**BCSE0131: Digital Image Processing Lab**

**Name:**  Saksham Sharma

**University Roll.No**: 191500698

**Sec/Class Roll.No**: C/23

**Experiment 01: Create command to familiarize with MATLAB & create the matrices and perform the various operations on them.**

[1 2 3]

1:5

1:2:5

a = [1 2 3 ; 4 5 6; 7 8 9]

a(2,3)

a(1,2)

a( :, 2)

a(2,:)

a(:,3)

a(3,:)

a(2:3, 2:3)

a(1:2:3, 2:3)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%mathematical functions:

%1. zeros()

%2. ones()

%3. eye()

%4. diag()

%5. size(a)

%6. length(a)

%7. det(a)

%8. inv(a)

%9. eig(a)

%10. rank(a)

a=[1 2; 3 4]

b=[2 2; 3 1]

c=a./b

d=a.\*b

e=a.^2

f=a.^b

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A = [1 2 3; 4 5 6; 7 8 9]

B = [-1 3 10; -9 5 25; 0 14 2]

S = [1 8 5 6 3]

T = [7 ; 0 ; 11]

a = A.+B

b = S.-T

e = inv(A)

c = A.\*e % A x (inverse of A) != I

d = S.\*S

A(2:3, 2:3)

g=min(min(A))

h=min(min(B))

i=min(min(S))

j=min(min(T))

k=max(max(A))

l=max(max(B))

m=max(max(S))

n=max(max(T))

%1.

abs(A)

%2.

sign(A)

%3.

sin(0)

%4.

cos(90)

%5. for exponent

exp(13)

%6.

round(mean(S))

%7.

floor(15.46)

%8.

ceil(15.46)

%9.

sort(S)

%10.

sum(S)

%11.

prod(S)

%12.

mean(S)

%13.

median(S)

**Experiment 2:** **Understanding image basic “image resize, image type conversion, extraction of color band, creating a synthesic image, pseudocolor image**

clc

clear all

img = imread('C:\Users\glau\Desktop\Radhika\img1.jpg');

img1 = rgb2gray(img);

img2 = imresize(imresize(img1,1/16),16);

imshow(img1)

subplot(2,3,1)

imshow(img)

title('original image')

subplot(2,3,2)

imshow(img1)

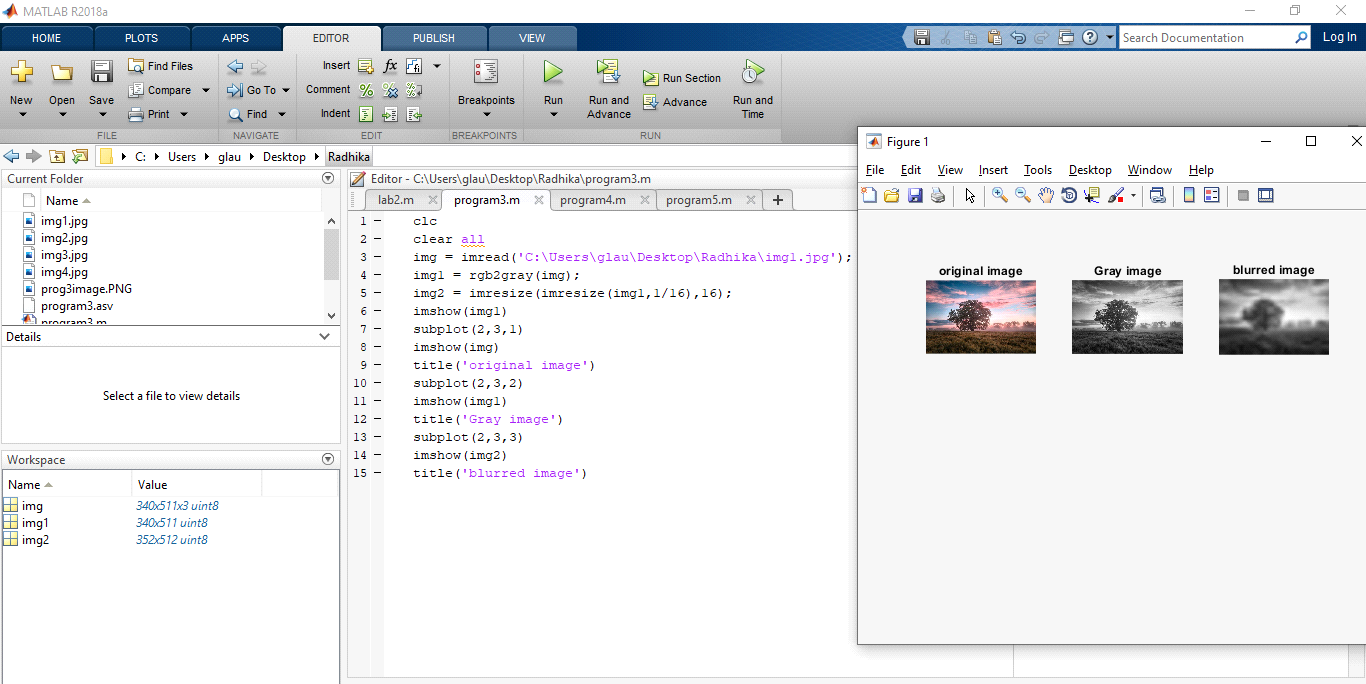
title('Gray image')

subplot(2,3,3)

imshow(img2)

title('blurred image')

**OUTPUT:-**



% Resizing the image-----------

clc

clear all

myimage = imread('C:\Users\admin\Downloads\download.jpg');

figure,imshow(myimage), title 'Original Image';

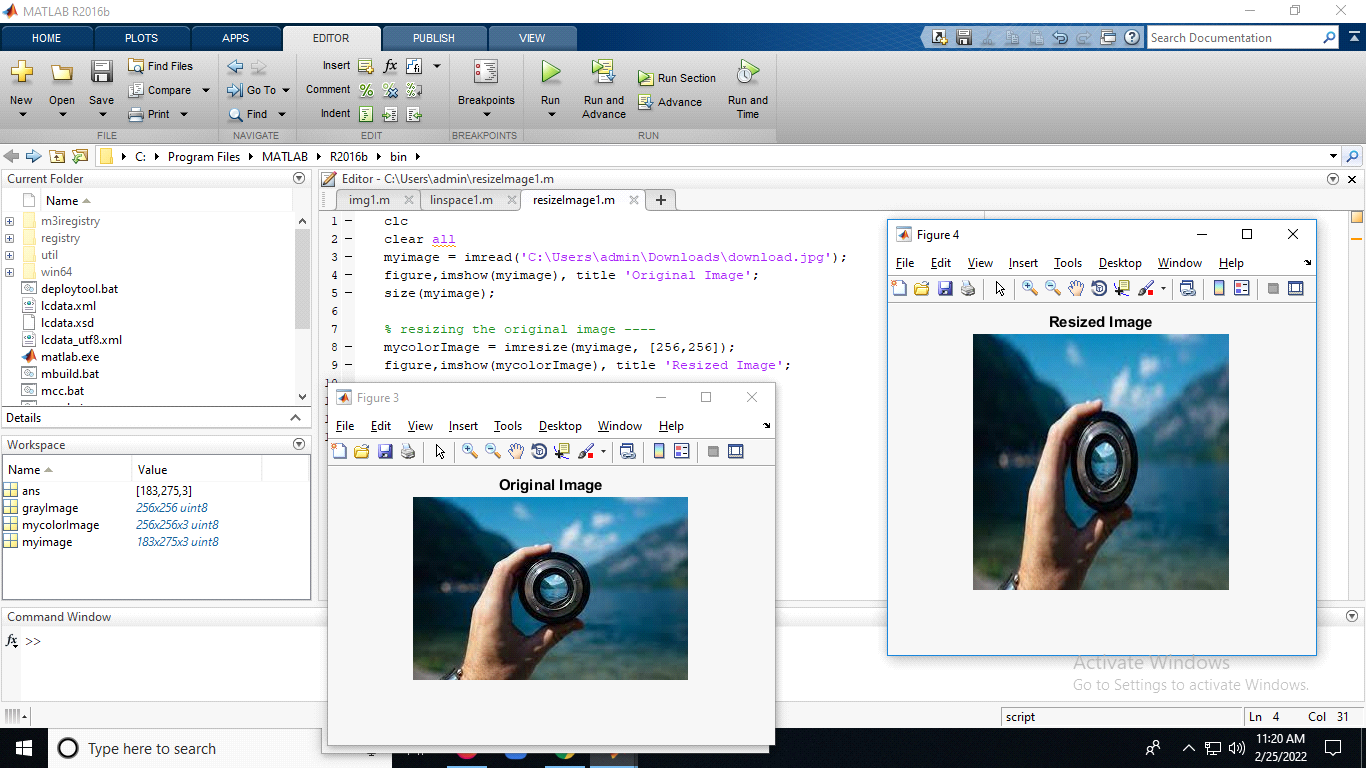
size(myimage);

% resizing the original image ----

mycolorImage = imresize(myimage, [256,256]);

figure,imshow(mycolorImage), title 'Resized Image';

**OUTPUT:-**



% bw, redcomp,grencomp,. bluecomp, -------

clc

clear all

mycolorImage = imread('C:\Users\admin\Downloads\download.jpg');

figure,imshow(mycolorImage), title 'Colored Image';

%black and white image------

bw = im2bw(mycolorImage);

figure,imshow(bw), title 'bw Image';

%red cmoponent image------

redcomp = mycolorImage(:,:,1);

figure,imshow(redcomp), title 'Red Image';

%green cmoponent image------

greencomp = mycolorImage(:,:,2);

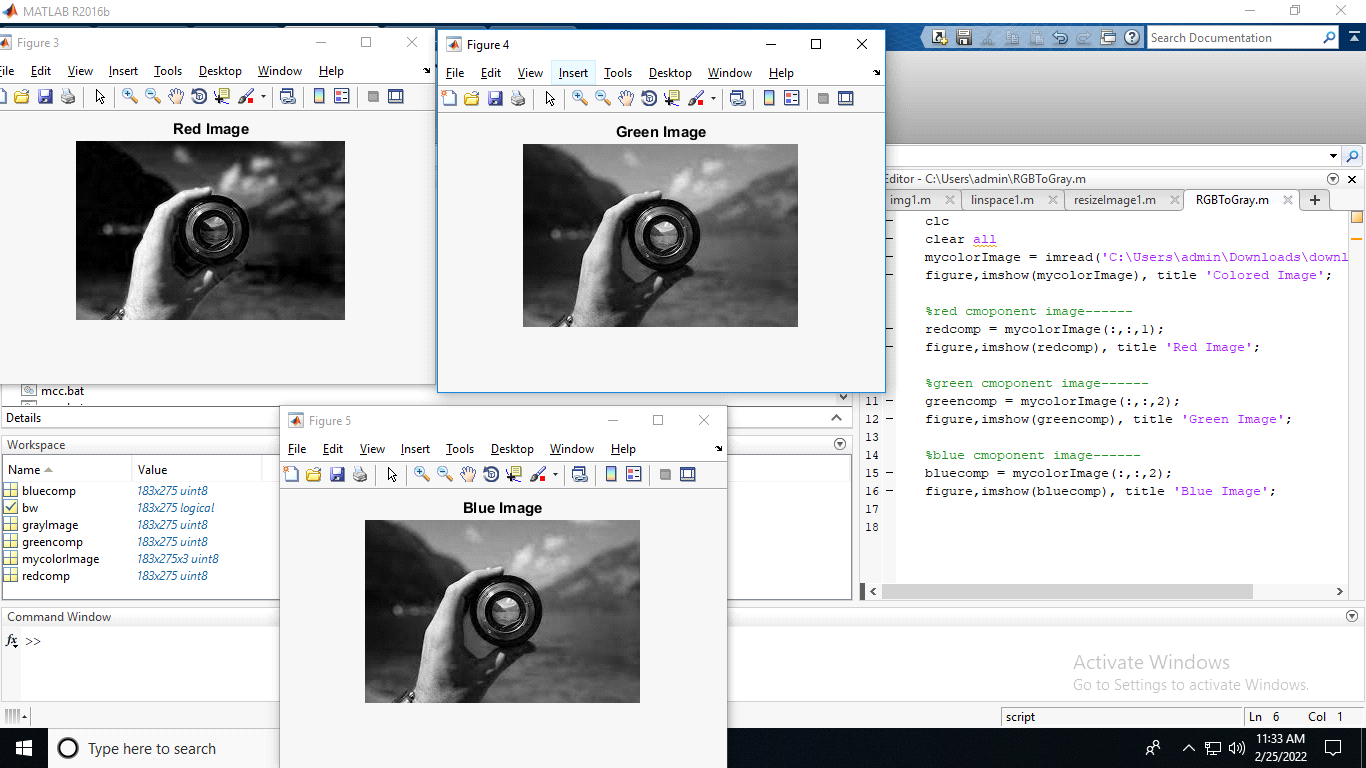
figure,imshow(greencomp), title 'Green Image';

%blue cmoponent image------

bluecomp = mycolorImage(:,:,2);

figure,imshow(bluecomp), title 'Blue Image';

**OUTPUT:-**



Experiment No 3: Perform various arithmetic operation logical operation (NOT, OR and XOR) on images

clc

clear all

img1 = imread('img3.jpg');

myimage1 = imresize(img1,[256,256])

img2 =imread('img4.jpg');

myimage2 = imresize(img2,[256,256])

bw1 = im2bw(myimage1);

bw2 = im2bw(myimage2);

and1 = bw1 & bw2;

or1 = bw1 | bw2;

xor1 = xor(bw1,bw2);

subplot(2,3,1)

imshow(myimage1)

title('Original Image 1');

subplot(2,3,2)

imshow(myimage2)

title('Original Image 2');

subplot(2,3,3)

imshow(bw1)

title('Black&white Image 1');

subplot(2,3,4)

imshow(bw2)

title('Black&white Image 2');

subplot(2,3,5)

imshow(and1)

title('And Operation');

subplot(2,3,6)

imshow(or1)

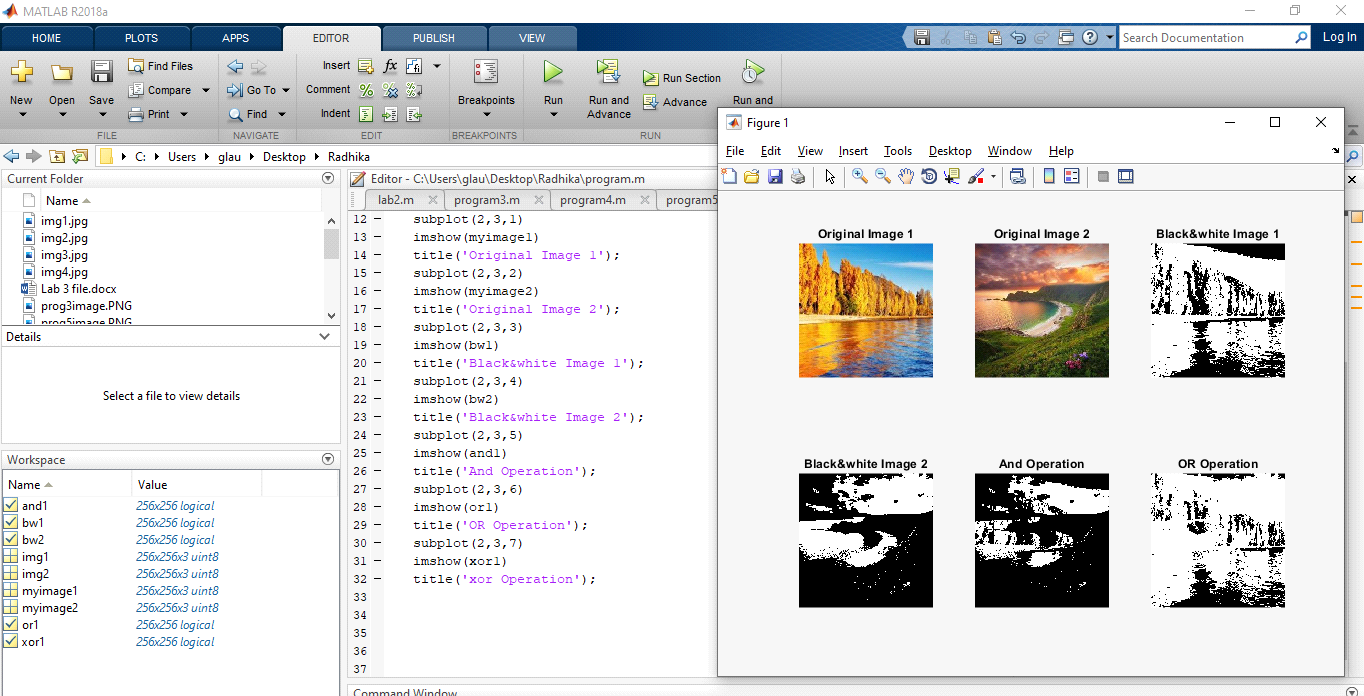
title('OR Operation');

subplot(2,3,7)

imshow(xor1)

title('xor Operation');

**OUTPUT:-**



Experiment No 4: Perform contrast operations, create histogram and perform histogram equalization.

% histogram using of loop.........

clc

clear all

gd = imread('C:\Users\GLAU\Downloads\goodContrast.jpg');

img1 = rgb2gray(gd);

h = zeros(1,300);

[r,c]=size(img1);

for i=1:r

for j=1:c

if(img1(i,j) == 0)

img1(i,j) = 1;

end

end

end

for i=1:r

for j=1:c

t = img1(i,j);

h(t) = h(t)+1;

end

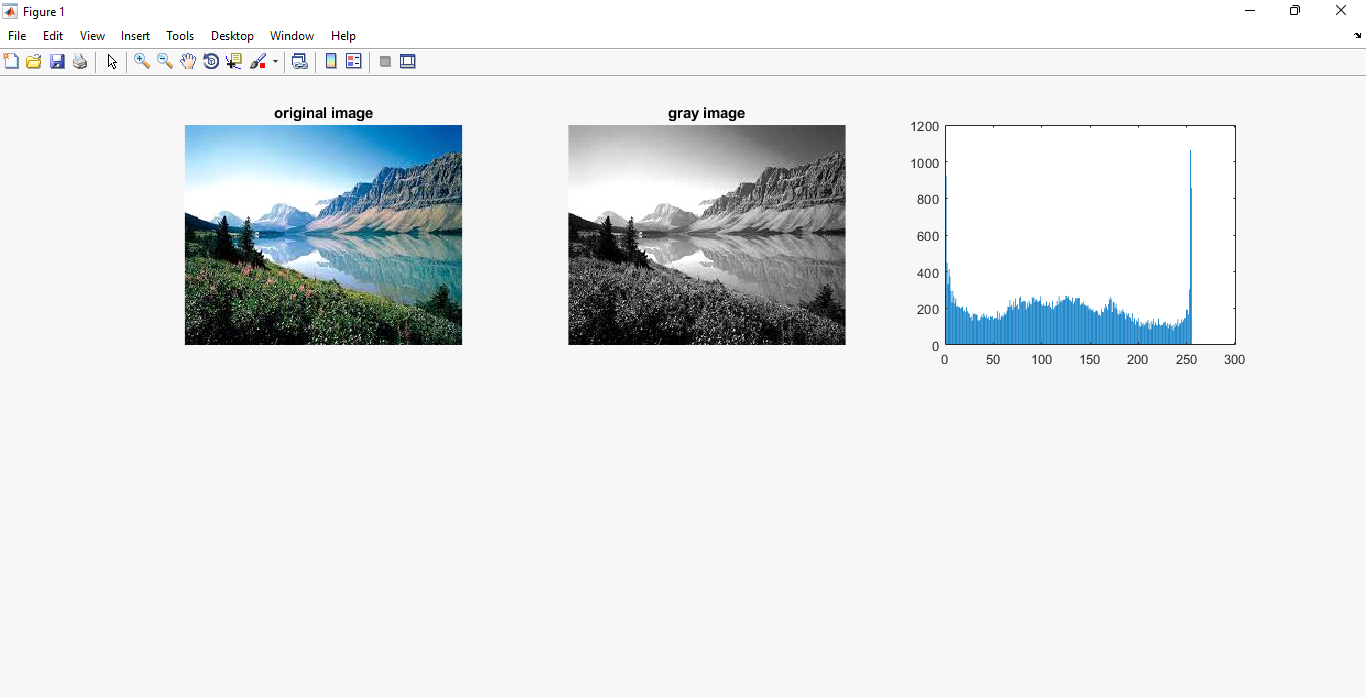
end

subplot(2,3,1); imshow(gd), title 'original image';

subplot(2,3,2); imshow(img1), title 'gray image';

subplot(2,3,3),bar(h);

**OUTPUT:-**



% Histogram Equalization..........

clc

clear all

goodC = imread('C:\Users\GLAU\Downloads\goodContrast.jpg');

img1 = rgb2gray(goodC);

subplot(3,4,1); imshow(img1), title 'good contrast image';

subplot(3,4,2); imhist(img1);

img2 = histeq(goodC);

subplot(3,4,3); imshow(img2), title 'equalized image';

subplot(3,4,4); imhist(img2);

highC = imread('C:\Users\GLAU\Downloads\poorContrast.jpg');

subplot(3,4,5); imshow(highC), title 'high contrast image';

subplot(3,4,6); imhist(highC);

img3 = histeq(highC);

subplot(3,4,7); imshow(img3), title 'equalized image';

subplot(3,4,8); imhist(img3);

poorC = imread('C:\Users\GLAU\Downloads\highContrast.jpg');

subplot(3,4,9); imshow(poorC), title 'low contrast image';

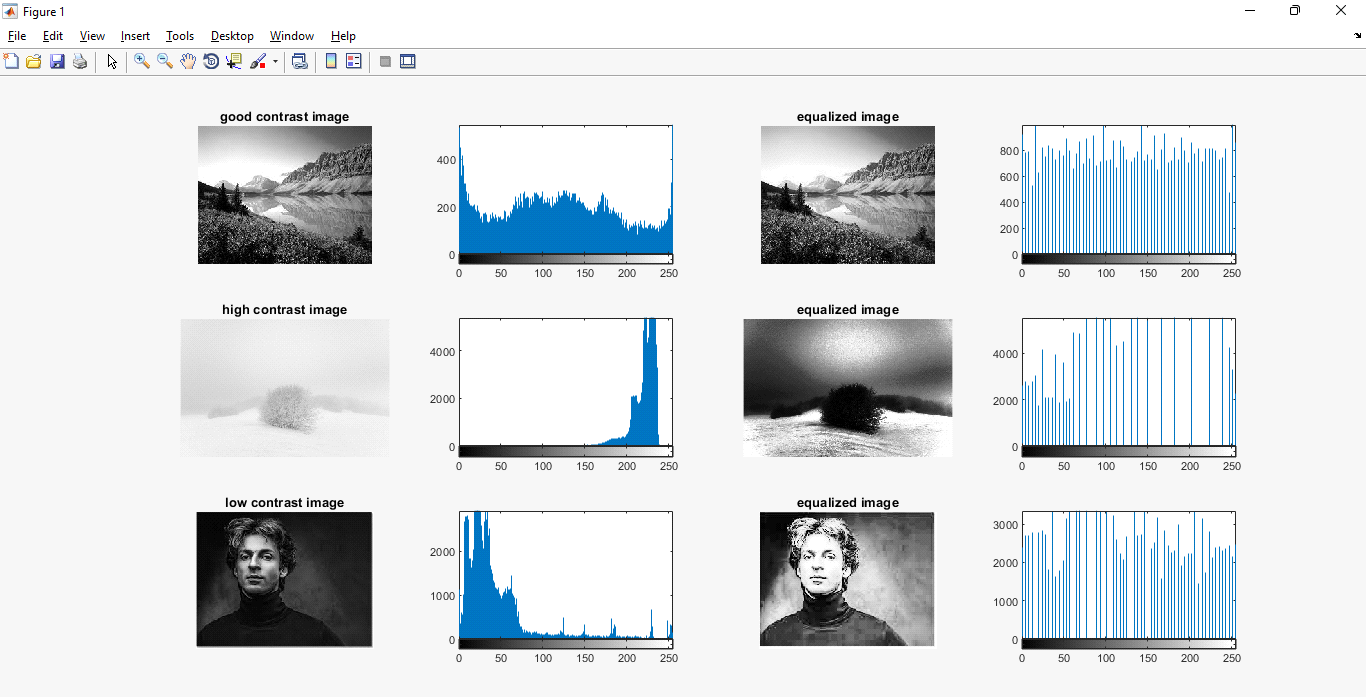
subplot(3,4,10); imhist(poorC);

img4 = histeq(poorC);

subplot(3,4,11); imshow(img4), title 'equalized image';

subplot(3,4,12); imhist(img4);

**OUTPUT:-**



clc

clear all

img1 = imread('C:\Users\glau\Desktop\Radhika\img1.jpg');

img2 = rgb2gray(img1);

h = zeros(1,300);

[r c] = size(img2);

for i = 1:r

for j = 1:c

if(img2(i,j)==0)

img2(i,j)=1;

end

end

end

for i = 1: r

for j = 1:c

t=i\*j;

t = img2(i,j);

h(t) = h(t)+1;

end

end

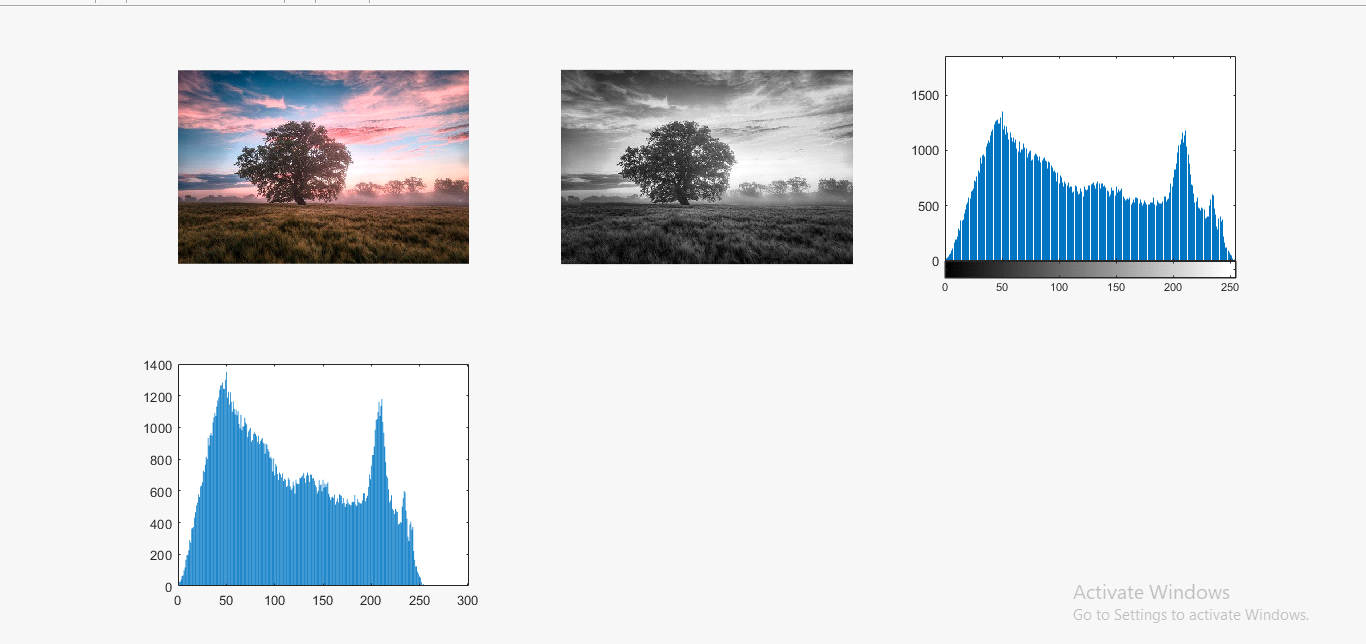
subplot(2,3,1), imshow(img1);

subplot(2,3,2), imshow(img2);

subplot(2,3,3), imhist(img2);

subplot(2,3,4),bar(h);

**OUTPUT:-**



Experiment No 5: Perform contrast stretching and gamma correction on image.

% Contrast stretching of image....

image = imread('satellite\_Image.png');

stretched\_Image = imadjust(image, stretchlim(image, [0.05, 0.95]),[]);

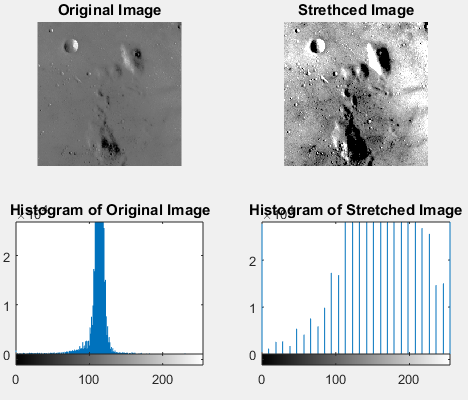
subplot(2,2,1), imshow(image), title('Original Image');

subplot(2,2,2), imshow(stretched\_Image), title('Strethced Image');

subplot(2,2,3), imhist(image), title('Histogram of Original Image');

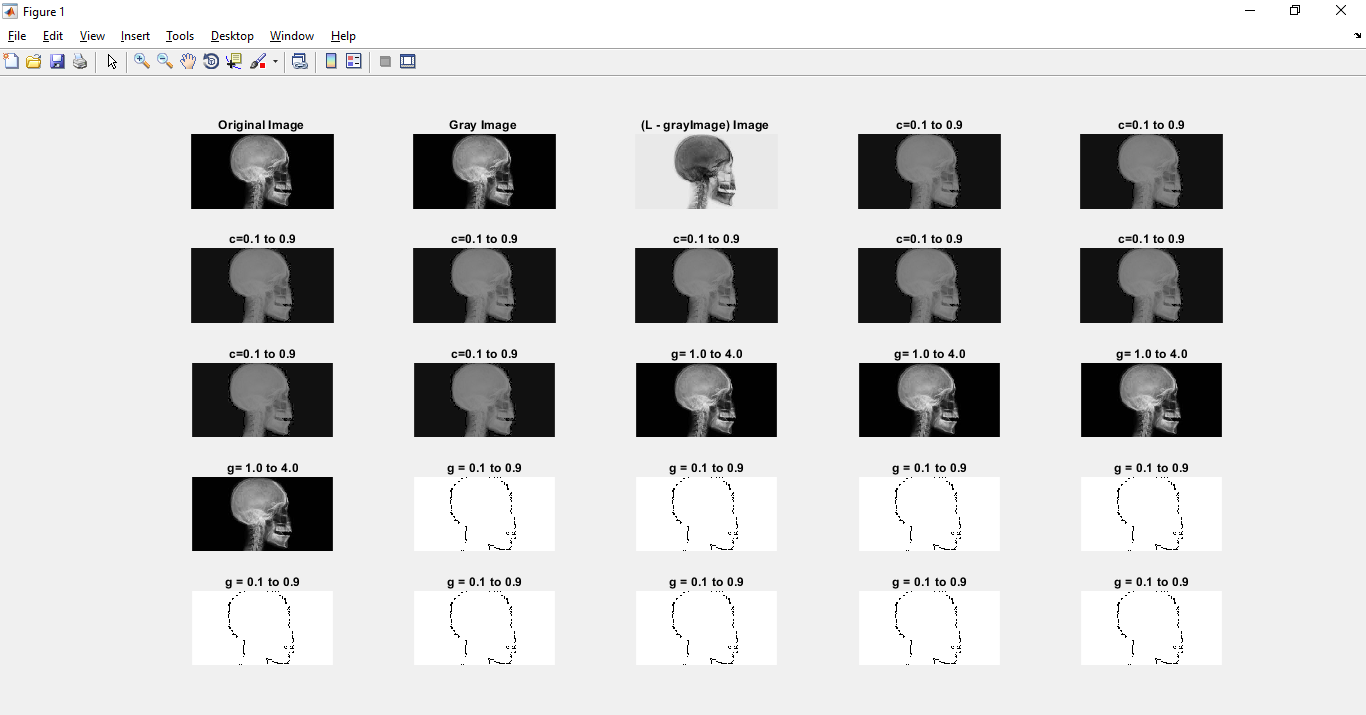
subplot(2,2,4), imhist(stretched\_Image), title('Histogram of Stretched Image');

**OUTOUT:-**

****

% Contrast stretching and gamma correction of an image....

Clc  
clear all  
  
img1 = imread('C:\Users\CL224\Downloads\1.jpeg');  
img2 = rgb2gray(img1);  
L = max(max(img2));  
img3 = L - img2;  
subplot(5,5,1), imshow(img1), title 'Original Image ';   
subplot(5,5,2), imshow(img2), title 'Gray Image ';   
subplot(5,5,3), imshow(img3), title '(L - grayimage) Image ';   
  
a = double(img2);  
c=1;  
r=9;  
k=4;  
  
for i=c:r  
    img4 = c ./ 10 .\* log(1+a);  
    subplot(5,5,k);  
    imshow(img4), title 'c=0.1 to 0.9';  
    k = k+1;  
end  
  
g=1;  
c=1  
q=4;  
k=13;  
for i=g:q  
    img5 = c .\* (img2 .^ g);  
    subplot(5,5,k);  
    imshow(img5), title 'g= 1.0 to 4.0';  
    k = k+1;  
end  
a = double(img2);  
g=1;  
q=9;  
c=1;  
k=17;  
for i=g:q  
    img5 = c .\* (a .^ (g ./ 10));  
    subplot(5,5,k);  
    imshow(img5), title 'g = 0.1 to 0.9';  
    k = k+1;  
end

**OUTPUT:-**

Experiment No 6: Perform Smoothing using linear and order statistics filter min, amx and med of various sizes and sharpen an image using Laplacian filter.

% Smoothing an image using linear filter....

clc

clear all

img1 = imread('C:\Users\glau\Desktop\Radhika\noisy.jpg');

img2 = rgb2gray(img1);

[m, n] = size(img2);

img2 = double(img2);

size1 = input('input fiter size');

f = ones(size1);

c = (size1+1)/2;

img3 = img2;

for i = c:m-c+1

for j = c:n-c+1

sum = 0;

for k = 1:size1

for l = 1:size1

sum = sum + img2(i-c+k,j-c+1) \* f(k,l);

end

end

img3(i,j) = sum / (size1^2);

end

end

subplot(2,2,1), imshow(uint8(img1));

title('Original Image');

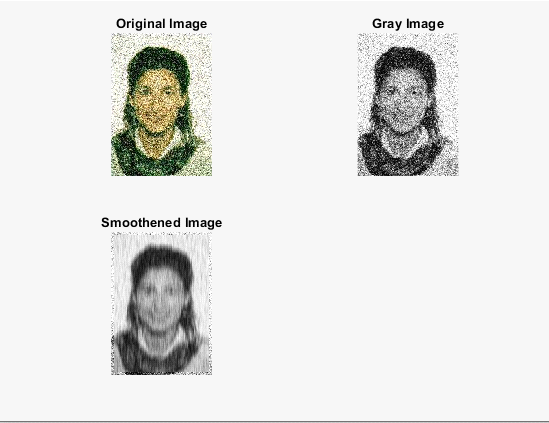
subplot(2,2,2), imshow(uint8(img2));

title('Gray Image');

subplot(2,2,3), imshow(uint8(img3));

title('Smoothened Image');

**OUTPUT:-**



% Smoothing of an image using average filter with normal image

clc

clear all

img= imread('C:\Users\CL224\Desktop\prachi\prachi.png');

img2=rgb2gray(img);

[m n] = size(img2);

img2=double(img2);

size1=input('input filter size');

f=ones(size1);

c=(size1+1)/2;

img3=img2;

for i=c: m-c+1

    for j=c: n-c+1

        sum=0;

        for k=1:size1

            for l=1:size1

               %sum=sum+img2(i-c+1,j-c+1)\*f(k,l);

                sum=sum+img2(i-c+k,j-c+l)\*f(k,l);

            end

        end

        img3(i,j)=sum/(size1^2);

    end

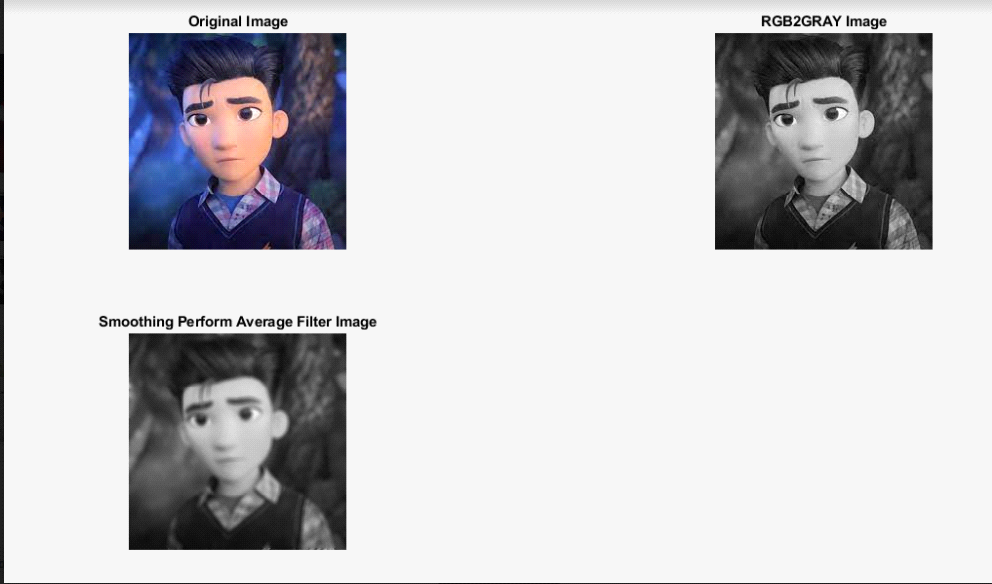
end

subplot(2,2,1),imshow(uint8(img)), title ('Original Image');

subplot(2,2,2),imshow(uint8(img2)),title ('RGB2GRAY Image');

subplot(2,2,3),imshow(uint8(img3)),title ('Smoothing Perform Average Filter Image');

**OUTPUT:-**



% Sharpening the image using Laplacian filter

clc

clear all

img1 = imread('C:\Users\glau\Desktop\Radhika\flower.jpg');

img2 = rgb2gray(img1);

[m, n] = size(img2);

img2 = double(img2);

c = 2;

img3 = img2;

f = [0 1 0;1 -4 1; 0 1 0];

for i = 2:m-1

for j = 2:n-1

sum = 0;

for k = 1:3

for l = 1:3

sum = sum+img2(i-2+k,j-2+l)\*f(k,l);

end

end

img3(i,j) = sum;

end

end

subplot(2,2,1), imshow(img1);

title('Original Image')

subplot(2,2,2), imshow(uint8(img3));

title('Sharpened Image')

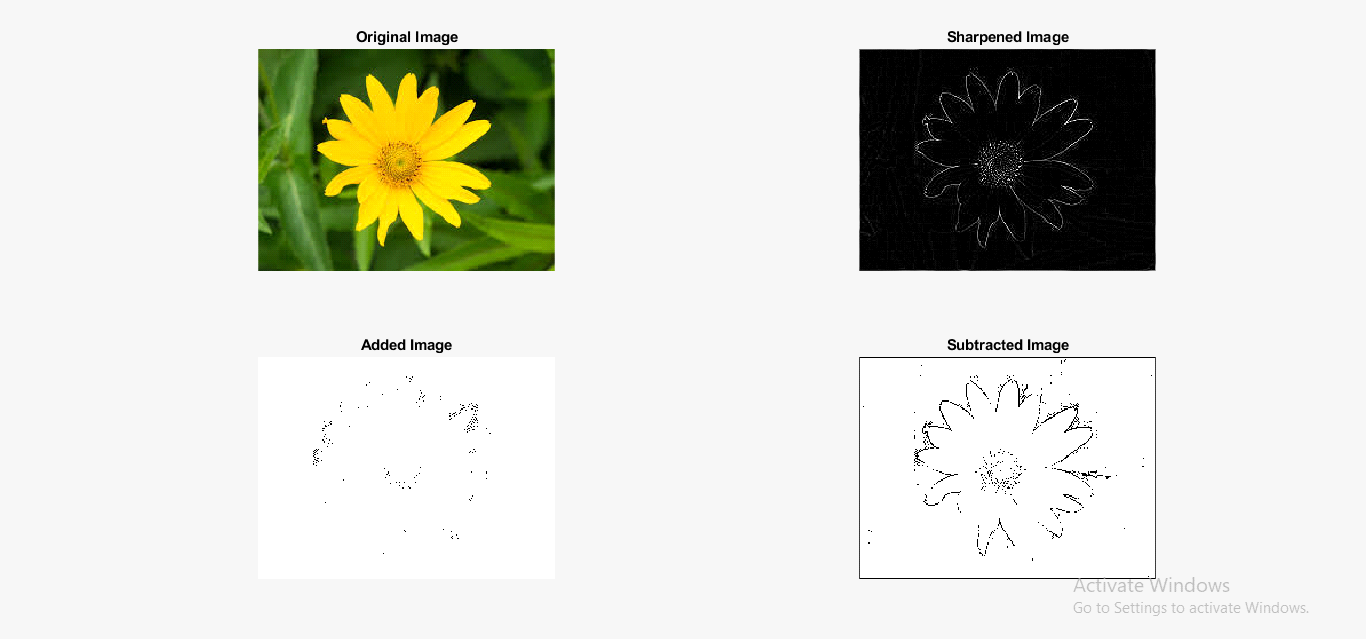
subplot(2,2,3),imshow(imadd(img2,img3));

title('Added Image')

subplot(2,2,4),imshow(imsubtract(img2,img3));

title('Subtracted Image')

**OUTPUT:-**



% Adding RGB2GRAY Image and Laplace Filter Image

img= imread('C:\Users\CL224\Desktop\prachi\prachi.png');

img2=rgb2gray(img);

[r c1]=size(img2);

img2=double(img2);

c=2;

img3=img2;

f=[0 1 0; 1 -4 1; 0 1 0]

for i=2:r-1

    for j=2:c1-1

        sum=0;

        for k=1:3

            for l=1:3

                sum=sum+img2(i-2+k,j-2+l)\*f(k,l);

            end

        end

        img3(i,j)=sum;

    end

end

subplot(2,2,1),imshow(uint8(img)), title ('Original Image');

subplot(2,2,2),imshow(uint8(img2)),title ('RGB2GRAY Image');

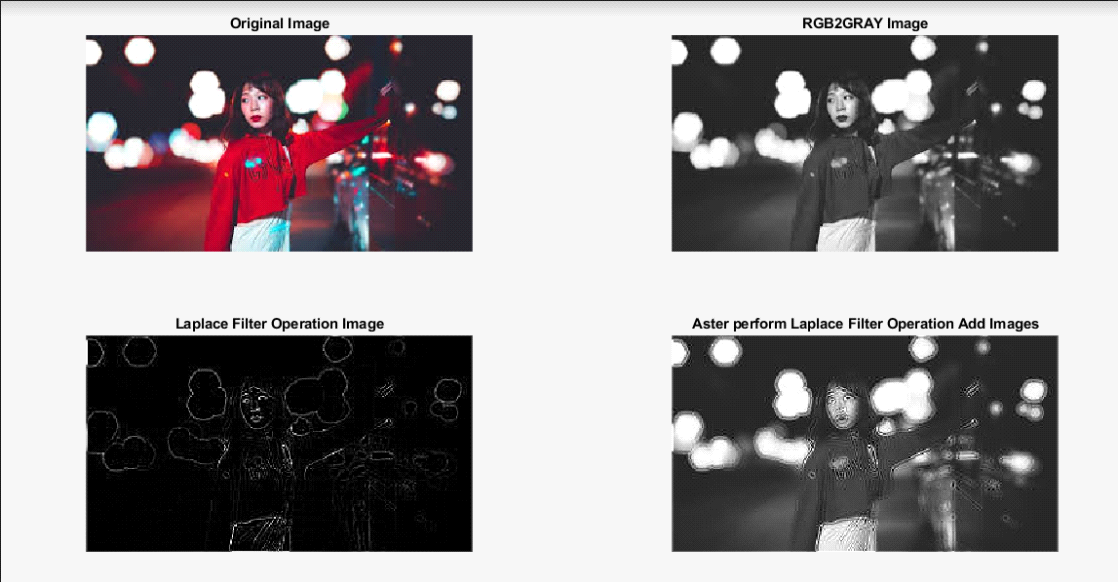
subplot(2,2,3),imshow(uint8(img3)),title ('Laplace Filter Operation Image');

img4=img2;

img4=img3+img2;

subplot(2,2,4),imshow(uint8(img4)),title ('After perform Laplace Filter Operation Add Images');

**OUTPUT:-**



Experiment 7: Perform various Fast Fourier Transformations (FFT) and frequency domain filtering on image using MATLAB.

% Inverse Fourier Transformation...

I = imread('C:\Users\glau\Desktop\Radhika\flower.jpg');

G = rgb2gray(I);

F = fft2(G);

lF = log(1+abs(F))

sf = fftshift(F)

slf = log(1+abs(sf))

ift = ifft2(F)

subplot(2,3,1);

imshow(G),title('Original Image');

subplot(2,3,2);

imshow(lF,[]),title('FFT');

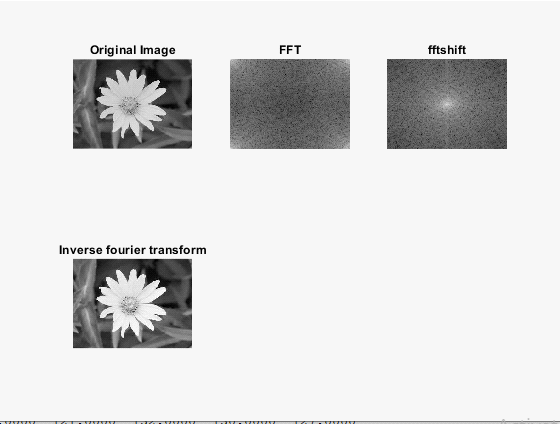
subplot(2,3,3);

imshow(slf,[]),title('fftshift');

subplot(2,3,4);

imshow(ift,[]),title('Inverse fourier transform');

**Output:-**



%LOW PASS FILTERING USING FOURIER TRANSFORM...........

I = imread('C:\Users\glau\Desktop\Radhika\flower.jpg');

A = rgb2gray(I);

B = fft2(A);

lb = log(1+abs(B))

[m, n] = size(A)

mask = zeros(m,n)

for i = 150:180

for j = 210:240

mask(i,j) = 1;

end

end

C = fftshift(mask);

D = B .\*C;

E = abs(D)

subplot(2,3,1),imshow(A),title('Original Image');

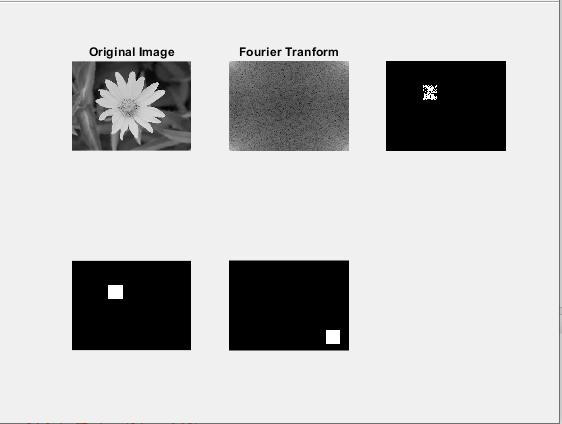
subplot(2,3,2),imshow(lb,[]),title('Fourier Tranform');

subplot(2,3,3),imshow(D);

subplot(2,3,4),imshow(E);

subplot(2,3,5),imshow(mask);

Output:-



%HIGH PASS FILTERING IN FOURIER TRANSFORM........

I = imread('C:\Users\glau\Desktop\Radhika\flower.jpg');

A = rgb2gray(I);

B = fft2(A);

lb = log(1+abs(B))

[m, n] = size(A)

mask = ones(m,n)

for i = 150:180

for j = 210:240

mask(i,j) = 0;

end

end

C = fftshift(mask);

D = B .\*C;

E = abs(D)

subplot(2,3,1),imshow(A),title('Original Image');

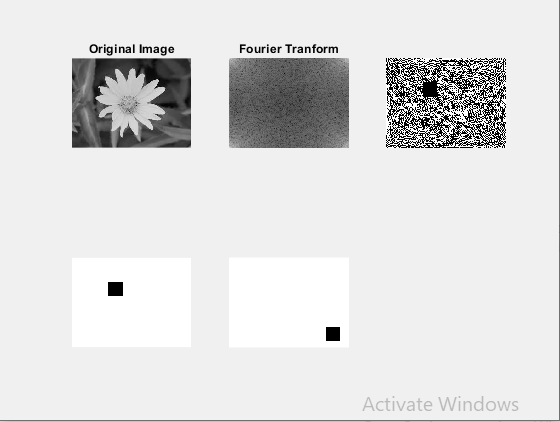
subplot(2,3,2),imshow(lb,[]),title('Fourier Tranform');

subplot(2,3,3),imshow(D);

subplot(2,3,4),imshow(E);

subplot(2,3,5),imshow(mask);

Output:-



Experiment 8: Perform various Morphological operations dilation, erosion, internal and external boundary extraction, thinning, thickening and skeletonizing of image and perform dilation, erosion, boundary extraction without using direct functions.

% EXTRACTING INTERNAL AND EXTERNAL BOUNDARIES FROM THE DILATED AND ERODED IMAGE

% Dilation is used to expand the image pixels

%Taking a b&w image

orgimg = imread('C:\Users\glau\Desktop\Radhika\Alphabet.jpg');

se = strel('square',15);%to define structuring element

%Applying dilation function

dilimg = imdilate(orgimg,se)

%Performing erosion

eroimg = imerode(orgimg,se)

subplot(2,3,1),imshow(orgimg),title('Orignal image');

subplot(2,3,2),imshow(dilimg),title('Dilated image');

subplot(2,3,3),imshow(eroimg),title('Eroded image');

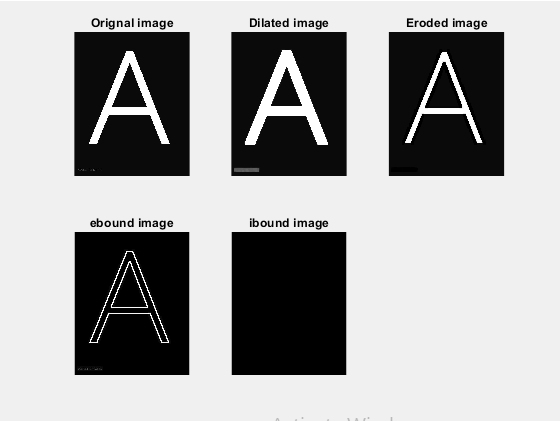
ebound = imsubtract(orgimg, eroimg)

ibound = imsubtract(orgimg, dilimg)

subplot(2,3,4),imshow(ebound),title('External Boundary image');

subplot(2,3,5),imshow(ibound),title('Internal boundary image');

output



% Performing Morphological Operations on the image(Thinning, thickening,skeleton)

% Dilation is used to expand the image pixels

%Taking a b&w image

orgimg = imread('C:\Users\glau\Desktop\Radhika\Alphabet.jpg');

se = strel('square',15);%to define structuring element

%Applying dilation function

dilimg = imdilate(orgimg,se);

%Performing erosion

eroimg = imerode(orgimg,se);

subplot(2,4,1),imshow(orgimg),title('Orignal image');

subplot(2,4,2),imshow(dilimg),title('Dilated image');

subplot(2,4,3),imshow(eroimg),title('Eroded image');

ebound = imsubtract(orgimg, eroimg);

ibound = imsubtract(orgimg, dilimg);

subplot(2,4,4),imshow(ebound),title('External Boundary image');

subplot(2,4,5),imshow(ibound),title('Internal boundary image');

%Thinning of the image

morphthin = bwmorph(im2bw(orgimg),'thin');

subplot(2,4,6),imshow(morphthin),title('Thinning');

%Thickening of the image

mthick = bwmorph(im2bw(orgimg),'thicken');

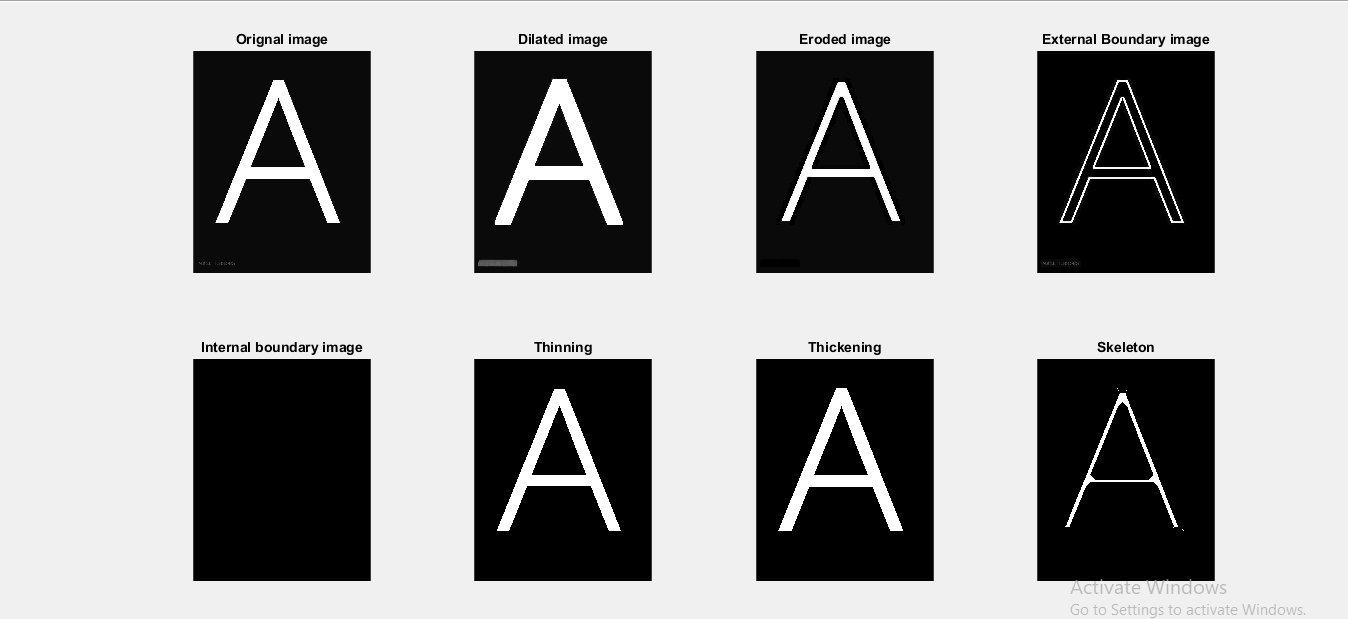
subplot(2,4,7),imshow(mthick),title('Thickening');

%Skeleton of the image

mskel = bwmorph(im2bw(orgimg),'skel',20);

subplot(2,4,8),imshow(mskel),title('Skeleton');

**OUTPUT:-**



% Dilation is used to expand the image pixels

%Taking a b&w image

orgimg = imread('C:\Users\glau\Desktop\Radhika\Alphabet.jpg');

se = strel('square',9);%to define structuring element

%Applying dilation function

dilimg = imdilate(orgimg,se)

%Performing erosion

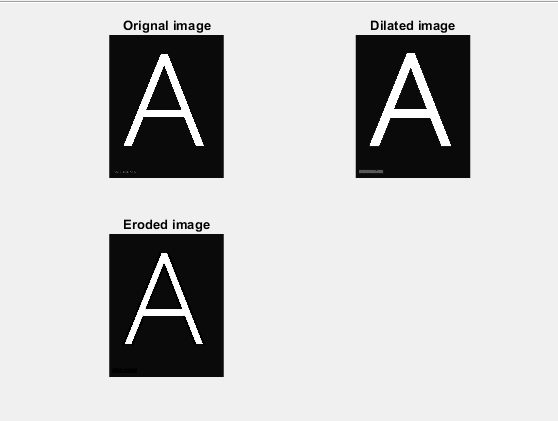
eroimg = imerode(orgimg,se)

subplot(2,2,1),imshow(orgimg),title('Orignal image');

subplot(2,2,2),imshow(dilimg),title('Dilated image');

subplot(2,2,3),imshow(eroimg),title('Eroded image');

Output:-



% Apply Erosion and Dilation in a black and white image using complement sign (~)

clc

clear all

img=imread('C:\Users\CL224\Desktop\prachi\fingerprint.png');

img=imresize(img,[512,512]);

subplot(3,3,1),imshow(img),title("Original Image");

img1=~im2bw(img);

subplot(3,3,2),imshow(img1),title("Black and white Image");

sc=strel('square',4);

dilimg=imdilate(img1,sc);

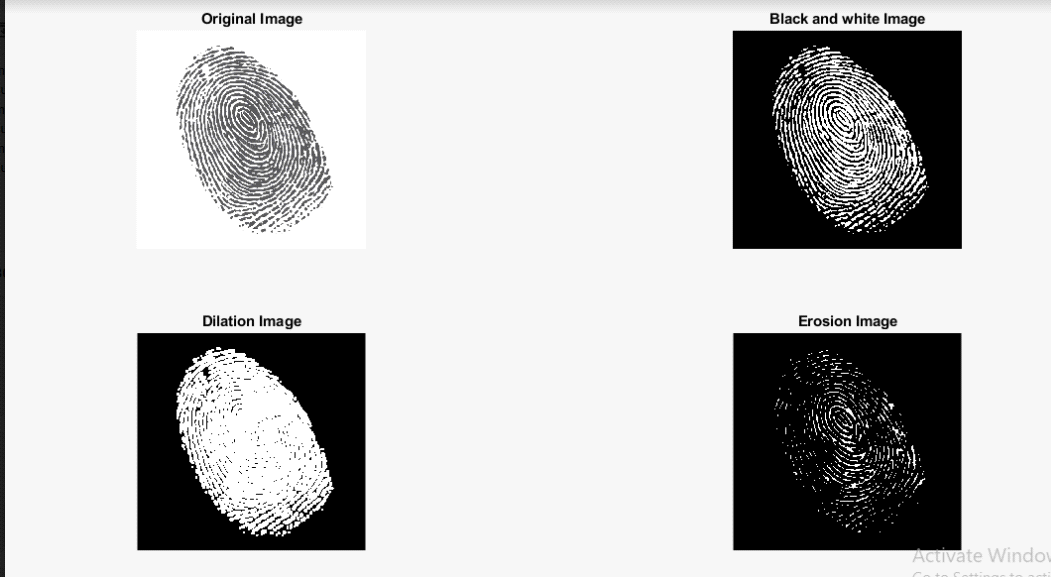
subplot(3,3,3),imshow(dilimg),title("Dilation Image");

eroing=imerode(img1,sc);

eroing=imresize(eroing,[512,512]);

subplot(3,3,4),imshow(eroing),title("Erosion Image");

**OUTPUT:-**



% Apply Erosion and Dilation in a black and white image

clc

clear all

img=imread('C:\Users\CL224\Desktop\prachi\alphabet.png');

img=imresize(img,[512,512]);

subplot(3,3,1),imshow(img),title("Original Image");

img1=im2bw(img);

subplot(3,3,2),imshow(img1),title("Black and white Image");

sc=strel('square',4);

dilimg=imdilate(img1,sc);

subplot(3,3,3),imshow(dilimg),title("Dilation Image");

eroing=imerode(img1,sc);

eroing=imresize(eroing,[512,512]);

subplot(3,3,4),imshow(eroing),title("Erosion Image");

ebound=imsubtract(img1,eroing);

subplot(3,3,5),imshow(ebound),title("Subratct original Image from Erosion image");

dbound=imsubtract(dilimg,img1);

subplot(3,3,6),imshow(dbound),title("Subratct original Image from Dilation image");

**OUTPUT:-**



Experiment 9:- Image Segmentation using Thresholding Function (Simple, Multiple, Adaptive and Optimal Thresholding).

% Image Segmentation using Thresholding Function.....

clc

clear all

A = imread('jupiter.jpg');

subplot(2,3,1),imshow(A),title('Original Image');

img2 = rgb2gray(A);

subplot(2,3,2),imshow(img2),title('Gray Image');

threshold = 128;

[r c] = size(img2);

for i=1:r

for j=1:c

if img2(i,j) <= threshold

b(i,j)=0;

else

b(i,j)=255;

end

end

end

subplot(2,3,3),imshow(b),title('Thresholded Image');

for i=1:r

for j=1:c

if img2(i,j) <= threshold

b2(i,j)=255;

else

b2(i,j)=0;

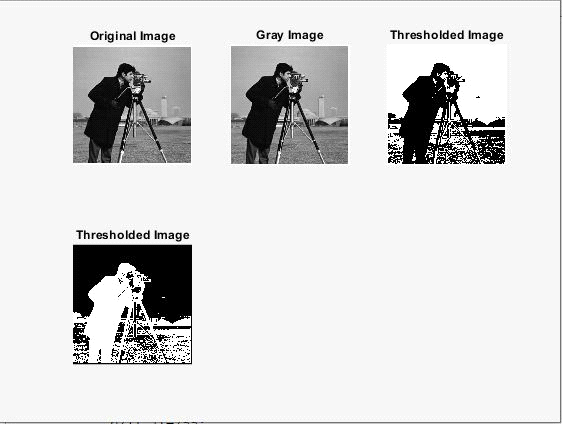
end

end

end

subplot(2,3,4),imshow(b2),title('Thresholded Image');

**Output:-**



% Image Segmentation using Thresholding Function

clc

clear all

a = imread('C:\Users\glau\Documents\radhika\dog.jpg'); %Gray Image

g = rgb2gray(a)

%figure, imshow(g),title('Gray Image');

th = 0.3;

thimg = im2bw(g,th); %Simple Thresholding Method 1

%figure, imshow(thimg),title('Simple Thresholded image');

%figure, imshow(g > 50),title('Simple Thresholded image-2'); %Method 2

%Optimal Threshholding

opt = graythresh(g);

optimg = im2bw(g,opt);

%figure,imshow(optimg),title('Optimal Threholded image');

%Multiple Threshholding

%mul = multithresh(0.3:0.6);

%mulImg = im2bw(g,mul);

%Adaptive Threshholding

%Here we compute mean and median with any user defined constant value

b = imread('C:\Users\glau\Documents\radhika\blurred.jpg');

b0 = rgb2gray(b);

b1 = medfilt2(b0,[20,20]);

ad = b1 + 2;

thresh = b0 - ad;

%imshow(thresh>0);

subplot(2,4,1),imshow(g),title('Gray image');

subplot(2,4,2),imshow(thimg),title('Simple Thresholded image');

subplot(2,4,3),imshow(g > 60),title('Simple Thresholded image-2');

subplot(2,4,4),imshow(optimg),title('Optimal Thresholded image');

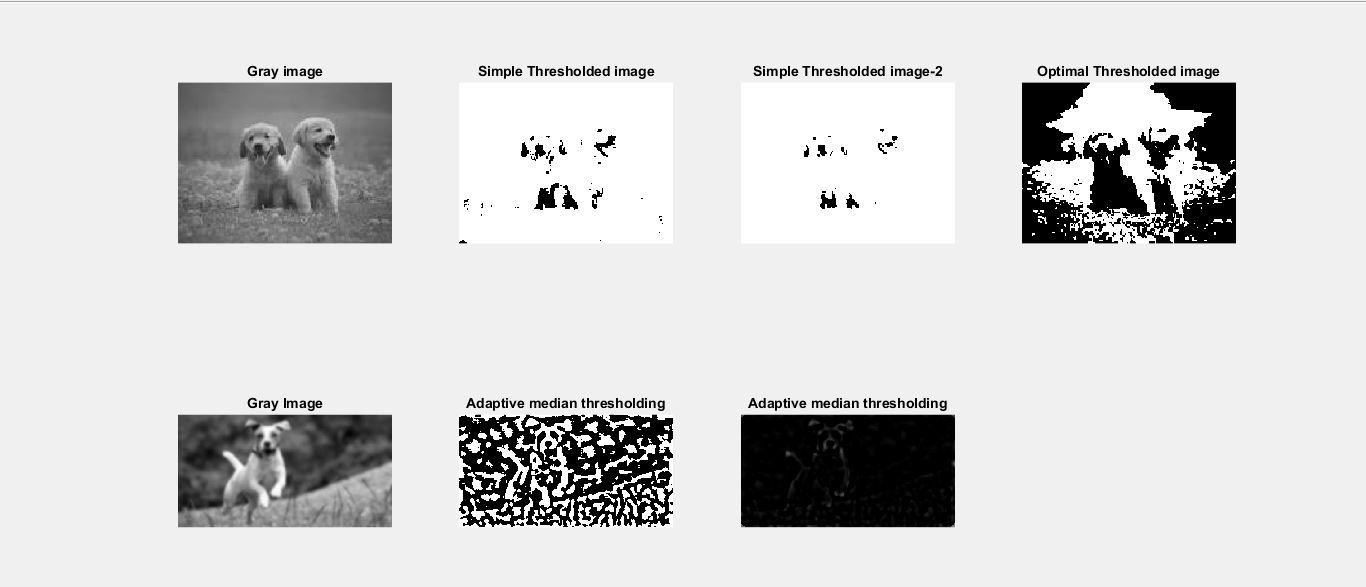
%subplot(2,3,5),imshow(mulImg), title('Multiple Threshholding');

subplot(2,4,5),imshow(b0),title('Gray Image');

subplot(2,4,6),imshow(thresh > 0),title('Adaptive median thresholding');

subplot(2,4,7),imshow(thresh),title('Adaptive median thresholding');

**OUTPUT:-**



Experiment 10:- Perform the various edge detection operations (ordinary, roberts, prewitts and sobel operators).

% Edge detection using **Ordinary** operators..

w1=[1 0; -1 0]

w2 = [1 -1; 0 0]

img = imread('C:\Users\GLAU\Pictures\boxes.jpg')  
img2=rgb2gray(img);  
[r c1]=size(img2);  
img2=double(img2);  
c=2;  
img3=img2;  
f=[1 0; -1 0]  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:2  
            for l=1:2  
                sum=sum+img2(i-2+k,j-2+l)\*f(k,l);  
            end  
        end  
        img3(i,j)=sum;  
    end  
end  
subplot(2,3,1),imshow(uint8(img)), title ('Original Image');  
subplot(2,3,2),imshow(uint8(img2)),title ('RGB2GRAY Image');  
subplot(2,3,3),imshow(uint8(img3)),title ('edge detected using ordinary filter');  
  
img4=img2;  
f=[1 -1; 0 0]  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:2  
            for l=1:2  
                sum=sum+img2(i-2+k,j-2+l)\*f(k,l);  
            end  
        end  
        img4(i,j)=sum;  
    end  
end  
subplot(2,3,4),imshow(uint8(img4)),title ('edge detected using ordinary filter 2');  
  
img5 = img3+img4;  
subplot(2,3,5),imshow(uint8(img5)),title ('Img3 + img4');

**OUTPUT:-**



% Edge detection using **Roberts** operators..

w3 = [1 0; 0 -1]

w4 = [0 1; -1 0]

img = imread('C:\Users\GLAU\Pictures\boxes.jpg')

img2=rgb2gray(img);

[r c1]=size(img2);

img2=double(img2);

c=2;

img3=img2;

f=[1 0; 0 -1]

for i=2:r-1

for j=2:c1-1

sum=0;

for k=1:2

for l=1:2

sum=sum+img2(i-2+k,j-2+l)\*f(k,l);

end

end

img3(i,j)=sum;

end

end

subplot(2,3,1),imshow(uint8(img)), title ('Original Image');

subplot(2,3,2),imshow(uint8(img2)),title ('RGB2GRAY Image');

subplot(2,3,3),imshow(uint8(img3)),title ('edge detected using Robert filter');

img4=img2;

f=[0 1; -1 0]

for i=2:r-1

for j=2:c1-1

sum=0;

for k=1:2

for l=1:2

sum=sum+img2(i-2+k,j-2+l)\*f(k,l);

end

end

img4(i,j)=sum;

end

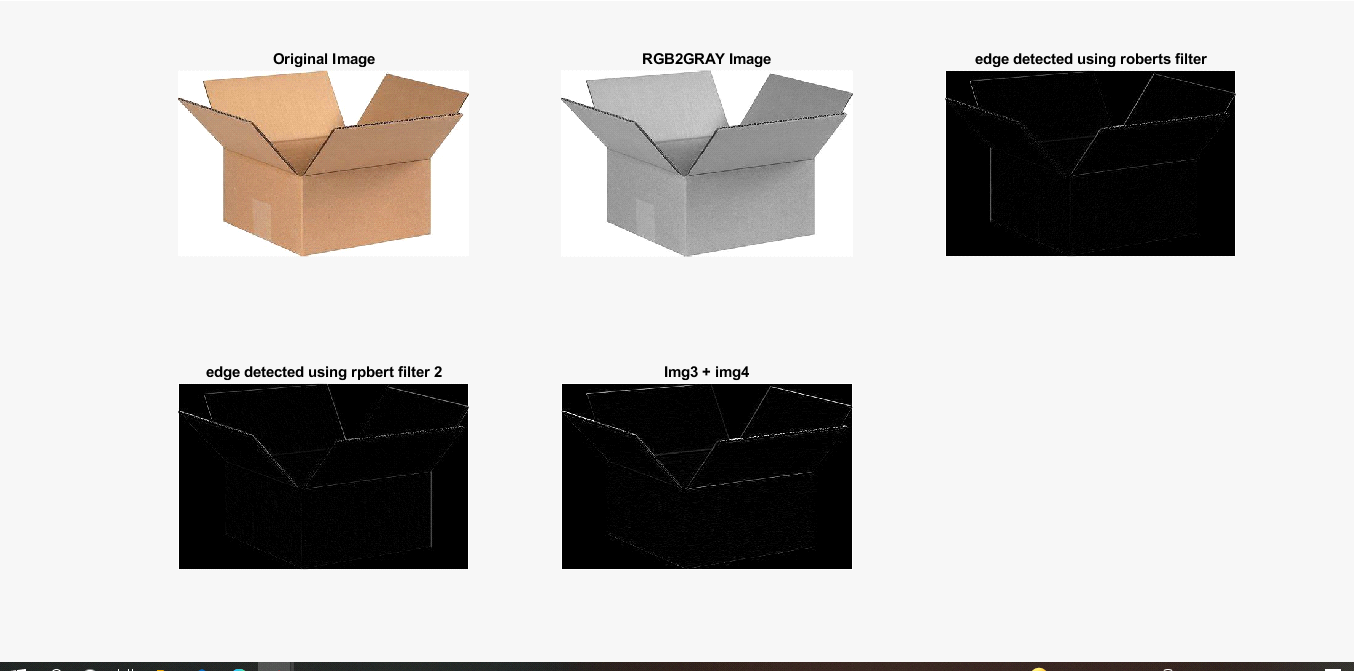
end

subplot(2,3,4),imshow(uint8(img4)),title ('edge detected using robert filter 2');

img5 = img3+img4;

subplot(2,3,5),imshow(uint8(img5)),title ('Img3 + img4');

**OUTPUT:-**



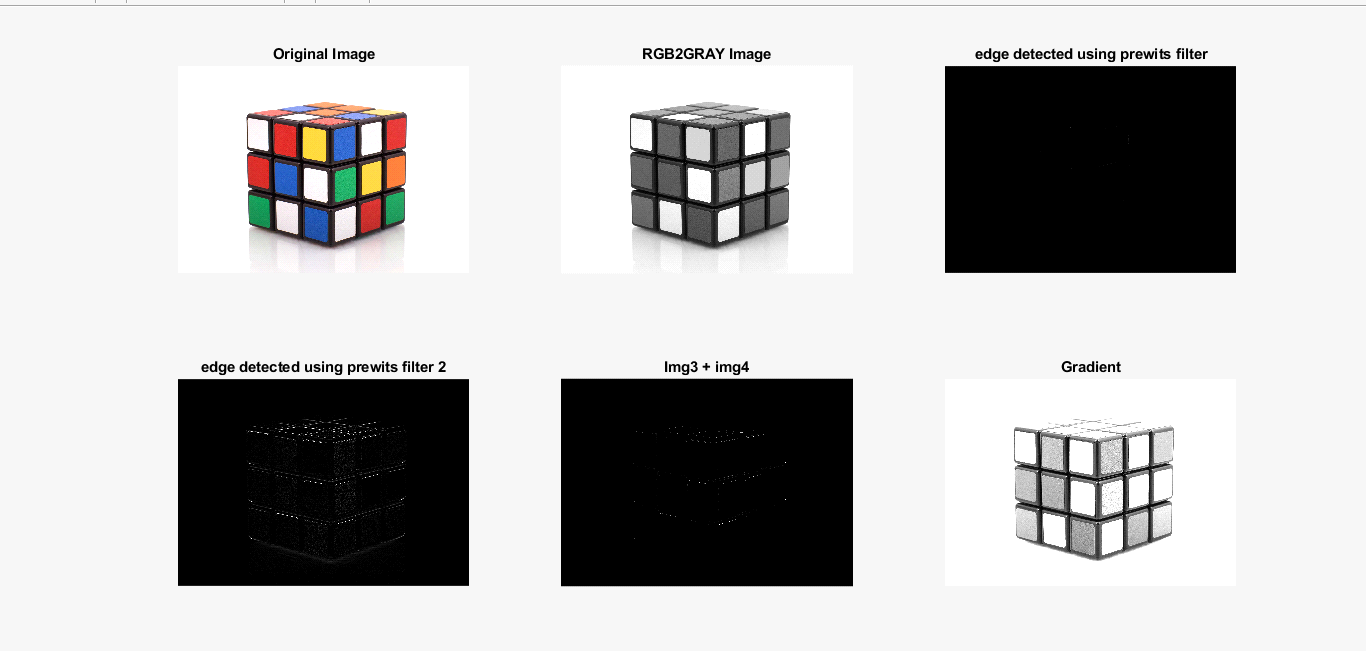
% Edge detection using **Prewitts** operators..

w5 = [-1 0 -1 ; -1 0 1; -1 0 1]

w6 = [-1 -1 -1; 0 0 0; 1 1 1]

img = imread('C:\Users\GLAU\Pictures\cube.jpg');  
img2=rgb2gray(img);  
[r c1]=size(img2);  
img2=double(img2);  
c=2;  
img3=img2;  
f=[-1 0 -1; -1 0 1; -1 0 1];  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:3  
            for l=1:3  
                sum=sum+img2(i-2+k,j-2+l)\*f(k,l);  
            end  
        end  
        img3(i,j)=sum;  
    end  
end  
subplot(2,3,1),imshow(uint8(img)), title ('Original Image');  
subplot(2,3,2),imshow(uint8(img2)),title ('RGB2GRAY Image');  
subplot(2,3,3),imshow(uint8(img3)),title ('edge detected using prewits filter');  
  
img4=img2;  
f2=[-1 -1 -1; 0 0 0; 1 1 1]  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:3  
            for l=1:3  
                sum=sum+img2(i-2+k,j-2+l)\*f2(k,l);  
            end  
        end  
        img4(i,j)=sum;  
    end  
end  
subplot(2,3,4),imshow(uint8(img4)),title ('edge detected using prewits filter 2');  
  
img5 = img3+img4;  
subplot(2,3,5),imshow(uint8(img5)),title ('Img3 + img4');  
  
img6 = sqrt(img3.^2+img4.^2);  
subplot(2,3,6),imshow(uint8(img6)),title ('Gradient');

**OUTPUT:-**



% Edge detection using **Sobel** operators..

w7 =  [-1 -2 -1; 0 0 0; 1 2 1]

w8 = [-1 0 1; -2 0 2; -1 0 1]

img = imread('C:\Users\GLAU\Pictures\boxes.jpg');  
img2=rgb2gray(img);  
[r c1]=size(img2);  
img2=double(img2);  
c=2;  
img3=img2;  
f=[-1 2 -1; 0 0 0; -1 2 1];  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:3  
            for l=1:3  
                sum=sum+img2(i-2+k,j-2+l)\*f(k,l);  
            end  
        end  
        img3(i,j)=sum;  
    end  
end  
subplot(2,3,1),imshow(uint8(img)), title ('Original Image');  
subplot(2,3,2),imshow(uint8(img2)),title ('RGB2GRAY Image');  
subplot(2,3,3),imshow(uint8(img3)),title ('edge detected using sobel filter');  
  
img4=img2;  
f2=[-1 0 1; -2 0 2; -1 0 1]  
for i=2:r-1  
    for j=2:c1-1  
        sum=0;  
        for k=1:3  
            for l=1:3  
                sum=sum+img2(i-2+k,j-2+l)\*f2(k,l);  
            end  
        end  
        img4(i,j)=sum;  
    end  
end  
subplot(2,3,4),imshow(uint8(img4)),title ('edge detected using sobel filter 2');  
  
img5 = img3+img4;  
subplot(2,3,5),imshow(uint8(img5)),title ('Img3 + img4');  
  
img6 = sqrt(img3.^2+img4.^2);  
subplot(2,3,6),imshow(uint8(img6)),title ('Gradient');

**OUTPUT:-**

