SCHOOL OF COMPUTING

FACULTY OF ENGINEERING

SECV3213-02 ASAS PEMPROSESAN IMEJ

(FUNDAMENTAL OF IMAGE PROCESSING)

SECTION 02

ASSIGNMENT 1

Lecturer : Dr. Md Sah bin Hj Salam

Team Members (Group ZBL):

1. Chiam Wooi Chin A19EC0034

2. Ng Jing Er A19EC0115

1. Introduction

Assignment 1 is about the Image Enhancement in Spatial Domain. In this assignment, we have developed a simple application using GUIDE in the Matlab. The user can darken a part of the image that is selected by using the application.

There are a few requirements of this assignment. First, the application enables the user to load any images from the library of the computer. The function of roipoly has been used to enable the user to choose multiple parts of the image, and darken the parts selected. The user is also able to save the image after the darken effect in their computer in the jpg format. We have created several buttons with specific functions to fulfil the requirements such as LoadImage button, SelectPart button, Darken button and SaveImage button. And the design of the application is simple and easy to look at which provides a good interface for the user.

2. Codes and Documentation

2.1 Load Image Button

```
% --- Executes on button press in LoadImage.
function LoadImage_Callback(hObject, eventdata, handles)
% hObject    handle to LoadImage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname] = uigetfile('*.*', 'Choose an image file to process');
img= imread([pathname filename]);
handles.currentData =img;
imshow(handles.currentData,'Parent',handles.axes1);
imshow(handles.currentData,'Parent',handles.axes2);

guidata(hObject,handles);
```

The **LoadImage** button is used to allow the users to load any image they want for the purpose of image processing. **uigetfile()** function is invoked and opens a modal dialog box that lists the files in the current folder of pathname. The **imread()** function is used to read the image from the file of any format specified by filename. Then, the **imshow()** function will display the image based on the image's data and properties.

2.2 Select Button

```
% --- Executes on button press in SelectPart.
function SelectPart_Callback(hObject, eventdata, handles)
% hObject    handle to SelectPart (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global bw;
global I;
I=handles.currentData;
bw= im2uint8(roipoly(I));

guidata(hObject,handles);
```

In the **SelectPart_Callback** function, **roipoly** is used to create an interactive polygon tool associated with the image displayed in the application. This function allows the user to select a polygonal region of interest within the image. **roipoly** returns a binary image which allows us to use as a mask for masked filtering while **im2unit8()** will convert the RGB image I to uint8 , rescaling or offsetting the data as necessary.

2.3 Darken Button

```
% --- Executes on button press in Darken.
function Darken_Callback(hObject, eventdata, handles)
% hObject    handle to Darken (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    global bw;
    global I;
    I = handles.currentData;
    I_dark = imdivide(I,1.5);

    bw_cmp = bitcmp(bw);
    roi = bitor(I_dark,bw_cmp);
    not_roi = bitor(I,bw);
    handles.currentData=I_dark;
    I_new = bitand(roi,not_roi);
    handles.currentData=I_new;
    imshow(I_new,'Parent',handles.axes2);

    guidata(hObject,handles);
```

For the **Darken_Callback** function, the arithmetic operation of **division** has been used to perform the darken effect. Based on the code above, **imdivide()** is used where the value of the image will be divided by 1.5 and returns the result **I_dark**. This results in the image value of the selected part being divided by 1.5 when the Darken button is being pressed each time, which means the selected part will become darker.

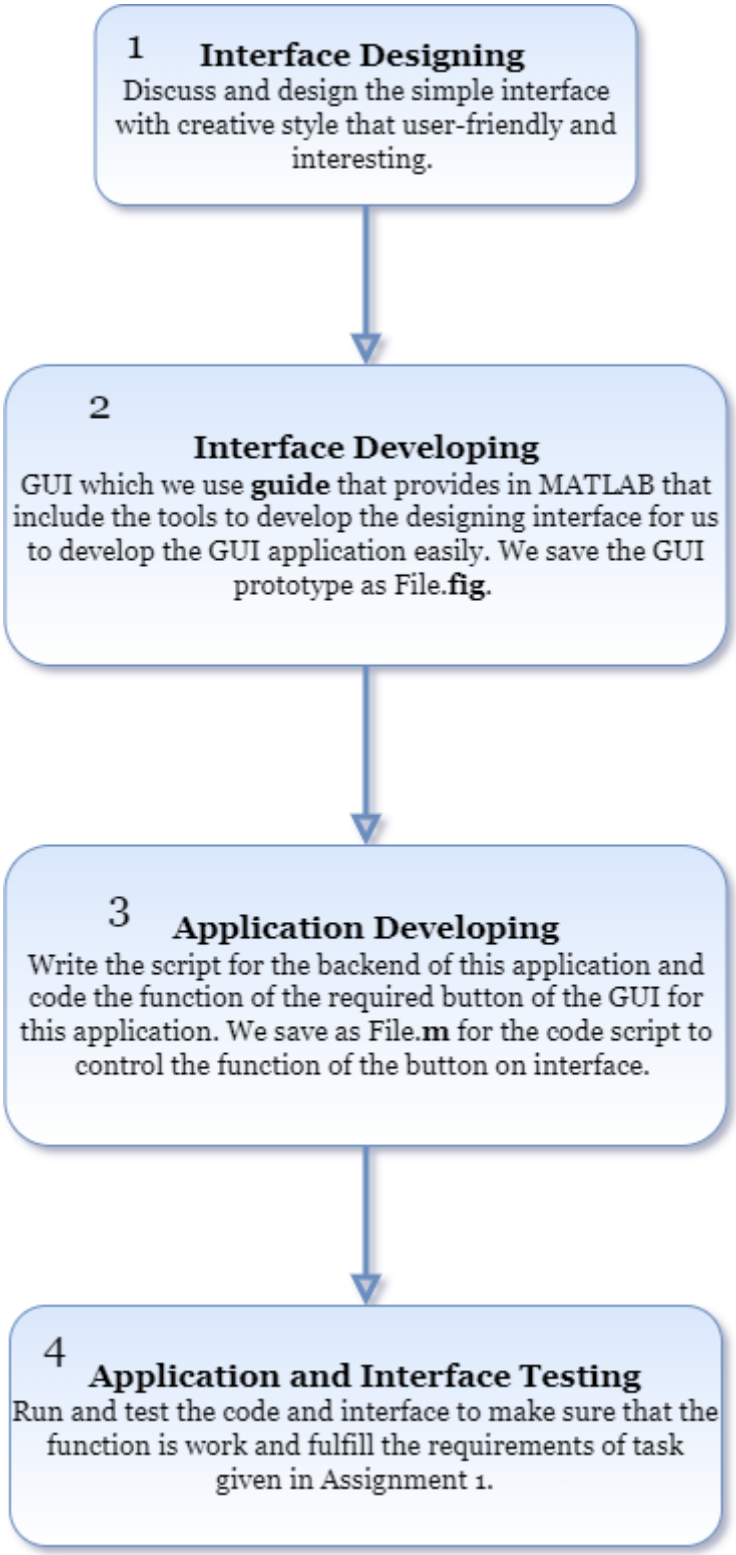
2.4 Save Button

```
% --- Executes on button press in SaveImage.
function SaveImage_Callback(hObject, eventdata, handles)
% hObject    handle to SaveImage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname] =uiputfile({'*.jpg'},'Choose a location to save image file');
fileName = fullfile(pathname,filename);
imwrite(handles.currentData,fileName);

guidata(hObject,handles);
```

In the SaveImage_Callback function, **[filename, pathname]= uiputfile** is to save the image to the specific path location which the **uiputfile({'*.jpg'},'Choose a location to save image file')** to input the file by naming the file in jpg and the path location that chooses to save the image. **fileName = fullfile(pathname,filename)** is to declare the location of the saved image and the name of the jpg file. **imwrite(handles.currentData,fileName)** is to create a new file that will export the output current image to the specific file with **fileName** and enter the image file format as jpg.

3. Flow and Process Chart



4. Example of the Image and Output

Figure 1 :
Example image before darken

Figure 2:
Example image after darken

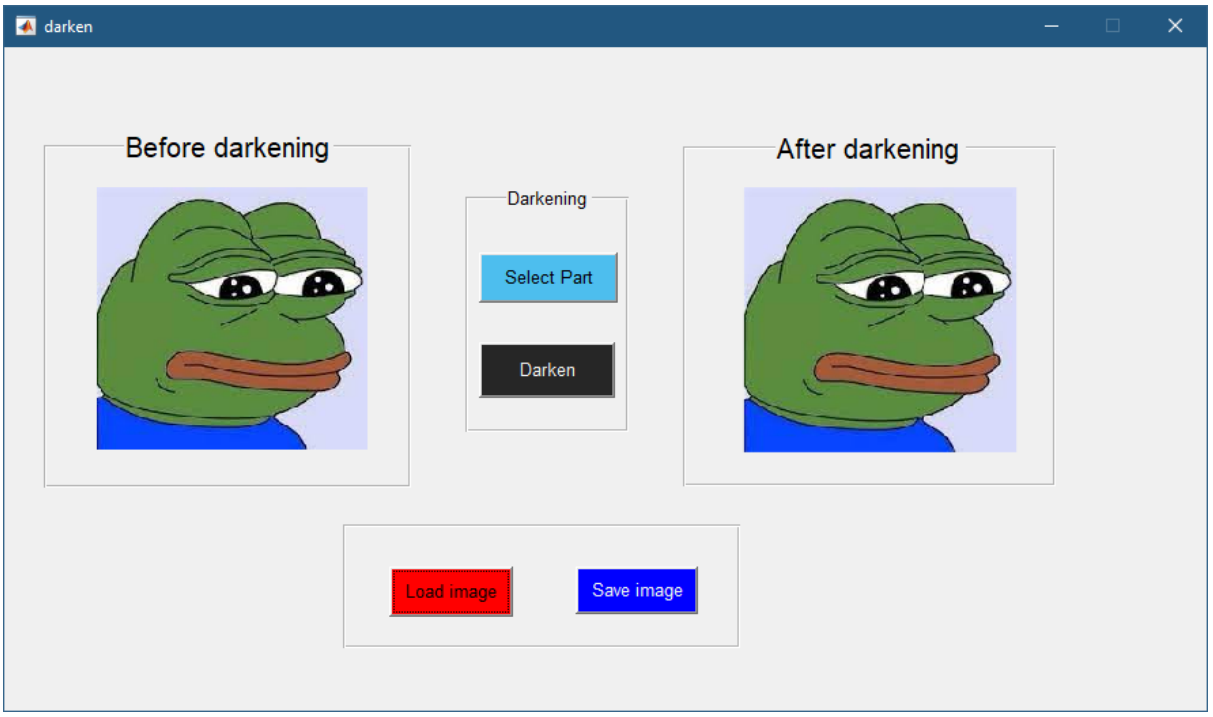


Figure 3 :
Example application before darkening the image

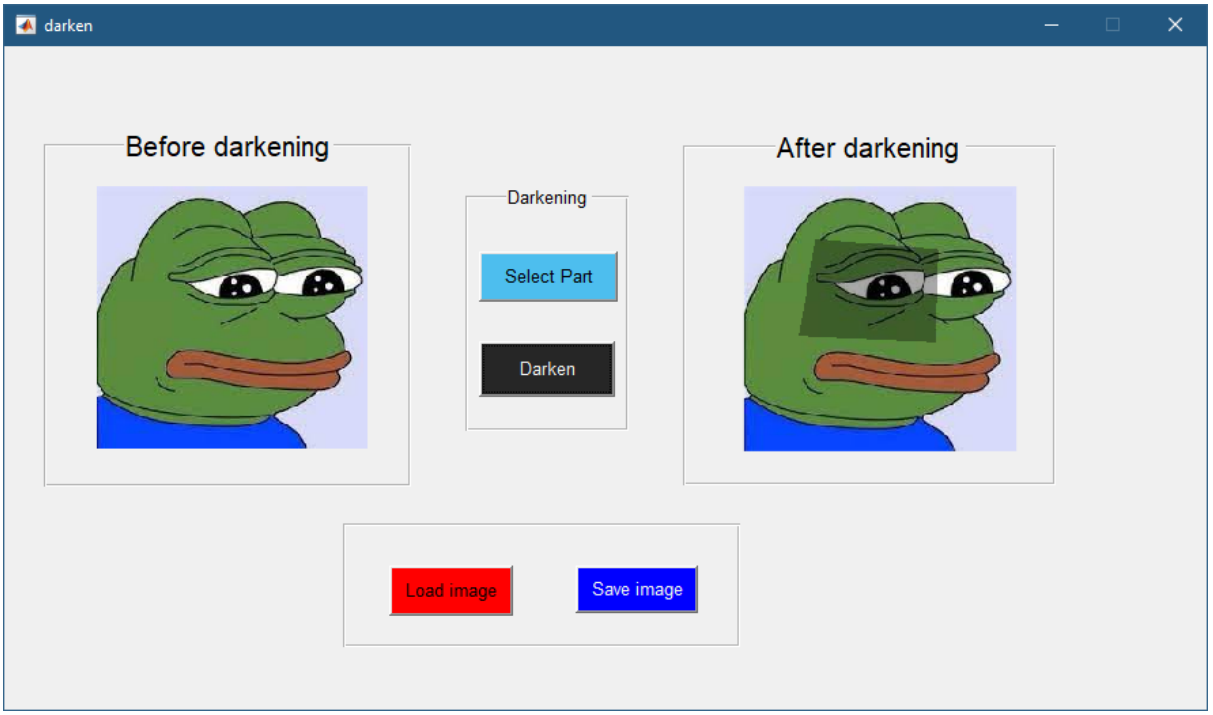


Figure 4 :
Example application after darkening the image

5. Video Explanation Link

https://youtu.be/ZVMmxIcAeSA