Image Processing and Computer Graphics

INTE 41312

Histogram Equalization and Histogram Normalization

Assignment 1

IM/2019/112-Sandushi Weraduwa

01.

a. What is a histogram of an image?

A histogram of an image is a graphical representation of the distribution of pixel intensities or grayscale values in that image. It shows the number of pixels for each intensity level from 0 to 255 for an 8-bit grayscale image. The x-axis represents the intensity values, while the y-axis represents the frequency of pixels at each intensity level.

b. Briefly explain what histogram equalization and histogram normalization are.

Histogram Equalization:

Histogram equalization is a technique used to improve the contrast of an image by redistributing the pixel intensity values so that they span the entire range of possible values more evenly. This process enhances the overall visibility of details in an image, making features more distinguishable.

Histogram Normalization:

Histogram normalization is a process that adjusts the range of pixel intensity values in an image to match a specified range. This technique ensures that the pixel values span the entire intensity range, for example, from 0 to 255 in an 8-bit image, improving the image's dynamic range and contrast. Unlike histogram equalization, which focuses on equalizing the distribution, normalization scales the values to fit within a desired range.

c. List down two (02) applications of histogram equalization and histogram normalization.

Applications of Histogram Equalization:

- **Medical Imaging:**

Enhancing the contrast of X-ray images, CT scans, or MRI images to improve the visibility of structures and anomalies, aiding in more accurate diagnosis.

- **Satellite Imagery:**

Improving the contrast of satellite images to better distinguish different land features, water bodies, and vegetation is useful in geographical mapping and environmental monitoring.

Applications of Histogram Normalization:

- **Image Preprocessing in Machine Learning:**

Normalizing images before feeding them into machine learning models to ensure consistent input data, which helps in improving the model's performance and training efficiency.

- **Digital Photography:**

Enhancing the overall appearance of digital photographs by adjusting the brightness and contrast to achieve a more visually appealing image with a better dynamic range.

02. Implement a MATLAB code for histogram equalization using in-build functions following the given instructions.

a. Provide the MATLAB code.

Editor - C:\Users\Sandushi\Documents\MATLAB\HistogramEqualization.m

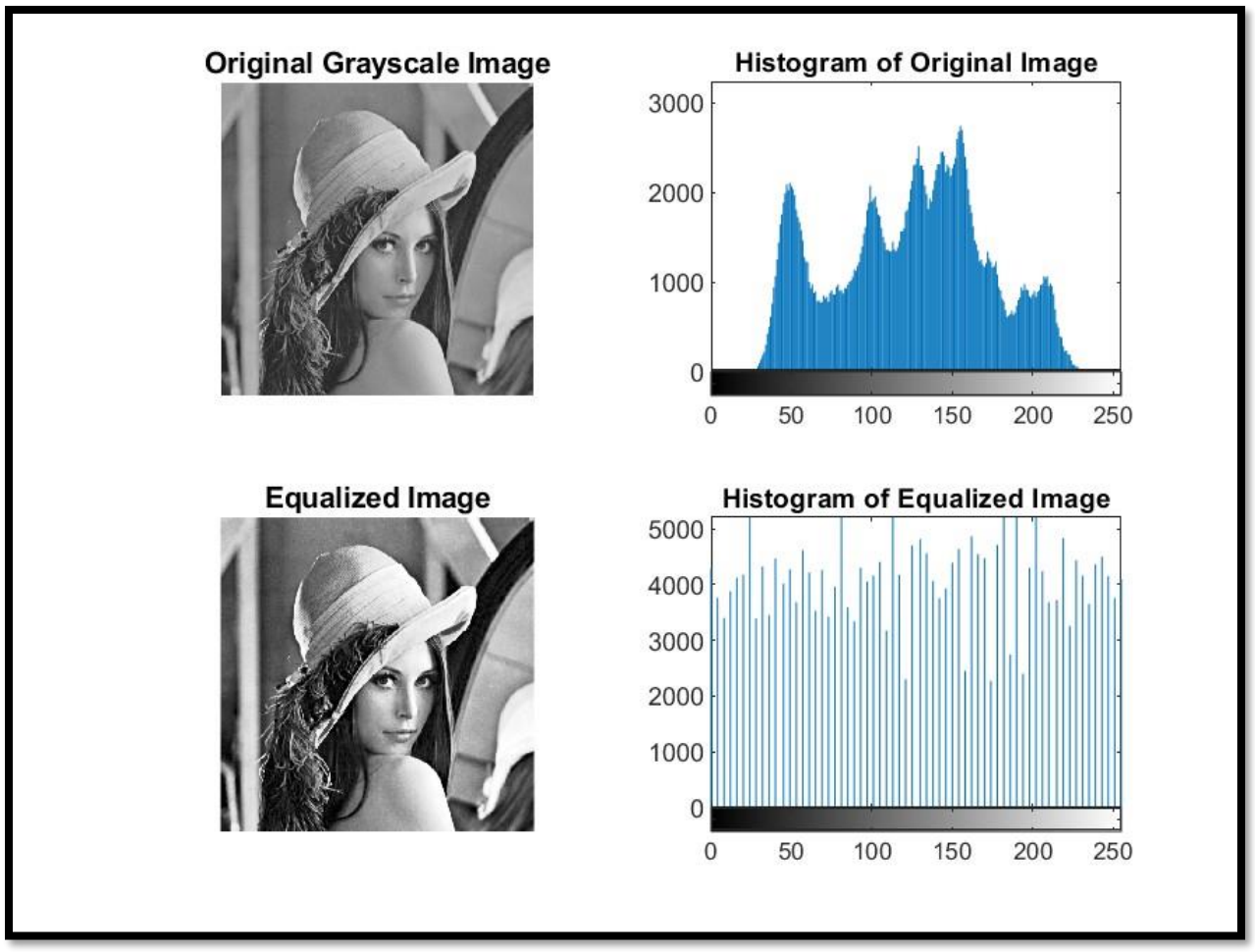
HistogramEqualization.m

```
% Clear the workspace
clear;
clc;
close all;

% Load the image
img = imread("C:\Users\Sandushi\Documents\4th year\4th Year 1st Sem\Image Processing\Lenna_(test_image).png");

% Convert the image to grayscale
gray_img = rgb2gray(img);

% Apply histogram equalization
equalized_img = histeq(gray_img);

% Create subplots
figure;

% Original image
subplot(2, 2, 1);
imshow(gray_img);
title('Original Grayscale Image');

% Histogram of the original image
subplot(2, 2, 2);
imhist(gray_img);
title('Histogram of Original Image');

% Equalized image
subplot(2, 2, 3);
imshow(equalized_img);
title('Equalized Image');

% Histogram of the equalized image
subplot(2, 2, 4);
imhist(equalized_img);
title('Histogram of Equalized Image');

% Save the figure as a JPEG image
saveas(gcf, 'C:\Users\Sandushi\Documents\4th year\4th Year 1st Sem\Image Processing\Figure1.jpeg');
```

b. Insert the saved JPEG image (Title: Figure 1).

This is the JPEG image showing the original grayscale image, its histogram, the equalized image, and its histogram, saved as "Figure 1":

Figure 1



03. Implement a customized MATLAB code for histogram equalization.

a. Provide the MATLAB code.

Editor - C:\Users\Sandushi\Documents\MATLAB\HistogramEqulaization_CustomCode.m

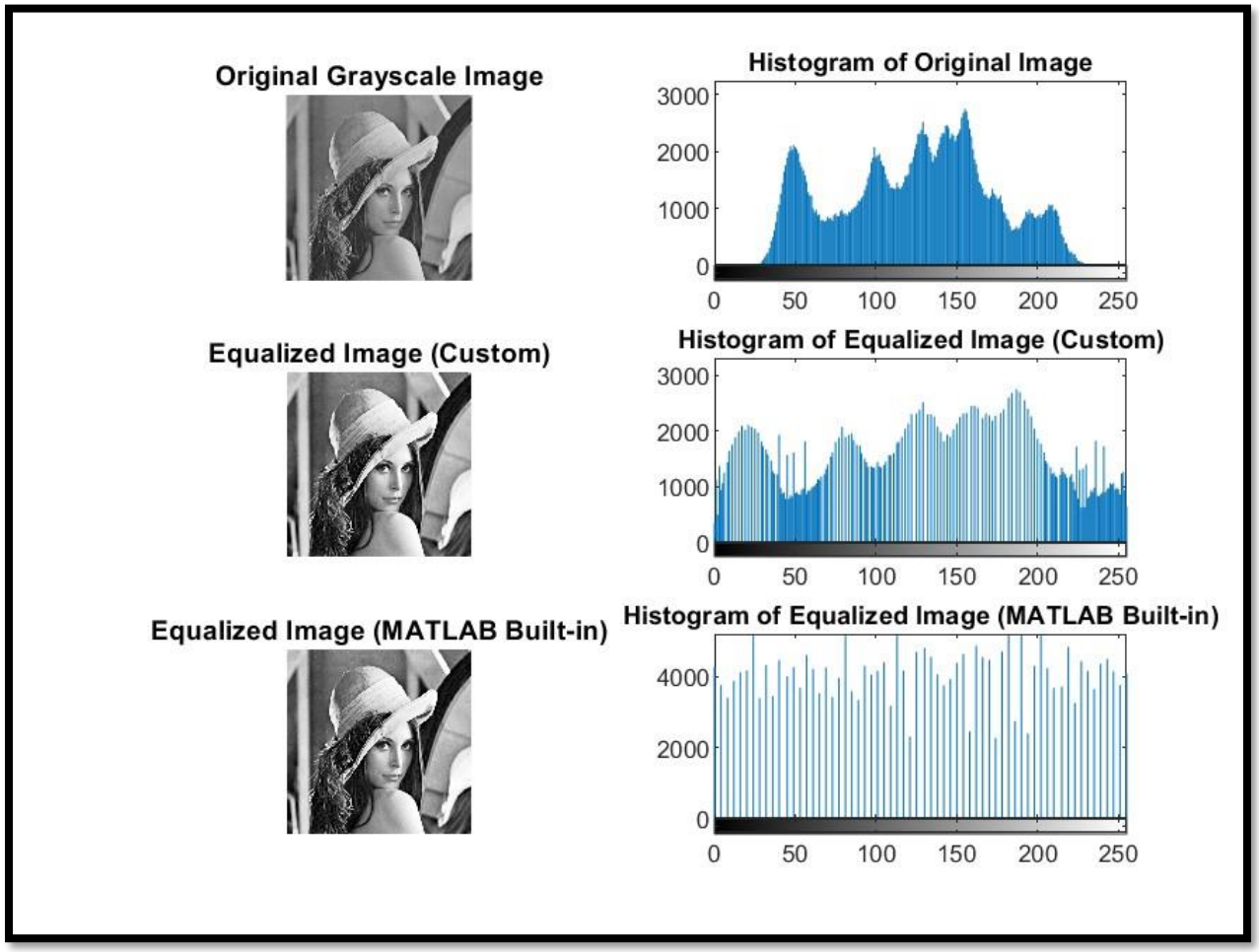
HistogramEqualization.m | HistogramEqulaization_CustomCode.m

```m
% Clear the workspace
clear;
clc;
close all;

% Load the image
img = imread("C:\Users\Sandushi\Documents\4th year\4th Year 1st Sem\Image Processing\Lenna_(test_image).png");

% Convert the image to grayscale
gray_img = rgb2gray(img);

% Define the number of rows and columns
[num_rows, num_cols] = size(gray_img);

% Calculate the total number of pixels
total_pixels = num_rows * num_cols;

% Initialize the final image and histogram vectors
equalized_img_custom = zeros(num_rows, num_cols);
pdf = zeros(256, 1);
cdf = zeros(256, 1);
histogram_original = zeros(256, 1);

% Calculate the histogram of the original image
for i = 1:num_rows
    for j = 1:num_cols
        intensity = gray_img(i, j);
        histogram_original(intensity + 1) = histogram_original(intensity + 1) + 1;
    end
end

% Calculate the PDF
pdf = histogram_original / total_pixels;

% Calculate the CDF
cdf(1) = pdf(1);
for k = 2:256
    cdf(k) = cdf(k - 1) + pdf(k);
end

% Normalize the CDF
cdf_normalized = cdf * 255;

% Apply the histogram equalization
for i = 1:num_rows
    for j = 1:num_cols
        equalized_img_custom(i, j) = cdf_normalized(gray_img(i, j) + 1);
    end
end

% Convert the equalized image to uint8
equalized_img_custom = uint8(equalized_img_custom);

% Apply histogram equalization using MATLAB built-in function
equalized_img_builtin = histeq(gray_img);

% Create subplots
figure;

% Original image
subplot(3, 2, 1);
imshow(gray_img);
title('Original Grayscale Image');

% Histogram of the original image
subplot(3, 2, 2);
imhist(gray_img);
title('Histogram of Original Image');

% Equalized image using customized code
subplot(3, 2, 3);
imshow(equalized_img_custom);
title('Equalized Image (Custom)');

% Histogram of the equalized image using customized code
subplot(3, 2, 4);
imhist(equalized_img_custom);
title('Histogram of Equalized Image (Custom)');

% Equalized image using MATLAB built-in function
subplot(3, 2, 5);
imshow(equalized_img_builtin);
title('Equalized Image (MATLAB Built-in)');

% Histogram of the equalized image using MATLAB built-in function
subplot(3, 2, 6);
imhist(equalized_img_builtin);
title('Histogram of Equalized Image (MATLAB Built-in)');

% Save the figure as a JPEG image
saveas(gcf, "C:\Users\Sandushi\Documents\4th year\4th Year 1st Sem\Image Processing\Figure2.jpeg");
```

b. Insert the saved JPEG image (Title: Figure 2).

Here is the JPEG image showing the original grayscale image, its histogram, the equalized image using the customized code, its histogram, the equalized image using MATLAB's built-in function, and its histogram, saved as "Figure 2":

Figure 2



c. Compare and contrast the output images and their histograms obtained using MATLAB in-built function and customized function.

Equalized Image (Custom) vs. Equalized Image (MATLAB Built-in):

- Both images show improved contrast compared to the original image, but the exact pixel intensity distribution might differ slightly due to the manual implementation details.
- The customized equalized image may have a less smooth appearance than the MATLAB built-in equalized image due to differences in the way the CDF (Cumulative Distribution Function) is calculated and applied.

Histograms:

- The histogram of the equalized image using the customized code shows a more spread-out distribution of pixel intensities compared to the original image, indicating improved contrast.
- The histogram of the equalized image using MATLAB's built-in function also shows a spread-out distribution but might be smoother and more evenly distributed due to the optimized algorithm used in the built-in function.

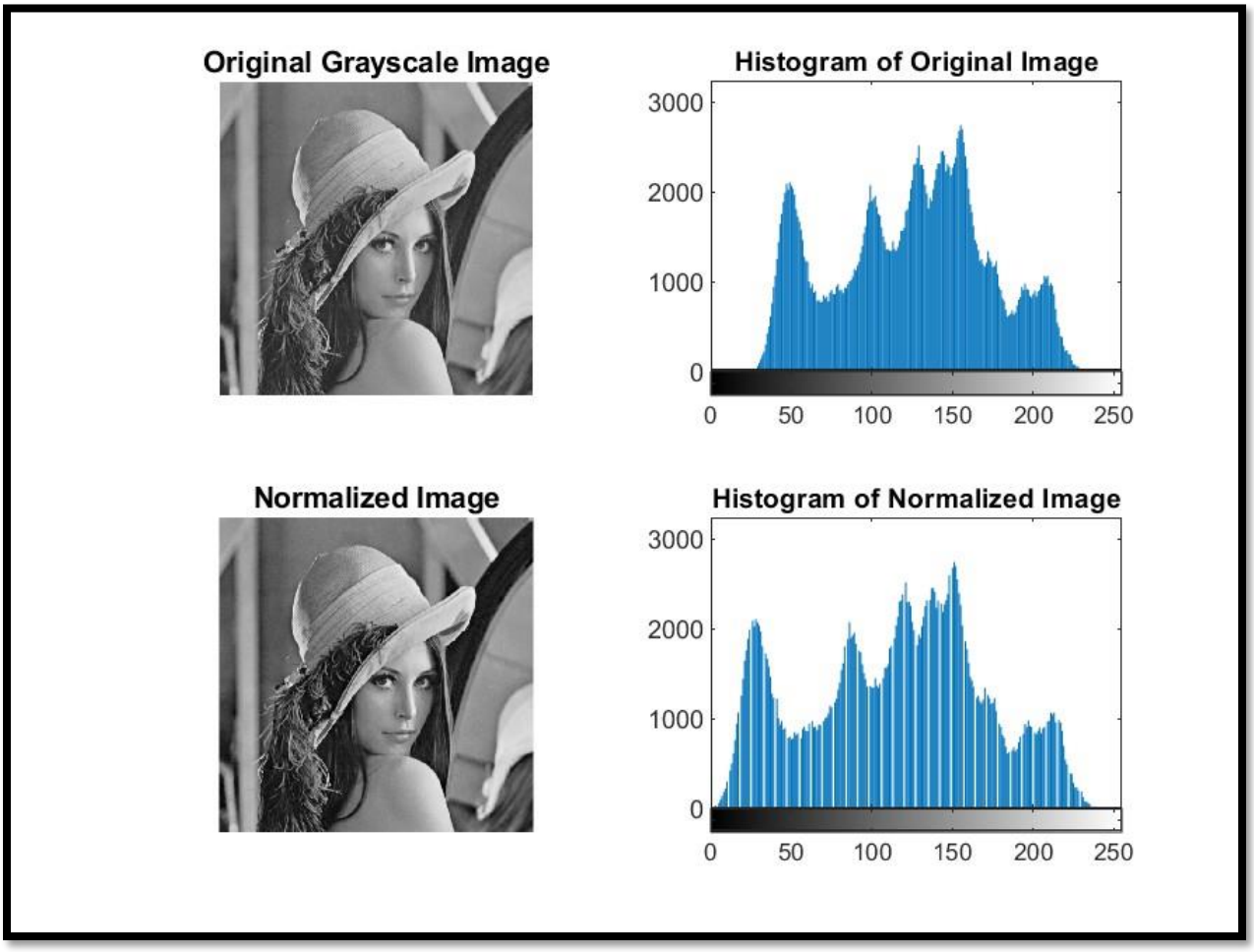
04. Implement a MATLAB code to normalize the histogram same image used in Q. 2 and Q. 3.

a. Provide the MATLAB code.

b. Insert the saved JPEG image (Title: Figure 3).

This is the JPEG image showing the original grayscale image, its histogram, the normalized image, and its histogram, saved as "Figure 3":

Figure 3



c. Compare and contrast the differences of the output images and histograms of histogram equalization and histogram normalization.

Histogram Equalization:

- Purpose: Enhance the contrast of the image by redistributing pixel intensities to achieve a uniform histogram.
- Effect on Image: Reveals more details by making dark areas lighter and light areas darker. The resulting image has more balanced contrast across all intensity levels.
- Histogram: The histogram of the equalized image is more uniform, with pixel values spread across the entire range (0 to 255).

Histogram Normalization:

- Purpose: Scale the pixel intensities to fit within a specified range (e.g., 0 to 255).

- Effect on Image: Improves the overall dynamic range of the image, making it appear more balanced in terms of brightness and contrast.
- Histogram: The histogram of the normalized image is stretched to cover the entire range of intensity values, but it doesn't necessarily have a uniform distribution like histogram equalization.

Differences:

- Contrast Enhancement: Histogram equalization specifically targets contrast enhancement by equalizing the distribution of pixel intensities, making it more effective for images with low contrast. Histogram normalization, on the other hand, primarily adjusts the range of pixel values, which improves dynamic range but may not enhance contrast as effectively.
- Uniformity of Histogram: Histogram equalization results in a more uniform histogram, whereas histogram normalization results in a stretched histogram that covers the full range of intensities but retains the original shape of the histogram.

Both techniques improve the visual quality of the image, but histogram equalization is generally better for enhancing contrast, while histogram normalization is useful for adjusting the dynamic range.