**Batch: A-3 Roll No.: 16010122104**

**Experiment No. 7**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title****:** Write a program to apply the global processing technique: Histogram equalization.on a digital image |

**Objective:** To learn and understand the concept of histogram stretching and equalization in image enhancement operations.

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
| **CO4** | Design & implement algorithms for digital image enhancement, segmentation & restoration. |

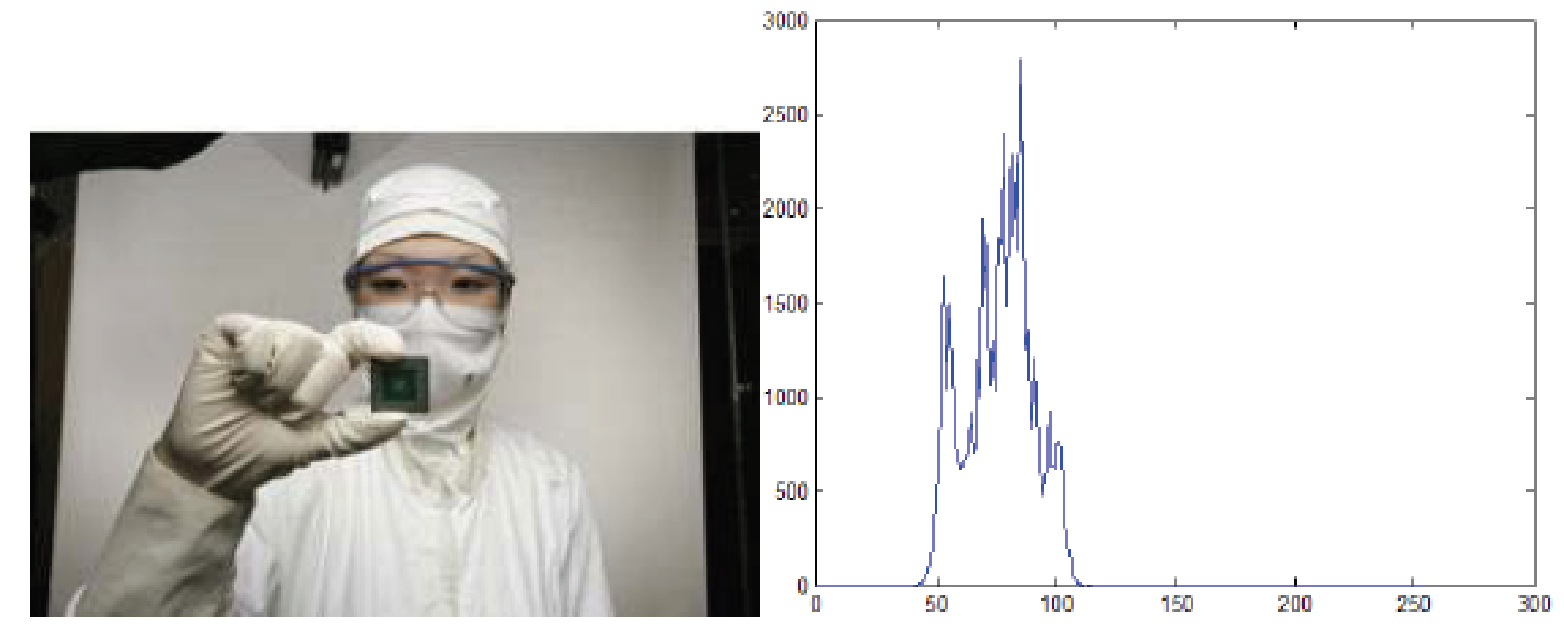
**Books/ Journals/ Websites referred:**

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

**Pre Lab/ Prior Concepts:**

**Image histogram:**

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit greyscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those greyscale values. Histograms can also be taken of color images either individual histogram of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count. The exact output from the operation depends upon the implementation it may simply be a picture of the required histogram in a suitable image format, or it may be a data file of some sort representing the histogram statistics.

****

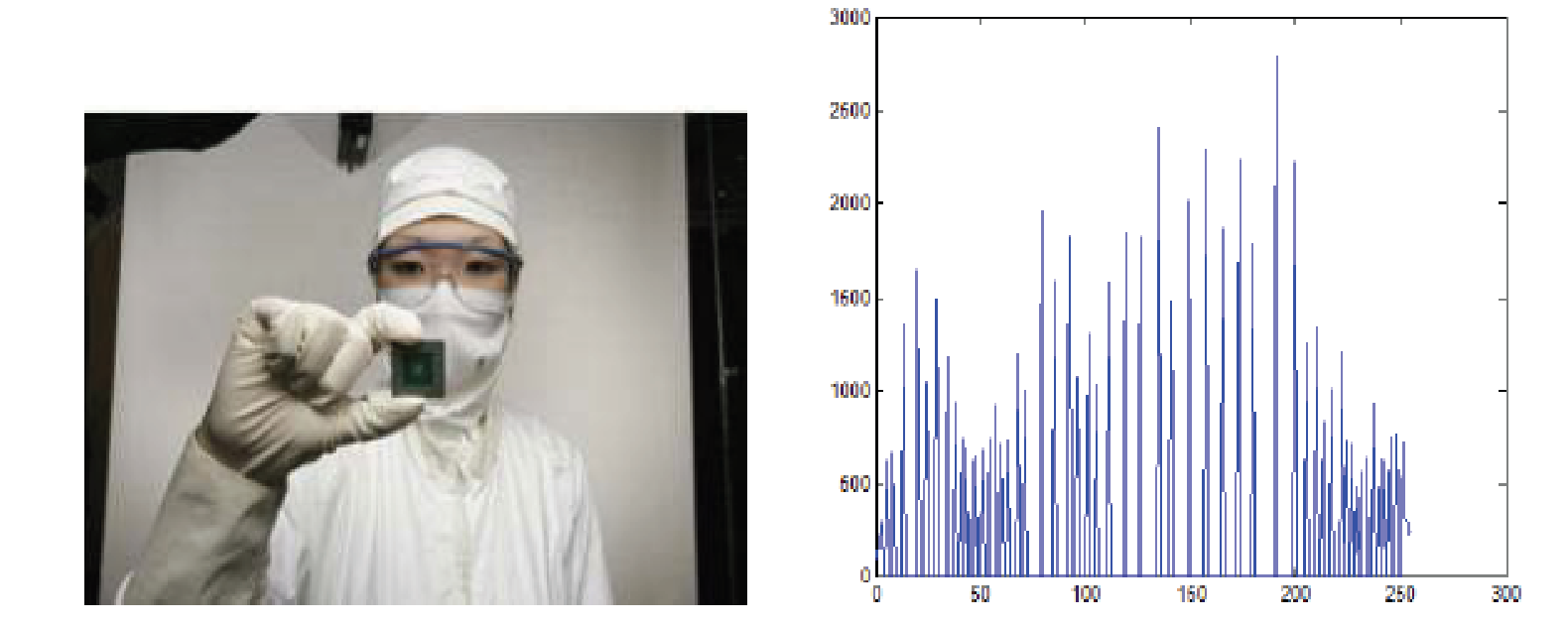
**Fig. 1 An image and its histogram**

**Histogram Equalization:**

A perfect image is one which has equal number of pixels in all its grey levels. hence our objective is not only to spread the dynamic range , but also to have equal pixels in all the grey levels. This technique is known as histogram equalization.

Basically the histogram equalization spreads out intensity values along the total range of values in order to achieve higher contrast. This method is especially useful when an image is

represented by close contrast values, such as images in which both the background and foreground are bright at the same time, or else both are dark at the same time. For example, the result of applying histogram equalization to the image in figure 1 is presented in figure 2.



**Fig. 2 New image and its equalized histogram**

**Description of cumulative histogram equalization:**

Here are the steps for implementing this algorithm.

1. Create the histogram for the image.

2. Calculate the cumulative distribution function histogram.

3. Calculate the new values through the general histogram equalization formula.

4. Assign new values for each gray value in the image.

Thus processed image is obtained by mapping each pixel with level rk into a corresponding pixel with level sk in o/p image. This transformation is called Histogram equalization

**Resources Used:** Matlab

**Implementation Details:**

**Write Algorithm and Matlab commands used**

% Read the image

img = imread('parrot.bmp');

if size(img, 3) == 3

% Convert to grayscale if it's RGB

img = rgb2gray(img);

end

% Get the dimensions of the image

[M, N] = size(img);

% Part 1: Forced Histogram

% Given pixel counts for each intensity level

forced\_counts = [50; 100; 100; 50; 200; 100; 300; 100];

% Define original gray levels

original\_gray\_levels = (0:7)';

% Calculate total pixels and PDF

total\_forced = sum(forced\_counts);

forced\_pdf = forced\_counts / total\_forced;

% Calculate CDF

forced\_cdf = zeros(8,1);

forced\_cdf(1) = forced\_pdf(1);

for i = 2:8

forced\_cdf(i) = forced\_cdf(i-1) + forced\_pdf(i);

end

% Scale CDF and define new gray levels

forced\_cdf\_scaled = round(forced\_cdf \* 7);

forced\_new\_gray = forced\_cdf\_scaled;

% Create table for forced histogram processing

Table1\_forced = table( ...

original\_gray\_levels, ...

forced\_counts, ...

forced\_pdf, ...

forced\_cdf, ...

forced\_cdf\_scaled, ...

forced\_new\_gray, ...

'VariableNames', ...

{'OriginalGrayLevel', 'PixelsPerIntensity', 'pdf', 'cdf', 'cdfTimes7', 'NewGrayLevel'} ...

);

% Display table

fprintf('Table 1 Histogram processing numerical\n');

disp(Table1\_forced);

% Calculate new pixel intensity distribution

forced\_new\_pixel\_intensity = zeros(8,1);

for val = 1:8

forced\_new\_pixel\_intensity(val) = sum(forced\_counts(forced\_new\_gray == (val - 1)));

end

% Create table for new pixel intensity distribution

Table2\_forced = table( ...

original\_gray\_levels, ...

forced\_new\_pixel\_intensity, ...

'VariableNames', ...

{'GrayLevel', 'PixelsPerIntensity'} ...

);

% Display table

fprintf('Table 2 New Pixel Intensity Dist\n');

disp(Table2\_forced);

% Part 2: Histogram Equalization

% Initialize histogram counts

hist\_counts = zeros(256, 1);

% Count pixel intensities in the original image

for i = 1:M

for j = 1:N

intensity = img(i, j) + 1;

hist\_counts(intensity) = hist\_counts(intensity) + 1;

end

end

% Calculate PDF and CDF

pdf = hist\_counts / (M \* N);

cdf\_val = cumsum(pdf);

% Scale CDF to get new gray levels

scaled\_cdf = round(cdf\_val \* 255);

new\_gray\_levels = scaled\_cdf;

% Apply histogram equalization

output\_img = img;

for i = 1:M

for j = 1:N

old\_intensity = img(i, j) + 1;

output\_img(i, j) = new\_gray\_levels(old\_intensity);

end

end

% Count pixel intensities in the equalized image

hist\_eq\_counts = zeros(256,1);

for i = 1:M

for j = 1:N

eq\_intensity = output\_img(i, j) + 1;

hist\_eq\_counts(eq\_intensity) = hist\_eq\_counts(eq\_intensity) + 1;

end

end

% Display images and histograms

figure;

subplot(2,2,1);

imshow(img);

title('Original Image');

subplot(2,2,2);

imshow(output\_img);

title('Histogram Equalized Image');

subplot(2,2,3);

bar(0:255, hist\_counts, 1, 'FaceColor','b','EdgeColor','none');

xlabel('Gray level');

ylabel('Pixel Count / gray level');

title('Histogram of Original Image');

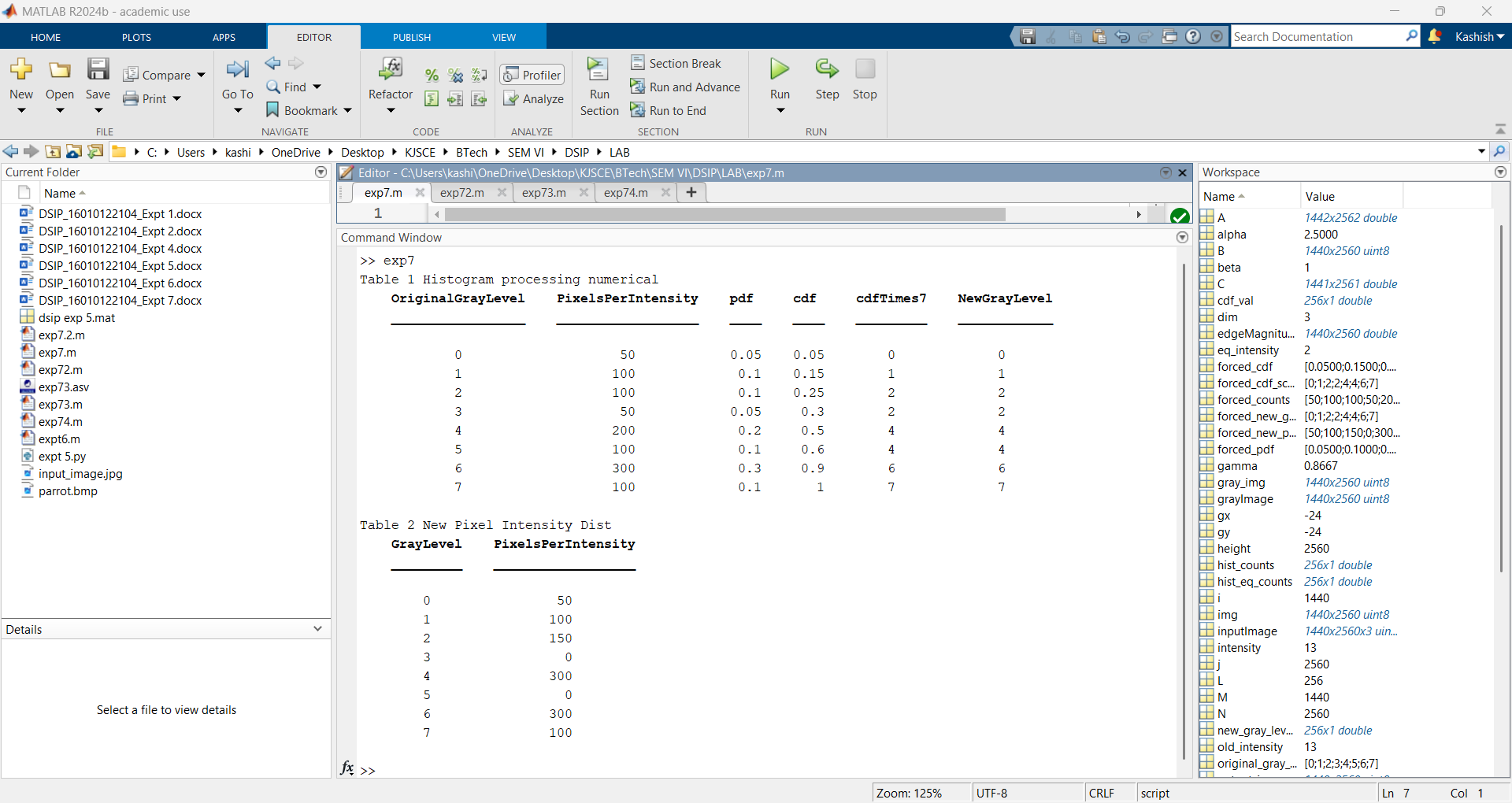
subplot(2,2,4);

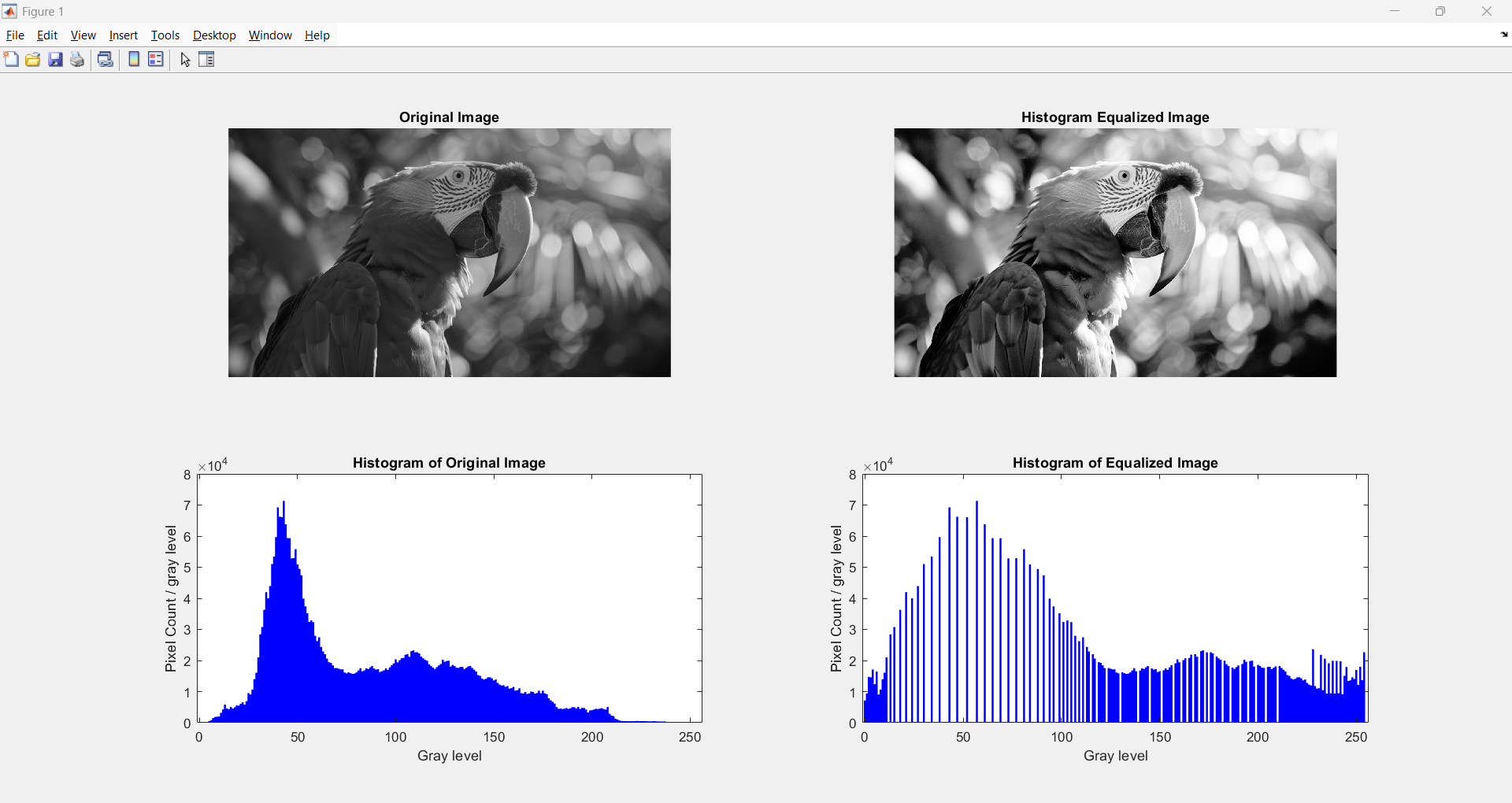
bar(0:255, hist\_eq\_counts, 1, 'FaceColor','b','EdgeColor','none');

xlabel('Gray level');

ylabel('Pixel Count / gray level');

title('Histogram of Equalized Image');

****

****

**Conclusion:-**

**We implemented a program to apply the global processing technique: Histogram equalization on a digital image.**

**Date: 11/03/2025 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

Compare between contrast stretching and histogram equalization.

Ans:

Both contrast stretching and histogram equalization are techniques used in image processing to enhance the visual quality of images. Here's a comparison of these two methods:

Contrast Stretching

Definition: Contrast stretching is a simple technique used to enhance the contrast of an image by stretching the range of pixel values to the full range of possible values (e.g., 0 to 255 for 8-bit images).

Method: It involves mapping the minimum and maximum pixel values of the original image to the minimum and maximum possible pixel values, effectively making dark areas darker and bright areas brighter.

Advantages:

Easy to implement.

Improves image contrast without altering the original histogram shape.

Disadvantages:

Does not handle cases where the histogram is heavily skewed or bimodal.

May not effectively enhance images with a narrow range of pixel values.

Histogram Equalization

Definition: Histogram equalization is a more sophisticated technique that aims to adjust the contrast of an image by modifying the pixel values to create a uniform distribution of intensities across the histogram.

Method: It involves calculating the cumulative distribution function (CDF) of the image histogram and using it to map each pixel value to a new value that corresponds to a uniform distribution.

Advantages:

Can handle images with skewed or bimodal histograms.

Provides a more balanced distribution of intensities, often resulting in visually pleasing images.

Disadvantages:

More complex to implement compared to contrast stretching.

May introduce artifacts if the original image has noise or if the histogram is highly irregular.

Key Differences

Effect on Histogram: Contrast stretching maintains the original shape of the histogram, while histogram equalization alters it to achieve a uniform distribution.

Complexity: Histogram equalization is computationally more intensive due to the need to calculate and apply the CDF.

Applicability: Contrast stretching is suitable for simple contrast enhancement, whereas histogram equalization is more effective for images with complex histograms or when a more uniform distribution is desired.

In summary, contrast stretching is a straightforward method for basic contrast enhancement, while histogram equalization offers a more powerful approach to achieve a balanced distribution of intensities, making it suitable for a wider range of image enhancement tasks.