```matlab
%Histogram Equalisation
%Created by Srinivas Vatsav

clc;
clear all;
close all;

% Generate a random 8x8 intensity matrix with values between 50 and 100
I = round(100 + (100 - 50) * rand(8)); % Values in the range [50, 150]

%Flatten the matrix into a single row vector
Ia = I(:)';

%Sort intensities
sortIa = sort(Ia);

%Find unique intensities and their frequencies
[uniqueNumbers, ~, idx] = unique(sortIa); % Unique intensities and indices
frequencies = accumarray(idx, 1); % Frequency of each intensity

%Calculate the cumulative distribution function (CDF)
cumulativeFrequencies = cumsum(frequencies);

% Total number of pixels in the image
totalPixels = numel(I);

% Minimum non-zero cumulative frequency
cdf_min = min(cumulativeFrequencies(cumulativeFrequencies > 0));

% Number of intensity levels (L = 256 for 8-bit grayscale)
L = 256;

%Apply the histogram equalization formula
h_v = round((((cumulativeFrequencies - cdf_min) / (totalPixels - cdf_min)) *
(L - 1)));

%Map the original intensities to equalized values
equalizedImage = zeros(size(I)); % Initialize equalized image
for i = 1:length(uniqueNumbers)
    equalizedImage(I == uniqueNumbers(i)) = h_v(i);
end

equalizedImage = uint8(equalizedImage);
I = uint8(I);

% Plot the original and equalized histograms for comparison
figure(1);
subplot(2, 1, 1);
bar(uniqueNumbers, frequencies, 'FaceColor', 'b');
title('Original Histogram');
xlabel('Intensity Values');
ylabel('Frequency');
```
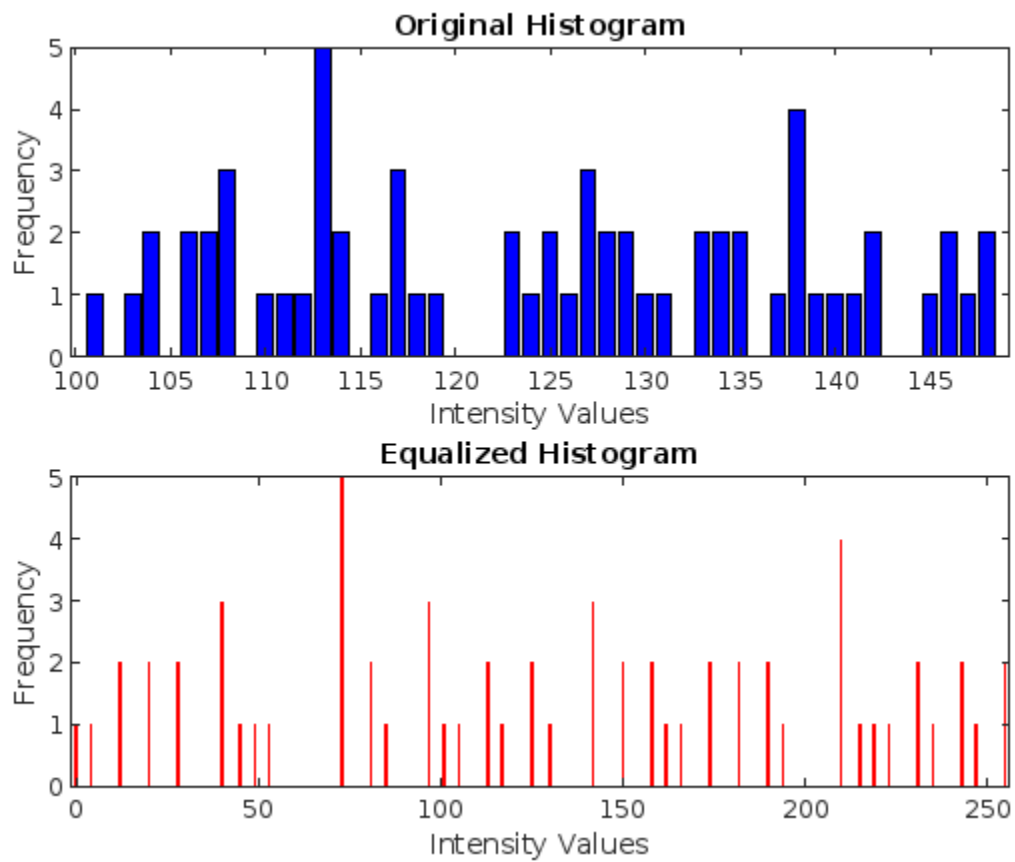
```
subplot(2, 1, 2);
bar(0:L-1, histcounts(equalizedImage(:), 0:L), 'FaceColor', 'r');
title('Equalized Histogram');
xlabel('Intensity Values');
ylabel('Frequency');

figure(2)
subplot(1,2,1)
imshow(I)
title("Original Image");

subplot(1,2,2)
imshow(equalizedImage)
title("Equalised Image");
```
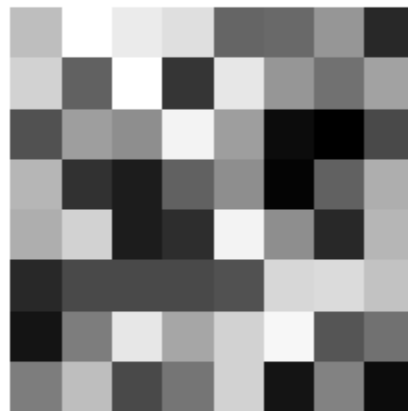
Original Image

Equalised Image

Published with MATLAB® R2024b