

Image Processing in Matlab by using GUI

Project Submitted

in Partial Fulfillment of the Requirements

for the Degree of

# Bachelor of Engineering

in

# Computer Science & Engineering

# 

By

# Sagar Bhandula

# 1811981268

**Submitted to Dr.Abhishek kumar Pandey**

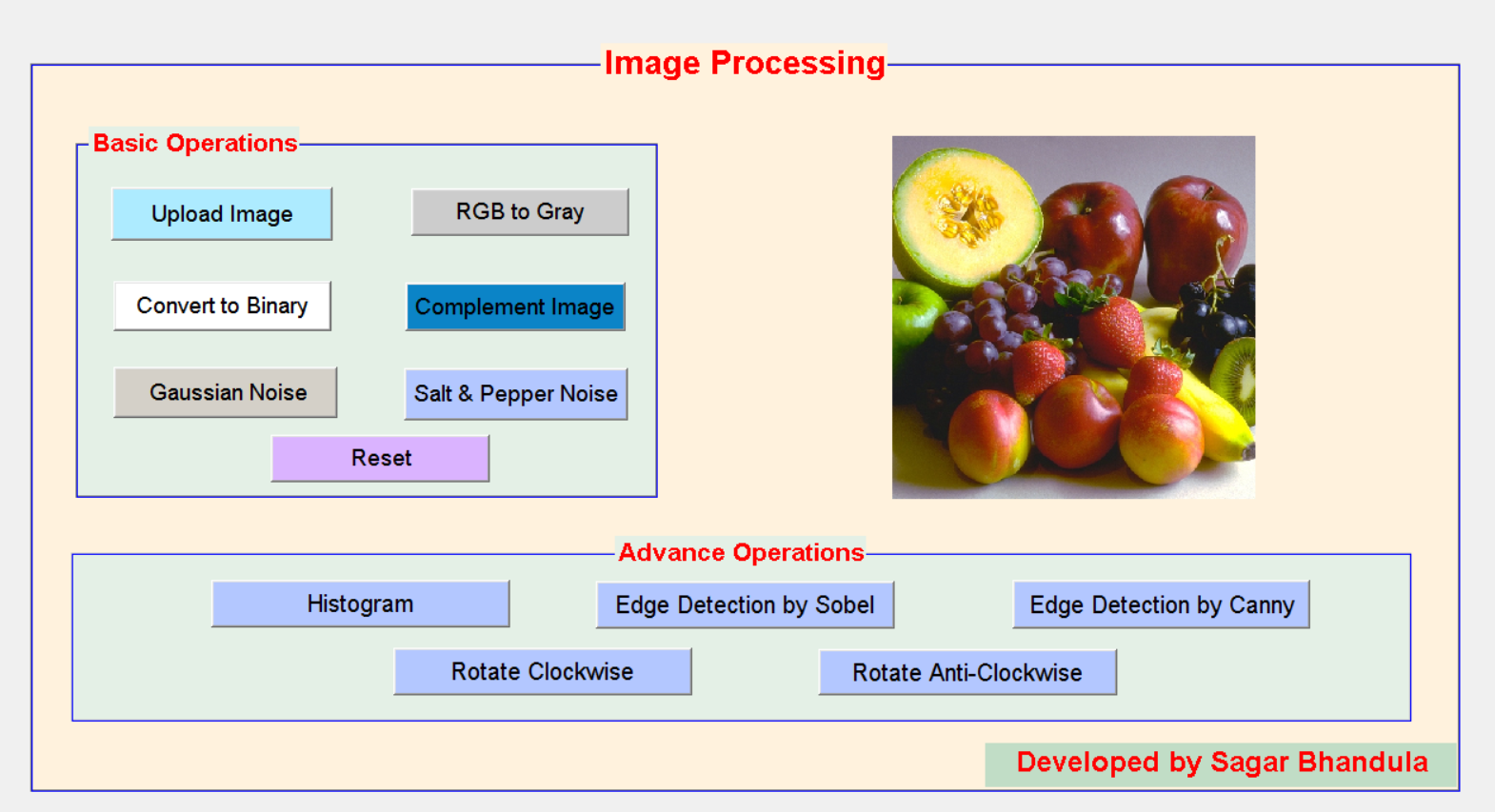
COMPUTER SCIENCE & ENGINEERING

CHITKARA SCHOOL OF ENGINEERING & TECHNOLOGY

CHITKARA UNIVERSITY, HIMACHAL PRADESH, INDIA

July,2020

GUI Of My Project:-



**Highlighted with Gray Background and with yellow color are Extra Things that I have added more to my project.**

function varargout = img\_processing\_GUI(varargin)

% IMG\_PROCESSING\_GUI M-file for img\_processing\_GUI.fig

% IMG\_PROCESSING\_GUI, by itself, creates a new IMG\_PROCESSING\_GUI or raises the existing

% singleton\*.

%

% H = IMG\_PROCESSING\_GUI returns the handle to a new IMG\_PROCESSING\_GUI or the handle to

% the existing singleton\*.

%

% IMG\_PROCESSING\_GUI('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in IMG\_PROCESSING\_GUI.M with the given input arguments.

%

% IMG\_PROCESSING\_GUI('Property','Value',...) creates a new IMG\_PROCESSING\_GUI or raises the

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

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

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to img\_processing\_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 img\_processing\_GUI

% Last Modified by GUIDE v2.5 05-Jul-2020 15:43:49

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @img\_processing\_GUI\_OpeningFcn, ...

'gui\_OutputFcn', @img\_processing\_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 img\_processing\_GUI is made visible.

function img\_processing\_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 img\_processing\_GUI (see VARARGIN)

% Choose default command line output for img\_processing\_GUI

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes img\_processing\_GUI wait for user response (see UIRESUME)

% uiwait(handles.figure1);

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

function varargout = img\_processing\_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 Sobel\_pushbutton7.

function Sobel\_pushbutton7\_Callback(hObject, eventdata, handles)

% hObject handle to Sobel\_pushbutton7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

aedge=a;

aedge=rgb2gray(aedge);

aedge=edge(aedge,'Sobel')'

axes(handles.axes1);

imshow(aedge);

% --- Executes on button press in histogram\_pushbutton8.

function histogram\_pushbutton8\_Callback(hObject, eventdata, handles)

% hObject handle to histogram\_pushbutton8 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

ahist=a;

ahist=rgb2gray(ahist);

axes(handles.axes1);

imhist(ahist);

% --- Executes on button press in Canny\_pushbutton9.

function Canny\_pushbutton9\_Callback(hObject, eventdata, handles)

% hObject handle to Canny\_pushbutton9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

aedge=a;

aedge=rgb2gray(aedge);

aedge=edge(aedge,'Canny')'

axes(handles.axes1);

imshow(aedge);

% --- Executes on button press in rotate\_clockwise\_pushbutton10.

function rotate\_clockwise\_pushbutton10\_Callback(hObject, eventdata, handles)

% hObject handle to rotate\_clockwise\_pushbutton10 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

aclock=a;

aclock=imrotate(aclock,270);

axes(handles.axes1);

imshow(aclock);

% --- Executes on button press in rotate\_anti\_clockwise\_pushbutton11.

function rotate\_anti\_clockwise\_pushbutton11\_Callback(hObject, eventdata, handles)

% hObject handle to rotate\_anti\_clockwise\_pushbutton11 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

aclock=a;

aclock=imrotate(aclock,90);

axes(handles.axes1);

imshow(aclock);

% --- Executes on button press in rgb\_gray\_pushbutton2.

function rgb\_gray\_pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to rgb\_gray\_pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

agray=rgb2gray(a);

axes(handles.axes1);

imshow(agray);

% --- Executes on button press in convert\_binary\_pushbutton2.

function convert\_binary\_pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to convert\_binary\_pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

abw=im2bw(a);

axes(handles.axes1);

imshow(abw);

% --- Executes on button press in complemtn\_imgpushbutton3.

function complemtn\_imgpushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to complemtn\_imgpushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

acomp=a;

acomp=imcomplement(acomp);

axes(handles.axes1);

imshow(acomp);

% --- Executes on button press in Upload\_Img\_pushbutton4.

function Upload\_Img\_pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to Upload\_Img\_pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=uigetfile('sagar.jpg')

a=imread(a);

axes(handles.axes1);

imshow(a);

setappdata(0,'a',a)

% --- Executes on button press in reset\_img\_pushbutton5.

function reset\_img\_pushbutton5\_Callback(hObject, eventdata, handles)

% hObject handle to reset\_img\_pushbutton5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

axes(handles.axes1);

imshow(a);

% --- Executes on button press in Gaussian\_noise\_imgpushbutton6.

function Gaussian\_noise\_imgpushbutton6\_Callback(hObject, eventdata, handles)

% hObject handle to Gaussian\_noise\_imgpushbutton6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

anoise=a;

anoise=imnoise(anoise,'gaussian',0.05);

axes(handles.axes1);

imshow(anoise);

% --- Executes on button press in salt&pepper pushbutton12.

function pushbutton12\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton12 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=getappdata(0,'a');

anoise=a;

anoise=imnoise(anoise,'salt & pepper',0.05);

axes(handles.axes1);

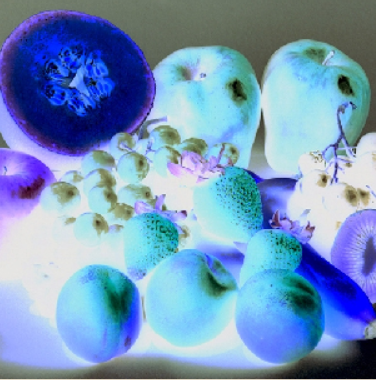
imshow(anoise);

**OUTPUT of My GUI Project**



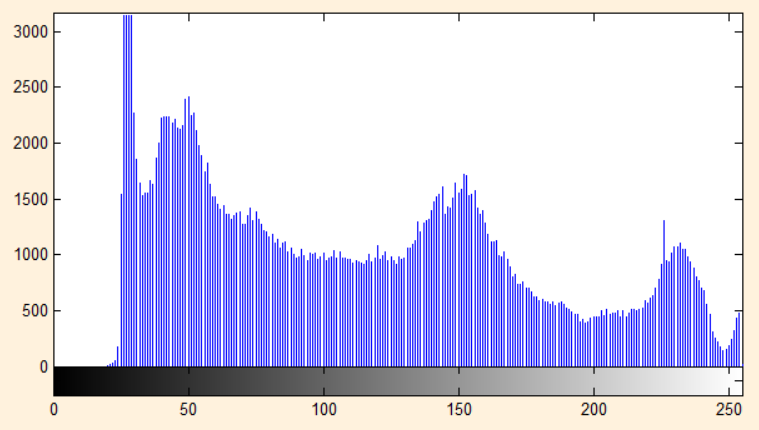


Original Image Gray Image Binary Image





Complement Image Gaussian Noise Image Salt & Pepper Noise Image



Clockwise Image Anti-Clockwise Image Histogram

Edge Detection by Sobel Edge Detection by Canny