

PART A

PROGRAM: 01

Write a Program to read a digital image. Split and display image into 4 quadrants, up, down, right, and left.

Without GUI

```
img = imread("C:\Users\Student\Music\dip\01.jpg");
figure,imshow(img);
title("Original image");
[r c l] = size(img);
img1 = img(1:r/2,1:c/2,:);
img2 = img(r/2:r,1:c/2,:);
img3 = img(1:r/2,c/2:c,:);
img4 = img(r/2:r,c/2:c,:);

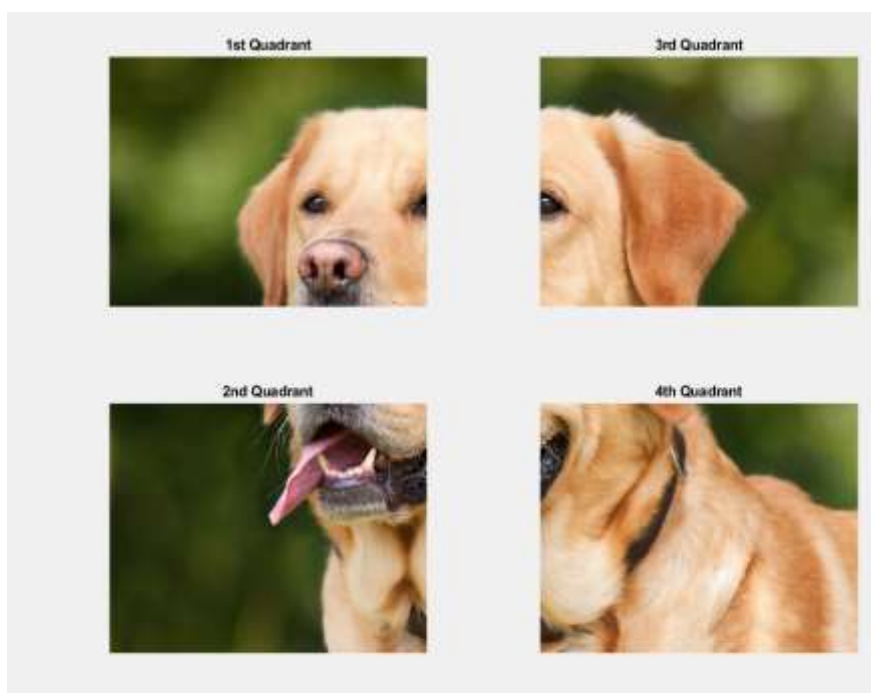
subplot(2,2,1),imshow(img1);
title("1st Quadrant");

subplot(2,2,3),imshow(img2);
title("2nd Quadrant");

subplot(2,2,2),imshow(img3);
title("3rd Quadrant");

subplot(2,2,4),imshow(img4);
title("4th Quadrant");
```

Output



With GUI

```
function pushbutton1_Callback(hObject,eventdata,handles)
```

```
global img
```

```
[file path]=uigetfile("*.jpg");
```

```
img=imread([path file]);
```

```
axes(handles.axes1),imshow(img);
```

```
function pushbutton2_Callback(hObject,eventdata,handles)
```

```
global img
```

```
[r c l]=size([img]);
```

```
img3=img(1:r/2,c/2:c,:);
```

```
img2=img(r/2:r,1:c/2,:);
```

```
img1=img(1:r/2,1:c/2,:);
```

```
img4=img(r/2:r,c/2:c,:);
```

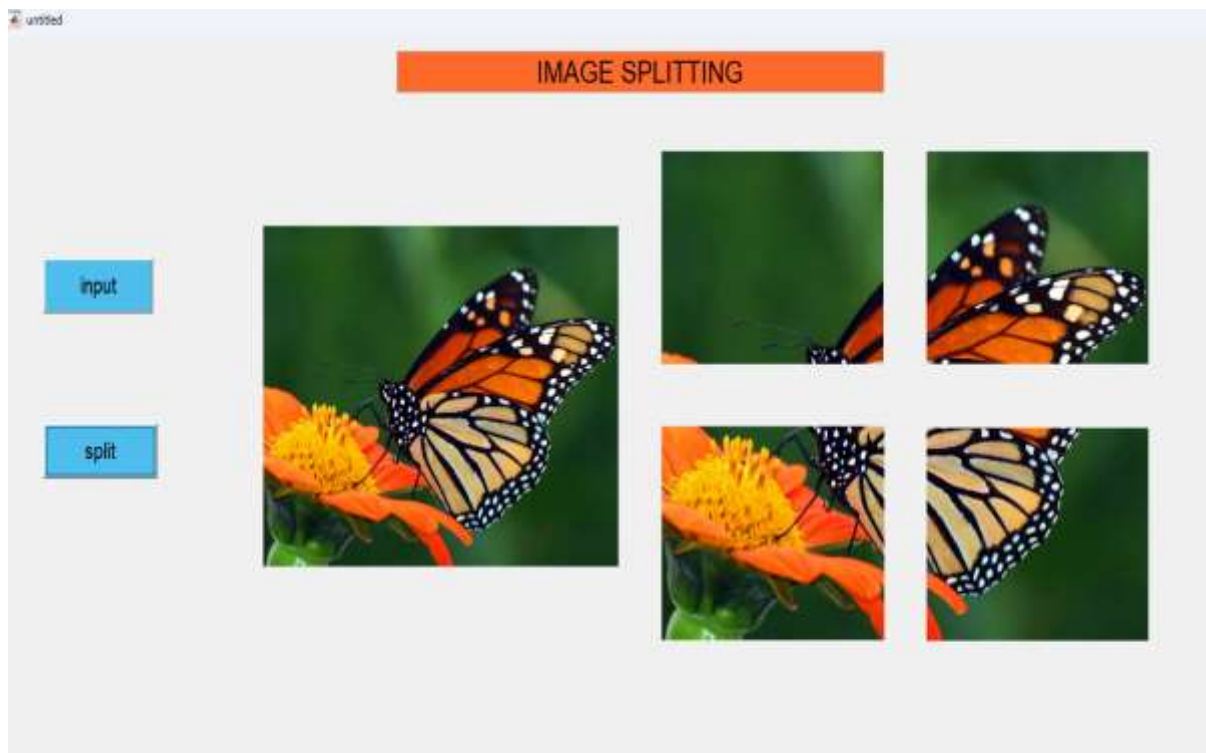
```
axes(handles.axes2),imshow(img1);
```

```
axes(handles.axes3),imshow(img2);
```

```
axes(handles.axes4),imshow(img3);
```

```
axes(handles.axes5),imshow(img4);
```

Output



PROGRAM: 02

Write a program to show rotation, scaling, and translation of an image

Without GUI

```
img=imread("bridge.jpg");
subplot(2,2,1);
imshow(img);
title("ORIGINAL IMAGE");
img1=imresize(img,0.1);
subplot(2,2,2);
imshow(img1);
title("SCALED IMAGE");
img2=imrotate(img,180);
subplot(2,2,3);
imshow(img2);
title("ROTATED IMAGE");
img3=imtranslate(img,[100,100],"FillValues",100);
subplot(2,2,4);
imshow(img3);
title("TRANSLATED IMAGE");
```

Output



With GUI

```
function pushbutton1_Callback(hObject,eventdata,handles)
```

```
global img
```

```
[file path]=uigetfile("*.jpg");
```

```
img=imread([path file]);
```

```
axes(handles.axes1),imshow(img);
```

```
function pushbutton2_Callback(hObject,eventdata,handles)
```

```
global img
```

```
img1=imresize(img,0.1);
```

```
img2=imrotate(img,180);
```

```
img3=imtranslate(img,[100,100],"FillValues",100);
```

```
axes(handles.axes2),imshow(img1);
```

```
title("SCALED IMAGE");
```

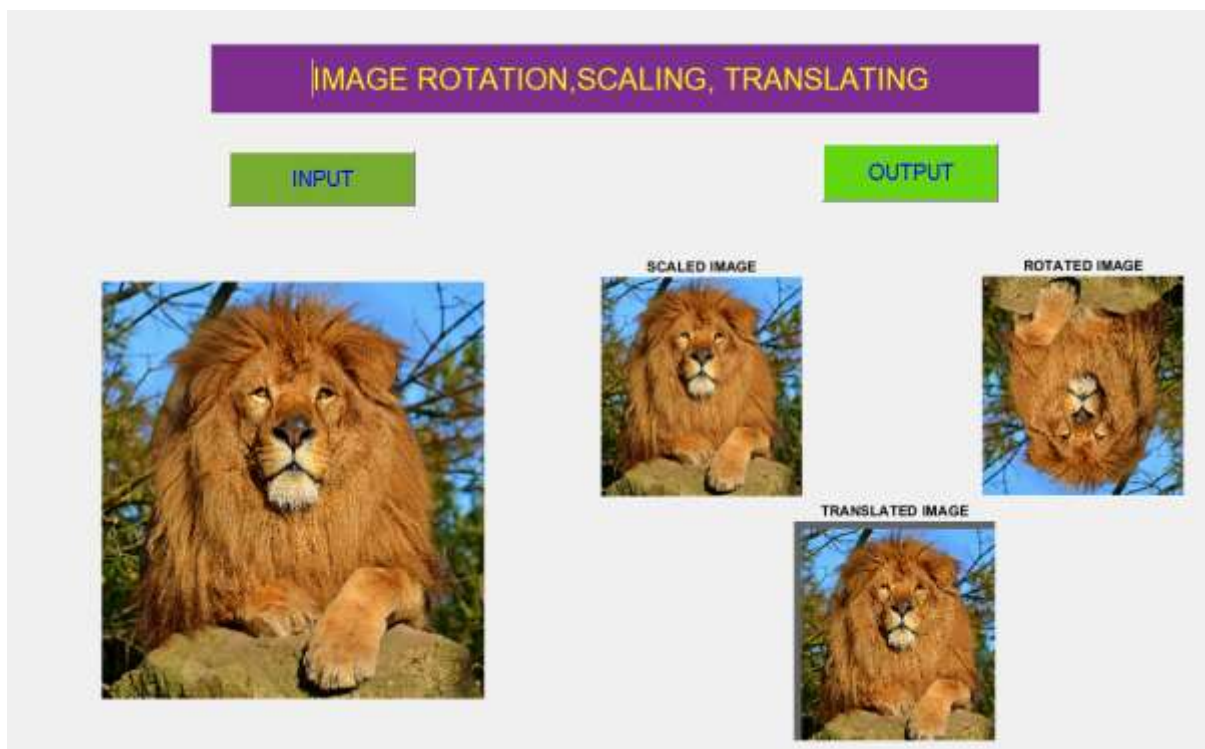
```
axes(handles.axes3),imshow(img2);
```

```
title("ROTATED IMAGE");
```

```
axes(handles.axes4),imshow(img3);
```

```
title("TRANSLATED IMAGE");
```

Output



PROGRAM: 03

Read an image, first apply erosion to the image and then subtract the result from the original. Demonstrate the difference in the edge image if you use dilation instead of erosion.

Without GUI

```
img=imread("firewatch-olly-moss-game-wallpaper.jpg");
subplot(2,2,1),imshow(img);
title("ORIGINAL IMG");
sel=strel('diamond',6);
eroded_img=imerode(img,sel);
subplot(2,2,2),imshow(eroded_img);
title('IMAGE AFTER EROSION');
sel=strel('disk',15);
dialated_img=imdilate(img,sel);
subplot(2,2,3),imshow(dialated_img);
title('IMAGE AFTER DIALATION');
```

Output



With GUI

```
function pushbutton1_Callback(hObject,eventdata,handles)
```

```
global img
```

```
[file path]=uigetfile("*.jpg");
```

```
img=imread([path file]);
```

```
axes(handles.axes1),imshow(img);
```

```
function pushbutton2_Callback(hObject,eventdata,handles)
```

```
global img
```

```
sel=strel('diamond',6);
```

```
eroded_img=imerode(img,sel);
```

```
sel=strel('disk',15);
```

```
dialated_img=imdilate(img,sel);
```

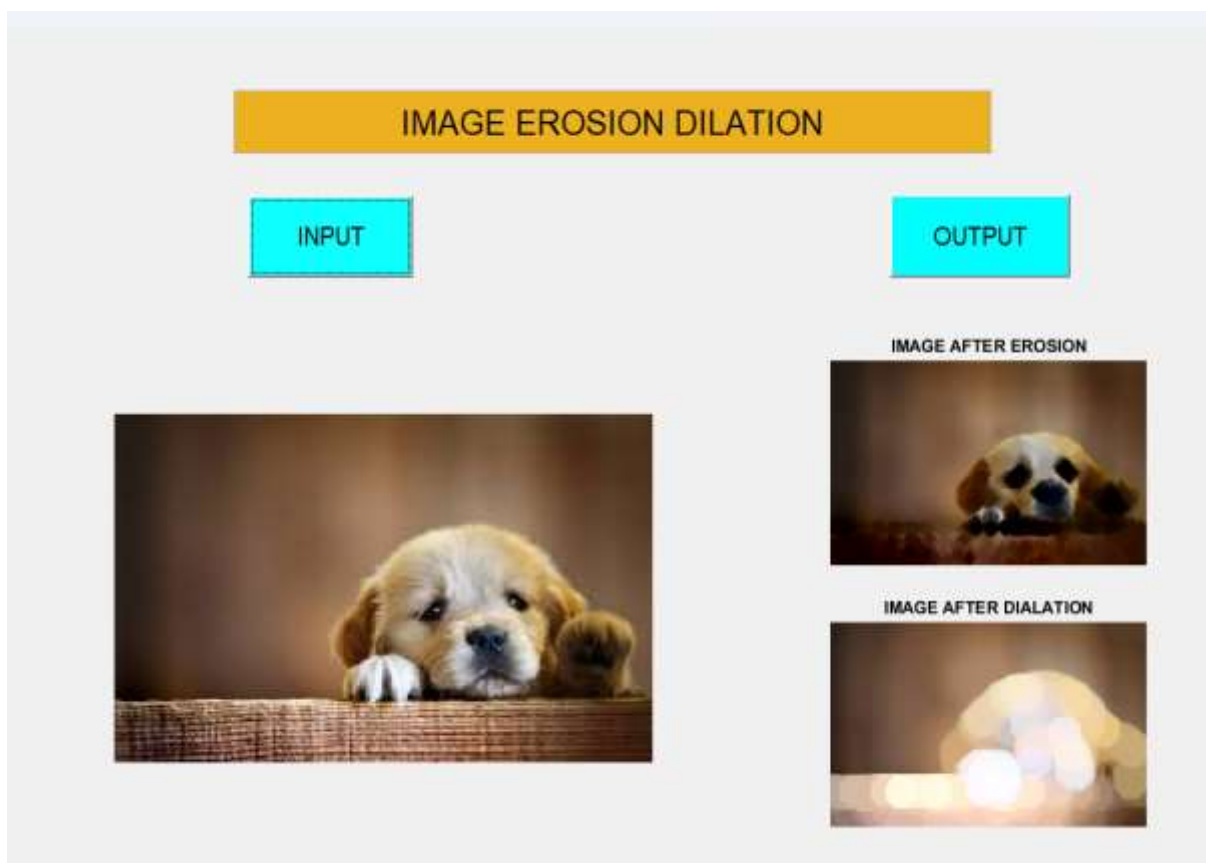
```
axes(handles.axes2),imshow(eroded_img);
```

```
title('IMAGE AFTER EROSION');
```

```
axes(handles.axes3),imshow(dialated_img);
```

```
title('IMAGE AFTER DIALATION');
```

Output



PROGRAM: 04

Read an image and extract and display low-level features such as edges, textures using filtering Techniques.

Without GUI

```
img=imread("number.jpg");  
subplot(2,2,1);  
imshow(img);  
title("ORIGINAL IMAGE");  
i=rgb2gray(img)  
b1=edge(i,"sobel");  
subplot(2,2,2)  
imshow(b1);  
title("SOBEL EDGE");  
b2=edge(i,"canny");  
subplot(2,2,3);  
imshow(b2);  
title("CANNY EDGE");
```

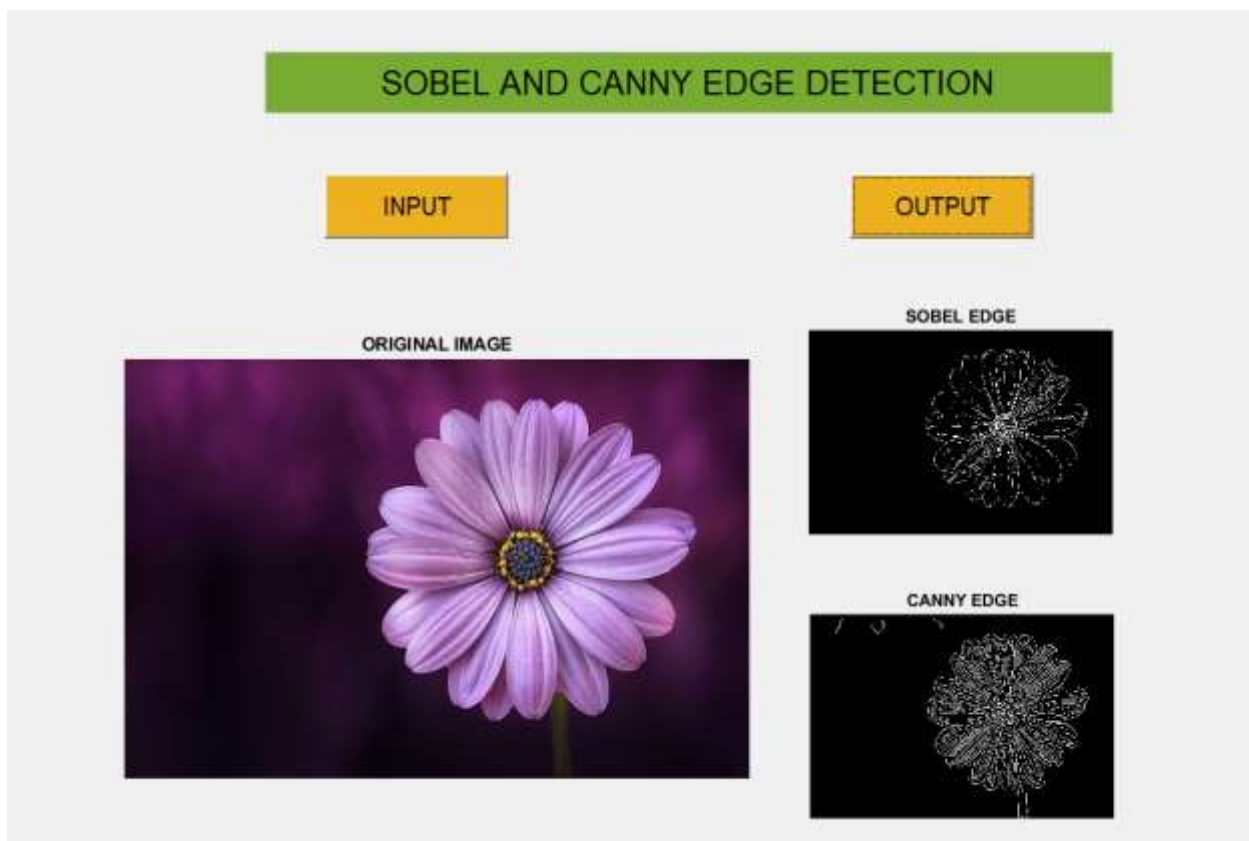
Output



With GUI

```
function pushbutton1_Callback(hObject,eventdata,handles)
global img
[file path]=uigetfile("*.jpg");
img=imread([path file]);
axes(handles.axes1),imshow(img);
title("ORIGINAL IMAGE");

function pushbutton2_Callback(hObject,eventdata,handles)
global img
i=rgb2gray(img)
b1=edge(i,"sobel");
b2=edge(i,"canny");
axes(handles.axes2),imshow(b1);
title("SOBEL EDGE");
axes(handles.axes3),imshow(b2);
title("CANNY EDGE");
```

Output:

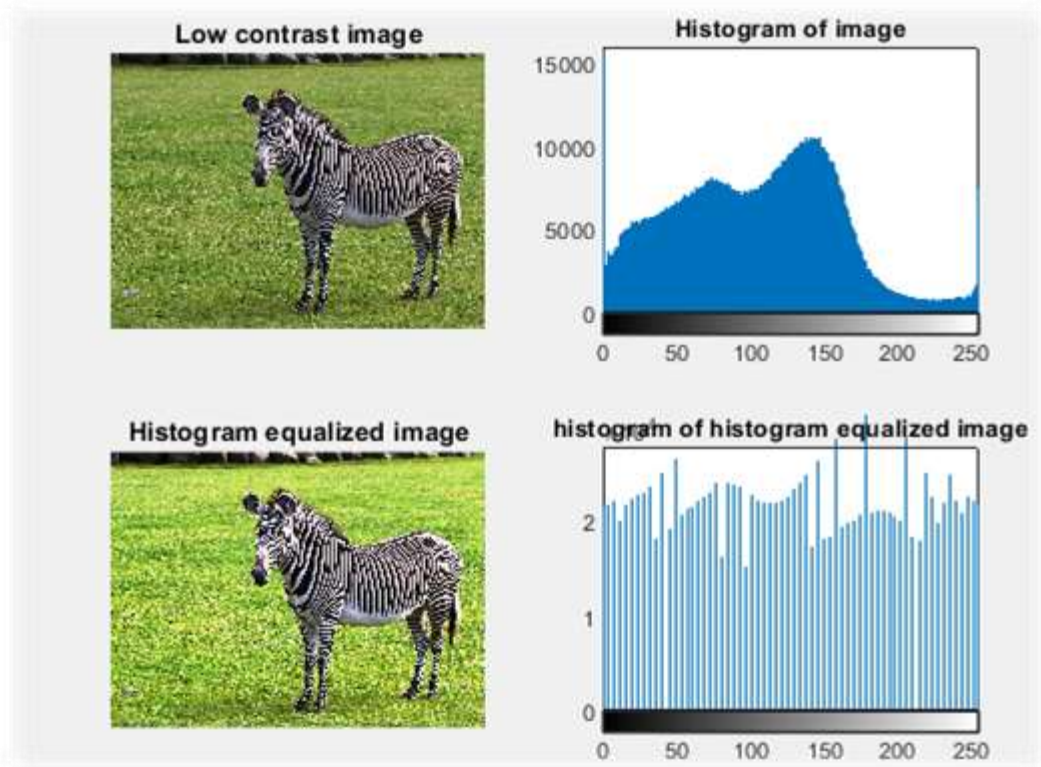
PROGRAM: 05

Demonstrate enhancing and segmenting low contrast 2D images.

Without GUI

```
img=imread("Zenitsu.jpg");  
subplot(2,2,1);  
imshow(img);  
title("Low contrast image");  
subplot(2,2,2);  
  
imhist(img);  
  
title("Histogram of image");  
hist_eqi_img=histeq(img);  
subplot(2,2,3);  
imshow(hist_eqi_img);  
title("Histogram equalized image");  
subplot(2,2,4);  
imhist(hist_eqi_img);  
title("histogram of histogram equalized image");
```

Output



With GUI

```
function varargout = untitled05(varargin)
% UNTITLED05 MATLAB code for untitled05.fig
% UNTITLED05, by itself, creates a new UNTITLED05 or raises the existing
% singleton*.
%
% H = UNTITLED05 returns the handle to a new UNTITLED05 or the handle to
% the existing singleton*.
%
% UNTITLED05('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in UNTITLED05.M with the given input arguments.
%
% UNTITLED05('Property','Value',...) creates a new UNTITLED05 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before untitled05_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to untitled05_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 untitled05

% Last Modified by GUIDE v2.5 21-Jun-2023 11:30:28

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @untitled05_OpeningFcn, ...
                  'gui_OutputFcn', @untitled05_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 untitled05 is made visible.
function untitled05_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 untitled05 (see VARARGIN)

% Choose default command line output for untitled05
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes untitled05 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = untitled05_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;

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

% --- 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 img
[file path]=uigetfile('*.jpg');
img=imread([path file]);
axes(handles.axes1),imshow(img);
title("LOW CONTRAST IMAGE");

% --- 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 img
axes(handles.axes2),imhist(img);
title("HISTOGRAM OF IMAGE");
hist_eqi_img=histeq(img);
axes(handles.axes3),imshow(hist_eqi_img);
title("HISTOGRAM OF EQUALIZED IMAGE");
```

```
axes(handles.axes4),imhist(hist_eqi_img);  
title("HISTOGRAM OF HISTOGRAM EQUALIZED IMAGE");
```

Output

