Khwopa Engineering College

Course Code :BEG 475 IP

Image Processing and Pattern recogntiion

Lab Report on Basics of MATLAB GUI

Author:
Rabi Raj Khadka

Roll Number:
700324

June 12, 2017

Contents

1 Theory:

1.1 Gray Level Transformation

There are three basic gray level transformation.
1. Linear
2. Logarithmic
3. Power – law

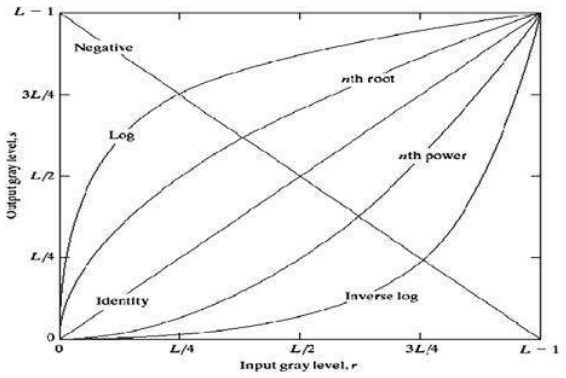

Figure 1: Gray Level Transformatins

The **Logarithmic transformations** can be defined by this formula

$$s = clog(r + 1).$$

Where s and r are the pixel values of the output
and the input image and c is a constant.

The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation, this result in image enhancement and the value of c in the log transform adjust the kind of enhancement we are looking for.

The **Power law transformations** which includes nth power and nth root transformation can be given by the expression:

$$s = cr^{\Gamma}$$

This symbol Γ is called gamma, due to which this transformation is also known as gamma transformation.
Variation in the value of Γ varies the enhancement of the images. Different display devices/monitors have their own gamma correction, that's why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.
Correcting gamma. $s = cr^{\Gamma}$
$s = cr^{(1/2.5)}$

1.2 MATLAB GUI

MATLAB also provides a developer or user to design their application with amazing graphical user interface consisting textbox, axes, push buttons,radio buttons, sliders, inputboxes, popup menus, toggle, table, listbox, panels and manymores.

Commands to enter in GUI mode also known as Graphical User Interface Design Environment is `guide`.

2 Code Description

2.1 Program to Demonstrate Gray Level Transformation

1.Code for Logarithmic and Power Law Transformations

```
%Title: To Demonstrate power and log transformation
%Author: Rabi Raj Khadka
%Date: 6th June 2017
%--------------------------------------------------------------------------
%Three Critical Statements
%--------------------------------------------------------------------------
close all;
clear variables;
clc;
%--------------------------------------------------------------------------
%INPUT
%--------------------------------------------------------------------------
image=imread('C:\Users\rabiraj\Desktop\ImageProcessingLab\neuromancer.jpg');
imshow(image);
imagedouble = im2double(image); %converting the image pixel to double value
x = imagedouble; %declaring an output image of size as size of input image
y = imagedouble;
[r,c] = size(x); %determining the size of row and column
factor = 1;
gamma = 20;
%--------------------------------------------------------------------------
%CALCULATION
%--------------------------------------------------------------------------
for i = 1:r
    for j = 1:c
        x(i,j) = factor*log(1+imagedouble(i,j)); %log transformation
        y(i,j) = factor*imagedouble(i,j)^gamma;  %power transformation
    end
end
%--------------------------------------------------------------------------
%OUTPUT
%--------------------------------------------------------------------------
subplot(1,3,1), imshow(imagedouble), title('Original Image');
subplot(1,3,2), imshow(x), title('Log Transformed Image');
subplot(1,3,3), imshow(y), title('Power Transformed Image');
```

2.2 Programs to Demonstrate Use of GUIDE

1.Function to load image using GUI

```
% --- Executes on button press in load.
function load_Callback(hObject, eventdata, handles)
% hObject    handle to load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global orgim img2
[path, user_cancel]=imgetfile();
if user_cancel
    msgbox(sprintf('Error Occured'),'Error','Error');
    return;
end
orgim=imread(path);
orgim=im2double(orgim);
img2=orgim;
global activewindow
activewindow = 'origim';
set(handles.brightnesslevel,'String','0');
set(handles.brightness_control,'Value',1);
axes(handles.opwindow);
imshow(orgim);
```

2. Function to convert image into B/W and negative

```
% --- Executes on button press in bw.
function bw_Callback(hObject, eventdata, handles)
global orgim
img_bw=im2bw(orgim);
global activewindow
activewindow=img_bw;
set(handles.brightnesslevel,'String','0');
set(handles.brightness_control,'Value',1);
axes(handles.opwindow);
imshow(img_bw);
```

```
% --- Executes on button press in negative.
function negative_Callback(hObject, eventdata, handles)
global orgim
img_negative=1-orgim;
axes(handles.opwindow);
global activewindow
activewindow=img_negative;
set(handles.brightnesslevel,'String','0');
set(handles.brightness_control,'Value',1);
axes(handles.opwindow);
imshow(img_negative);
```

3. Function to convert image into grayscale, reset original image and control brightness of loaded image

```
% --- Executes on button press in grayscale.
function grayscale_Callback(hObject, eventdata, handles)
global orgim
img_grayscale=rgb2gray(orgim);
global activewindow
activewindow=img_grayscale;
set(handles.brightnesslevel,'String','0');
set(handles.brightness_control,'Value',1);
axes(handles.opwindow);
imshow(img_grayscale);
% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
global img2
global activewindow
activewindow=img2;
set(handles.brightnesslevel,'String','0');
set(handles.brightness_control,'Value',1);
axes(handles.opwindow);
imshow(img2);
% --- Executes on slider movement.
function brightness_control_Callback(hObject, eventdata, handles)
global activewindow
val=get(hObject,'Value')-1;
set(handles.brightnesslevel,'String',num2str(val));
imbright= activewindow+val;
axes(handles.opwindow);imshow(imbright);
```

3 Result and Discussion

Gray Level Transformation is used for Image enhancement and we used logarithmic and power law transformations to see how image enhancement is done using Gray level Transformation.
The GUI in the MATLAB is easy to use, just by entering "`guide`" command. There are many functions and interfaces we can use by just dragging and dropping it into design interface. Image is loaded into the UI axes:opwindow and then various operation like converting to black and white, grayscale, negative are done on the copy of original loaded images, for user convenience all active images brightness can be changed through the slider in th right botom of the UI window.
Outputs:

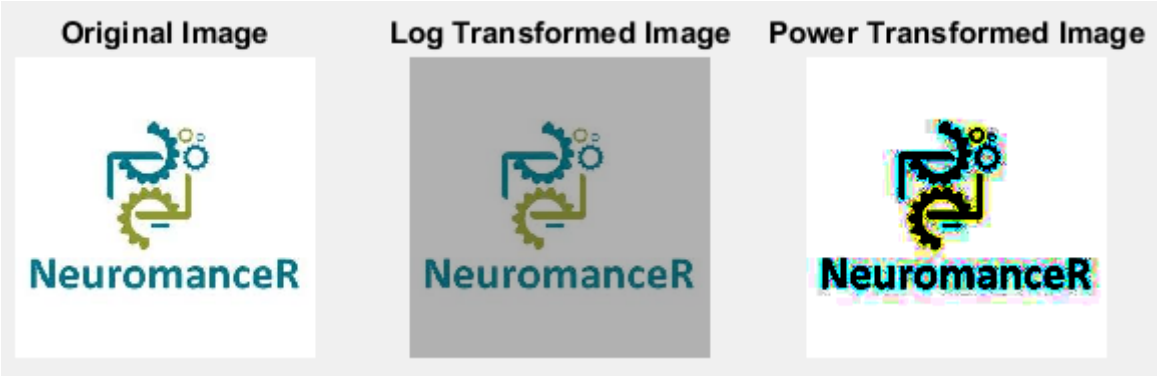

Figure 2: Gray Level Transformations Example

Figure 3: GUI Initial Loaded Window

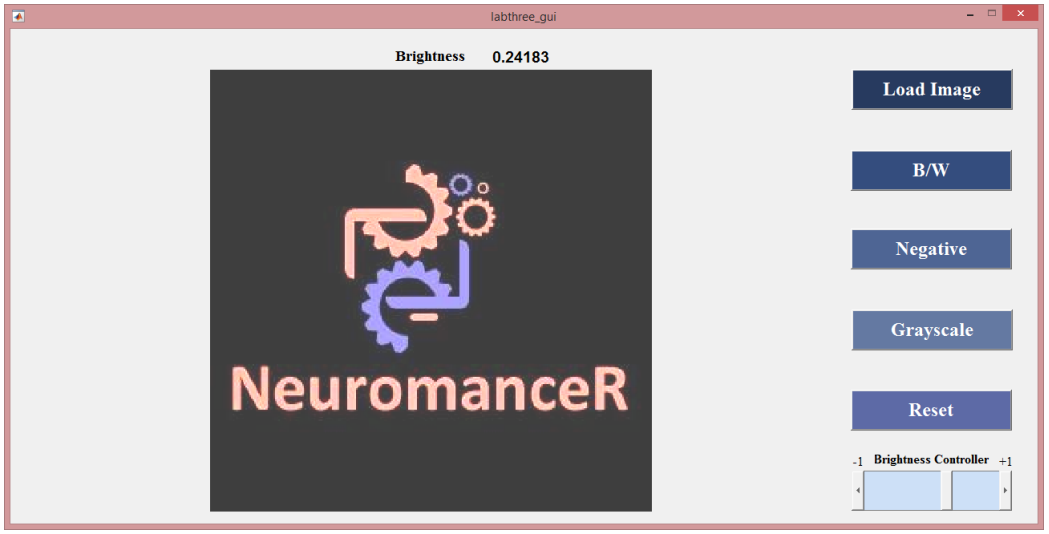

Figure 4: GUI example in action

4 Conclusion

Hence,
We are familiarized with the image enhancement techniques and learned about making an GUI based application for windows platform by the use of Graphical User Interface Design Environment of the MATLAB application.