
```
clc;
clear all;
close all;

% Read the input image
InputImg = imread("MED.png");

% Convert to grayscale only if the image is RGB
if size(InputImg,3) == 3
    GrayImg = rgb2gray(InputImg);
else
    GrayImg = InputImg;
end
```

----- Manual Histogram Equalization Implementation -----

```
% Determine image dimensions
[Rows, Cols] = size(GrayImg);

% Total number of pixels in the image
TotalPixels = Rows * Cols;

% Initialize frequency array for intensity values (0 to 255)
PixelCount = zeros(256,1);

% Compute histogram manually
for r = 1:Rows
    for c = 1:Cols
        PixelValue = GrayImg(r,c);
        PixelCount(PixelValue + 1) = PixelCount(PixelValue + 1) + 1;
    end
end

% Compute Probability Density Function (PDF)
ProbDist = PixelCount / TotalPixels;

% Compute Cumulative Distribution Function (CDF)
CumDist = zeros(256,1);
CumDist(1) = ProbDist(1);

for idx = 2:256
    CumDist(idx) = CumDist(idx - 1) + ProbDist(idx);
end

% Define number of intensity levels
NumLevels = 256;

% Create intensity mapping using CDF
IntensityMap = round((NumLevels - 1) * CumDist);
```

```
% Create output image matrix with proper data type
EqualizedImg = zeros(Rows, Cols, 'uint8');

% Apply the mapping to each pixel
for r = 1:Rows
    for c = 1:Cols
        OldIntensity = GrayImg(r,c);
        EqualizedImg(r,c) = IntensityMap(OldIntensity + 1);
    end
end
```

----- Display Results -----

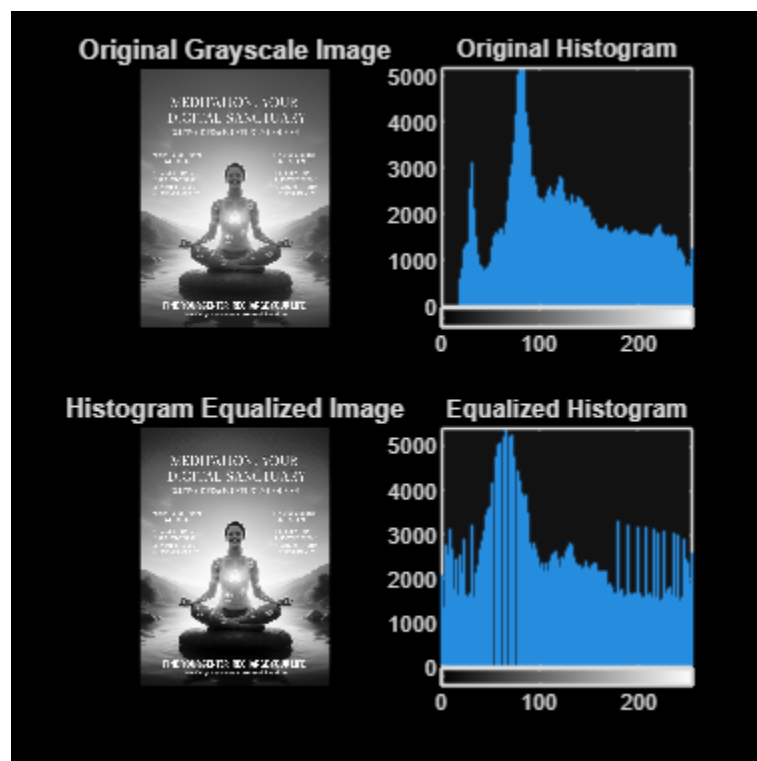
```
figure;

subplot(2,2,1);
imshow(GrayImg);
title('Original Grayscale Image');

subplot(2,2,2);
imhist(GrayImg);
title('Original Histogram');

subplot(2,2,3);
imshow(EqualizedImg);
title('Histogram Equalized Image');

subplot(2,2,4);
imhist(EqualizedImg);
title('Equalized Histogram');
```



Published with MATLAB® R2025b