COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

Assignment # 01 - Digital Image Processing (CSC331)

Submitted by:

Laiba Binte Tahir

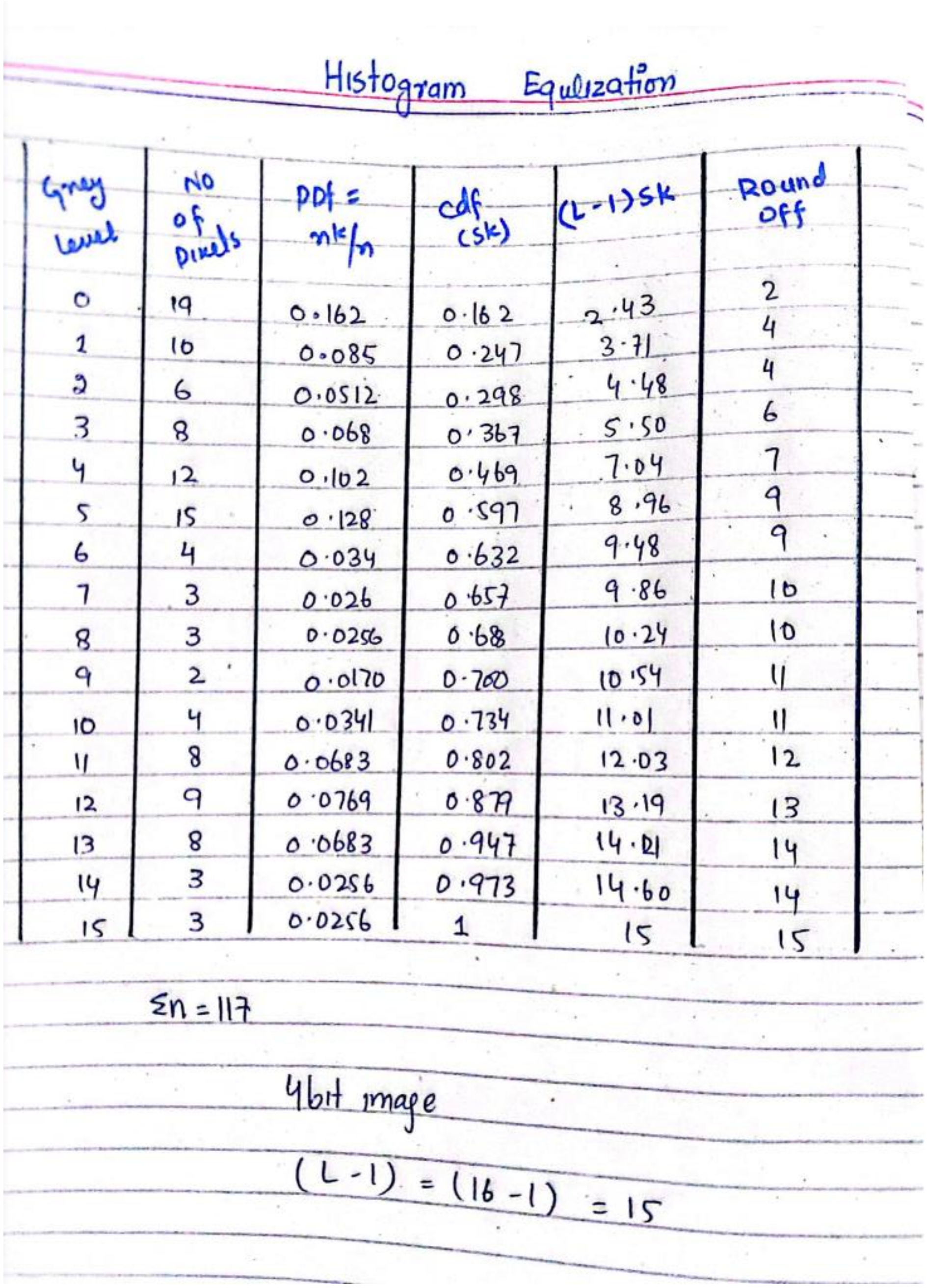
FA21-BSE-019-6A

Submitted To:

Dr. Mubashir Ahmad

Histogram Equalization:

A: Please calculate the histogram equalization of the following 13x9 grayscale 4-bit image with a graphical representation. Please follow all the steps in histogram equalization.



Histogram Equlization

Grey level	No of Pixels	PDf = nk/n	cdf (sk)	(L-1)sk	Round off
0	19	0.162	0.162	2.43	2
1	10	0.085	0.247	3.71	4
2	6	0.0512	0.298	4.48	4
3	8	0.068	0.367	5.50	6
4	12	0.102	0.469	7.04	7
5	15	0.128	0.597	8.96	9
6	4	0.034	0.632	9.48	9
7	3	0.026	0.657	9.86	10
8	3	0.0256	0.68	10.24	10
9	2	0.0170	0.700	10.54	11
10	4	0.0341	0.734	11.01	11
11	8	0.0683	0.802	12.03	12
12	9	0.0769	0.879	13.19	13
13	8	0.0683	0.947	14.21	14
14	3	0.0256	0.973	14.60	14
15	3	0.0256	1	15	15

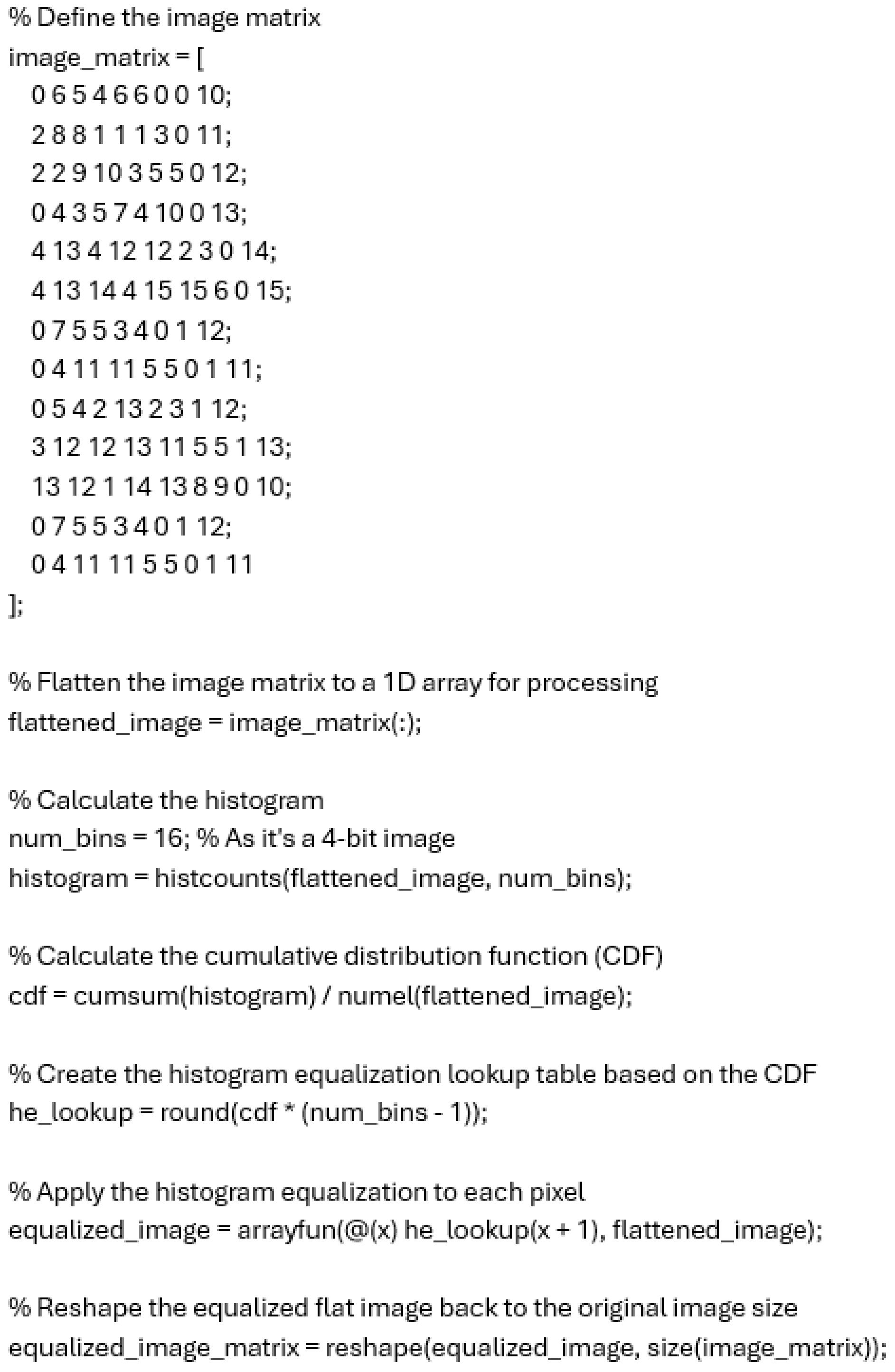
$\Sigma n = 117$

4bit image

$(L-1) = (16-1) = 15$

B: Implement the histogram equalization in MATLAB. Please write a code and generate the results of the above metrics with a graph that should show the histogram before and after HE.

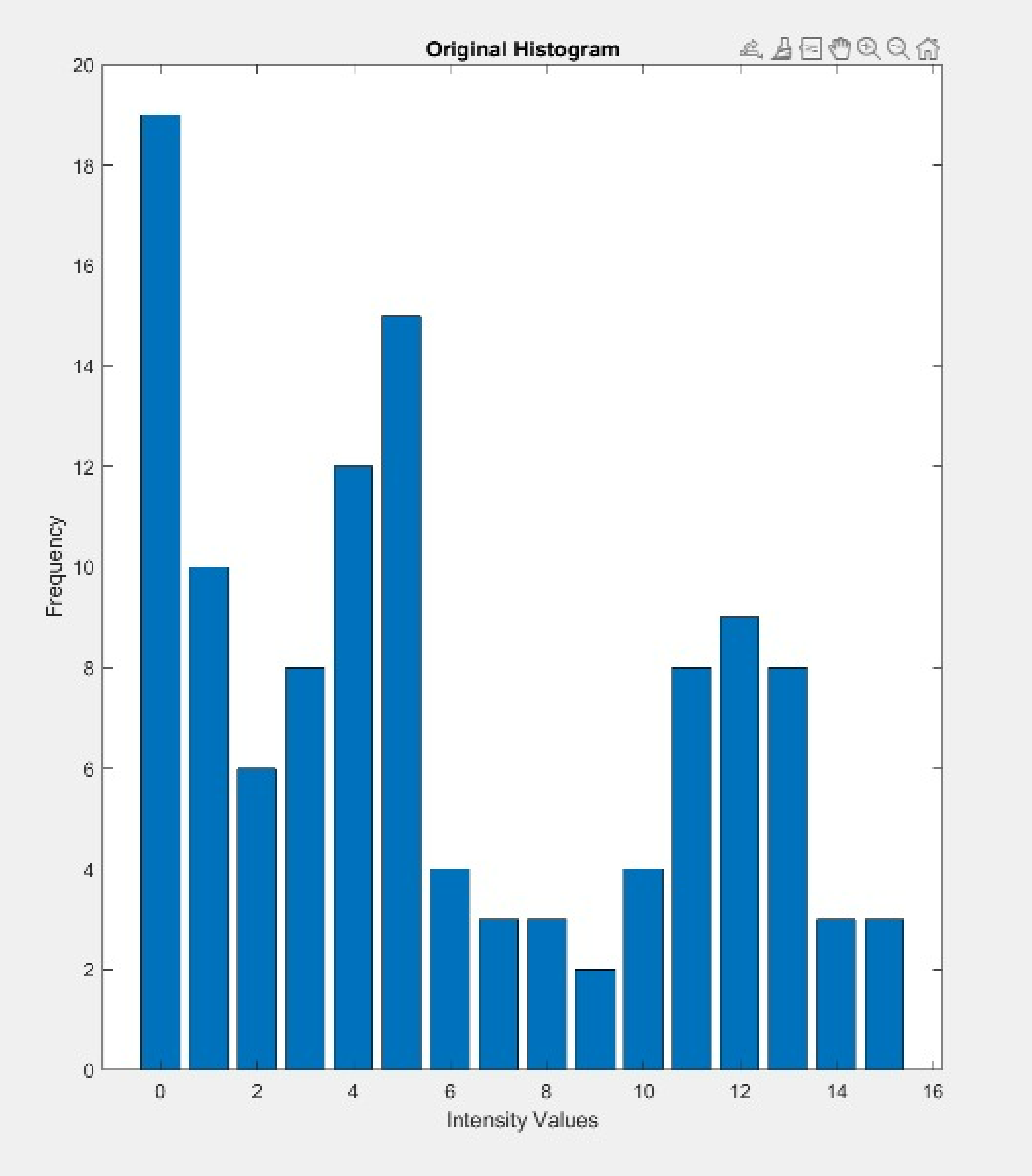
Code:



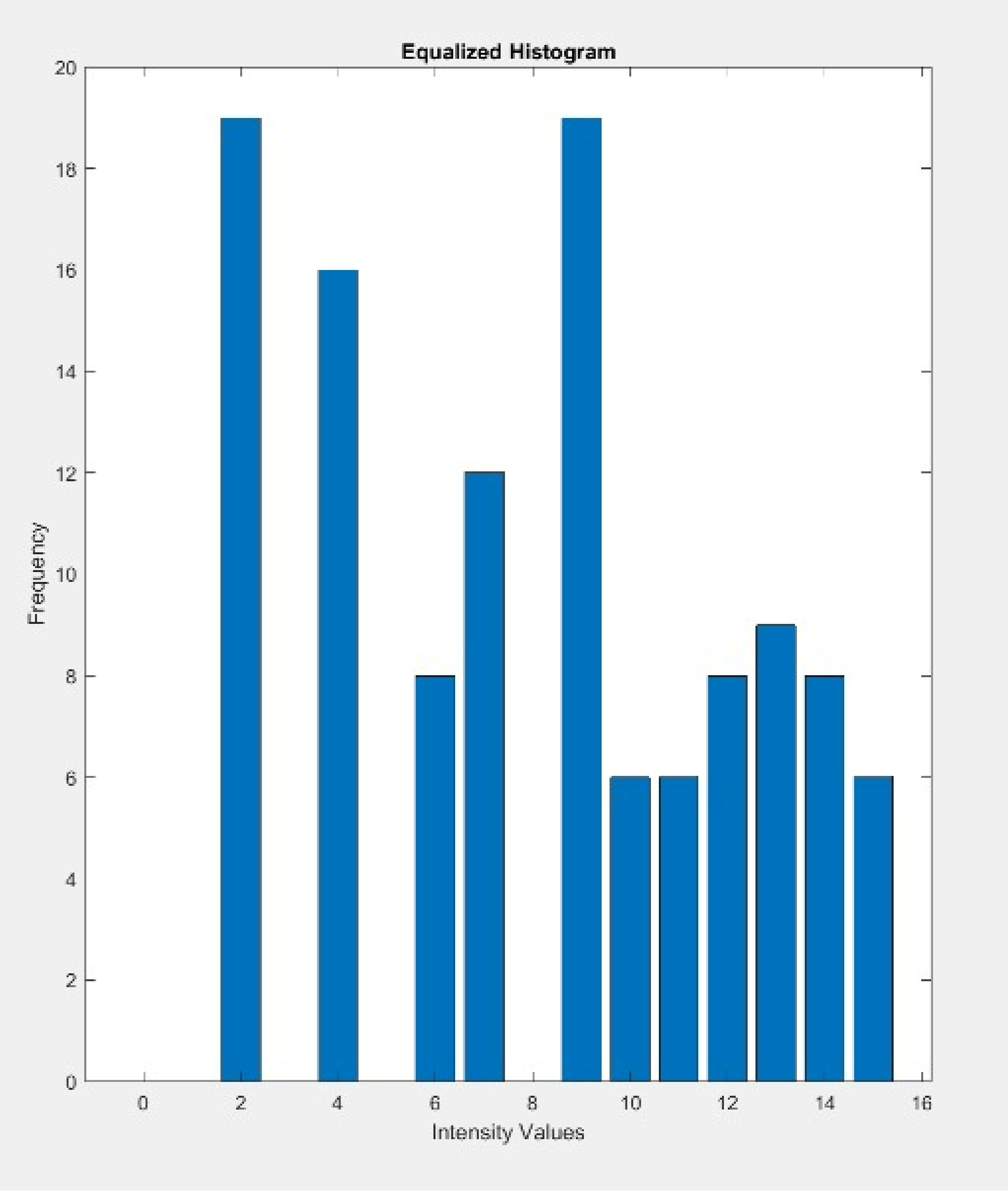
```
% Define the image matrix
image_matrix = [
   0 6 5 4 6 6 0 0 10;
   2 8 8 1 1 1 3 0 11;
   2 2 9 10 3 5 5 0 12;
   0 4 3 5 7 4 10 0 13;
   4 13 4 12 12 2 3 0 14;
   4 13 14 4 15 15 6 0 15;
   0 7 5 5 3 4 0 1 12;
   0 4 11 11 5 5 0 1 11;
   0 5 4 2 13 2 3 1 12;
   3 12 12 13 11 5 5 1 13;
   13 12 1 14 13 8 9 0 10;
   0 7 5 5 3 4 0 1 12;
   0 4 11 11 5 5 0 1 11
];

% Flatten the image matrix to a 1D array for processing
flattened_image = image_matrix(:);

% Calculate the histogram
num_bins = 16; % As it's a 4-bit image
histogram = histcounts(flattened_image, num_bins);

% Calculate the cumulative distribution function (CDF)
cdf = cumsum(histogram) / numel(flattened_image);

% Create the histogram equalization lookup table based on the CDF
he_lookup = round(cdf * (num_bins - 1));

% Apply the histogram equalization to each pixel
equalized_image = arrayfun(@(x) he_lookup(x + 1), flattened_image);

% Reshape the equalized flat image back to the original image size
equalized_image_matrix = reshape(equalized_image, size(image_matrix));
```

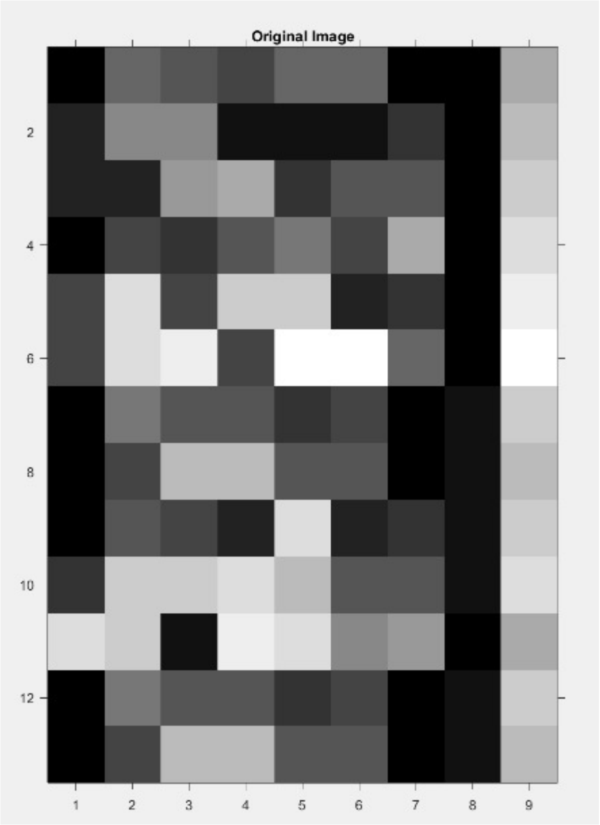
```
% Calculate the histogram of the equalized image
equalized_histogram = histcounts(equalized_image, num_bins);

% Plot the original and equalized images in a separate figure
figure('Name', 'Images Comparison', 'NumberTitle', 'off', 'Position', get(0, 'Screensize'));

subplot(1, 2, 1);
imshow(image_matrix, [0 num_bins-1]);
title('Original Image');
colormap('gray');
axis on;

subplot(1, 2, 2);
imshow(equalized_image_matrix, [0 num_bins-1]);
title('Equalized Image');
colormap('gray');
axis on;

% Plot the histograms in a separate figure with new color scheme and design
figure('Name', 'Histograms Comparison', 'NumberTitle', 'off', 'Position', get(0, 'Screensize'));

subplot(1, 2, 1);
bar(0:num_bins-1, histogram, 'FaceColor', [0,1,1]);
title('Original Histogram');
xlabel('Intensity Values');
ylabel('Frequency');
xlim([-0.5, num_bins-0.5]);

subplot(1, 2, 2);
bar(0:num_bins-1, equalized_histogram, 'FaceColor', [0,1,1]);
title('Equalized Histogram');
xlabel('Intensity Values');
ylabel('Frequency');
xlim([-0.5, num_bins-0.5]);
```

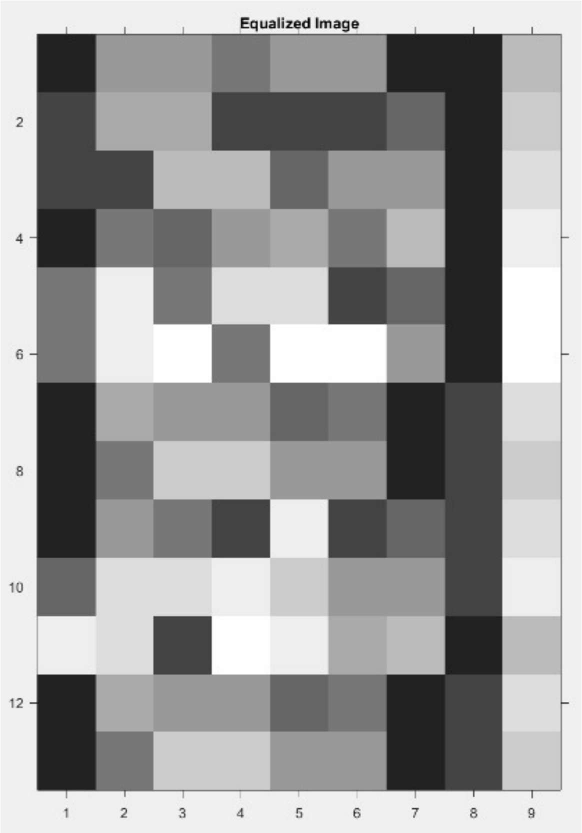
Original histogram (Before Equalization)



Equalized histogram (After Equalization)




Original Image
2
4
6
8
10
12
1
2
3
4
5
6
7
8
9


Equalized Image
2
4
6
8
10
12
1
2
3
4
5
6
7
8
9