
```
% Clear workspace, command window, and close all figures
clc;
clearvars;
close all;

% Load the color image and convert it to grayscale
inputColorImage = imread("p1.jpg"); % Load the color image
grayScaleImg = rgb2gray(inputColorImage); % Convert the image to grayscale

% Display the grayscale and color images side-by-side
figure;
montage({grayScaleImg, inputColorImage}, 'Size', [1 2]);
title('Grayscale Image (Left) and Original Color Image (Right)');

% Initialize variables
numGrayLevels = 256; % Number of gray levels (8-bit image)
[rows, cols] = size(grayScaleImg); % Dimensions of the grayscale image
intensityHistogram = zeros(1, numGrayLevels); % Preallocate histogram array

% Compute the histogram manually
for rowIdx = 1:rows
    for colIdx = 1:cols
        intensityValue = grayScaleImg(rowIdx, colIdx); % Get pixel intensity
        intensityHistogram(intensityValue + 1) =
intensityHistogram(intensityValue + 1) + 1;
    end
end

% Normalize the histogram
totalPixelCount = sum(intensityHistogram); % Total number of pixels
in the image
normalizedHist = intensityHistogram / totalPixelCount; % Probability of each
intensity

% Compute the cumulative distribution function (CDF)
cumulativeDistFunc = cumsum(normalizedHist); % Cumulative sum of the
normalized histogram
minCdfValue = min(cumulativeDistFunc(cumulativeDistFunc > 0)); % Minimum non-
zero value in the CDF

% Normalize the CDF to the range [0, numGrayLevels-1]
scaledCdf = round((cumulativeDistFunc - minCdfValue) / (1 - minCdfValue) *
(numGrayLevels - 1));

% Apply the equalized intensity values to create the new image
equalizedImg = uint8(scaledCdf(grayScaleImg + 1));

% Compute the histogram of the equalized image
equalizedHist = imhist(equalizedImg);

% Plot the original grayscale histogram
figure;
```

```

bar(0:numGrayLevels-1, intensityHistogram, 'BarWidth', 1, 'FaceColor', [0.5
0.5 0.5]);
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency');
title('Histogram of Original Grayscale Image');
grid on;

% Plot the equalized histogram
figure;
bar(0:numGrayLevels-1, equalizedHist, 'BarWidth', 1, 'FaceColor', 'g');
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency');
title('Histogram of Equalized Image');
grid on;

% Display the original and equalized images side-by-side
figure;
montage({grayScaleImg, equalizedImg}, 'Size', [1 2]);
title('Original Grayscale Image (Left) vs Equalized Image (Right)');

% Overlay the CDF on the histogram of the original grayscale image
figure;
yyaxis left;
bar(0:numGrayLevels-1, intensityHistogram, 'BarWidth', 1, 'FaceColor', [0.5
0.5 0.5]);
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency');
title('Histogram and CDF of Original Grayscale Image');
grid on;

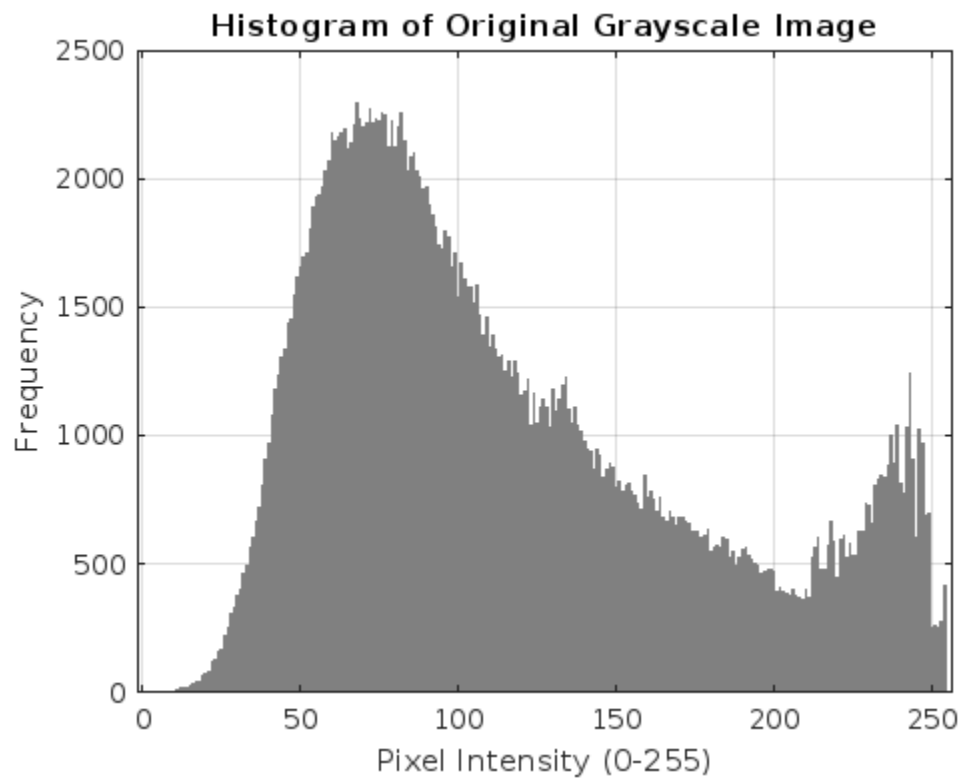
yyaxis right;
plot(0:numGrayLevels-1, cumulativeDistFunc, 'r', 'LineWidth', 2);
ylabel('Cumulative Distribution Function (CDF)');
legend('Histogram', 'CDF');

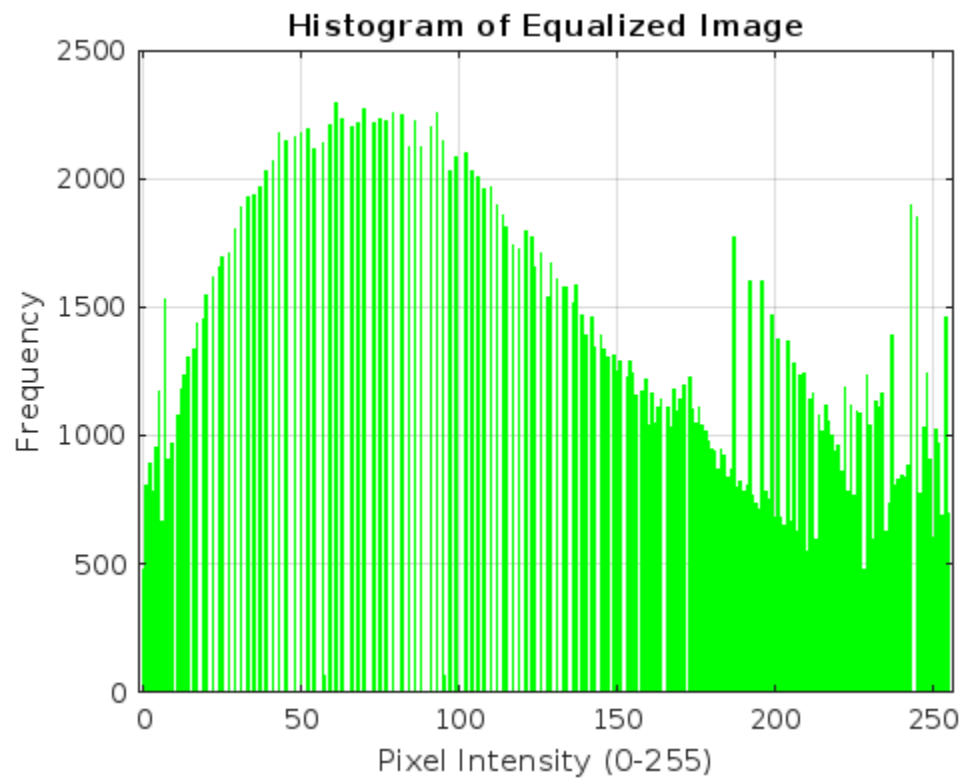
% Compute the CDF of the equalized image
equalizedCdf = cumsum(equalizedHist / totalPixelCount);

% Plot the histogram and CDF of the equalized grayscale image in the same
graph
figure;
bar(0:numGrayLevels-1, equalizedHist, 'BarWidth', 1, 'FaceColor', 'g'); %
Plot histogram
hold on;
plot(0:numGrayLevels-1, equalizedCdf * max(equalizedHist), 'r', 'LineWidth',
2); % Plot scaled CDF
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency / Scaled CDF');
title('Histogram and CDF of Equalized Image');
legend('Equalized Histogram', 'Equalized CDF (Scaled)');
grid on;

```

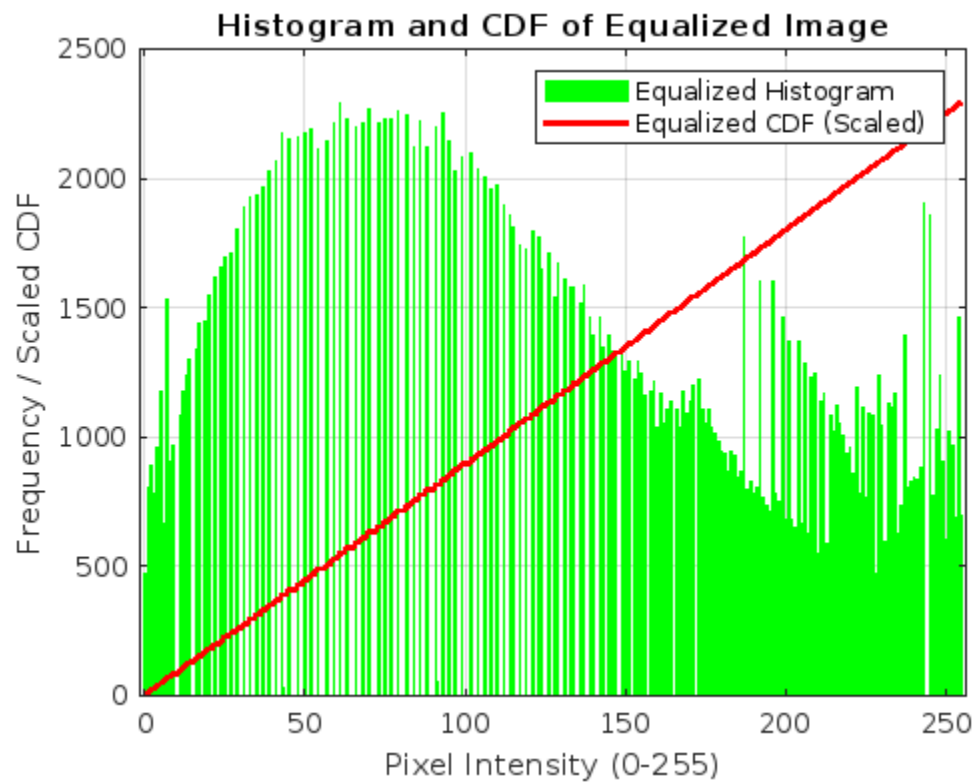
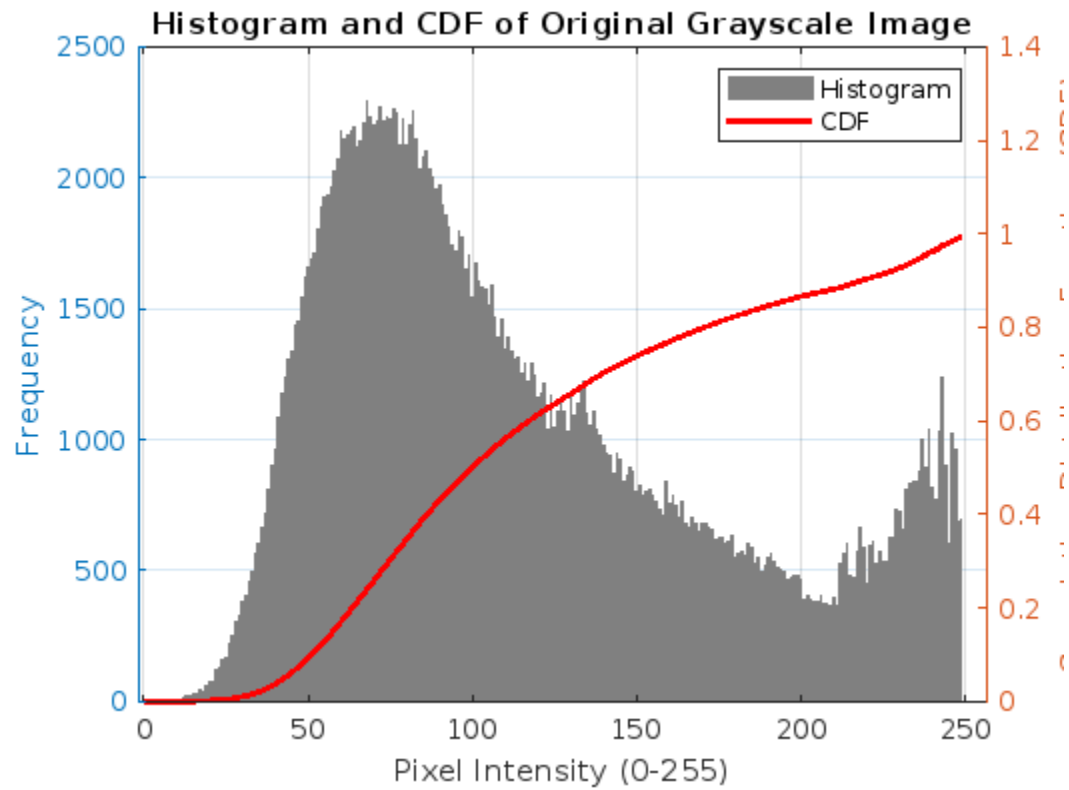
Grayscale Image (Left) and Original Color Image (Right)





Original Grayscale Image (Left) vs Equalized Image (Right)





Published with MATLAB® R2024b