

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
JIMMA UNIVERSITY
%%%%%%%%%
%%%%%%%%%
JIMMA INSTITUTE OF TECHNOLOGY
%%%%%%%%%
%%%%%%%%%
DEPARTEMENT OF BIOMEDICAL ENGINEERING
%%%%%%%%%
%%%%%%%%%
BIOMEDICAL IMAGE PROCESSING AND ANALYSIS-I
%%%%%%%%%
%%%%%%%%%
Assignment-II
%%%%%%%%%
%%%%%%%%%
SUBMITTED BY: Salih Aliyi
%%%%%%%%%
%%%%%%%%%
SUBMITTED TO: Dr.Gizeaddis.L

```

```

function varargout = gui_mini_project(varargin)
% GUI_MINI_PROJECT MATLAB code for gui_mini_project.fig
% GUI_MINI_PROJECT, by itself, creates a new GUI_MINI_PROJECT or raises the
existing
% singleton*.
%
% H = GUI_MINI_PROJECT returns the handle to a new GUI_MINI_PROJECT or the handle
to
% the existing singleton*.
%
% GUI_MINI_PROJECT('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_MINI_PROJECT.M with the given input arguments.
%
% GUI_MINI_PROJECT('Property','Value',...) creates a new GUI_MINI_PROJECT or
raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui_mini_project_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui_mini_project_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 gui_mini_project

% Last Modified by GUIDE v2.5 05-May-2019 16:43:51

```

```

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @gui_mini_project_OpeningFcn, ...
                  'gui_OutputFcn',    @gui_mini_project_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 gui_mini_project is made visible.
function gui_mini_project_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 gui_mini_project (see VARARGIN)


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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = gui_mini_project_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;


% --- 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)
smoothing_filter      % TO SELECT THE SMOOTHING FILTER IN SPATIAL DOMAIN 3X3,5X5,9X9

% --- Executes on button press in pushbutton2.
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)
sharpening           % TO SELECT GUI SHARPEN IMAGE IN SPATIAL DOMAIN BY USING LAPLACIAN
AND SOBEL OPERATOR

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
simplefrequencyD      % PUSH BUTTON TO SELECT SMOOTHING AND SHARPENING IN FREQUENCY
DOMAIN

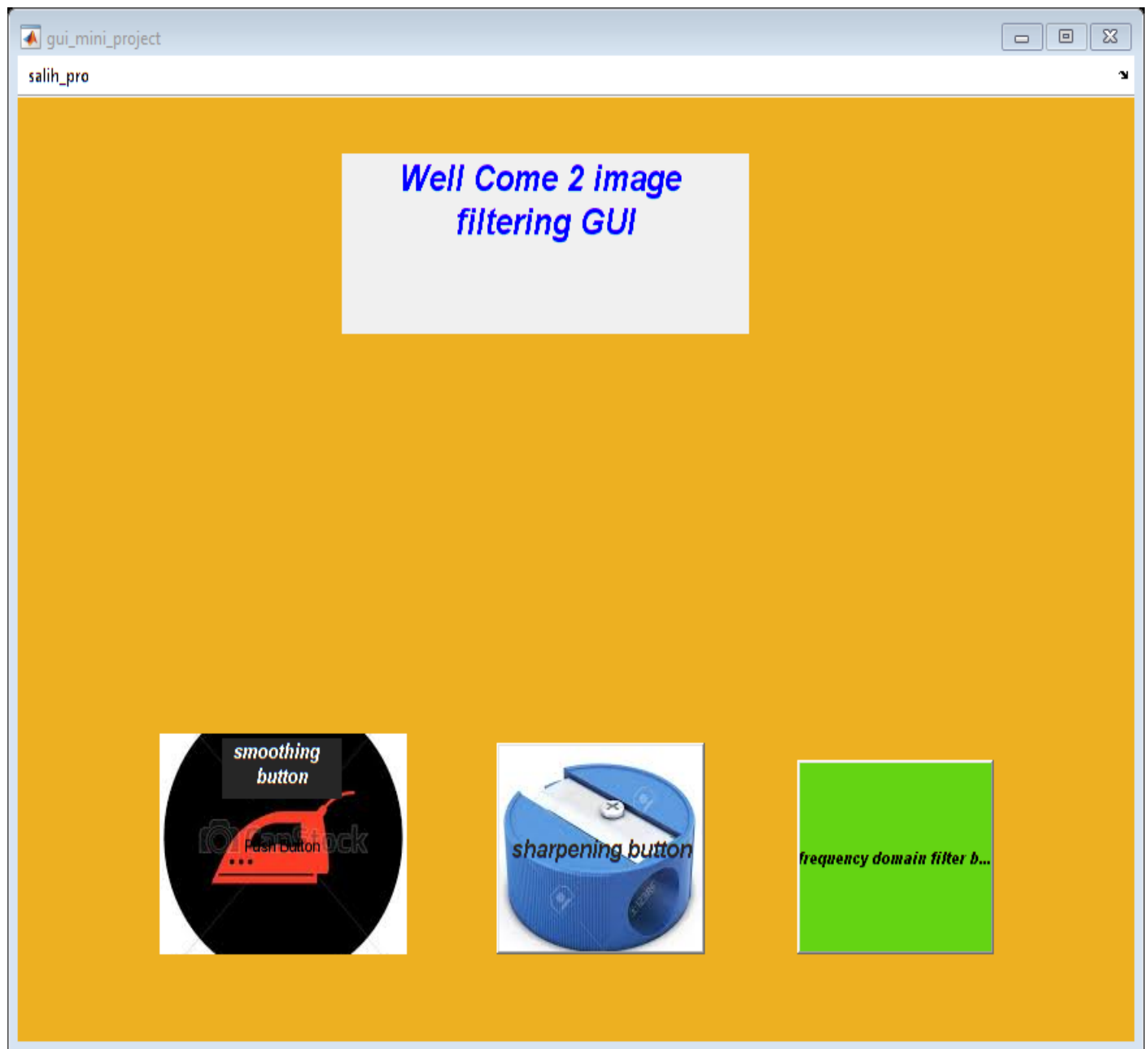
% --- Executes on key press with focus on pushbutton2 and none of its controls.
function pushbutton2_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  structure with the following fields (see MATLAB.UI.CONTROL.UICONTROL)
%           Key: name of the key that was pressed, in lower case
%           Character: character interpretation of the key(s) that was pressed
%           Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)

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

```

```
% -----  
function salih_pro_Callback(hObject, eventdata, handles)  
% hObject    handle to salih_pro (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles     structure with handles and user data (see GUIDATA)  
  
% -----  
function about_me_Callback(hObject, eventdata, handles)  
% hObject    handle to about_me (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles     structure with handles and user data (see GUIDATA)  
msgbox(sprintf('Name:Salih Aliyi\n\n\n ID.no:RM7438/11-0\n\n\n Biomedical Imaging\nstudent@JiT'),'About_me');
```



1. Apply smoothing filter to an image with filter mask options of 3x3, 5x5 and 9x9

```
function varargout = smoothing_filter(varargin)
% SMOOTHING_FILTER MATLAB code for smoothing_filter.fig
%     SMOOTHING_FILTER, by itself, creates a new SMOOTHING_FILTER or raises the
existing
%     singleton*.
%
%     H = SMOOTHING_FILTER returns the handle to a new SMOOTHING_FILTER or the handle
to
%     the existing singleton*.
%
%     SMOOTHING_FILTER('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in SMOOTHING_FILTER.M with the given input arguments.
%
%     SMOOTHING_FILTER('Property','Value',...) creates a new SMOOTHING_FILTER or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before smoothing_filter_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to smoothing_filter_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 smoothing_filter

% Last Modified by GUIDE v2.5 05-May-2019 17:24:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @smoothing_filter_OpeningFcn, ...
                  'gui_OutputFcn',  @smoothing_filter_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 smoothing_filter is made visible.
function smoothing_filter_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 smoothing_filter (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = smoothing_filter_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;

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a
[im ,path]=imgetfile();           % to select a given image from file or load image
if path
    msgbox(sprintf('invalid selection'),'error','warn');
    return
end
r=imread(im);
a=rgb2gray(r);                   % not used if the image we use already in gray scale
axes(handles.axes4);
imshow(a);
title('gray level image');

% --- 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 a
h=zeros(3,3);
for i=1:3           % index start from 1 in matlab
    for j=1:3
        h(i,j)=1/50;
        b=imfilter(a,h);
    end
end
axes(handles.axes1);
imshow(b);
title('filtred image by 3x3 filter mask');

% --- Executes on button press in pushbutton2.
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)
%[im ,path]=uigetfile('*.','enter the image');
%im=strcat(path,im);
%a=imread(im);
global a
h=zeros(5,5);
for i=1:5
    for j=1:5
        h(i,j)=1/50;
        b=imfilter(a,h);
    end
end
axes(handles.axes2);
imshow(b);
title('filtred image by 5x5 filter mask');

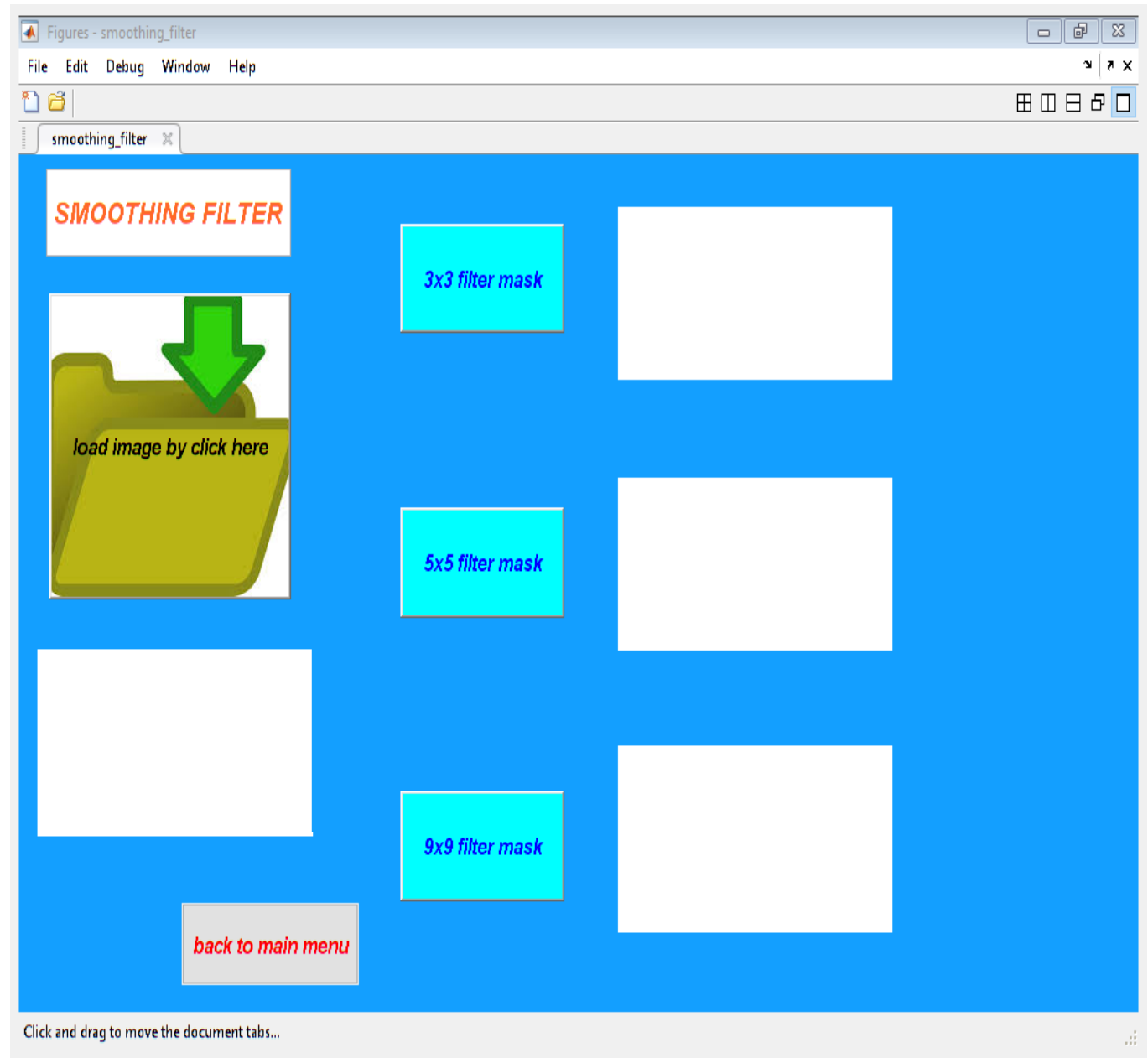
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a
%[im ,path]=uigetfile('*.','enter the image');
%im=strcat(path,im);
%a=imread(im);
h=zeros(9,9);
for i=1:9
    for j=1:9
        h(i,j)=1/50;

```

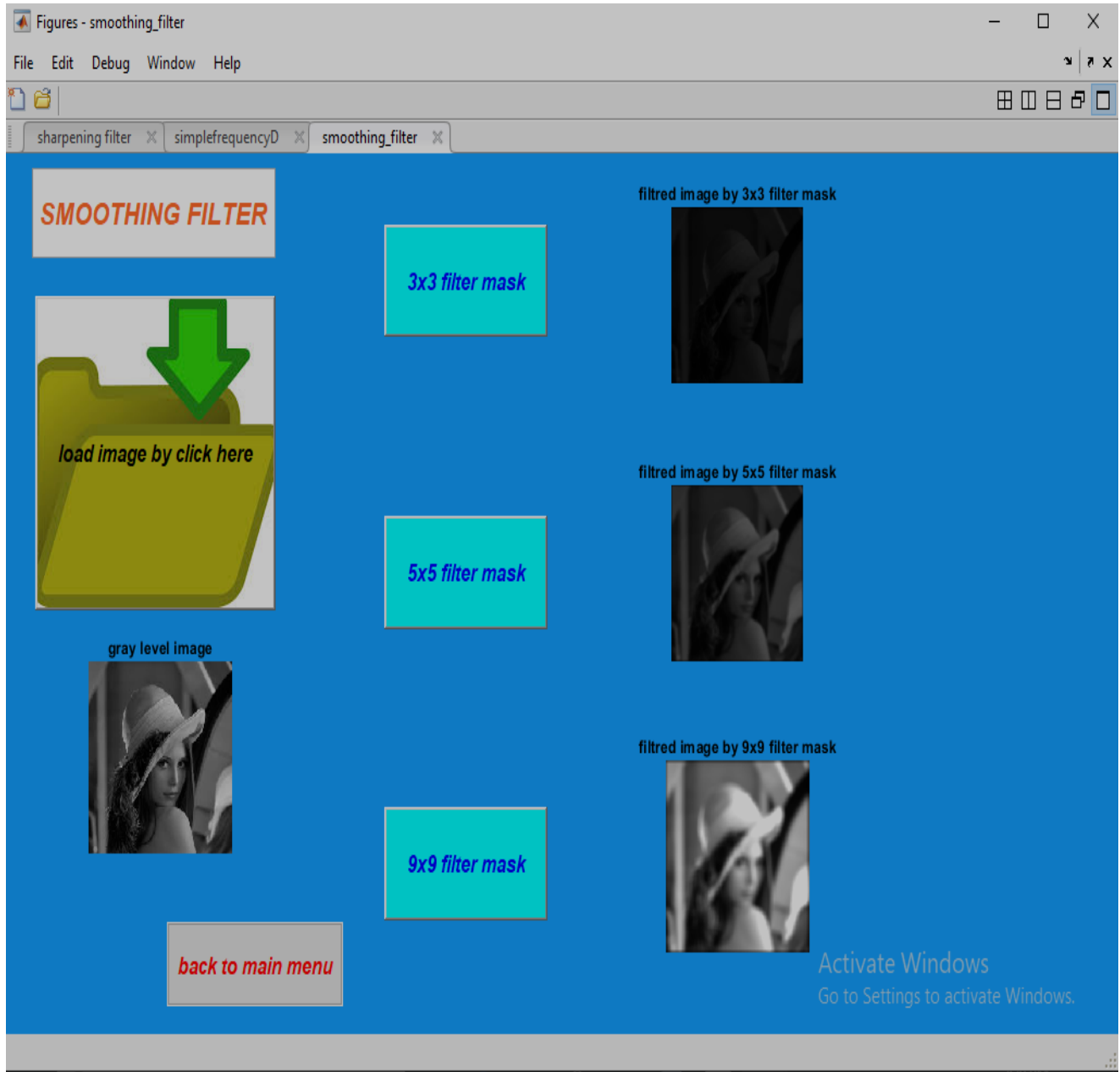


```
        b=imfilter(a,h);
    end
end
axes(handles.axes3);
imshow(b);
title('filtred image by 9x9 filter mask');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
gui_mini_project
```



THE RESULT AFTER LOAD IMAGE AND APPLY CHOICES:



2. Apply sharpening filter using Laplacian and Sobel operator

```
function varargout = sharpening(varargin)
% SHARPENING MATLAB code for sharpening.fig
%     SHARPENING, by itself, creates a new SHARPENING or raises the existing
%     singleton*.
%
%     H = SHARPENING returns the handle to a new SHARPENING or the handle to
%     the existing singleton*.
%
%     SHARPENING('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in SHARPENING.M with the given input arguments.
%
%     SHARPENING('Property','Value',...) creates a new SHARPENING or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before sharpening_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to sharpening_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 sharpening

% Last Modified by GUIDE v2.5 05-May-2019 17:33:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @sharpening_OpeningFcn, ...
                  'gui_OutputFcn',  @sharpening_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 sharpening is made visible.
function sharpening_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 sharpening (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = sharpening_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;

% --- Executes on button press in pushbutton2.
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 b

[im, cancel]=imgetfile();
if cancel
    msgbox(sprintf('invalid selection'),'error','error');
    return
end
a=imread(im);
axes(handles.axes1);
imshow(a);
title('ORIGINAL IMAGE');
b=rgb2gray(a);
axes(handles.axes2);
imshow(b);

```

```

title('gray scale image');

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
sigma=0.4;      % sigma characterizes the amplitude of edges in image.
alpha=0.5;      % alpha controls highlight of fine detail details
c=locallapfilt(b,sigma,alpha);    %built in function
axes(handles.axes3);
imshow(c);
title('sharpened image using laplacian');

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
d=edge(b,'sobel');      % built in function
axes(handles.axes4);
imshow(d);
title('sharpened by sobel operator');

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

% Hints: get(hObject,'String') returns contents of sigma as text
%        str2double(get(hObject,'String')) returns contents of sigma as a double

% --- Executes during object creation, after setting all properties.
function sigma_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sigma (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

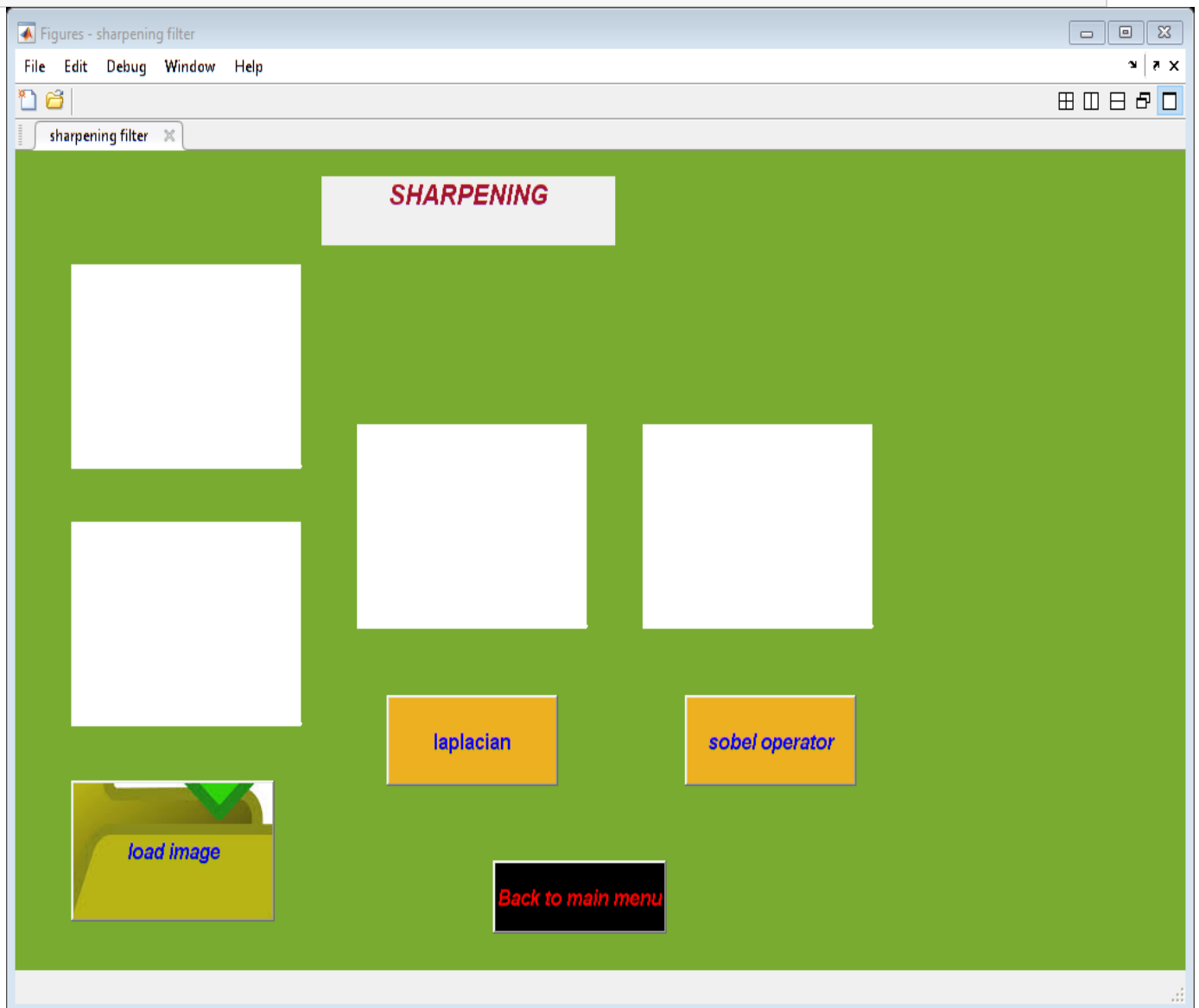
```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

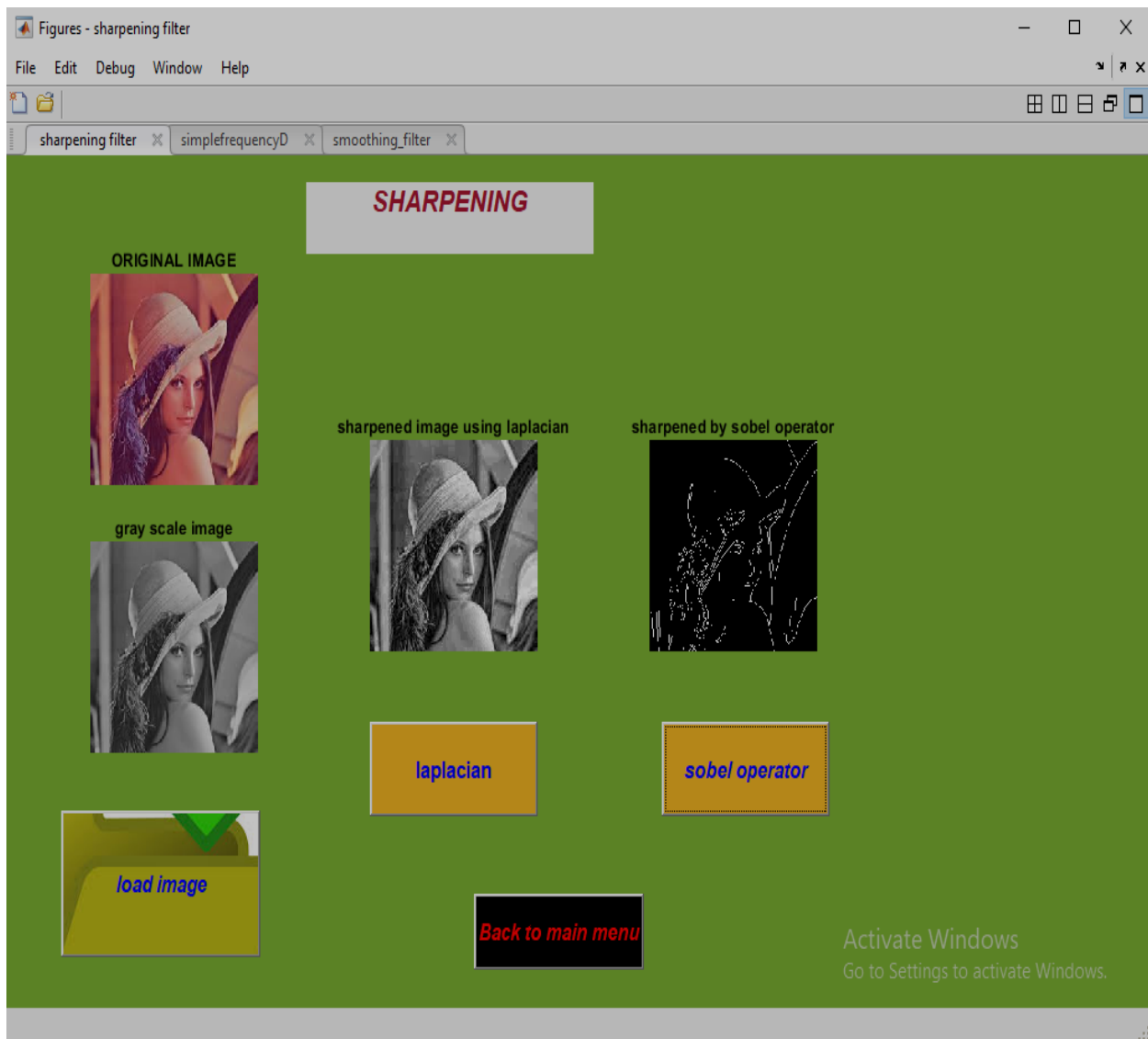
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
gui_mini_project    % use to connect with main menu

```



Published with MATLAB® R2018b

THE RESULT AFTER LOAD IMAGE AND APPLY CHOICES:



3. Apply frequency domain image processing using Butterworth low pass, Butterworth high pass, Gaussian low pass and Gaussian high pass with different cut-off frequencies.

```
function[H]=btwlpfilt(im,d,n) %create function for butterworth low pass filter
h=size(im,1); %height of the image that perform filtering on it
w=size(im,2); %width of the image that perform filtering on it
[x ,y]=meshgrid(-floor(w/2):floor(w-1)/2,-floor(h/2):floor(h-2)/2); % used to equalize
size of filer with image

H=1./ (1.+((x.^2+y.^2).^0.5)./d).^ (2.*n)); % formula of BLPF  $D(u,v)=(x.^2+y.^2).^0.5$ 

%D0=d

end
```

```
function[H]=btwhpfilt(im,d,n) %create function for butterworth highpass filter
h=size(im,1); %height of the image that perform filtering on it
w=size(im,2); %width of the image that perform filtering on it
[x ,y]=meshgrid(-floor(w/2):floor(w-1)/2,-floor(h/2):floor(h-2)/2);
H=1./ (1.+(d./ ((x.^2+y.^2).^0.5)).^ (2.*n)); % formula of BHPF

end
```

```
function[H]=gausslpfilt(im,d,n) %create function for butterworth highpass filter
h=size(im,1); %height of the image that perform filtering on it
w=size(im,2); %width of the image that perform filtering on it
[x ,y]=meshgrid(-floor(w/2):floor(w-1)/2,-floor(h/2):floor(h-2)/2);
H=exp(-(x.^2+y.^2).^2./2*d.^2); % formula of GLPF

end
```

```
function[H]=gausshpfilt(im,d,n) %create function for butterworth highpass filter
h=size(im,1); %height of the image that perform filtering on it
w=size(im,2); %width of the image that perform filtering on it
[x ,y]=meshgrid(-floor(w/2):floor(w-1)/2,-floor(h/2):floor(h-2)/2);
H=1.-(exp(-(x.^2+y.^2).^2./2*d.^2)); % formula of GHPF

end
```

```

function varargout = simplefrequencyD(varargin)
% SIMPLEFREQUENCYD MATLAB code for simplefrequencyD.fig
%     SIMPLEFREQUENCYD, by itself, creates a new SIMPLEFREQUENCYD or raises the
existing
%     singleton*.
%
%     H = SIMPLEFREQUENCYD returns the handle to a new SIMPLEFREQUENCYD or the handle
to
%     the existing singleton*.
%
%     SIMPLEFREQUENCYD('CALLBACK',hObject,eventData,handles,...) calls the local
function named CALLBACK in SIMPLEFREQUENCYD.M with the given input arguments.
%
%     SIMPLEFREQUENCYD('Property','Value',...) creates a new SIMPLEFREQUENCYD or
raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before simplefrequencyD_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to simplefrequencyD_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 simplefrequencyD

% Last Modified by GUIDE v2.5 05-May-2019 17:44:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @simplefrequencyD_OpeningFcn, ...
                  'gui_OutputFcn',  @simplefrequencyD_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 simplefrequencyD is made visible.
function simplefrequencyD_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 simplefrequencyD (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = simplefrequencyD_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;

% --- 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 b
[im, cancel]=imgetfile();      % used to read image from file
if cancel
    msgbox(sprintf('invalid selection'),'error','error'); % if cut the path without
    selecting image it return...
    return % message box
end
a=imread(im);
b=rgb2gray(a); % if the input image is first in gray scale no need to
convert
axes(handles.axes1);
imshow(b);
title('gray scale image');

```

```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
d=get(handles.cutoff,'string'); % to user enter cut off frequency on edit text

ft=fft2(b); % image transformed to frequency domain
c=fftshift(ft); % centered image
h=btwlpfilt(b,d,2); % btwlpfilt is the funtion that i created to create filter for
corresponding image
g=h.*c; % multiplication of filter and image in frequency domain

R=ifft2(g); % inverse fourior transform to get image in spatial domain
axes(handles.axes2); % display image on second(2)axis
imshow(R, []);
title('filtered image by BLPF');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
d=get(handles.cutoff,'string');
c=fftshift(fft2(b));
h=btwhpfilt(b,d,2); % btwhpfilt is the funtion that i created to create filter
for corresponding image
g=h.*c;
R=(ifft2(g));
axes(handles.axes3);
fftshift(imshow(R, []));
title('filtered image by BHPF');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
d=get(handles.cutoff,'string');
c=fftshift(fft2(b));
h=gausslpfilt(b,d,2); % call function "gausslpfilt" already created
g=h.*c;
R=(ifft2(g));
axes(handles.axes4);
fftshift(imshow(R, []));

```

```

title(' filtered image  by GLPF');

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
d=get(handles.cutoff,'string');
c=fftshift(fft2(b));
h=gausshpfilt(b,d,2); % call function gauss hp filter
g=h.*c;
R=(ifft2(g));
axes(handles.axes5);
fftshift(imshow(R,[]));
title('filtered  image by GHPF');

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

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

% Hints: get(hObject,'String') returns contents of cutoff as text

```

```

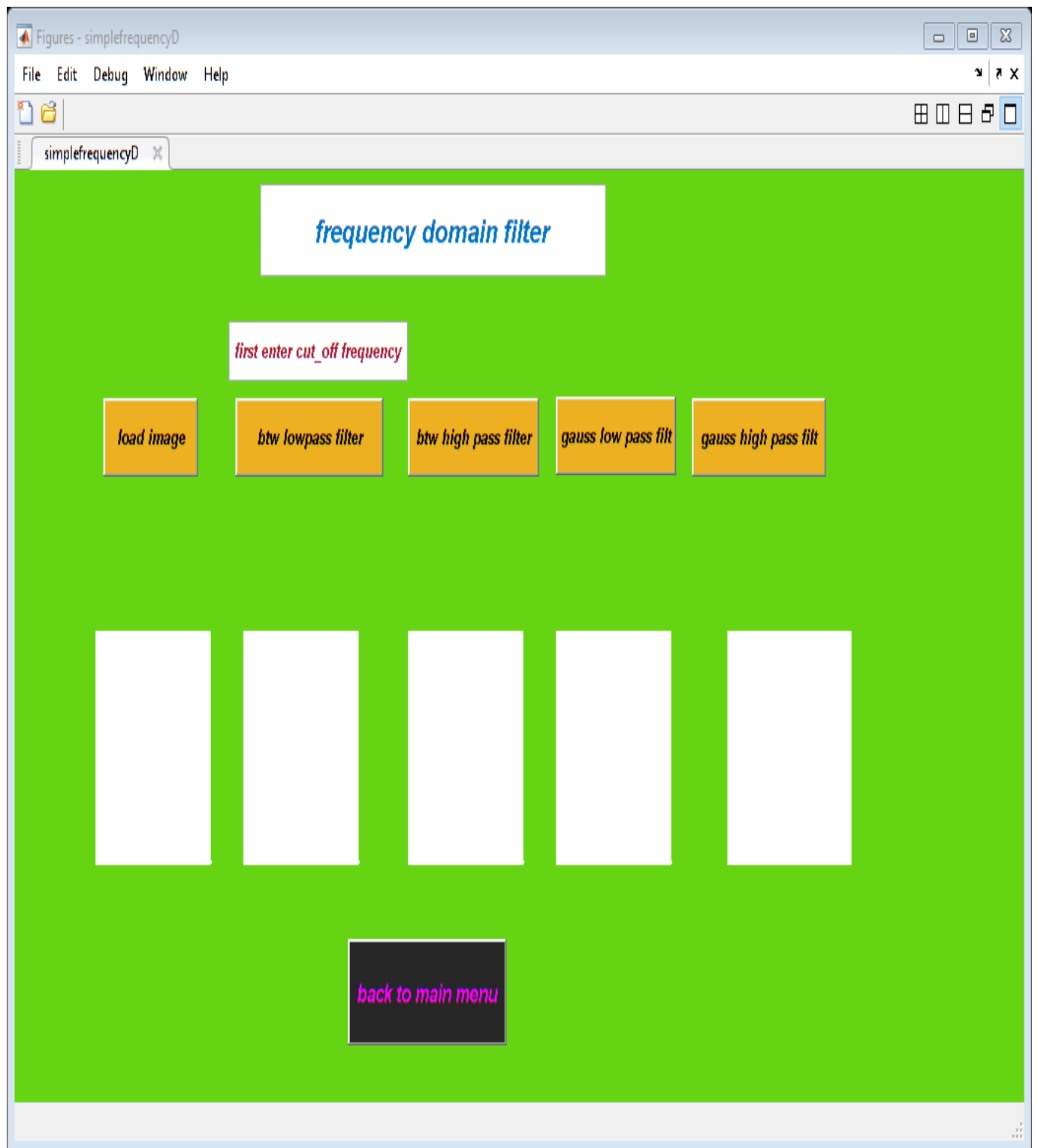
%         str2double(get(hObject,'String')) returns contents of cutoff as a double

% --- Executes during object creation, after setting all properties.
function cutoff_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cutoff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
gui_mini_project    % use to go back to main menu

```



THE RESULT AFTER LOAD IMAGE AND APPLY CHOICES:

