LAB REPORT - 1

IMAGE PROCESSING

October 19, 2018

Submitted to

Desire SIDIBE

ASRAF ALI Abdul Salam Rasmi

Masters in Computer Vision

Centre Universitaire Condorcet

Universite de Bourgogne

Chapter 1

Contrast Enhancement

Contrast Enhancement is a process by which an user can extract more features from an image. Usually Contrast enhancement is done when the user couldn't acquire necessary information from the input image. Contrast manipulation involves in changing the the range of Intensity values in an image in order to increase contrast. Some techniques of Contrast enhancement like Point Transformation(Power Law), Histogram Equalization and Modification has been implemented in MATLAB.

Chapter 2

Power Law Transformation

2.1 Introduction

Power Law Transformation is one among the Point Transformation technique of Contrast Enhancement. It is generally called as Gamma Correction because the exponential of intensity of each pixel of the image is calculated by a power value $gamma(\gamma)$, such that

$$S = C.r^{\gamma}$$

where,

- C is a constant.
- r is the input image.
- γ is the Power value.

For increasing the brightness of the image γ must be < 1 and for decreasing the brightness γ must be > 1. A function named $powerLawTrans(I, C, gamma)$ has been implemented as follows.

2.2 Codes

```
1 function [P] = powerLawTrans(I,C, gamma)
2 %% POWERLAWTRANS - This fuction enhances the contranst of brighter
     regions of
3 %an image, I.
```

```
4  %   Input  :     I - Input Image
5  %                C - is a constant
6  %                gamma - The power value (generally called gamma
       correction)
7  %
8  %   Output :     P - Output of Power Law Tranformation
9
10 %% The Function starts here
11 I=im2double(I); % Convert Image to Double
12 [r,c]=size(I); % Read the size of image
13
14 P=zeros(r,c); % Intialize the Output Image
15
16 for i=1:r
17     for j=1:c
18         P(i,j)=C*I(i,j).^gamma; % Implementation of Gamma Correction
19     end
20 end
21
22 % Display the result
23 subplot(1,2,1), imshow(I);
24 title('Original Image')
25 subplot(1,2,2), imshow(P);
26 title('Enhanced Image (Gamma Correction)')
27
28 end
```

2.3 OUTPUT

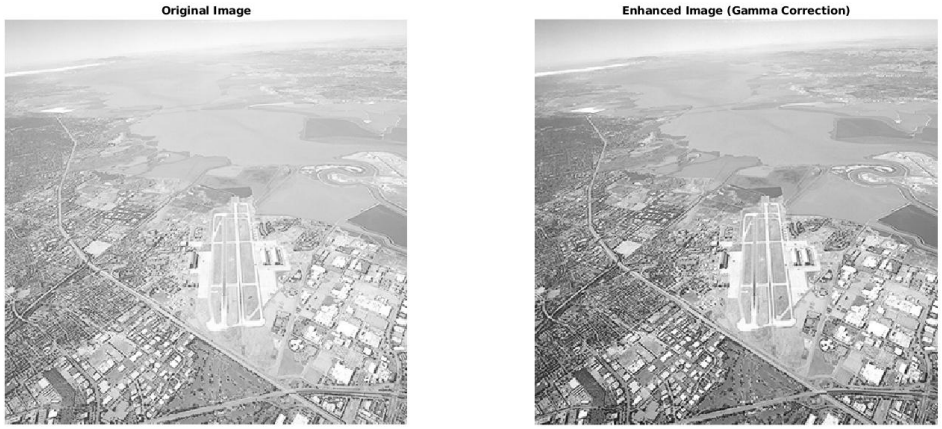

Figure 2.1: Power Law Transformation ($\gamma > 1$)

Figure 2.2: Power Law Transformation ($\gamma < 1$)

Chapter 3

Histogram Stretching

3.1 Introduction

The histogram stretching algorithm is implemented in such a way that the minimum (a) and maximum (b) intensity value of the pixel in the histogram is stretched out (from 0 to 255) on the through the axis of the histogram. A function named $histStretch(I)$ has been implemented as follows. A sample result of the output image is displayed below with its histogram in comparison with an inbuilt MATLAB function $imadjust$

3.2 Codes

```
function [S] = histStretch(I)
%% HISTSTRETCH - This fuction performs Histogram Equalization of the
    given Image, I.
%
%   Input  :    I - Input Image
%
%   Output :    S - Histogram Stretched Image

%% The Function starts here

I = double(I); % Convert Image to Double
[r,c]=size(I); % Read the size of image

```

```
a=min(I(:)); % Minimum Intensity level of given image
b=max(I(:)); % Maximum Intensity level of given image

S=zeros(r,c); % Intialize the Output Image

% The formula for Histogram Stretching has been implemented as follows
for i=1:r
    for j=1:c
        if I(i,j)<a
            S(i,j)=0;
        elseif I(i,j)>b
            S(i,j)=255;
        else
            S(i,j) = 255*(I(i,j)-a)/(b-a);
        end
    end
end

% Display the result
subplot(2,2,1), imshow(I, []);
title('Original Image');
subplot(2,2,2), imshow(S, []);
title('Histogram Stretched Image');
subplot(2,2,3), imhist(uint8(I));
title('Histogram of Original Image');
subplot(2,2,4), imhist(uint8(S));
title('Histogram of Stretched Image');

end
```

3.3 OUTPUT

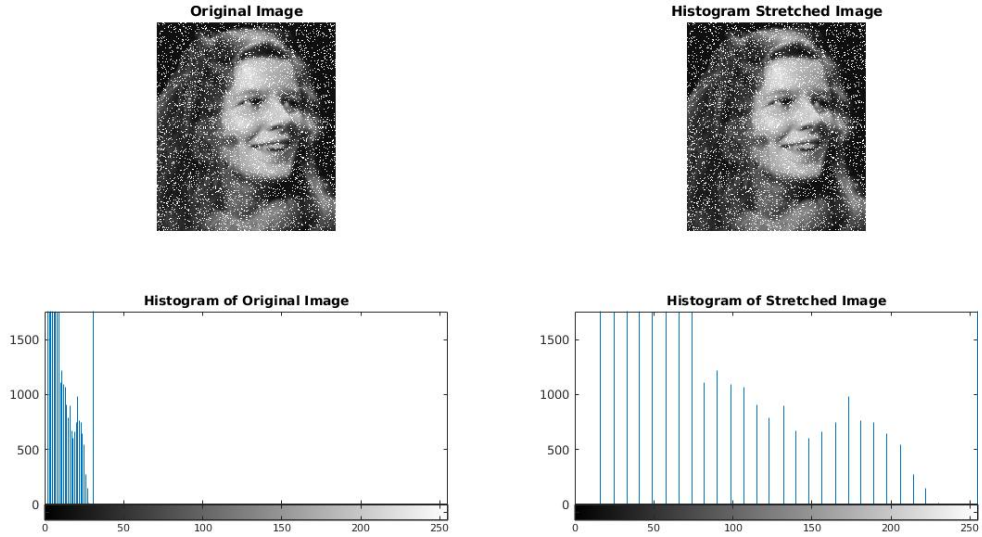

Figure 3.1: Histogram Stretching

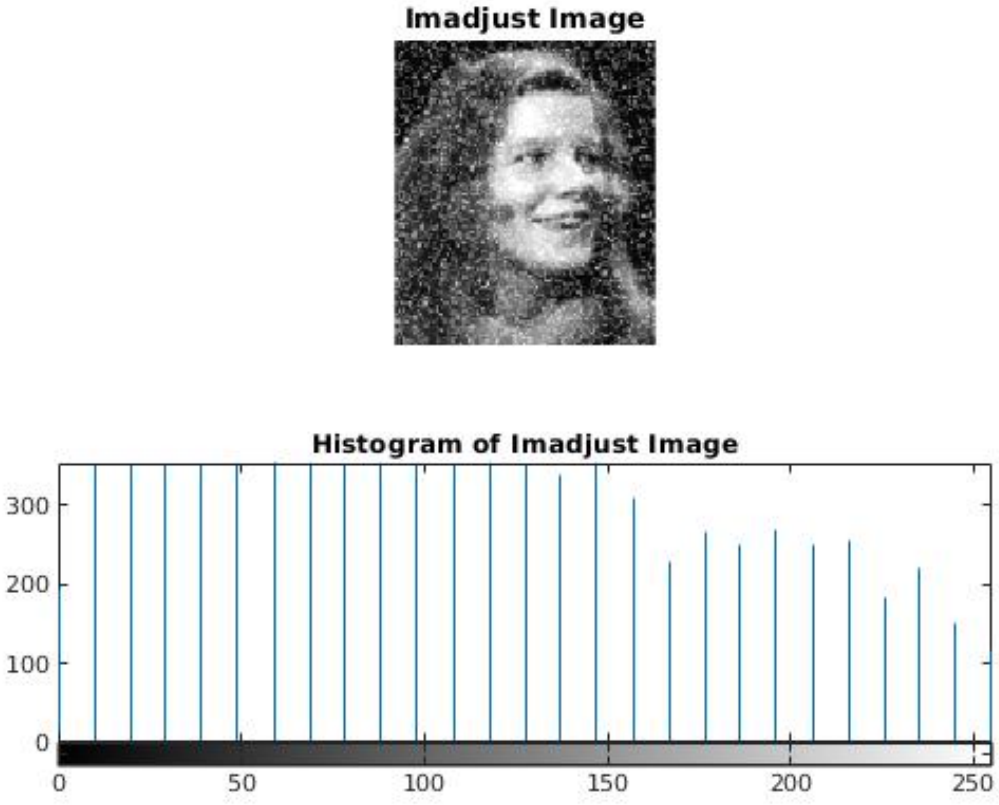

Figure 3.2: Histogram Stretching ($imadjust$)

Chapter 4

Histogram Equalization

4.1 Introduction

Histogram equalization is another technique for adjusting image intensities to enhance contrast.In this technique the Intensities over the pixels of an image will be widespread throughout the image (cumulative intensity).A function named $myHistEq(I)$ has been implemented as follows. A sample result of the output image is displayed below with its histogram in comparison with an inbuilt MATLAB function $histeq$

4.2 Codes

```
function [histequal] = myHistEq(I)
%% MYHISTEQ - This fuction performs Histogram Equalization of the given
    Image, I.
%
%   Input  :    I - Input Image
%
%   Output :    histeq - Histogram Equalized Image

%% The Function starts here

[r,c]=size(I);% Read the size of Image
histequal=uint8(zeros(r,c)); % Intialize the Output Image
n=r*c; % Total no.of Pixels(row X column)
```

```
13
14 % The following variables are initialized as a vector the size 256
15 f=zeros(256,1);
16 pdf=zeros(256,1);
17 cdf=zeros(256,1);
18 cum=zeros(256,1);
19 out=zeros(256,1);
20
21 % To Compute the pdf for each Intensity Value from (0-255)
22 for i=1:r
23     for j=1:c
24         val=I(i,j); % The intensity value of current pixel stored in
               val
25         f(val+1)=f(val+1)+1; %  Icrement the count
26         pdf(val+1)=f(val+1)/n; % pdf
27     end
28 end
29
30 sum=0;
31 L=255; % The total no. of intensity values (-1)
32
33 % To Compute the cdf for each Intensity Value from (0-255)
34 for i=1:size(pdf)
35     sum=sum+f(i);
36     cum(i)=sum;
37     cdf(i)=cum(i)/n; % cdf
38     out(i)=(cdf(i)*L); % cdf X the total intensity value
39 end
40
41 % To feed the equalized value back in input image
42 for i=1:r
43     for j=1:c
44         histequal(i,j)=out(I(i,j)+1);
45     end
46 end
```

```
47
48 % Display the result
49 subplot(1,3,1), imshow(I);
50 title('Original Image');
51 subplot(1,3,2), imshow(histequal);
52 title('Histogram Equalized Image');
53 subplot(1,3,3), imshow(histeq(I));
54 title('Histogram Equalized Image (Inbuilt)');
55
56 end
```

4.3 OUTPUT

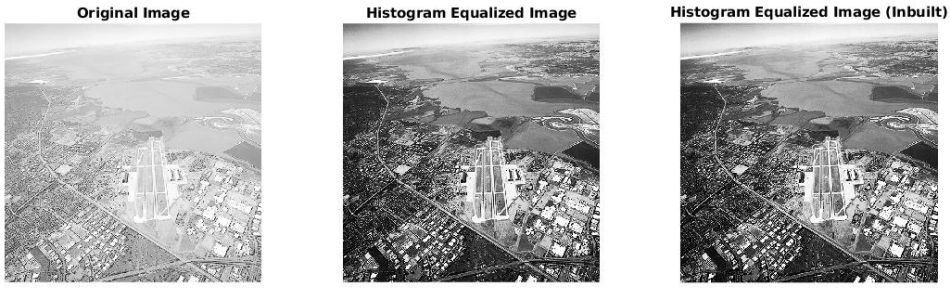

Figure 4.1: Histogram Equalization (aerial.tif)

Figure 4.2: Histogram Equalization (spine.tif)