**ASSIGNMENT #04**

** DIGITAL IMAGE PROCESSING**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA**

**SUBMITTED TO:**

Dr. ALI JAVED

**SUBMITTED BY:**

M. NAEEM UR REHMAN

(17-SE-01)

RAJA FAHAD HAMEED (17-SE-99)

**SUBMISSION DATE:**

26 July ,2020

**Question: Implement proposed Algorithm of Semester Project.**

**CODE Implemented:**

**GUI.m file:**

function varargout = GUI(varargin)

% GUI MATLAB code for GUI.fig

% GUI, by itself, creates a new GUI or raises the existing

% singleton\*.

%

% H = GUI returns the handle to a new GUI or the handle to

% the existing singleton\*.

%

% GUI('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in GUI.M with the given input arguments.

%

% GUI('Property','Value',...) creates a new GUI or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before GUI\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to GUI\_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

% Last Modified by GUIDE v2.5 22-Dec-2015 10:26:59

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @GUI\_OpeningFcn, ...

'gui\_OutputFcn', @GUI\_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 is made visible.

function GUI\_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 (see VARARGIN)

% Choose default command line output for GUI

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = GUI\_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)

[filename,pathname] = imgetfile();

global myimg1;

myimg1=imread(filename);

%img{i} = imread(list(i).name);

if pathname

msgbox(sprintf('Error'),'Error','Error');

return

end

I = imread('image.jpg');

grayimage=rgb2gray(I);

%ad=imadjust(grayimage,[0.1,0.9],[0.0,1.0]);

ad=imadjust(grayimage);

%filtered2=fspecial('average',[3 3]);

%filtered=imfilter(grayimage,filtered2);

filtered=imnoise(ad,'salt & pepper',0);

%sharpen= imsharpen(I);

%contrast=imcontrast(filtered);

%pixel\_avg=mean(filtered(:));

%perprocessed=pixel\_avg-grayimage;

SE=strel('disk',3);

%filtered=imopen(grayimage,SE);

erosion=imerode(filtered,SE);

dilation=imdilate(filtered,SE);

diff=dilation-erosion;

%image\_dilate\_diff=dilation-grayimage;

%image\_erode\_diff=erosion-grayimage;

gradient\_image=filtered-diff;

SE2=strel('diamond',3);

new\_dilation=imdilate(gradient\_image,SE2);

axes(handles.axes1);

imshow(I);

c=edge(new\_dilation,'sobel');

axes(handles.axes2);

imshow(c);

c2=edge(new\_dilation,'prewitt');

axes(handles.axes3);

imshow(c2);

c3=edge(new\_dilation,'canny',0.2);

axes(handles.axes4);

imshow(c3);

**Dip\_final\_proj.m File:**

I = imread('image.jpg');

grayimage=rgb2gray(I);

%ad=imadjust(grayimage,[0.1,0.9],[0.0,1.0]);

ad=imadjust(grayimage);

%filtered2=fspecial('average',[3 3]);

%filtered=imfilter(grayimage,filtered2);

filtered=imnoise(ad,'salt & pepper',0);

%sharpen= imsharpen(I);

%contrast=imcontrast(filtered);

%pixel\_avg=mean(filtered(:));

%perprocessed=pixel\_avg-grayimage;

SE=strel('disk',3);

%filtered=imopen(grayimage,SE);

erosion=imerode(filtered,SE);

dilation=imdilate(filtered,SE);

diff=dilation-erosion;

%image\_dilate\_diff=dilation-grayimage;

%image\_erode\_diff=erosion-grayimage;

gradient\_image=filtered-diff;

SE2=strel('diamond',3);

new\_dilation=imdilate(gradient\_image,SE2);

c=edge(new\_dilation,'sobel');

figure(1);

imshow(c);

c2=edge(new\_dilation,'prewitt');

figure(2);

imshow(c2);

c3=edge(new\_dilation,'canny',0.2);

figure(3);

imshow(c3);

**OUTPUT:**



