Department of Electronics Engineering

Image enhancement- Histogram Equalization

Aim:- To write a program in MATLAB/PYTHON editor to plot histogram of the image & perform histogram equalization on it.

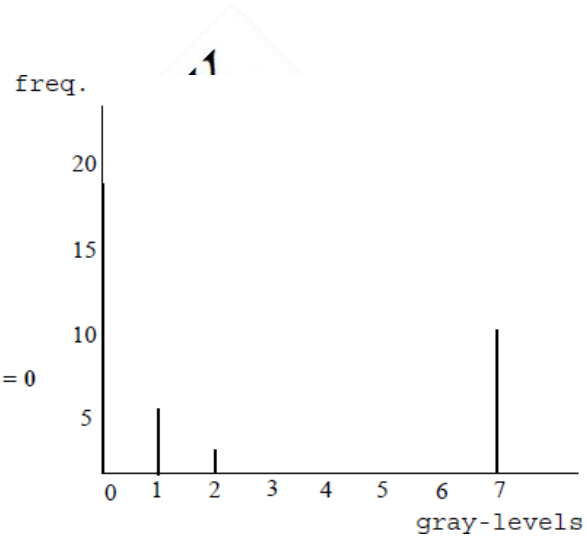
Theory:-

Histogram Equalization

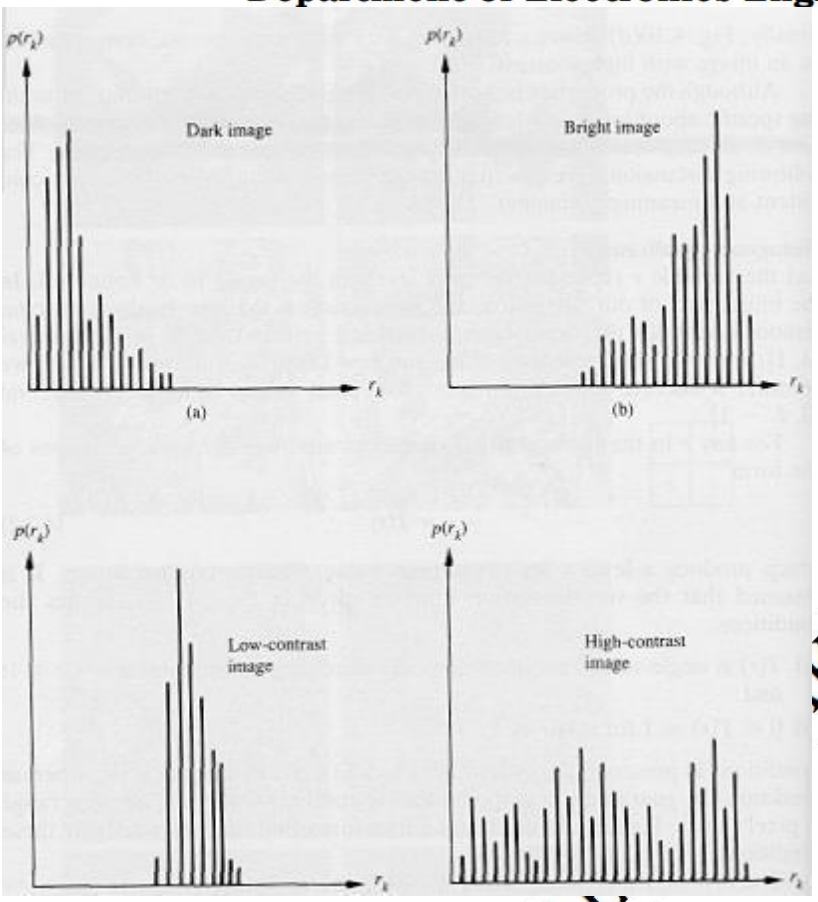
Low contrast images are usually mostly dark, mostly light, or mostly gray.
- *High contrast* images have large regions of dark and large regions of white (e.g., someone inside a room, stading in front of a window on a sunny day).
- *Good contrast* images exhibit a wide range of pixel values (i.e., no single gray level dominates the image).

0	0	1	0	2	0
1	0	7	7	7	0
0	7	0	0	7	0
1	0	0	7	2	0
0	0	7	1	0	1
1	0	7	7	7	0

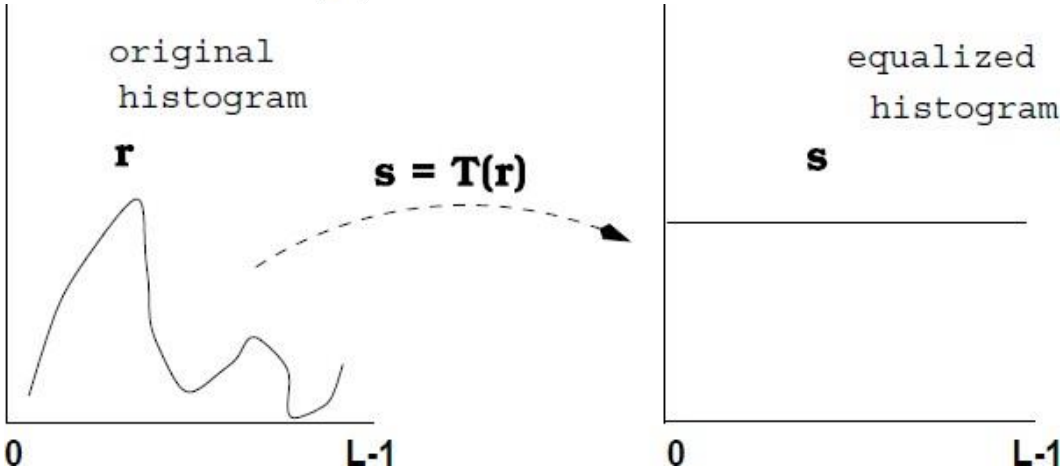
frequencies
f(0)= 18
f(1)= 6
f(2)= 2
f(3)=f(4)=f(5)=f(6) = 0
f(7) = 10



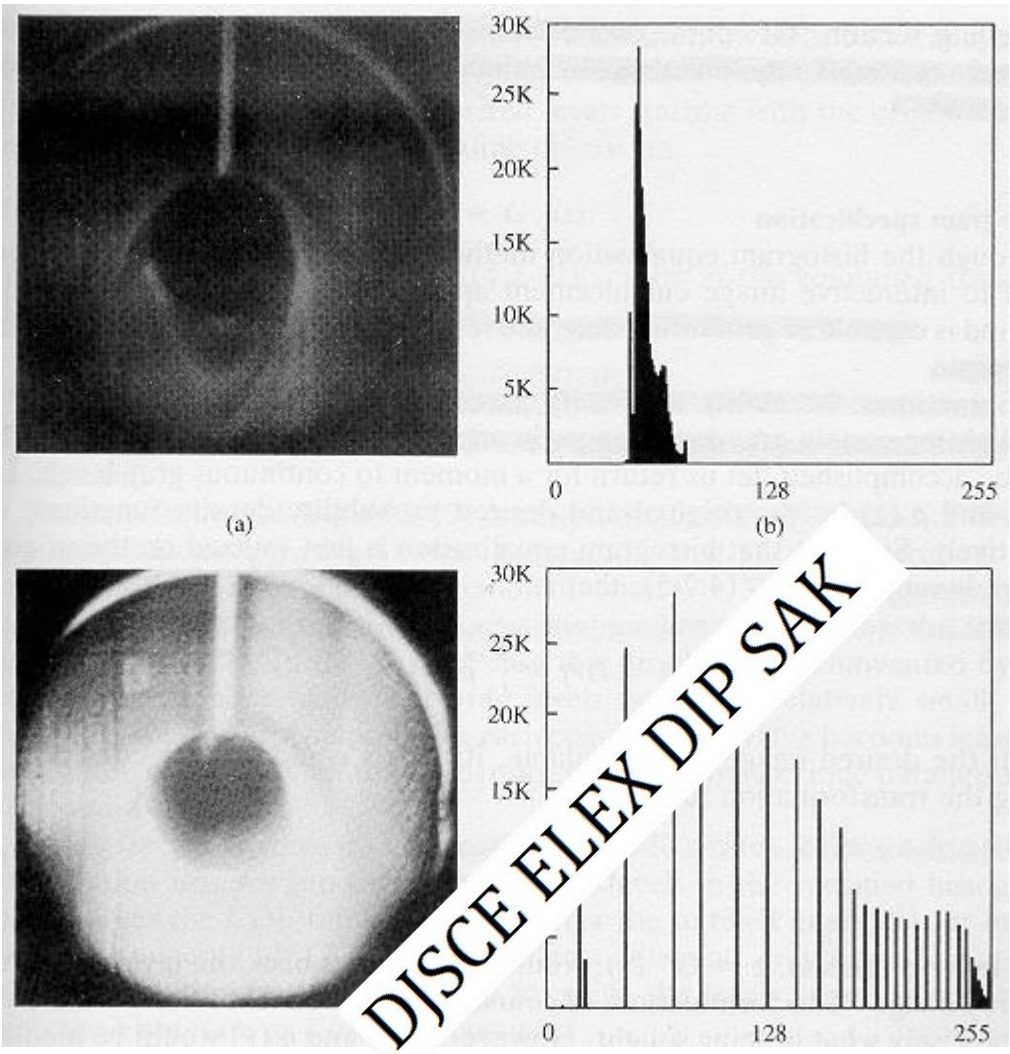
The histogram of an image (i.e., a plot of the gray-level frequences) provides important information regarding the contrast of an image.
* Histogram with a small spread: low contrast image
* Histogram with wide spread: high contrast image
* Histogram clustered at the low end: dark image
* Histogram clustered at the high end: bright image



- Histogram equalization is a transformation that stretches the contrast by redistributing the gray-level values uniformly.
- It is fully automatic compared to other contrast stretching techniques.



- In practice, the histogram might not become totally flat !



Implementation Instructions:

1) Read any low contrast gray scale image. Plot its histogram without using direct function available in MATLAB/PYTHON.
2) Perform histogram equalization using CDF function & plot the equalized histogram as well as equalized image.

1. **Using direct Histogram Function**

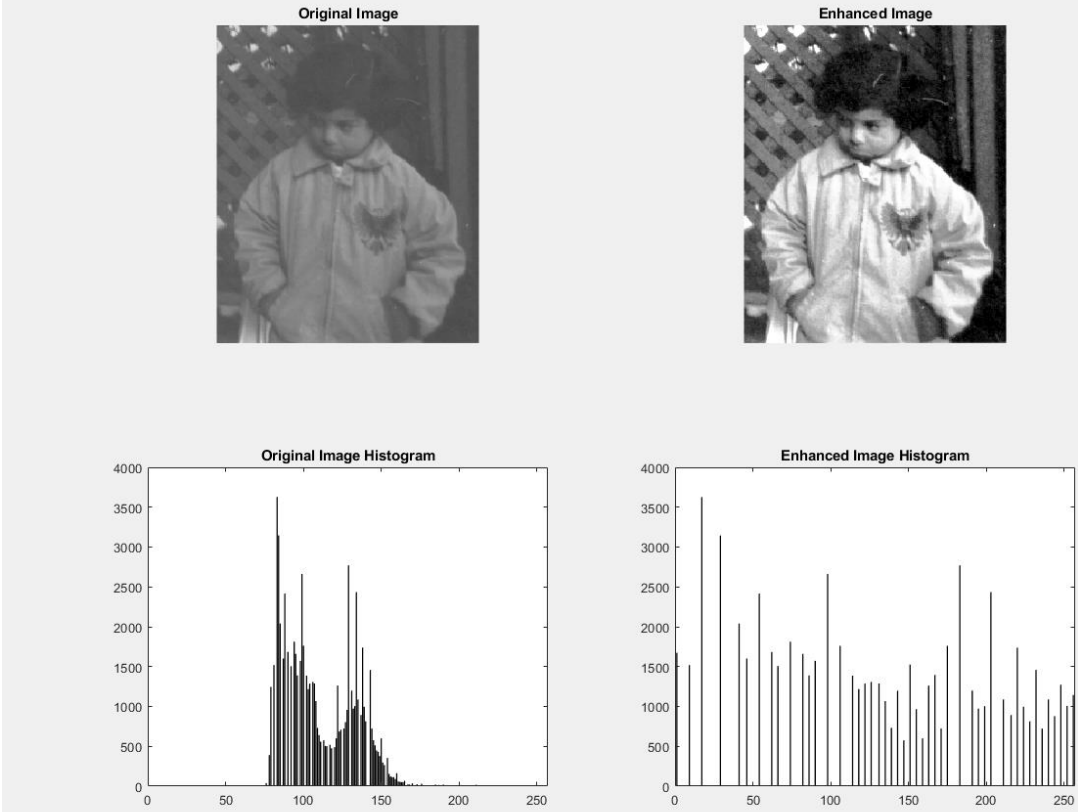
Code

exp4_dip.m

```
% Experiment 4 - Image Enhancement Histogram Equalization
% Krisha Lakhani - 60001200097

clc;
% Step 1: Load the image
image = imread('pout.tif');
% Step 2: Display the original image
subplot(2, 3, 1);
imshow(image);
title('Original Image');
% Step 3: Compute the histogram
hist_original = imhist(image);
% Step 4: Display the original image's histogram
subplot(2, 3, 4);
bar(hist_original, 'k');
title('Original Image Histogram');
% Step 5: Perform Histogram Equalization
equalized_image = histeq(image);
% Step 6: Display the enhanced image
subplot(2, 3, 2);
imshow(equalized_image);
title('Enhanced Image');
% Step 7: Compute the histogram of the enhanced image
hist_equalized = imhist(equalized_image);
% Display the histogram of the enhanced image
subplot(2, 3, 5);
bar(hist_equalized, 'k');
title('Enhanced Image Histogram');
% Step 8: Apply contrast stretching
min_intensity = min(equalized_image(:));
max_intensity = max(equalized_image(:));
stretched_image = uint8(255 * double(equalized_image - min_intensity) / (max_intensity - min_intensity));
% Step 9: Display the stretched image
subplot(2, 3, 3);
imshow(stretched_image);
title('Stretched Image');
% Step 10: Compute the histogram of the stretched image
hist_stretched = imhist(stretched_image);
% Display the histogram of the stretched image
subplot(2, 3, 6);
bar(hist_stretched, 'k');
title('Stretched Image Histogram');
% Adjust figure size and layout
set(gcf, 'Position', get(0, 'Screensize'));
sgtitle('Histogram Equalization with Contrast Stretching');
```

Output


Original Image
Enhanced Image
Original Image Histogram
Enhanced Image Histogram
0
500
1000
1500
2000
2500
3000
3500
4000
0
50
100
150
200
250

2. Without direct Histogram Function

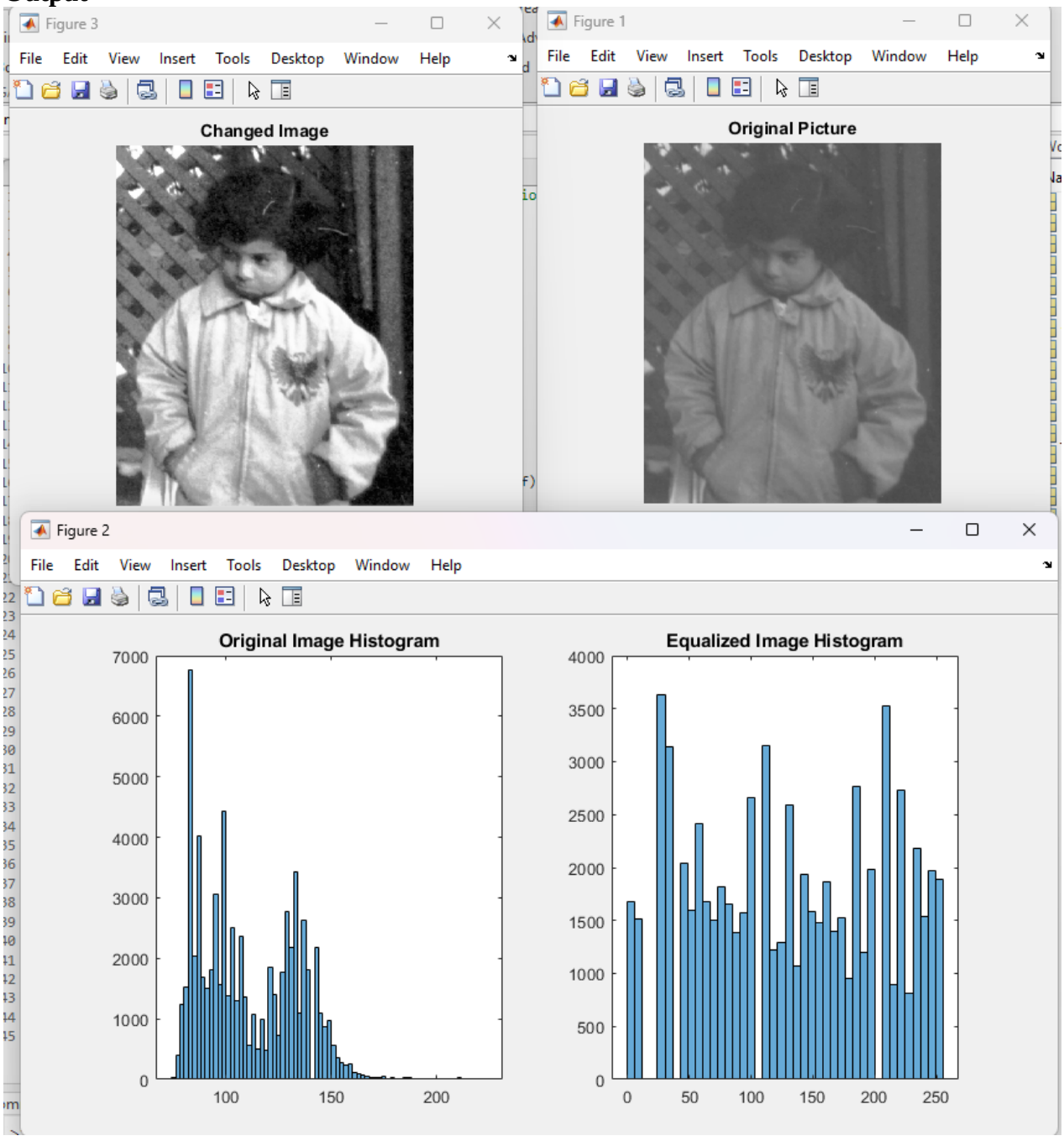
Code

exp4_dip.m | exp4_1_dip.m

```
% Experiment 4 - Image Enhancement Histogram Equalization
% Krisha Lakhani - 60001200097

clc;
a = imread("pout.tif");
figure(1);
imshow(a);
title('Original Picture');
% Calculate histogram
[count, ~] = imhist(a);
nr = sum(count);
pdf = double(count) / nr;
% Calculate CDF
cdf = cumsum(pdf);
% Normalize CDF
cdf_normalized = (cdf - min(cdf)) / (max(cdf) - min(cdf));
% Map intensity values
f = round(cdf_normalized * 255);
% Calculate new grayscale levels
ngl = zeros(1, 256);
for i = 1:255
    ngl(i) = count(i) + ngl(i + 1);
    if f(i+1) == f(i)
        ngl(i) = ngl(i) + count(i+1);
    end
end
% Plot histogram of equalized image
figure(2);
subplot(1, 2, 1);
histogram(a);
title('Original Image Histogram');
subplot(1, 2, 2);
histogram(f(a + 1), 'BinLimits', [0, 255]); % Adjust index by +1
title('Equalized Image Histogram');
% Apply histogram equalization to the original image
changed_image = zeros(size(a), 'uint8');
for i = 1:size(a, 1)
    for j = 1:size(a, 2)
        changed_image(i, j) = f(a(i, j) + 1); % Adjust index by +1
    end
end
% Display the equalized image
figure(3);
imshow(changed_image);
title('Changed Image');
```

Output



Conclusion:

Hence, we have successfully implemented histogram equalization on an image using direct histogram function as well as without direct histogram function and printed the changed image and changed histogram after equalization.