The project leverages MATLAB's robust image processing toolbox to analyze MRI images of the brain and identify potential tumor regions.

2. Project Objectives

- To create a user-friendly MATLAB GUI for loading and displaying brain MRI images.
- To implement image processing algorithms to detect the presence of brain tumors.
- To visualize the detected tumors on the MRI images.
- To provide information about the project and its developers through the GUI.

3. Methodology

The project is divided into several key components, each focusing on a specific aspect of the brain tumor detection process:

1. **GUI Development:**
    - Design a GUI using MATLAB's GUIDE (Graphical User Interface Development Environment).
    - Create buttons for loading images, processing images, and displaying results.
    - Include menus for additional information about the project and its developers.
2. **Image Loading:**
    - Implement functionality to load MRI images from the user's file system.
    - Display the loaded MRI image on the GUI.
3. **Image Processing for Tumor Detection:**
    - Convert the loaded image to grayscale and then to a binary image using a threshold.
    - Label connected components in the binary image.
    - Compute properties of labeled regions to identify potential tumor regions based on solidity and area.
    - Highlight the detected tumor region on the original MRI image.
4. **Display and Visualization:**
    - Show the original MRI image and the processed image with detected tumor regions highlighted.
    - Provide visual cues (e.g., colored boundaries) to indicate the presence of a tumor.

4. Implementation

The implementation involves several MATLAB functions and GUI callbacks. Below is a detailed explanation of the key functions and their roles:

4.1. GUI Initialization (BrainTumorDetection)

function varargout = BrainTumorDetection(varargin)

This function initializes the GUI and sets up the necessary configurations. It also handles input arguments and determines if the GUI is being created or an existing instance is being used.

4.2. GUI Opening Function (BrainTumorDetection_OpeningFcn)

function BrainTumorDetection_OpeningFcn(hObject, eventdata, handles, varargin)

This function executes just before the GUI becomes visible. It initializes the handles structure, clears previous images from the axes, and sets up default outputs.

4.3. Image Loading Button (pushbutton1_Callback)

function pushbutton1_Callback(hObject, eventdata, handles)

This callback function handles the image loading process. It allows the user to select an MRI image file, loads the image, and displays it on the GUI.

4.4. Tumor Detection Button (pushbutton2_Callback)

function pushbutton2_Callback(hObject, eventdata, handles)

This function processes the loaded MRI image to detect brain tumors. It converts the image to binary, labels connected components, and identifies potential tumor regions based on solidity and area. It then visualizes the detected tumor in the image.

5. Results

The developed GUI successfully loads and displays brain MRI images. The image processing algorithms effectively detect and highlight potential tumor regions based on predefined criteria. The visualization clearly indicates the presence of a tumor, if detected, using colored boundaries.

6. Conclusion

This project demonstrates the application of digital image processing techniques for brain tumor detection using MATLAB. The developed GUI provides a user-friendly interface for loading MRI images and detecting tumors, making it a valuable tool for medical professionals. Future enhancements could include more advanced image processing algorithms, integration with machine learning models for improved accuracy, and support for 3D MRI images.

7. Future Work

- **Advanced Image Processing:** Implement more sophisticated techniques such as edge detection, morphological operations, and region growing to improve tumor detection accuracy.
- **Machine Learning Integration:** Integrate machine learning models to enhance the detection capability and reduce false positives.
- **3D MRI Support:** Extend the tool to handle 3D MRI images for more comprehensive analysis.
- **User Feedback:** Collect feedback from medical professionals to refine the tool's usability and functionality.

8. Acknowledgments

This project was completed as part of the Digital Image Processing course by Hajra Mehmood and Muhammad. Special thanks to the course instructor, Sir Dr Muhammad Tahir and colleagues for their support and feedback.

9. References

- MATLAB Documentation: MATLAB Image Processing Toolbox
- Digital Image Processing Books and Resources, class lectures' slides
- YouTube Channel (SelfLab by Faraz) (https://www.youtube.com/watch?v=QjfaCnCz6Bc&list=WL&index=6)
- YouTube Channel (The Transworker) (https://www.youtube.com/watch?v=lApGpwF6-KM&list=WL&index=5)
- YouTube Channel (GEEK LEAK) (https://www.youtube.com/watch?v=UeFRo7uALhM&list=WL&index=4)
- ChatGPT 3.5
- MathWorks (https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html)

Appendix: Code Listing

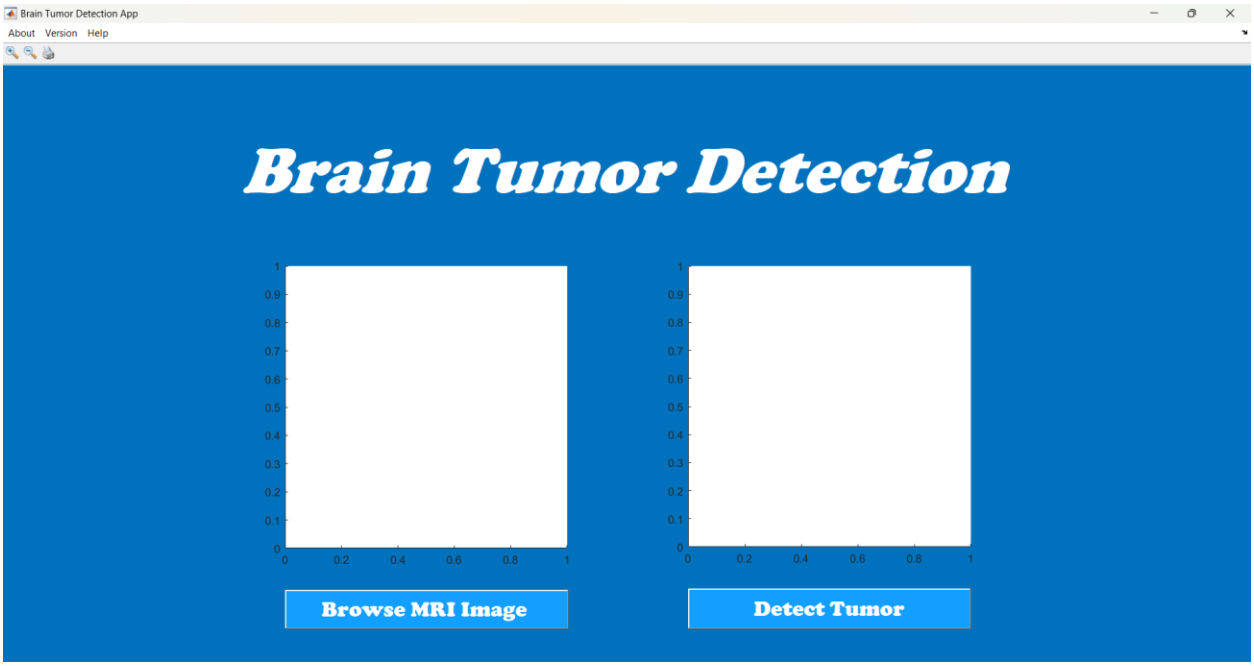
The full source code for the project is provided below:

Main GUI Functions

```
function varargout = BrainTumorDetection(varargin)
% BRAINTUMORDETECTION MATLAB code for BrainTumorDetection.fig
%      BRAINTUMORDETECTION, by itself, creates a new BRAINTUMORDETECTION or raises the existing
%      singleton*.
%
%      H = BRAINTUMORDETECTION returns the handle to a new BRAINTUMORDETECTION or the handle to
%      the existing singleton*.
%
%      BRAINTUMORDETECTION('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in BRAINTUMORDETECTION.M with the given input arguments.
%
%      BRAINTUMORDETECTION('Property','Value',...) creates a new BRAINTUMORDETECTION or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before BrainTumorDetection_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to BrainTumorDetection_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help BrainTumorDetection

% Last Modified by GUIDE v2.5 02-Jun-2024 02:32:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @BrainTumorDetection_OpeningFcn, ...
                   'gui_OutputFcn',  @BrainTumorDetection_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


% --- Executes just before BrainTumorDetection is made visible.
function BrainTumorDetection_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to BrainTumorDetection (see VARARGIN)

% Choose default command line output for BrainTumorDetection
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

cla(handles.axes1, 'reset');
cla(handles.axes2, 'reset');

% UIWAIT makes BrainTumorDetection wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = BrainTumorDetection_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% UIWAIT makes BrainTumorDetection wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global img1 img2

[path, nofile] = imgetfile();

if nofile
    msgbox('Image not found!!!','Error','warn');
    return
end

img1 = imread(path);
img1 = im2double(img1);
img2 = img1;

% Clear axes2
%cla(handles.axes2, 'reset');

%title(handles.axes2, '');

axes(handles.axes1);
imshow(img1);

title('Brain MRI', 'FontSize', 20, 'Color', [1, 1, 1]);
```

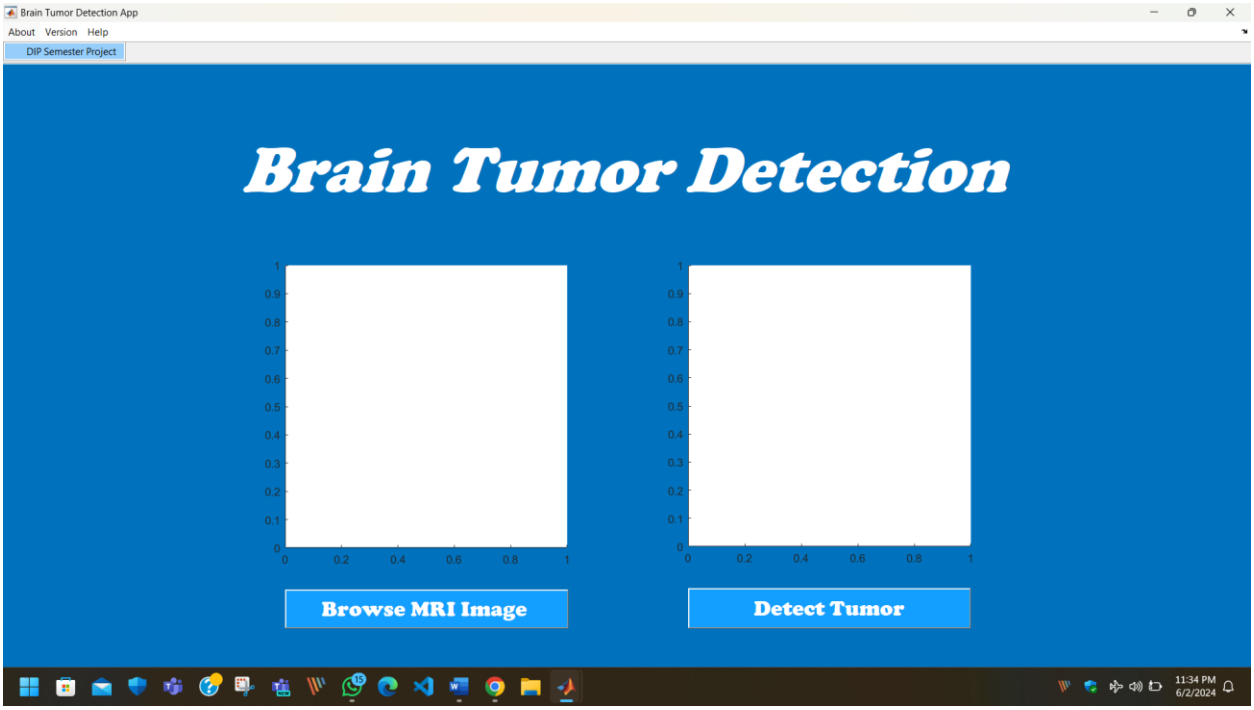
```
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global img1
axes(handles.axes2);
bw = im2bw(img1, 0.7);
label = bwlabel(bw);

stats = regionprops(label, 'Solidity', 'Area');
density = [stats.Solidity];
area = [stats.Area];
high_dense_area = density > 0.5;

% Set a minimum area threshold to filter out small areas
min_area_threshold = 100; % Adjust as needed

if ~any(high_dense_area) || max(area(high_dense_area)) < min_area_threshold
    % No tumor detected
    imshow(img1);
    title('\fontsize{20}\color[rgb]{0.996, 0.0, 0.0} Tumor Not Detected');
    return;
end

max_area = max(area(high_dense_area));
tumor_label = find(area == max_area);
tumor = ismember(label, tumor_label);

se = strel('square', 5);
tumor = imdilate(tumor, se);

Bound = bwboundaries(tumor, 'noholes');

imshow(img1);
hold on

for i = 1:length(Bound)
    plot(Bound{i}(:,2), Bound{i}(:,1), 'y', 'linewidth', 1.75)
end

title('\fontsize{20}\color[rgb]{0.0, 0.996, 0.0} Tumor Detected!');

hold off
axes(handles.axes2)



% --------------------------------------------------------------------
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
msgbox('The end semester project of Digital Image Processing, done by Hajra Mehmood and Muhammad.','About US','help','modal');


% --------------------------------------------------------------------
function Untitled_3_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
msgbox('Version: 1.00','Version','help','replace');


% --------------------------------------------------------------------
function Untitled_4_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
msgbox(sprintf('Muhammmad: zamuranimuhammed@gmail.com\nHajra Mehood: hajramehmood24@gmail.com'),'Contact US','help','non-modal');

```

Brain Tumor Detection Application Graphical User interface (GUI)

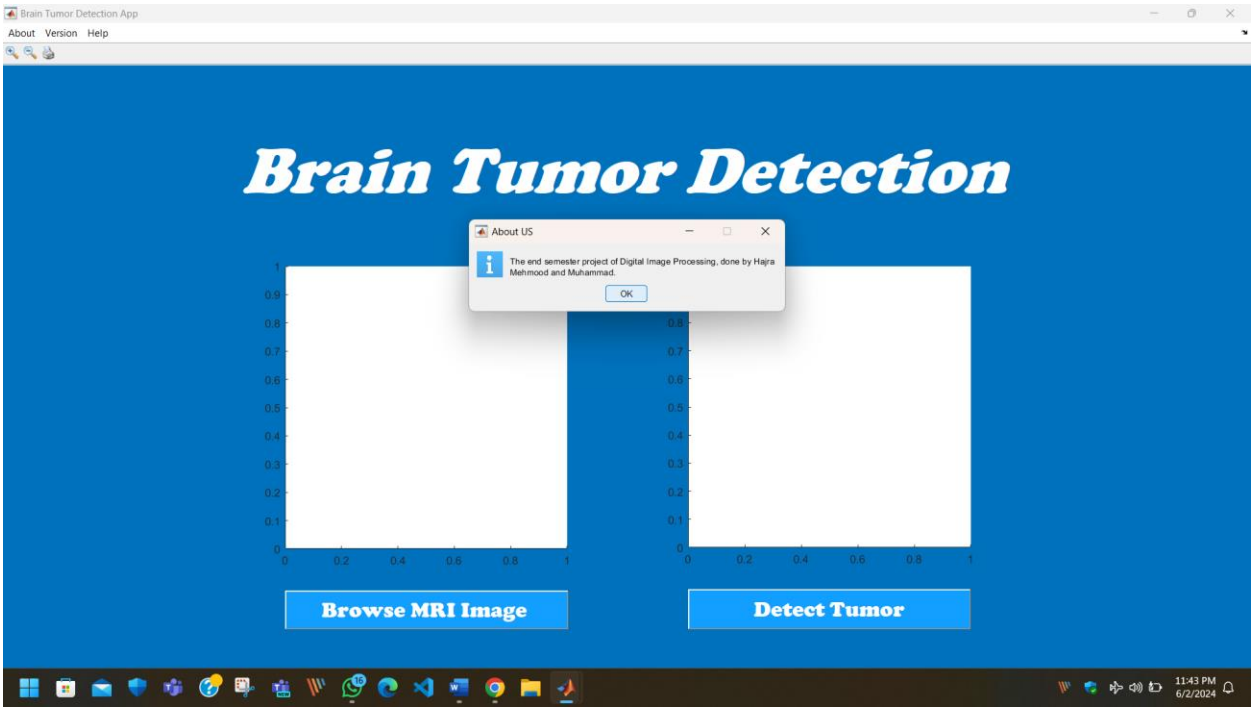


The main interface of the application with a menu bar with the follow fields (*About, Version and Help*), a toolbar for (*zoom-in, zoom-out and print*), a text that '*Brain Tumor Detection*', and two axes, one to *browse MRI image* and the second one for *detecting tumor* with the respective push buttons for their functionalities.

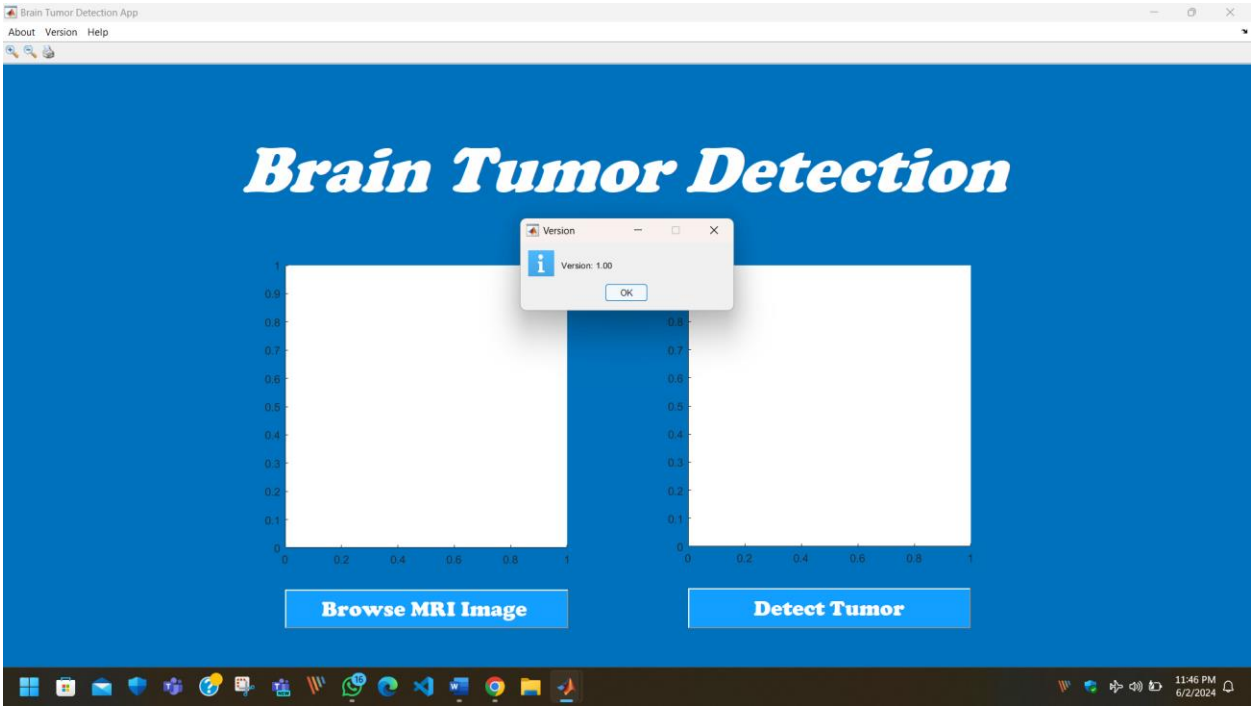
'About' Menu Bar



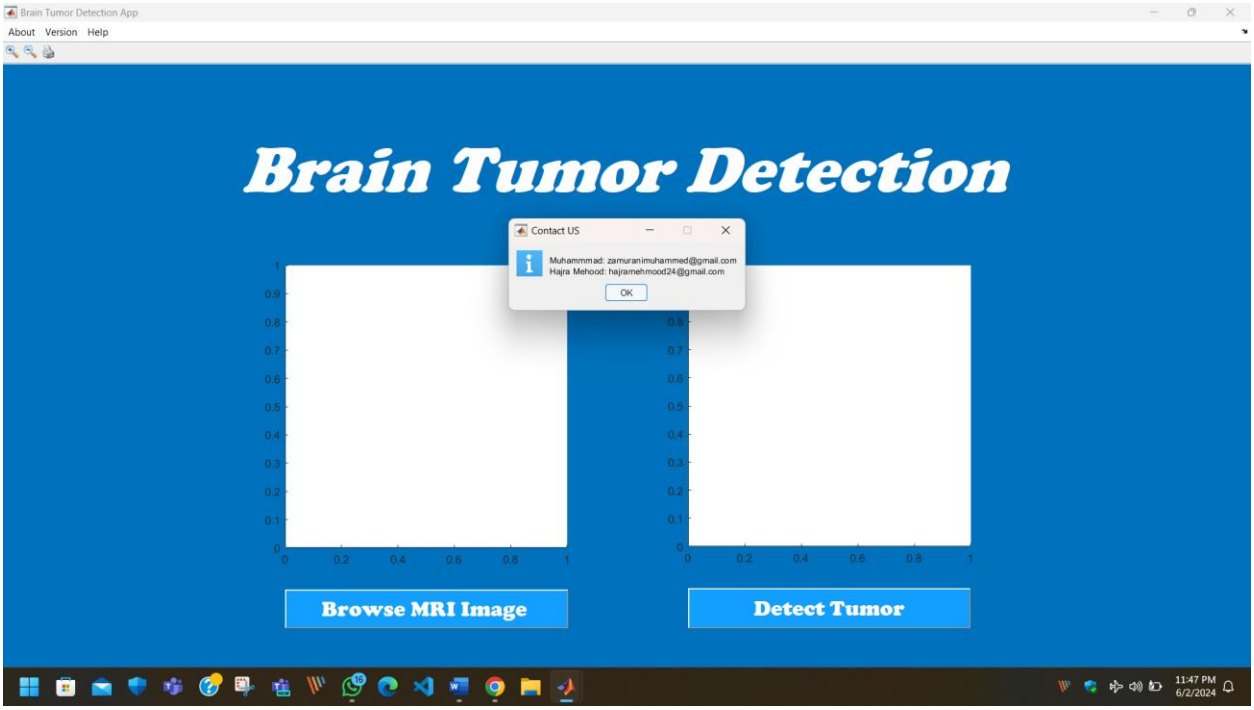
It shows another sub menu called '**DIP Semester Project**' when clicked shows the following message box with the following message '*The end semester project of Digital Image processing, done by Hajra Mehmood and Muhammad*'.



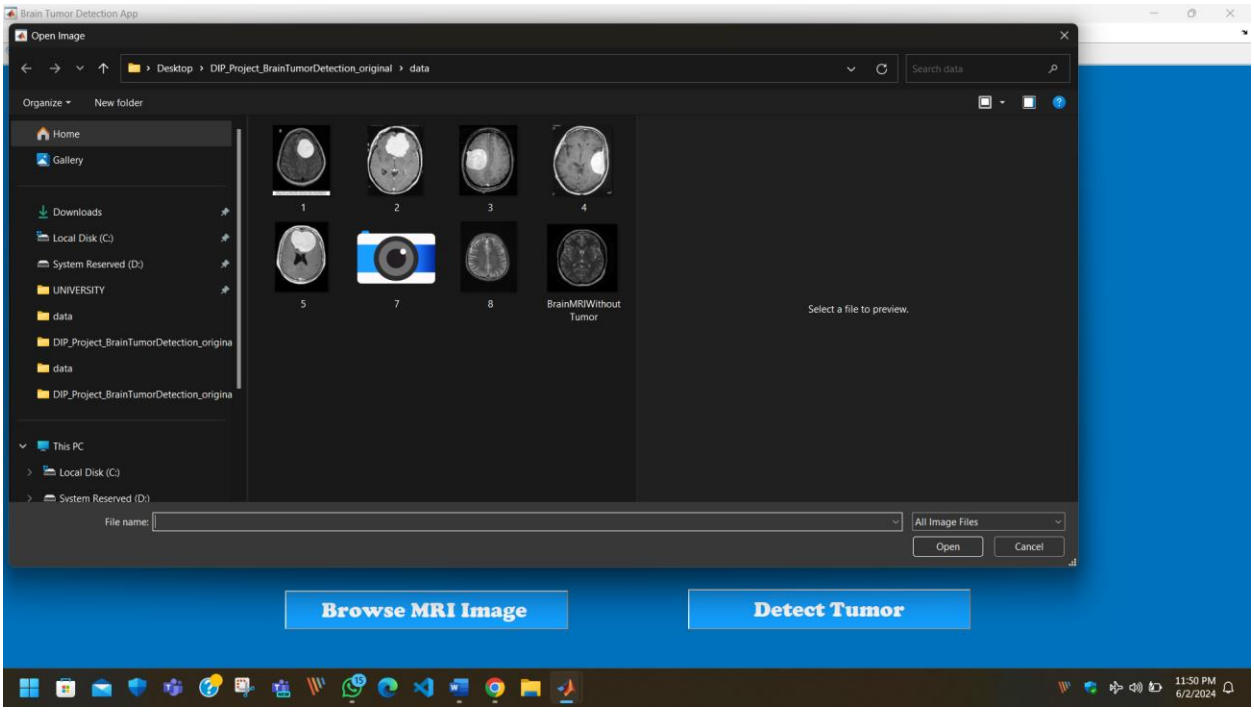
The **Version Menu Bar** shows the version of the current application built through MATLAB GUIDE (*Version: 1.00*).



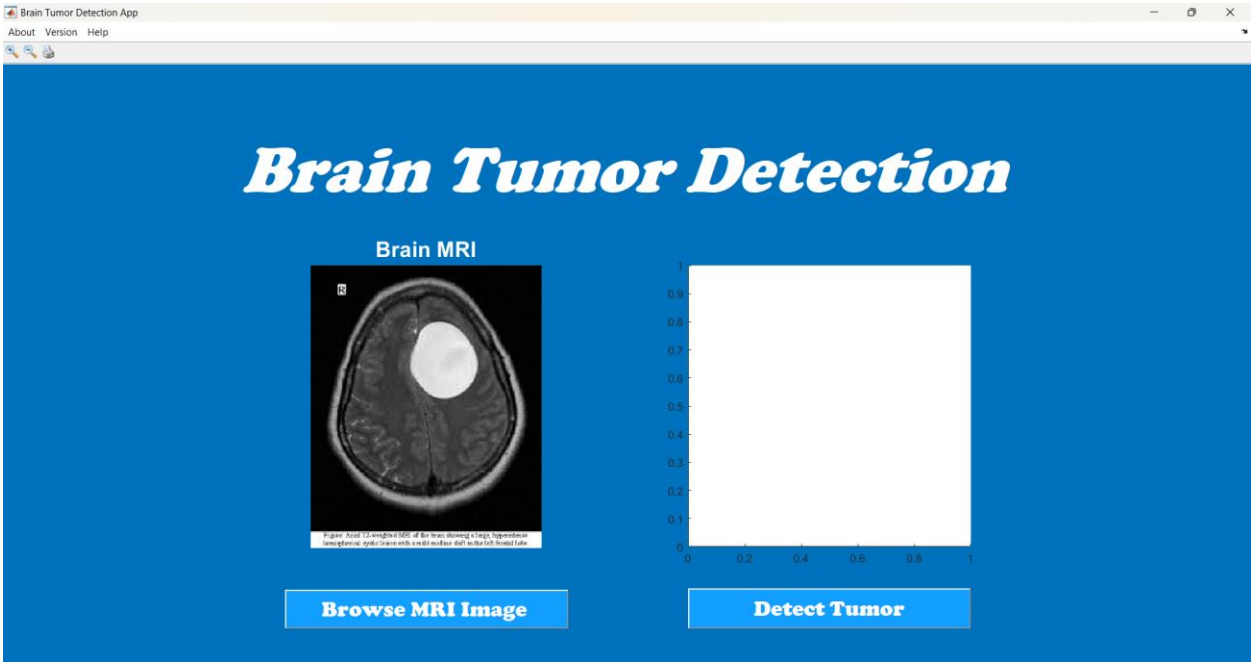
And the **Help Menu Bar** shows our contact information for any communication or the usage of the application.



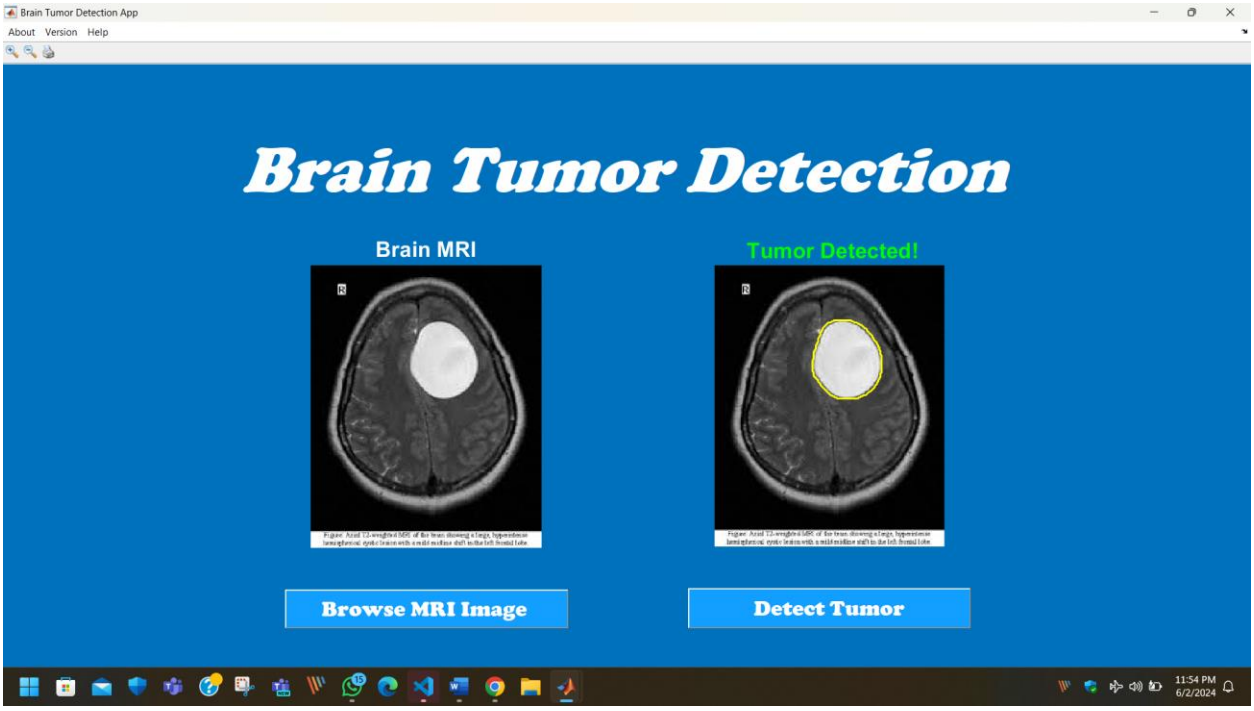
When you push the Browse MRI Image button then it opens a window to select an image from any directory.



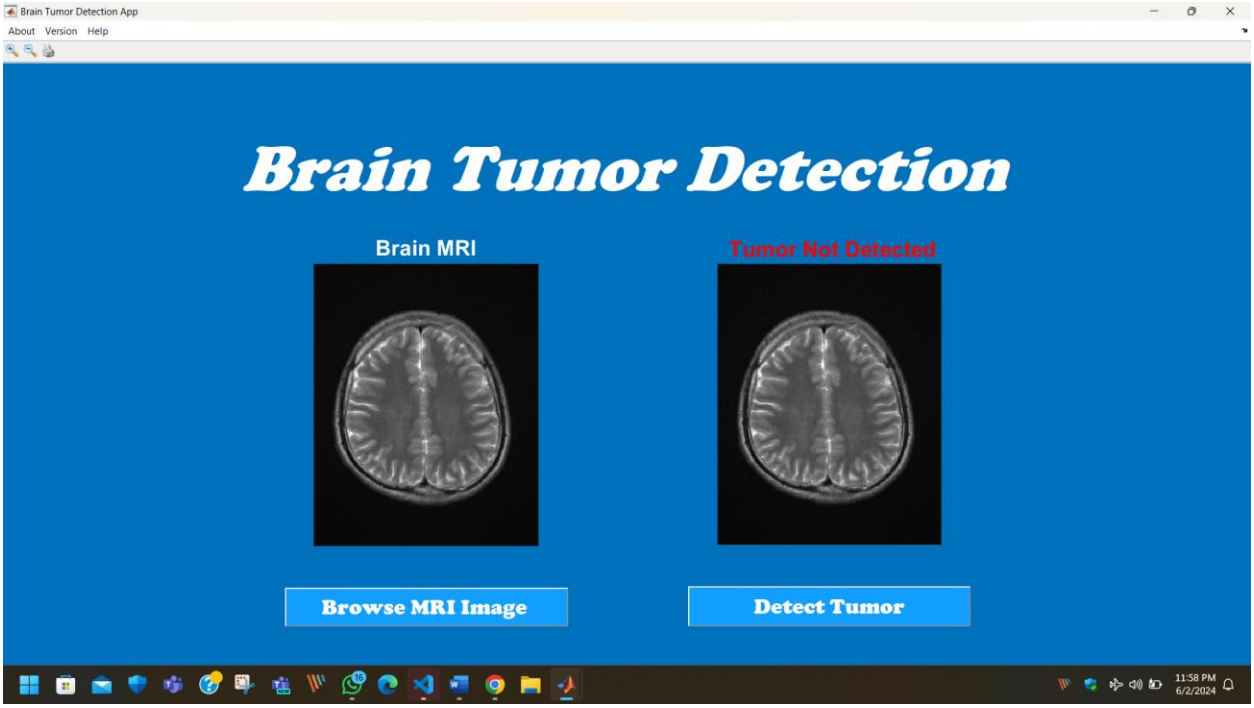
When an image is selected for processing then it would be shown in the axes 01 with a text above it as '*Brain MRI*', as given below.



If we push the button **'Detect Tumor'**, then it is shown below how a tumor is detected and highlighted, followed by a green text that '*Tumor Detected!*'.



What if an MRI image of brain does not keep tumor? The answer is that the system is also flexible for various cases in such a case then, the output would be as mentioned below that '*Tumor Not Detected*' with a red text.


Brain Tumor Detection App
About Version Help
Brain Tumor Detection
Brain MRI
Tumor Not Detected
Browse MRI Image
Detect Tumor
11:58 PM
6/2/2024

THE END!