
```
% Created on 19/01/25
% Created by Samar Patel, BT22ECE075
% Image Histogram Equalization

% Prompt to select an image file
[fileName, filePath] = uigetfile({'*.jpg;*.jpeg;.png;.bmp', 'Image Files
(*.jpg, *.jpeg, *.png, *.bmp)'}, 'Select an Image File');

% Exit if no file is selected
if isequal(fileName, 0)
    disp('No file selected. Exiting...');
    return;
end

% Construct the full path to the selected image file
imageFilePath = fullfile(filePath, fileName);

% Read the input image
imageOriginal = imread(imageFilePath);

% Convert to grayscale if the image is in RGB format
if size(imageOriginal, 3) == 3
    imageGrayscale = rgb2gray(imageOriginal);
else
    imageGrayscale = imageOriginal;
end

% Get image dimensions
[imageHeight, imageWidth] = size(imageGrayscale);

% Initialize histogram for the grayscale image
histOriginal = zeros(256, 1);

% Compute the histogram for the grayscale image
for row = 1:imageHeight
    for col = 1:imageWidth
        pixelValue = imageGrayscale(row, col);
        histOriginal(pixelValue + 1) = histOriginal(pixelValue + 1) + 1;
    end
end

% Compute the PDF (Probability Density Function) for the grayscale image
pdfOriginal = histOriginal / (imageHeight * imageWidth);

% Compute the CDF (Cumulative Distribution Function) for the grayscale image
cdfOriginal = cumsum(pdfOriginal);

% Map pixel intensities based on the equalized CDF
intensityMapping = round(cdfOriginal * 255);

% Create the equalized image using the intensity mapping
equalizedImage = zeros(size(imageGrayscale));
```

```

for row = 1:imageHeight
    for col = 1:imageWidth
        equalizedImage(row, col) = intensityMapping(imageGrayscale(row, col)
+ 1);
    end
end

% Convert the equalized image to uint8 for proper display
equalizedImage = uint8(equalizedImage);

% Initialize histogram for the equalized image
histEqualized = zeros(256, 1);

% Compute the histogram for the equalized image
for row = 1:imageHeight
    for col = 1:imageWidth
        pixelValue = equalizedImage(row, col);
        histEqualized(pixelValue + 1) = histEqualized(pixelValue + 1) + 1;
    end
end

% Compute the PDF for the equalized image
pdfEqualized = histEqualized / (imageHeight * imageWidth);

% Compute the CDF for the equalized image
cdfEqualized = cumsum(pdfEqualized);

% Display the grayscale image
figure;
subplot(1, 2, 1);
imshow(imageGrayscale);
title('Grayscale Image');

% Display the equalized image
subplot(1, 2, 2);
imshow(equalizedImage);
title('Equalized Image');

% Display the histogram of the grayscale image
figure;
subplot(1, 2, 1);
imhist(imageGrayscale);
title('Histogram of Grayscale Image');

% Display the histogram and CDF of the equalized image
subplot(1, 2, 2);
imhist(equalizedImage);
hold on;
plot(cdfEqualized * max(histEqualized), 'r', 'LineWidth', 2); % Scale CDF for
visualization
legend('Histogram', 'CDF');
title('Histogram and CDF of Equalized Image');

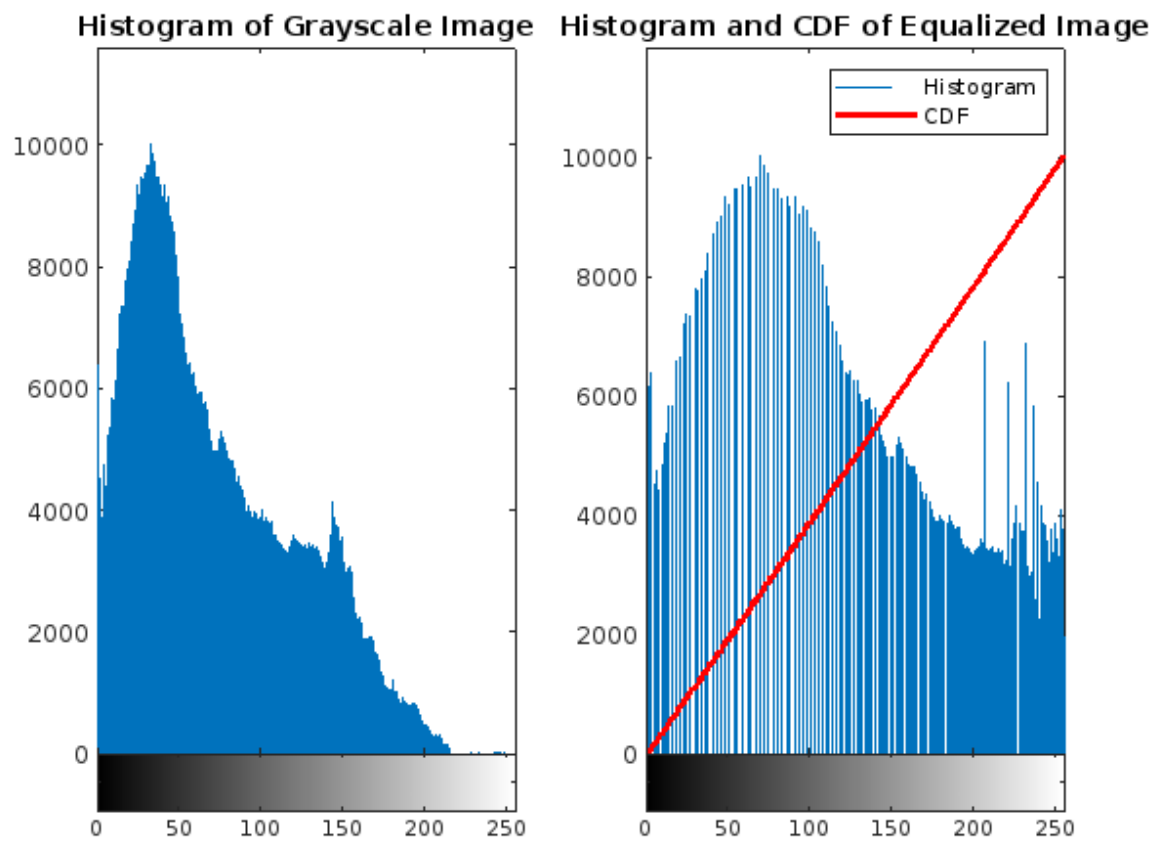
```

Grayscale Image



Equalized Image





Published with MATLAB® R2024b