

---

## Table of Contents

Problem 1 - Histogram Equalization. ....	1
Problem 2 - Loading image. ....	2
Problem 2a - Edge Detection: Roberts Operator. ....	2
Problem 2b - Edge Detection: Prewitt Operator. ....	3
Problem 2c - Edge Detection: Sobel Operator. ....	4
Problem 3 - Laplacian Edge Sharpening. ....	5
Functions. ....	6

Dan Otieno. EE 384 -> Spring '24. Classwork 6. Due date: 02/27/24. Credit to Sayan Samanta for assistance with Matlab scripts.

## Problem 1 - Histogram Equalization.

```
close all;clear all;clc
qla = imread('lowcontrast.jpg');
figure(1);
imshow(qla);('Display of original image.');
```

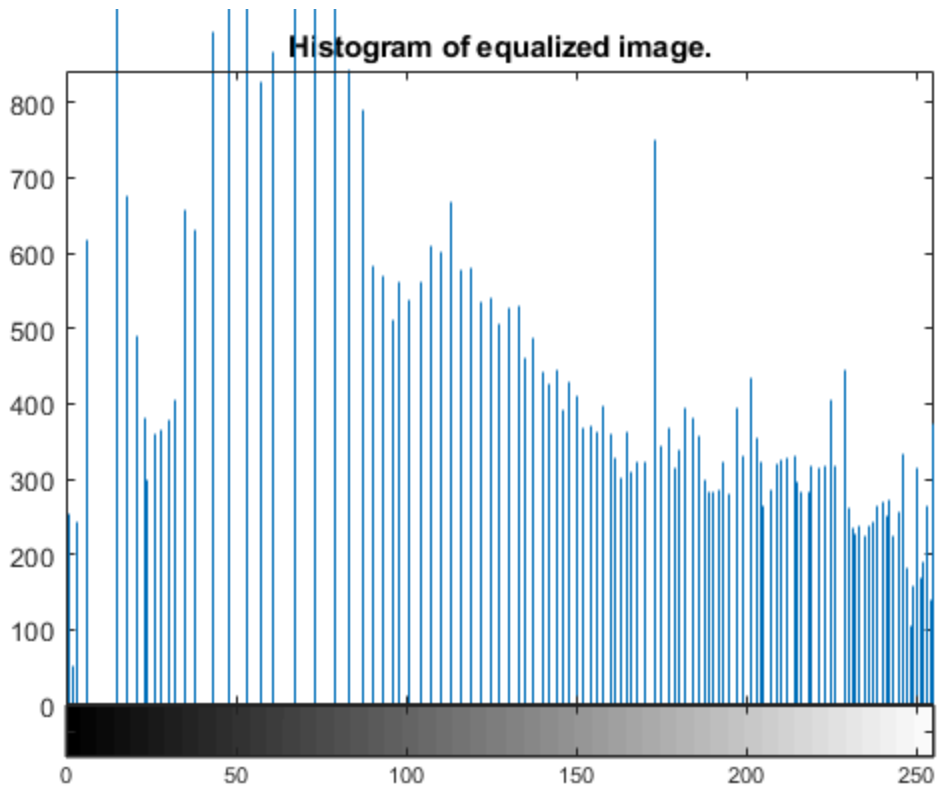
```
figure(2);
imhist(qla);title('Histogram of original image.');
```

```
histoeq = fnHisteq(qla);
figure(3);
imshow(histoeq);title('Display of equalized image.');
```

```
figure(4);
imhist(histoeq);title('Histogram of equalized image.');
```

Display of equalized image.





## Problem 2 - Loading image.

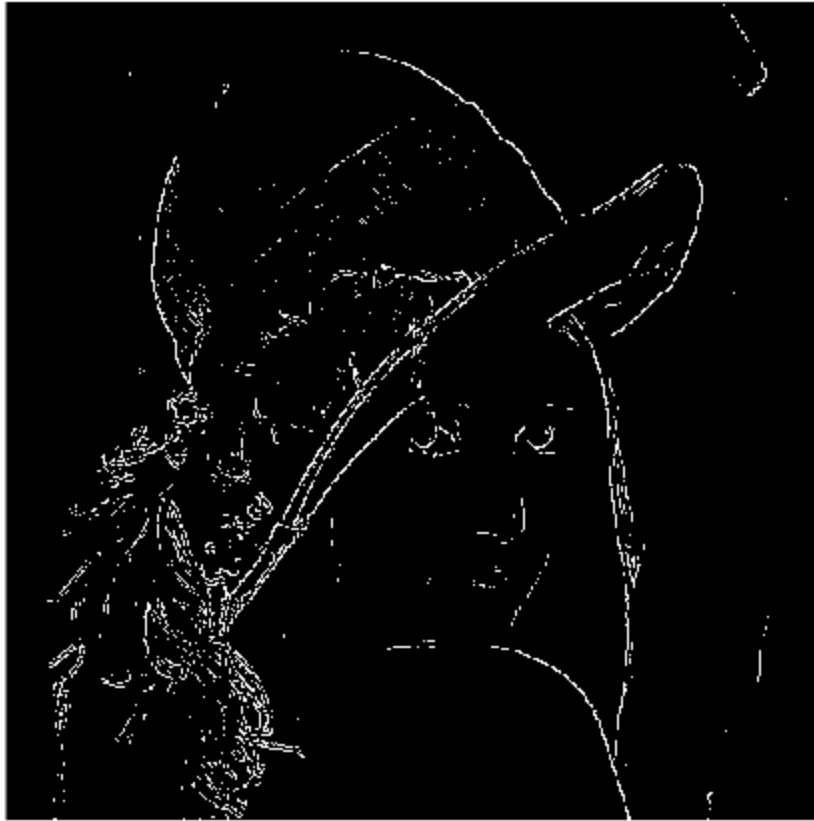
```
q2 = imread('lena.bmp');  
q2Gray = rgb2gray(q2);
```

## Problem 2a - Edge Detection: Roberts Operator.

```
q2a = edge(q2Gray, "roberts");  
figure(5);  
imshow(q2a);title('Edge Detection: Roberts.');
```

---

Edge Detection: Roberts.



## Problem 2b - Edge Detection: Prewitt Operator.

```
q2b = edge(q2Gray, "prewitt");  
figure(6);  
imshow(q2b);title('Edge Detection: Prewitt.');
```

---

Edge Detection: Prewitt.



## Problem 2c - Edge Detection: Sobel Operator.

```
q2c = edge(q2Gray, "sobel");  
figure(7);  
imshow(q2c);title('Edge Detection: Sobel.');
```

---

Edge Detection: Sobel.



## Problem 3 - Laplacian Edge Sharpening.

```
q3 = imread('blur_image.jpg');  
C = im2double(q3);  
fnLaPlacian(C);
```



## Functions.

```
function value = fnHisteq(imageIn)
    l = 256;
    histo = imhist(imageIn);
    cdf = cumsum(histo);
    norm = cdf / numel(imageIn);
    val = l*norm(double(imageIn) + 1);
    value = uint8(val);
end

function imOut = fnLaPlacian(imIn)
    mask = [-1 -1 -1; -1 8 -1; -1 -1 -1];
    convq3 = imfilter(imIn, mask);

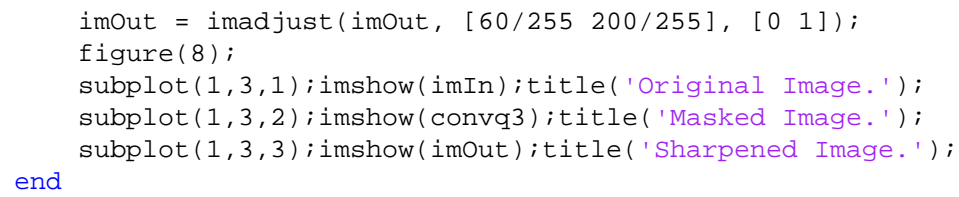
    % normalize image.
    minCov = min(convq3(:));
    maxCov = max(convq3(:));
    convq3 = (convq3 - minCov) / (maxCov - minCov);

    % Sharpening image.
    sharpImg = imIn + convq3;
    minSharp = min(sharpImg(:));
    maxSharp = max(sharpImg(:));
    imOut = (sharpImg - minSharp) / (maxSharp - minSharp);
```

---

---

```
imOut = imadjust(imOut, [60/255 200/255], [0 1]);  
figure(8);  
subplot(1,3,1);imshow(imIn);title('Original Image.');
```



The figure consists of three subplots arranged horizontally. The first subplot, titled 'Original Image.', shows a grayscale image of a person's face. The second subplot, titled 'Masked Image.', shows the same image with a central rectangular region masked out, appearing black. The third subplot, titled 'Sharpened Image.', shows the original image with enhanced edges, making features like the nose and hair more prominent.

```
subplot(1,3,2);imshow(convq3);title('Masked Image.');
```

The third subplot shows the result of applying a sharpening filter (convq3) to the masked image. The edges of the face are more defined, and the overall contrast is higher.

```
subplot(1,3,3);imshow(imOut);title('Sharpened Image.');
```

```
end
```

*Published with MATLAB® R2023a*