Contents


```
% Date       : 19/01/2026
% Created by : Abhishek Kumar Jayswal
% Description: Manual implementation of Histogram Equalization
%              for grayscale images without using built-in functions

clc;            % Clear command window
clear;          % Clear workspace variables
close all;      % Close all figure windows
```

Reading the input image

```
I = imread('C:\Users\Abhishek\Desktop\DIP\Histogram_Equalisation\Input_Image.jpg');

% Check if the image is RGB
% If yes, convert it to grayscale
if size(I,3) == 3
    I = rgb2gray(I);
end

% Ensure image is in uint8 format (0-255 intensity range)
I = uint8(I);

% Get image dimensions
[M, N] = size(I);
```

STEP 1: Compute Histogram of the Image

Initialize histogram array for 256 gray levels

```
hist_count = zeros(256,1);

% Count occurrence of each intensity value
for i = 1:M
    for j = 1:N
        intensity = I(i,j);                  % Pixel intensity
        hist_count(intensity + 1) = ...
```

```
            hist_count(intensity + 1) + 1;    % Increment count
    end
end
```

STEP 2: Compute Probability Density Function (PDF)

PDF = histogram / total number of pixels

```
pdf = hist_count / (M * N);
```

STEP 3: Compute Cumulative Distribution Function (CDF)

```
cdf = zeros(256,1);
cdf(1) = pdf(1);    % Initialize first value

for k = 2:256
    cdf(k) = cdf(k-1) + pdf(k);   % Cumulative sum of PDF
end
```

STEP 4: Normalize the CDF

Find the minimum non-zero CDF value to avoid intensity shift

```
cdf_min = min(cdf(cdf > 0));

% Normalize CDF to map intensity values into range [0, 255]
cdf_norm = round((cdf - cdf_min) / (1 - cdf_min) * 255);
```

STEP 5: Map Old Intensities to New Intensities

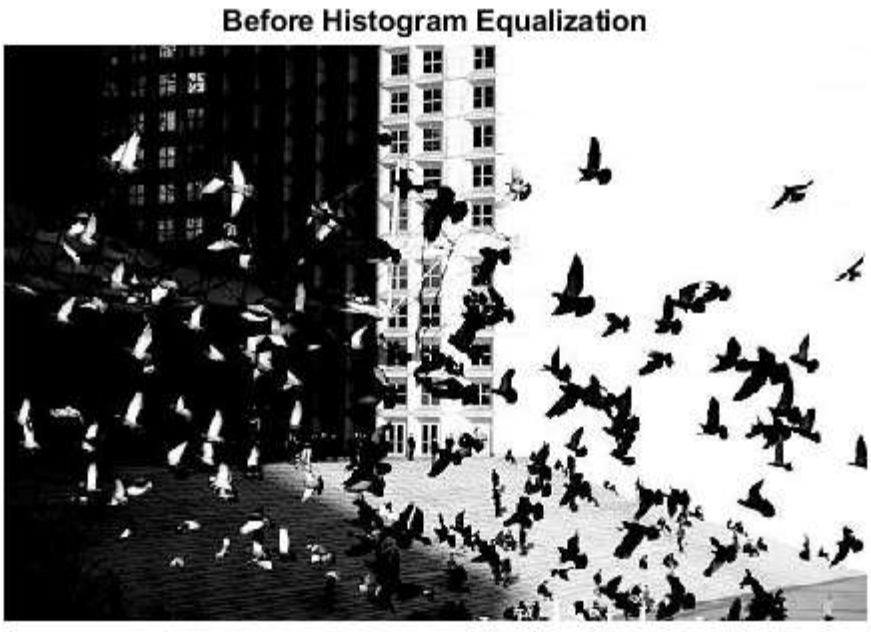
Initialize equalized image

```
Ieq = zeros(M, N, 'uint8');

% Replace each pixel using normalized CDF mapping
for i = 1:M
    for j = 1:N
        Ieq(i,j) = cdf_norm(I(i,j) + 1);
    end
end
```

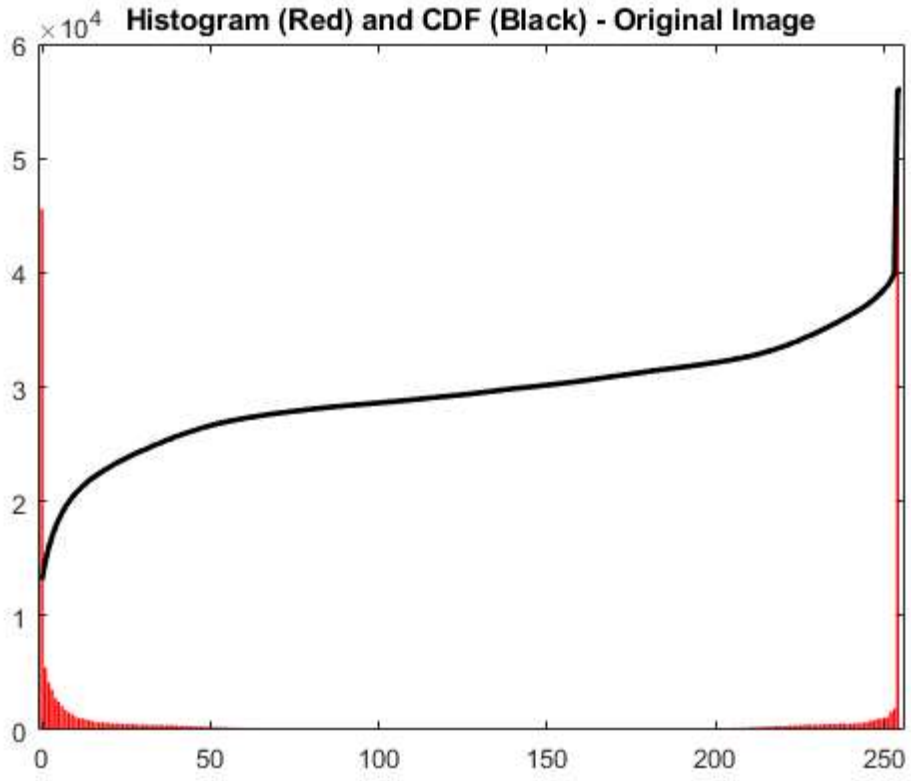
Display Original Image

```
figure;
imshow(I);
title('Before Histogram Equalization');
```



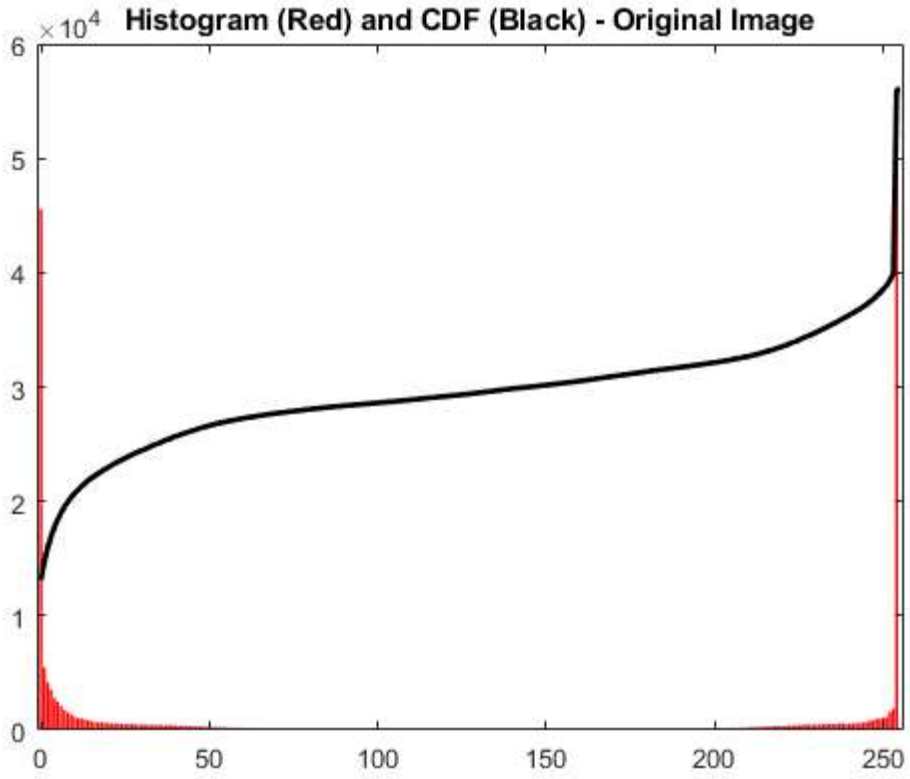
Display Histogram and CDF of Original Image

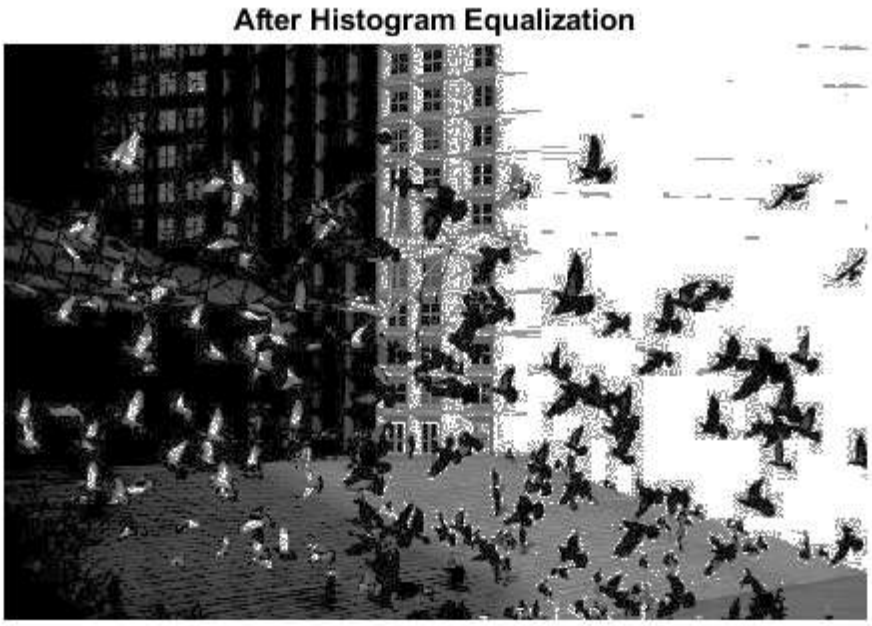
```
figure;
bar(0:255, hist_count, 'r');          % Histogram (red bars)
hold on;
plot(0:255, cdf * max(hist_count), ...
     'k', 'LineWidth', 2);            % Scaled CDF (black curve)
hold off;
title('Histogram (Red) and CDF (Black) - Original Image');
```



Display Equalized Image

```
figure;
imshow(Ieq);
title('After Histogram Equalization');
```





Compute Histogram of Equalized Image

```
hist_eq = zeros(256,1);

for i = 1:M
    for j = 1:N
        hist_eq(Ieq(i,j) + 1) = ...
            hist_eq(Ieq(i,j) + 1) + 1;
```

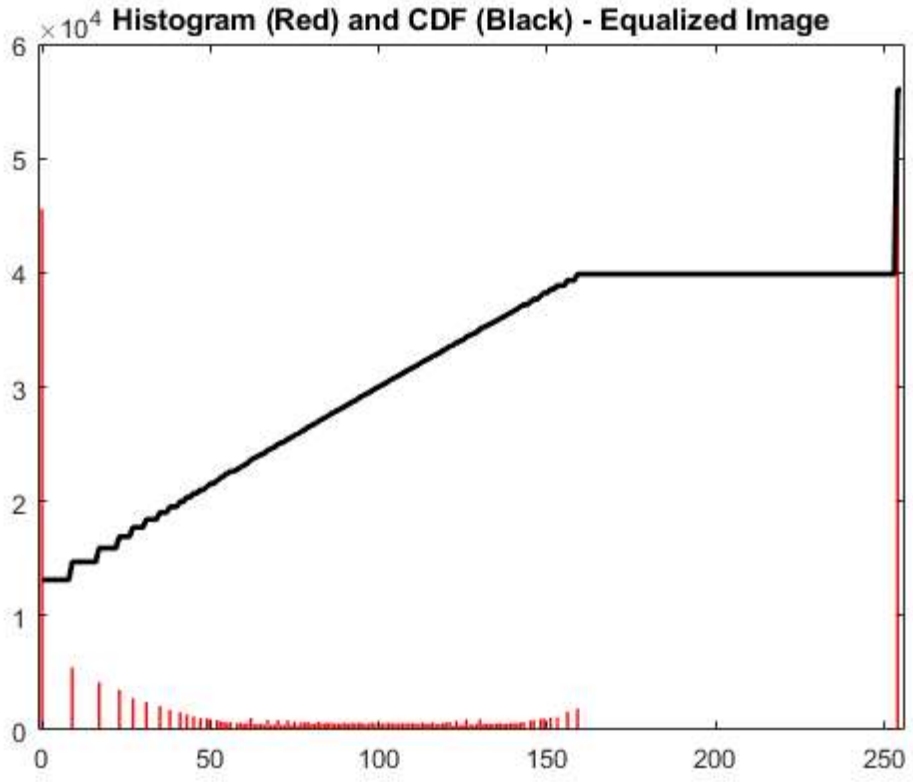
```
    end
end
```

Compute CDF of Equalized Image

```
cdf_eq = cumsum(hist_eq) / (M * N);
```

Display Histogram and CDF of Equalized Image

```
figure;
bar(0:255, hist_eq, 'r');              % Histogram (red bars)
hold on;
plot(0:255, cdf_eq * max(hist_eq), ...
     'k', 'LineWidth', 2);             % Scaled CDF (black curve)
hold off;
title('Histogram (Red) and CDF (Black) - Equalized Image');
```



Published with MATLAB® R2021a