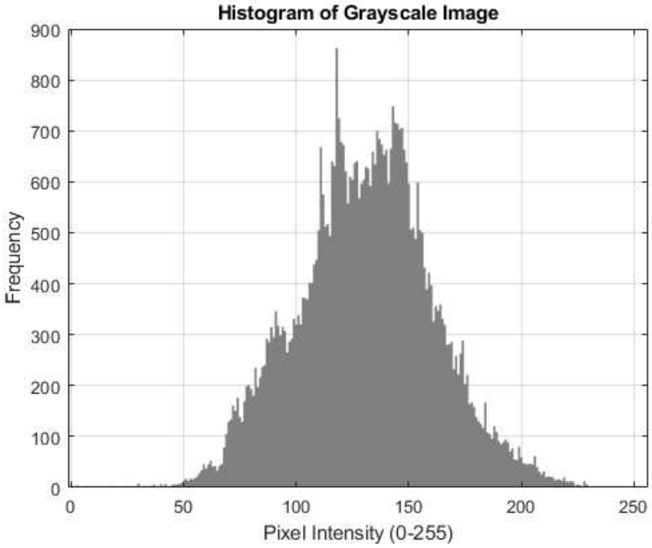
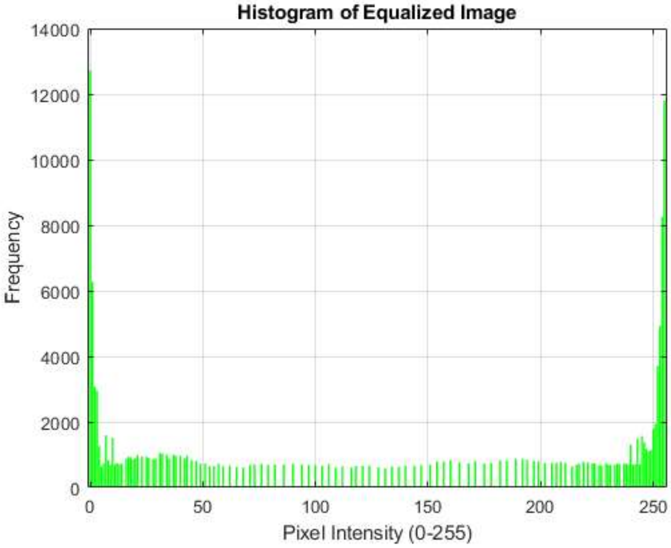
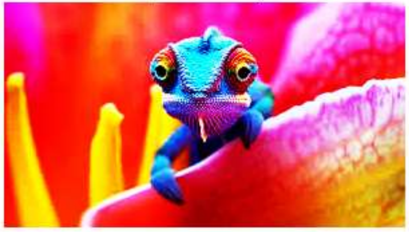
```
clc
clear all;
close all;

clc
clear all;
close all;

colorimg=imread("download.jpg");
grayImage=rgb2gray(colorimg);
montage({grayImage,colorimg}); % Showing the images side by side

L = 256; % Number of intensity levels for 8-bit images
histogram = zeros(1, L); % Array to store counts of each intensity

% Step 3: Count pixel intensities
[M, N] = size(grayImage); % Get image dimensions
for i = 1:M
    for j = 1:N
        pixel_value = grayImage(i, j); % Get pixel intensity
        histogram(pixel_value + 1) = histogram(pixel_value + 1) + 1; % Increment count
    end
end

% Step 4: Plot the histogram
figure;
bar(0:L-1, histogram, 'BarWidth', 1, 'FaceColor', [0.5 0.5 0.5]);
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency');
title('Histogram of Grayscale Image');
grid on;

clc
clear all;
close all;

colorimg=imread("download.jpg");
grayImage=rgb2gray(colorimg);
montage({grayImage,colorimg});

L = 256;
histogram = zeros(1, L);

[M, N] = size(grayImage);
for ii = 1:M
    for jj = 1:N
        pixel_value = grayImage(ii, jj); % Get pixel intensity
        histogram(pixel_value + 1) = histogram(pixel_value + 1) + 1;
    end
end


total_pixels = sum(histogram);
normalized_histogram = histogram / total_pixels;
cdf = cumsum(normalized_histogram);

cdf_min = min(cdf(cdf > 0));
normalized_cdf = round((cdf - cdf_min) / (1 - cdf_min) * (L - 1));


equalized_image = uint8(normalized_cdf(colorimg + 1));


equalized_histogram = imhist(equalized_image);




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



figure;
bar(0:L-1, equalized_histogram, 'FaceColor', 'g');
xlabel('Pixel Intensity (0-255)');
ylabel('Frequency');
title('Histogram of Equalized Image');
grid on;


figure;
imshow(equalized_image);
title('Equalized Image');
```


Histogram of Grayscale Image


Histogram of Equalized Image


Equalized Image

*Published with MATLAB® R2020b*