**Experiments**

1. Create command to familiarize with MATLAB & Create the matrices & perform the various operations on them.
2. Perform Various Arithmetic Operation (Image Addition & complement) & Logical Operation (NOT, OR and XOR) on images.
3. Perform various Histogram Operations histogram plot, histogram Equalization, Contrast Stretching & gamma correction on images & plot histogram without using imhist function.
4. Perform smoothing using linear and order statistics filters min, max & med of varying sizes and Sharpen an image using Laplacian filter.
5. Find the DFT of [0 1 2 1] without twiddle matrix.
6. Find the DFT of [0 1 2 1] using twiddle matrix.
7. Perform various Morphological operation dilation, erosion, internal & external boundary Extraction; Thinning, Thickening & Perform Dilation, erosion, boundary Extraction using direct function.
8. Perform various Morphological operation dilation, erosion, internal & external boundary Extraction, Thinning, Thickening & Perform Dilation, erosion, boundary Extraction without using direct function.
9. Perform the various Edge Detection Operators (Ordinary, Roberts, Prewitts & Sobel Operator).
10. Perform dilation, erosion, opening and closing operation on image and consider [1 1 1;1 1 1;1 1 1] as structural element

**1.Create command to familiarize with MATLAB & Create the matrices & perform the various operations on them.**

%Creating a new 2D array a=[1,2,3;4,5,6;7,8,9];

%size of matrix ‘a’ will return two values -> row in ‘r’ and column in ‘c’;

[r,c]=size(a);

%Creating a zero matrix a1=zeros(4,2);

%initialize two matrices for performing arithmetic operations

A = [10 20 30; 11 12 13; 40 50 60];

B = [51 52 53; 21 23 21; 44 54 64];

%addition of two matrix

addition =A+B;

%subtraction of two matrix

subtraction = A-B;

%multiplication of two matrix multiplication = A\*B;

%division of two matrix

division = A/B;

%finding length of 1D matrix

l = length(a);

% Find diagonal Elements in the matrix diagonal\_ele = diag(A);

%Find identity matrix

identity\_mat = eye(3,3);

%find sin value of matrix

sin\_value = sin(A);

% For Transpose of matrix

T=transpose(A);

% Using Sum function

S = sum(T);

%Inverse of a matrix

inverse\_mat = inv(A);

% For determinant of matrix detminant\_mat = det(A);

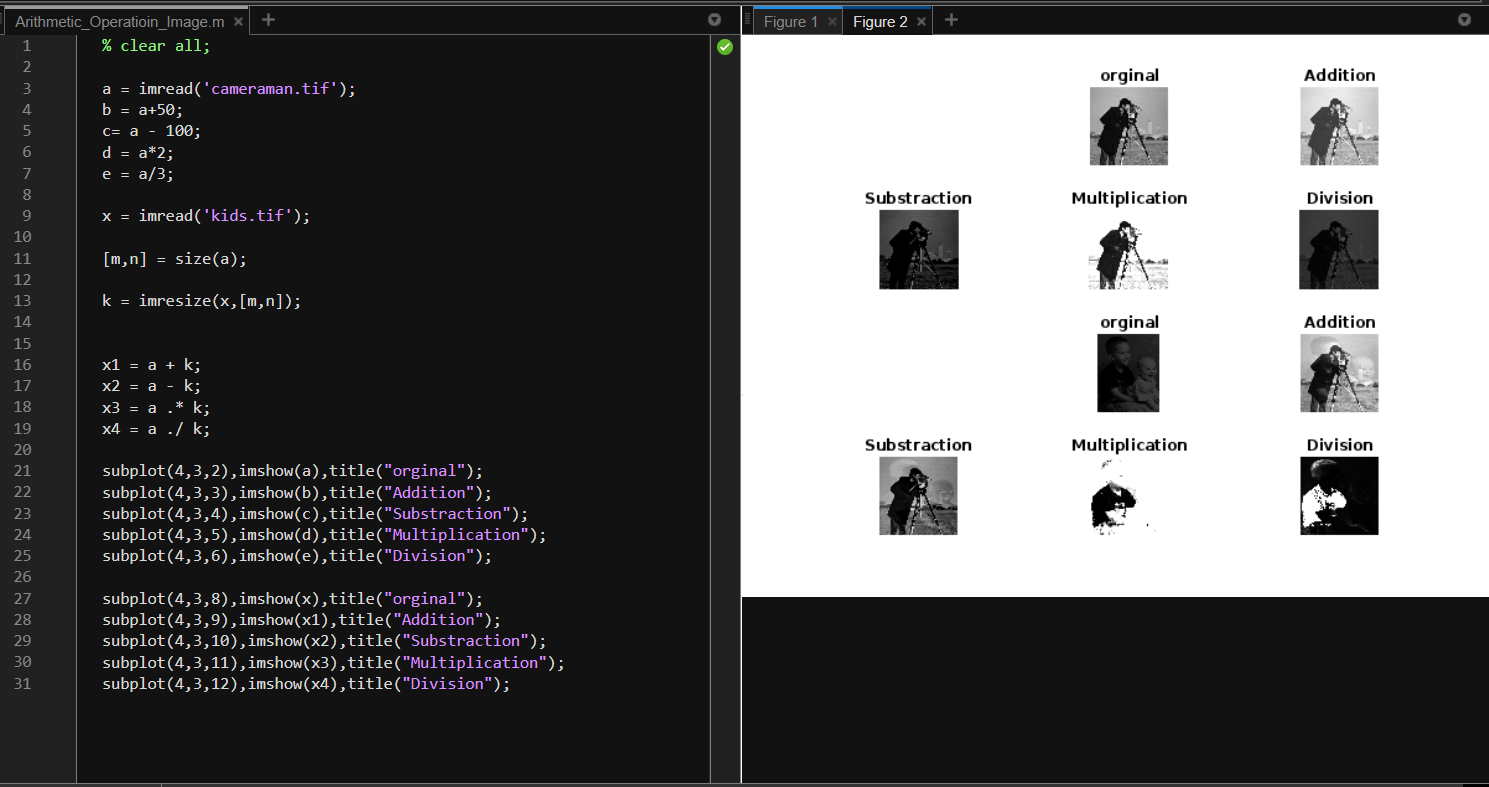
%Find the absolute value of matrix of complex elements

C = [2+3i 1+9i; 6+7i 9+i];

c1 = abs©;

%Matrix of random elements

R = rand(3,4)



**2. Perform Various Arithmetic Operation (Image Addition & Logical Operation (NOT, OR and XOR) on images.**

**% Perform Arithmetic Operations on Pixel Image Value**

% clear all;

a = imread('cameraman.tif');

b = a+50;

c= a - 100;

d = a\*2;

e = a/3;

x = imread('kids.tif');

[m,n] = size(a);

k = imresize(x,[m,n]);

x1 = a + k;

x2 = a - k;

x3 = a .\* k;

x4 = a ./ k;

subplot(4,3,2),imshow(a),title("orginal");

subplot(4,3,3),imshow(b),title("Addition");

subplot(4,3,4),imshow(c),title("Substraction");

subplot(4,3,5),imshow(d),title("Multiplication");

subplot(4,3,6),imshow(e),title("Division");

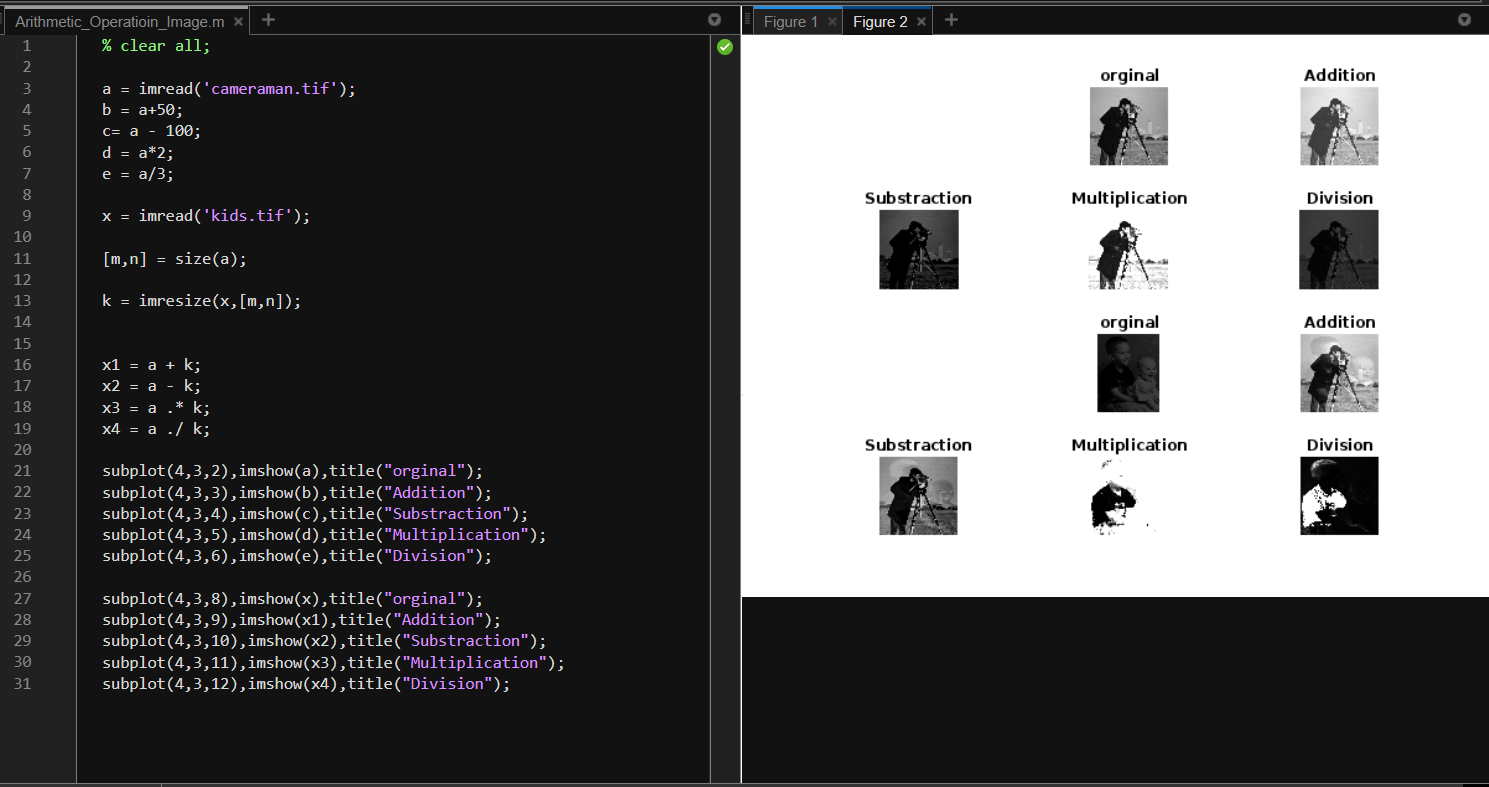
subplot(4,3,8),imshow(x),title("orginal");

subplot(4,3,9),imshow(x1),title("Addition");

subplot(4,3,10),imshow(x2),title("Substraction");

subplot(4,3,11),imshow(x3),title("Multiplication");

subplot(4,3,12),imshow(x4),title("Division");



% im2bw – with the help of this function we can convert into binary

clear all;

a=imread('kids.tif');

b=imread('cameraman.tif');

[m,n] = size(a);

b1 = imresize(b,[m,n]);

% c = rgb2gray(a);

% subplot()

% imshow(a);

% imshow(b);

first = im2bw(a);

second =im2bw(b1);

% binary = "0111100000111";

a1 = a&b1;

a2 = a|b1;

a3 = ~a;

a4 = ~b1;

% Logarithm operatioon

d1 = double(a);

c = (1\*log(1+d1))\*256;

s1=uint8(c);

% Power Law ........................

% output pixels = c\*inputpixels ^ gamma;

x = 2\*d1.^10;

x1 = uint8(x);

% b = double(a);

% b0 = mod(b,2);

% b1 = mod(floor(b/2),2);

% b2 = mod(floor(b/4),2);

subplot(3,4,2),imshow(a),title("First Image");

subplot(3,4,3),imshow(b),title("Second Image");

subplot(3,4,4),imshow(a1),title("And");

subplot(3,4,5),imshow(a2),title("OR");

subplot(3,4,6),imshow(a3),title("Not");

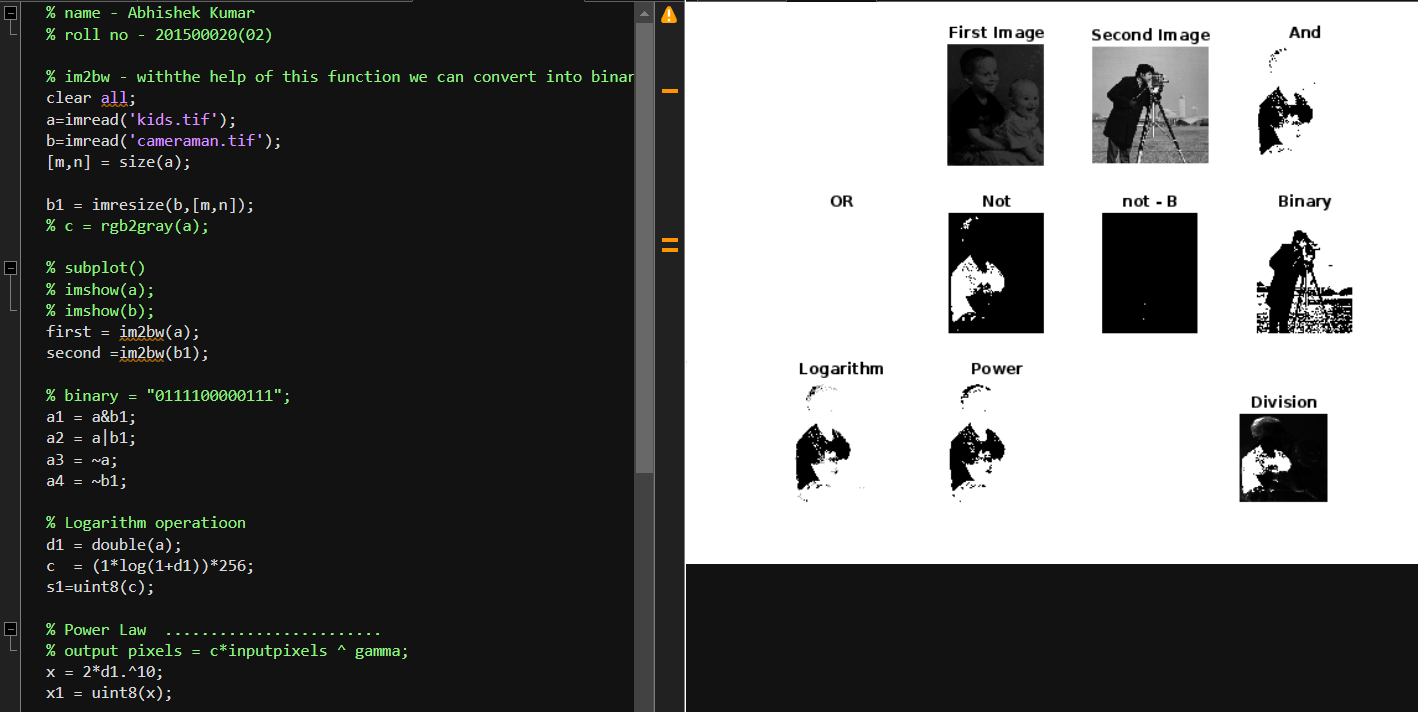
subplot(3,4,7),imshow(a4),title("not - B");

subplot(3,4,8),imshow(second),title("Binary");

subplot(3,4,9),imshow(s1),title("Logarithm");

subplot(3,4,10),imshow(x1),title("Power");

% rgb2gray it is used to change in gray scale



**3. Perform various Histogram Operations histogram plot, histogram Equalization, Contrast Stretching & gamma correction on images & plot histogram without using imhist function.**

a = imread('cameraman.tif');

% imshow(a);

[m,n] = size(a);

arr1 = zeros(256,1);

for i=1:m

for j=1:n

val = a(i,j);

arr1(val+1) = arr1(val+1) + 1;

end

end

bar(arr1);

b = histeq(a);

x = imadjust(a,[0.6,0.9],[0,1]);

% imhist(a);

% arr1;

% h = histogram(arr1);

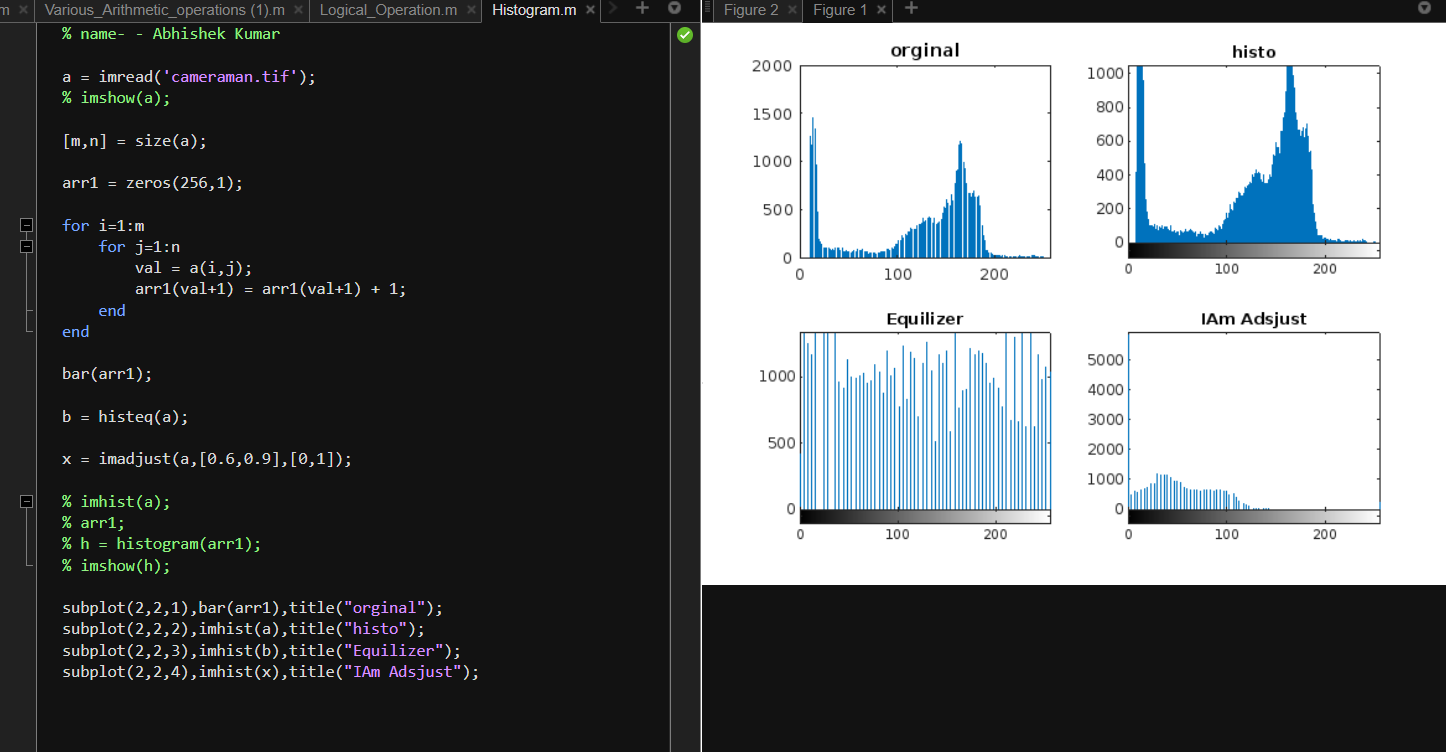
% imshow(h);

subplot(2,2,1),bar(arr1),title("orginal");

subplot(2,2,2),imhist(a),title("histo");

subplot(2,2,3),imhist(b),title("Equilizer");

subplot(2,2,4),imhist(x),title("IAm Adsjust");

****

**4. Perform smoothing using linear and order statistics filters min, max & med of varying sizes and Sharpen an image using Laplacian filter.**

**Perform smoothing using linear and order statistics filters of varying sizes.**

% Linear smoothing

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

size = input('size of filter (odd number): ');

f = ones(size);

c = (size + 1)/2;

i2 = i1;

for i = c:m-c+1

for j = c:n-c+1

sum = 0;

for k = 1:size

for l = 1:size

sum = sum + i1(i-c+k, j-c+l)\*f(k,l);

end

end

i2(i,j)= sum / (size ^2);

end

end

figure(1), subplot(1,2,1), imshow(uint8(i1);

figure(1), subplot(1,2,2), imshow(uint8(i2));

# Perform smoothing image using Max Min & Med filtering

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

size = input('size of filter (odd number): ');

f = ones(size);

c = (size + 1)/2;

maxi = i1;

mini = i1;

medi = i1;

for i = c:m-c+1

for j = c:n-c+1

maxi(i,j) = max(max(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

mini(i,j) = min(min(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

medi(i,j) = median(median(i1(i-c+1:i-c+size,j-c+1: j-c+size)));

end

end

figure(1), subplot(2,2,1), imshow(uint8(i1)), title('Original');

figure(1), subplot(2,2,2), imshow(uint8(maxi)), title('Max Filtered');

figure(1), subplot(2,2,3), imshow(uint8(mini)), title('Min Filtered');

figure(1), subplot(2,2,4), imshow(uint8(medi)), title('Median Filtered');

# Sharpen an image using Laplacian filter

%Laplacian

clear all

clc

i1 = imread('peppers.png');

i1 = rgb2gray(i1);

[m n] = size(i1);

i1 = double(i1);

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

s = i1;

for i = 2:m-1

for j = 2:n-1

sum = 0;

for k = 1:3

for l = 1:3

sum = sum + i1(i-2+k, j-2+l)\*f(k,l);

end

end

s(i,j) = sum;

end

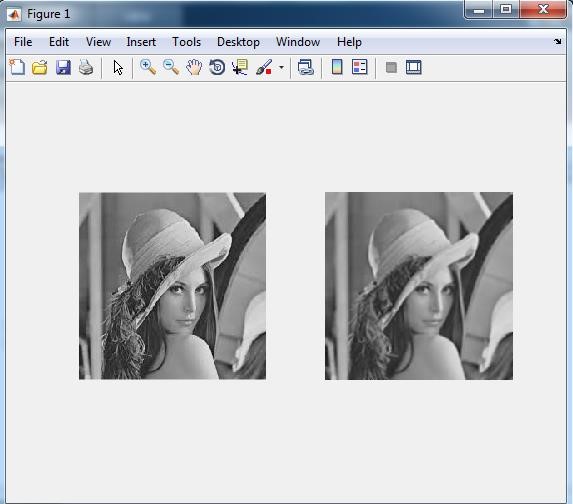
end

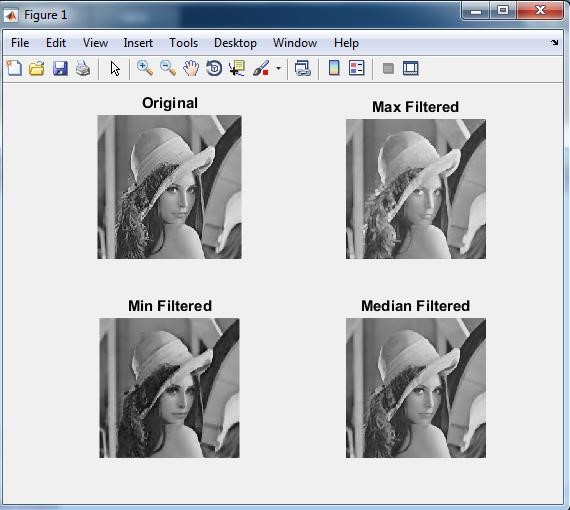
sm = i1 - s;

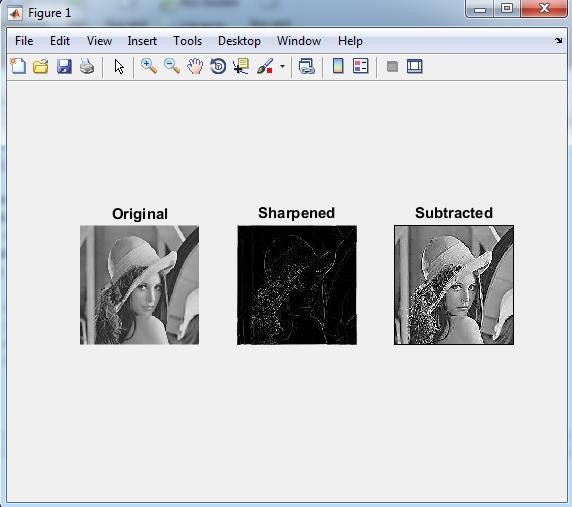
figure(1), subplot(1,3,1), imshow(uint8(i1)), title('Original');

figure(1), subplot(1,3,2), imshow(uint8(s)), title('Sharpened');

figure(1), subplot(1,3,3), imshow(uint8(sm)), title('Subtracted');







**5. Find the DFT of [0 1 2 1] without twiddle matrix**

% Data fourier transform without using twiddle

f(1:4) = [0 1 2 1];

F = zeros(1:4);

for u=1:4

for x = 1:4

F(u) = F(u) + f(x) \* (cos(2 \* pi \* (u - 1) \* (x-1) / 4) - 1 \* i \* sin(2 \* pi \* (u - 1) \* (x - 1) / 4));

end

end

F = 1/4 \* F;

imshow(F);

**6. Find the DFT of [0 1 2 1] twiddle matrix**

% Name - Abhishek Kumar

% Roll no - 201500020

% Section - A (02)

f = [0 1 2 1];

for u =0:3

for x = 0:3

val = exp(- i \* 2 \* pi \*u \* x/4);

t(u+1,x+1) = val;

end

end

F = 1/4 \* t \* transpose(f);

F;

**7. Perform various Morphological operation dilation, erosion, internal & external boundary Extraction; Thinning, Thickening & Perform Dilation, erosion, boundary Extraction using direct function**

clc;

close all;

clear all;

**% Read the test Image**

**% Convert the image to binary image**

myorigimg = imread('test.jpg');

myorigimg = im2bw(rgb2gray(myorigimg));

subplot(3, 3, 1);

imshow(myorigimg);title('Originalimage');

**% Create Structuring Element**

se = strel('disk', 9);

**% Perform dilation operation using imdilate command**

**% Display the dilated image**

mydilatedimg = imdilate(myorigimg, se);

subplot(3, 3, 2);

imshow(mydilatedimg);title('Dilated image');

**% Perform Erosion operation using imerode command**

**% Display the Eroded image**

myerodedimg = imerode(myorigimg, se);

subplot(3, 3, 3);

imshow(myerodedimg);title('Eroded image');

**% Find Internal Boundary**

**% Internal Boundary = Dilated Image AND Not of Eroded Image**

**% Display Internal Boundary**

internalboundimg = mydilatedimg & ~ myerodedimg;

subplot(3, 3, 4);

imshow(internalboundimg,[]);title('Internal Boundary');

**% Find External Boundary**

**% External Boundary = Dilated Image AND Not of Eroded Image**

**% Display External Boundary**

externalboundimg = mydilatedimg & ~myorigimg;

subplot(3, 3, 5);

imshow(externalboundimg,[]);title('External Boundary');

**% Find Morphological Gradient**

**% Morphological Gradient = Dilated Image AND Not of Eroded Image**

**% Display External Boundary**

mymorphgradimg = imsubtract(myorigimg,myerodedimg);

subplot(3, 3, 6);

imshow(mymorphgradimg,[]);title('Morphological Gradient');

**% Perform Thinning operation using bwmorph() command**

**% Display the dilated image**

thinf = bwmorph(myorigimg,'thin');

subplot(3,3,7);

imshow(thinf);title('Thinning of the Image');

**% Perform Thickening operation using bwmorph()command**

**% Display the dilated image**

thickf = bwmorph(myorigimg,'thicken');

subplot(3,3,8);

imshow(thickf);title('Thickening of the Image');

**% Perform Skeletonozation operation using bwmorph()command**

**% with 8 iterations and display the dilated image**

skelf100 = bwmorph(myorigimg,'skel',9);

subplot(3,3,9);

imshow(skelf100);title('Skeletonization - 9 iterations');

**8. Perform Dilation, erosion, boundary Extraction without using direct function**

clear all

clc

a=imread('test1.jpg');

p=size(a);

w=[1 1 1; 1 1 1; 1 1 1];

for x=2:1:p(1)-1

for y=2:1:p(2)-1

a1=[w(1)\*a(x-1,y-1) w(2)\*a(x-1,y) w(3)\*a(x-1,y+1) w(4)\*a(x,y-1) w(5)\*a(x,y) w(6)\*a(x,y+1) w(7)\*a(x+1,y-1) w(8)\*a(x+1,y) w(9)\*a(x+1,y+1)];

A(x,y)=min(a1);%Erosion

B(x,y)=max(a1);%dilation

Sharp(x,y)=a(x,y)-A(x,y);

end

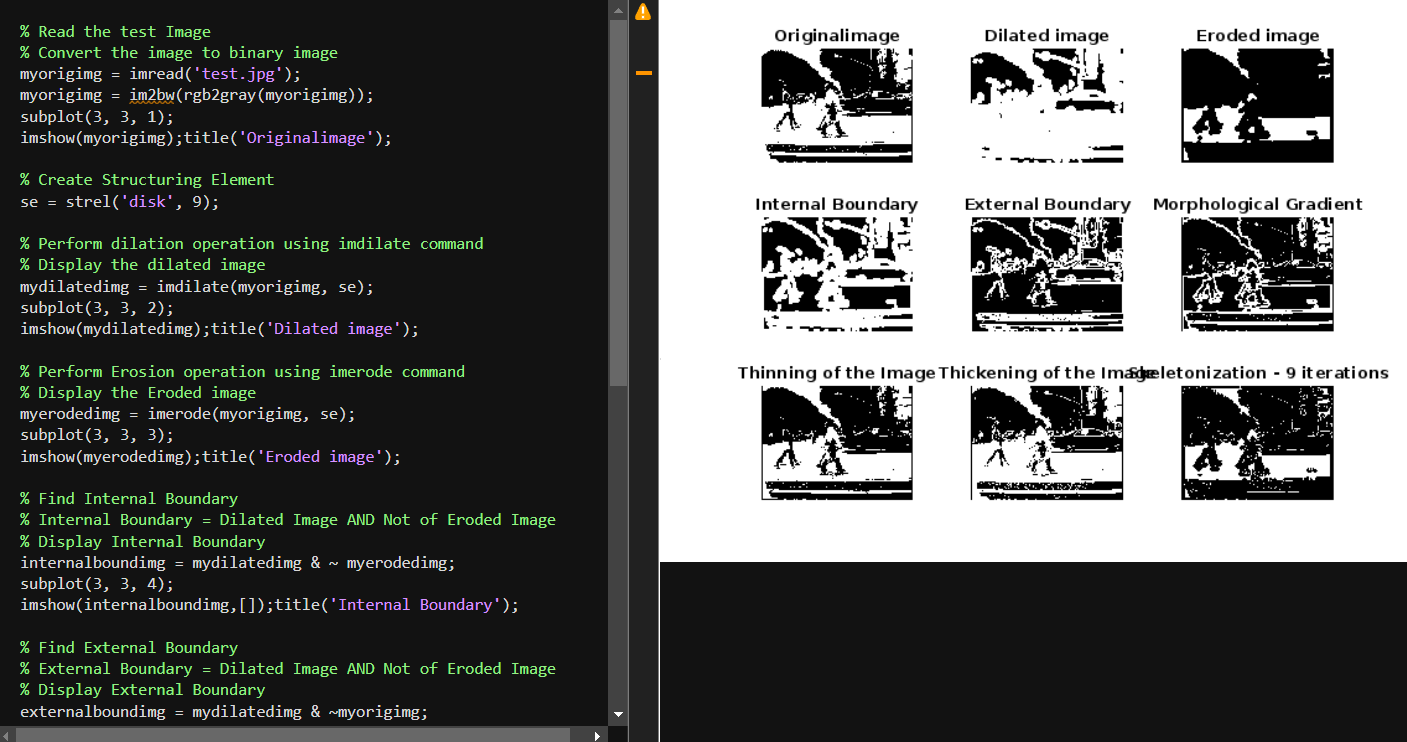
end

subplot(2,2,1),imshow(a),title('orignal image');

subplot(2,2,2),imshow(A),title('erosion');

subplot(2,2,3),imshow(B),title('Dilation');

subplot(2,2,4),imshow(Sharp),title('boundary extracted');



**9. Perform the various Edge Detection Operators (Ordinary, Roberts, Prewitts & Sobel Operator)**

clear all

clc

a=imread('shape456.png');

%a=double(aa);

[row col]=size(a);

**% ordinary operator**

w1=[1 0; -1 0];

w2=[1 -1; 0 0];

**% Roberts operator**

w3=[1 0; 0 -1];

w4=[0 1; -1 0];

**% prewitts operator**

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

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

**%sobel operator**

w7=[-1 -2 -1;0 0 0; 1 2 1];

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

for x=2:1:row-1

for y=2:1:col-1

a1(x,y)=w1(1)\*a(x,y)+w1(2)\*a(x,y+1)+w1(3)\*a(x+1,y)+w1(4)\*a(x+1,y+1);

a2(x,y)=w2(1)\*a(x,y)+w2(2)\*a(x,y+1)+w2(3)\*a(x+1,y)+w2(4)\*a(x+1,y+1);

a3(x,y)=w3(1)\*a(x,y)+w3(2)\*a(x,y+1)+w3(3)\*a(x+1,y)+w3(4)\*a(x+1,y+1);

a4(x,y)=w4(1)\*a(x,y)+w4(2)\*a(x,y+1)+w4(3)\*a(x+1,y)+w4(4)\*a(x+1,y+1);

a5(x,y)=w5(1)\*a(x-1,y-1)+w5(2)\*a(x-1,y)+w5(3)\*a(x-1,y+1)+w5(4)\*a(x,y-1)+w5(5)\*a(x,y)+w5(6)\*a(x,y+1)+w5(7)\*a(x+1,y-1)+w5(8)\*a(x+1,y)+w5(9)\*a(x+1,y+1);

a6(x,y)=w6(1)\*a(x-1,y-1)+w6(2)\*a(x-1,y)+w6(3)\*a(x-1,y+1)+w6(4)\*a(x,y-1)+w6(5)\*a(x,y)+w6(6)\*a(x,y+1)+w6(7)\*a(x+1,y-1)+w6(8)\*a(x+1,y)+w6(9)\*a(x+1,y+1);

a7(x,y)=w7(1)\*a(x-1,y-1)+w7(2)\*a(x-1,y)+w7(3)\*a(x-1,y+1)+w7(4)\*a(x,y-1)+w7(5)\*a(x,y)+w7(6)\*a(x,y+1)+w7(7)\*a(x+1,y-1)+w7(8)\*a(x+1,y)+w7(9)\*a(x+1,y+1);

a8(x,y)=w8(1)\*a(x-1,y-1)+w8(2)\*a(x-1,y)+w8(3)\*a(x-1,y+1)+w8(4)\*a(x,y-1)+w8(5)\*a(x,y)+w8(6)\*a(x,y+1)+w8(7)\*a(x+1,y-1)+w8(8)\*a(x+1,y)+w8(9)\*a(x+1,y+1);

end

end

A1=a1+a2;

A2=a3+a4;

A3=a5+a6;

A4=a7+a7;

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

figure(1),subplot(2,3,2),imshow(uint8(A1)),title('ordinary operator');

figure(1),subplot(2,3,3),imshow(uint8(A2)),title('Roberts Operator');

figure(1),subplot(2,3,4),imshow(uint8(A3)),title('Prewitts Operator');

figure(1),subplot(2,3,5),imshow(uint8(A4)),title('Sobel Operatos');

**10. Perform dilation, erosion, opening and closing operation on image and consider**

**[1 1 1;1 1 1;1 1 1] as structural element**

clear all;

% Importing the image

I = imread("cameraman.tif");

subplot(2, 3, 1),

imshow(I);

title("Original image");

% Dilated Image

se = strel("line", 7, 7);

dilate = imdilate(I, se);

subplot(2, 3, 2),

imshow(dilate);

title("Dilated image");

% Eroded image

erode = imerode(I, se);

subplot(2, 3, 3),

imshow(erode);

title("Eroded image");

% Opened image

open = imopen(I, se);

subplot(2, 3, 4),

imshow(open);

title("Opened image");

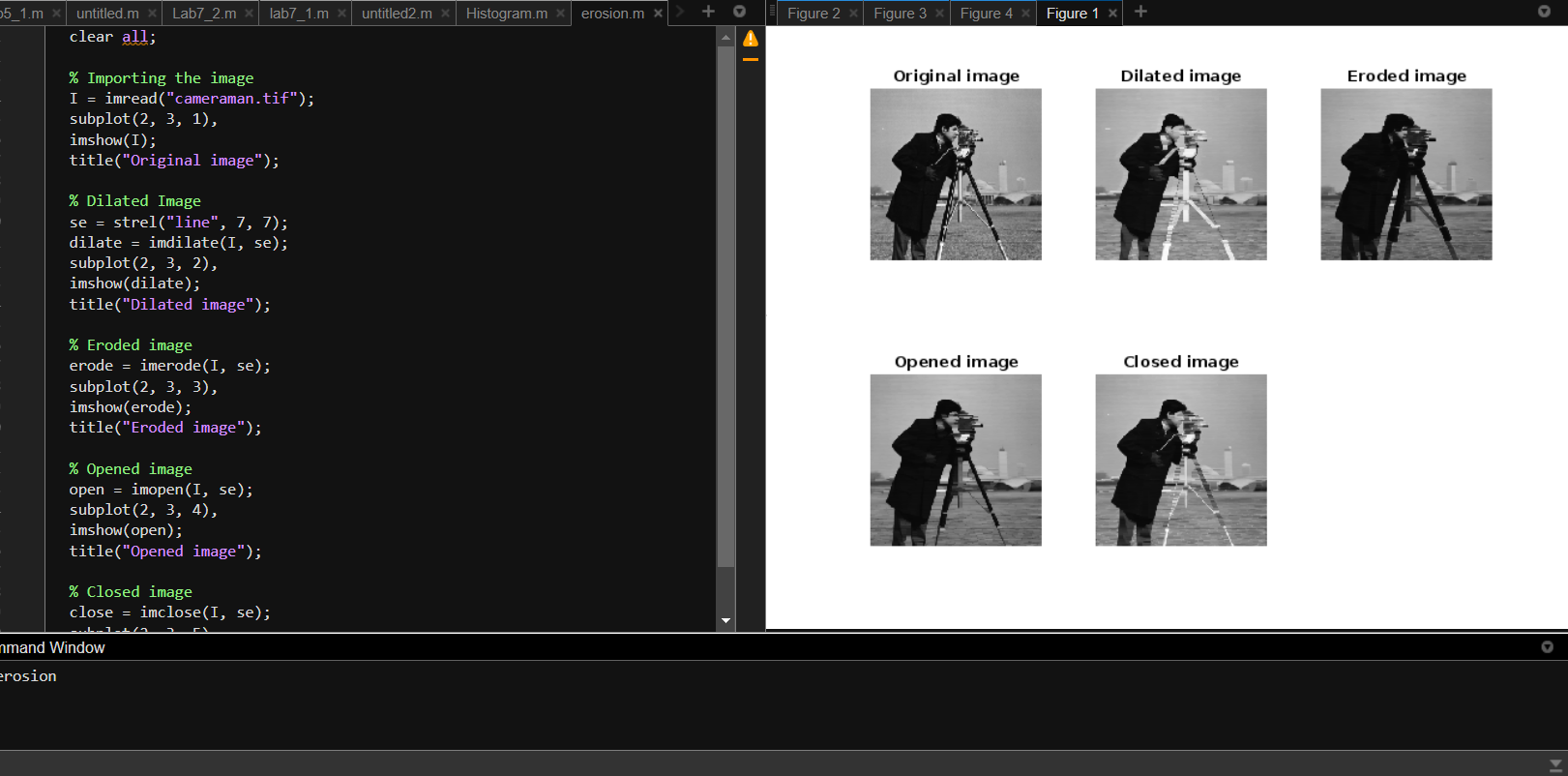
% Closed image

close = imclose(I, se);

subplot(2, 3, 5),

imshow(close);

title("Closed image");

****