**18CSE469J IMAGE PROCESSING AND PATTERN RECOGNITION**

**SEMESTER VI**

**CLASS :** B.Tech. [U.G]

**YEAR / SEM. :** III Year / VI Semester CSBS

**SOFTWARE REQUIREMENT : SCI LAB OR MATLAB**

****

**DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under SECTION 3 of the UGC Act, 1956)

S.R.M NAGAR, KATTANKULATHUR – 603203

KANCHEEPURAM DISTRICT

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp no.** | **Experiment name** | **Page no.** | **Date of experiment** | **Faculty sign** |
| **1** | Read, access, and display digital image | 3 | 11/1/2022 |  |
| **2** | Sampling | 5 | 28/1/2022 |  |
| **3** | Neighbourhood metrics | 7 | 4/2/2022 |  |
| **4** | Image enhancement | 10 | 11/2/2022 |  |
| **5** | Histogram | 12 | 18/2/2022 |  |
| **6** | Smoothing | 15 | 4/3/2022 |  |
| **7** | Thresholding | 18 | 11/3/2022 |  |
| **8** | Edge detection | 21 | 25/3/2022 |  |
| **9** | Hough transform | 24 | 1/4/2022 |  |
| **10** | Features extraction | 26 | 8/4/2022 |  |
| **11** | Dilation and Erosion Operators | 27 | 22/4/2022 |  |

|  |  |
| --- | --- |
| **Ex.No:** 1 | Read, access, and display digital image |
| **Date: 11/1/2022** |

**Aim:**

To read , access and display digital image

**Program 1:**

img= imread ('D:\flower.jpg');

figure;

title('Original image')

imshow(img);

img1=rgb2gray(a);

figure;

title('Grayscle image')

imshow(img1);

**Output 1:**

****



**Result:** Thus the program to read, access and displaying the images are executed successfully.

|  |  |
| --- | --- |
| **Ex.No:** 2 | Sampling |
| **Date: 28/1/2022** |

**Aim:**

To sample the spatial resolution of digitized image.

**Program 1:**

clc;

clear all;

close all;

n=8;

img = rgb2gray(imread('D:\flower.jpg'));

a=size(img);

w=a(2);

h=a(1);

im=zeros(100);

for i=1:n:h

for j=1:n:w

for k=0:n-1

for l=0:n-1

im(i+k,j+l)=img(i,j);

end

end

end

end

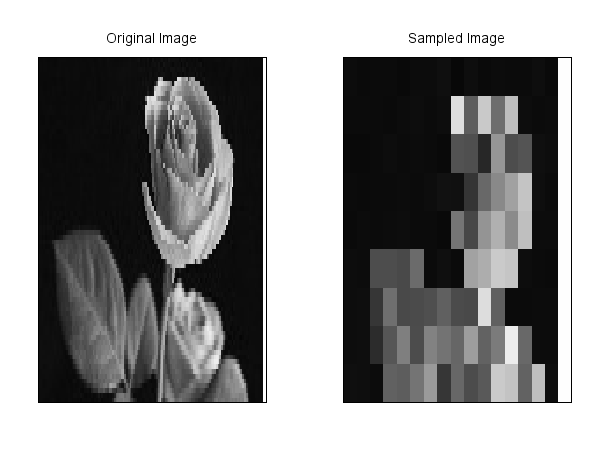
subplot(1,2,1);

imshow(uint8(img));title('Original Image');

subplot(1,2,2);

imshow(uint8(im));title('Sampled Image');

**Output :**



**Result:**

Thus the program for sampling has been executed successfully

|  |  |
| --- | --- |
| **Ex.No:** 3a | Neighbourhood metrics |
| **Date: 4/2/2022** |

**Aim:**

1. Neighborhood Operations - To learn about neighborhood operations and use them for i) Linear filtering ii) Non-linear filtering
2. Neighborhood Operations –To study the effect of the size of neighborhood on the result of processing

**Program 1:**

I = rgb2gray(imread('D:\flower.jpg'));

I\_noise=imnoise(I,'salt & pepper');

subplot(2,3,1);

title('original image')

imshow(I)

subplot(2,3,2);

title('noisy image')

imshow(I\_noise)

flinear1=1/25\*ones(5,5);

Ilinear1=imfilter(I\_noise,flinear1);

subplot(2,3,3);

title('Linear average filtered ')

imshow(Ilinear1)

hsize=[5,5];

sigma=1;

flinear2=fspecial('gaussian',hsize,sigma );

Ilinear2=imfilter(I\_noise,flinear2);

subplot(2,3,4);

title('Linear Gaussian Filtered')

imshow(Ilinear2)

fnonlinear=[3,3];

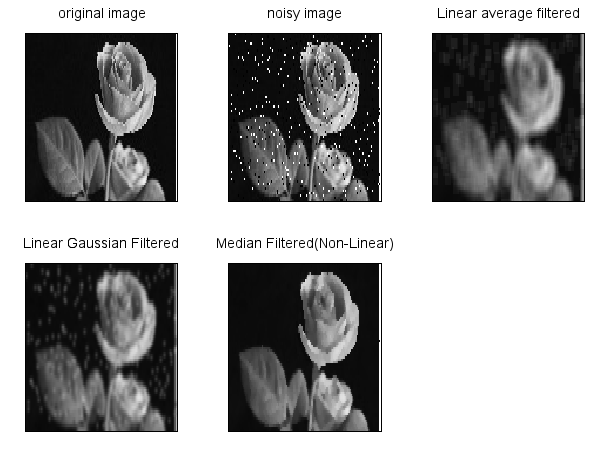
Inonlinear=immedian(I\_noise,fnonlinear);

subplot(2,3,5);

title('Median Filtered(Non-Linear)')

imshow(Inonlinear)

**Output 1:**



**Program 2:**

clc;

clear all;

close all;

I = rgb2gray(imread('D:\flower.jpg'));

I\_noise=imnoise(I,'salt & pepper');

FilterSize = [3 3];

I\_3x3 = immedian ( I\_noise , FilterSize ) ;

I\_5x5 = immedian ( I\_noise ,[5 5]) ;

I\_7x7 = immedian ( I\_noise ,[7 7]) ;

I\_9x9 = immedian ( I\_noise ,[9 9]) ;

subplot(2,3,1);

title('original image')

imshow(I)

subplot(2,3,2);

title('noisy image')

imshow(I\_noise)

subplot(2,3,3);

title('Filter size 3x3')

imshow(I\_3x3)

subplot(2,3,4);

title('Filter size 5x5')

imshow(I\_5x5)

subplot(2,3,5);

title('Filter size 7x7')

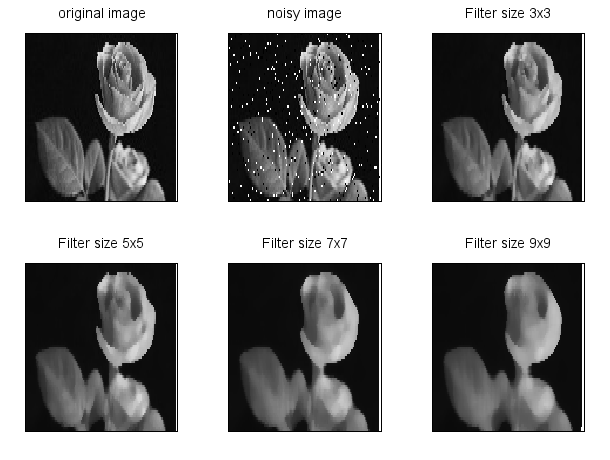
imshow(I\_7x7)

subplot(2,3,6);

title('Filter size 9x9')

imshow(I\_9x9)

**Output 2:**

****

**Result:** Thus the neighborhood operation on the pixels has been executed successfully

|  |  |
| --- | --- |
| **Ex.No:** 4 | Image Enhancement |
| **Date: 11/1/2022** |

**Aim:**

To learn image enhancement through Image negative, Gamma Transformation and Contrast Enhancement.

**Program:**

clc ;

clear;

close;

img= imread ('D:\cameraman.jpg');

img=rgb2gray(img);

I =im2double(img) ;

J = imcomplement(I); // Image Negative

subplot(2,3,1);

title('Original Image');

imshow(img);

subplot(2,3,2);

title('Image Negative');

imshow(J);

gamma=1.5

k=I.^gamma; // Gamma Transformation

subplot(2,3,3);

title('Gamma transformation');

imshow(k);

contrast1=1./(1+(0.2./(I+%eps)).^4); // Contrast Enhancement

contrast2=1./(1+(0.5./(I+%eps)).^5);

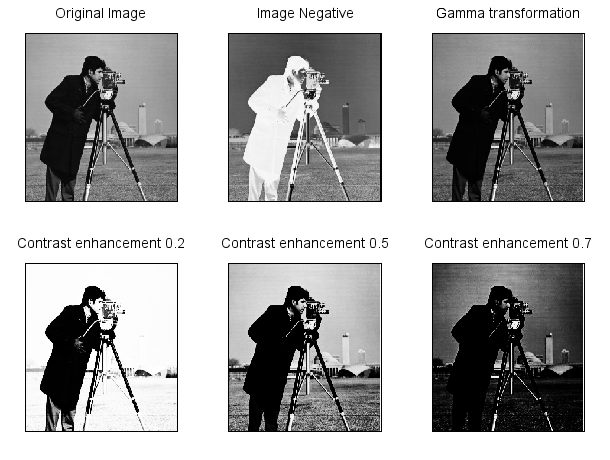
contrast3=1./(1+(0.7./(I+%eps)).^10);

subplot(2,3,4),imshow(contrast1);title('Contrast enhancement 0.2');

subplot(2,3,5),imshow(contrast2);title('Contrast enhancement 0.5');

subplot(2,3,6),imshow(contrast3);title('Contrast enhancement 0.7');

**Output:**

****

**Result:** Thus the image enhancement with different methods has been executed successfully.

|  |  |
| --- | --- |
| **Ex.No:** 5 | Histogram |
| **Date: 18/2/2022** |

**Aim:**

1. To understand how frequency distribution can be used to represent an image.
2. To study the correlation between the visual quality of an image with its histogram.

**Program 1:**

clc ;

clear;

close;

img= imread ('D:\cameraman.jpg');

img=rgb2gray(img);

[count,cells ]= imhist (img); // compute histogram

subplot(2,2,1);

title('Original image');

imshow(img);

subplot(2,2,2);

plot2d3 ('gnn' , cells , count )

title('Histogram plot for original image');

Iheq = imhistequal(img);

[count,cells ]= imhist (Iheq); // compute histogram equalization

subplot(2,2,3);

title('Histogram Equalized image');

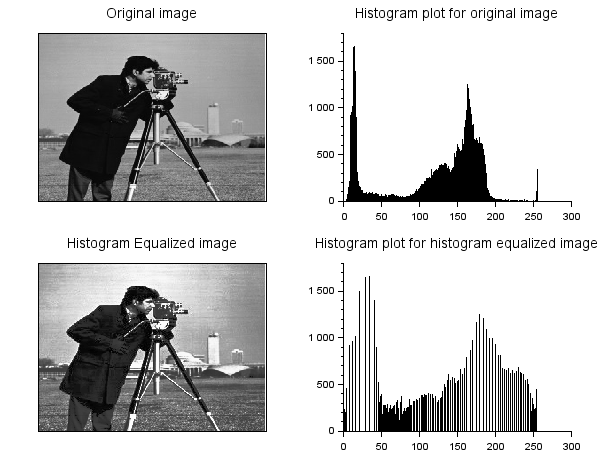
imshow(Iheq);

subplot(2,2,4);

plot2d3 ('gnn' , cells , count )

title('Histogram plot for histogram equalized image');

**Output 1:**



**Program 2:**

clc ;

clear;

close;

img= imread ('D:\cameraman.jpg');

img=rgb2gray(img);

//I = imresize (img ,[256 ,256]) ;

[ count , cells ]= imhist (img) ;

Iheq = imhistequal(img);

[count1,cells1 ]= imhist (Iheq);

// correlation between original image and Histogram equalized image

corrbsameimg = corr2(img,Iheq)

disp(corrbsameimg);

// correlation between the histograms of original image

x = xcorr ( count , count ) ;

//correlation between the histogram of original image and equalized image

x1 = xcorr ( count , count1 ) ;

subplot(2,1,1);

plot2d3 ( 'gnn' ,1: length ( x ) ,x ,2);

title('correlation b/w histograms of original image');

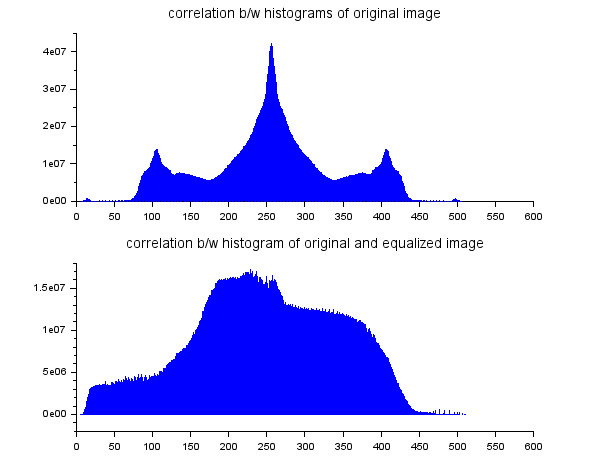
subplot(2,1,2);

plot2d3 ('gnn' ,1: length ( x1 ) ,x1 ,2);

title('correlation b/w histogram of original and equalized image')

**Output 2:**

Corrbsameimg = 0.9390109



**Result:** Thus the frequency distribution and correlation between the images using histogram has been executed successfully.

|  |  |
| --- | --- |
| **Ex.No:** 6 | Smoothing |
| **Date: 4/3/2022** |

**Aim:** To smoothen an image using median filter

**Program:**

clc ;

close ;

a= imread( 'D:\flower.jpeg' );

subplot(2,3,1)

title('Original image');

imshow(a)

b= imresize(a ,[256 ,256]) ;

b= imnoise (b,'salt & pepper',0.1);

subplot(2,3,2)

title('noise image');

imshow(b)

[m n]= size (b);

R=b(: ,: ,1);

G=b(: ,: ,2);

B=b(: ,: ,3);

exec('D:\scilab-6.1.1\experiments\Func\_medianall.sci')

Out\_R = Func\_medianall(R,N);

Out\_G = Func\_medianall(G,N);

Out\_B = Func\_medianall(B,N);

Out\_Image (: ,: ,1)= Out\_R ;

Out\_Image (: ,: ,2)= Out\_G ;

Out\_Image (: ,: ,3)= Out\_B ;

b = uint8 (b);

Out\_Image = uint8(Out\_Image);

subplot(2,3,3)

title( '3x3 median filtered' )

imshow(Out\_Image)

**Function func\_medianall**

function [Out\_Imag]=Func\_medianall(a, N)

a= double (a);

[m n]= size (a);

Out\_Imag =a;

if( modulo (N ,2) ==1)

Start =(N +1) /2;

End = Start ;

else

Start =N /2;

End = Start +1;

end

if( modulo (N ,2) ==1)

limit1 =(N -1) /2;

limit2 = limit1 ;

else

limit1 =(N/2) -1;

limit2 = limit1 +1;

end

for i= Start :(m-End +1) ,

for j= Start :(n-End +1) ,

I =1;

for k=- limit1 :limit2 ,

for l=- limit1 :limit2 ,

mat (I)=a(i+k,j+l);

I=I+1;

end

end

mat = gsort ( mat ); *// So r t the e l eme n t s to*

if( modulo (N ,2) ==1)

Out\_Imag (i,j)=( mat ((( N ^2) +1) /2) );

else

Out\_Imag (i,j)=( mat ((N ^2) /2)+ mat ((( N^2)/2)+1) ) /2;

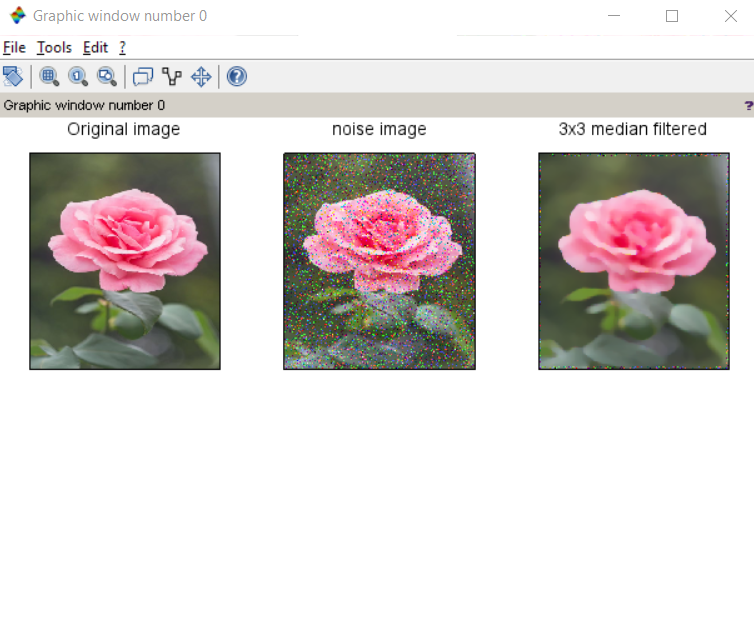
end

end

end

endfunction

**Output:**



**Result:** Image was smoothened using median filter

|  |  |
| --- | --- |
| **Ex.No:** 7 | Thresholding |
| **Date: 11/3/2022** |

**Aim:** To find the threshold of a image and segment it

**Program:**

RGB = imread ("D:\DSC\teaset.jfif");

Image = rgb2gray(RGB);

InvertedImage = uint8(255 \* ones(size(Image,1), size(Image,2))) - Image;

Histogram=imhist(InvertedImage);

figure();plot(0:255, Histogram')

xgrid(color('black'),1,8)

LogicalImage = im2bw(InvertedImage, 100/255);

f1=scf(1);f1.name='Original Image';

imshow(Image);

f2=scf(2);f2.name='Inverted Image';

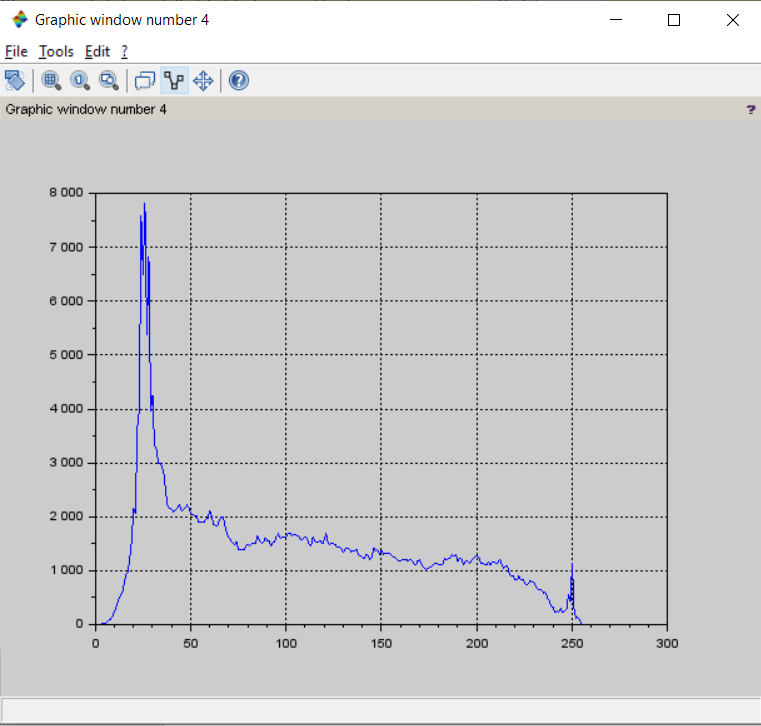
imshow(InvertedImage);

f3=scf(3);f3.name='Result of Thresholding';

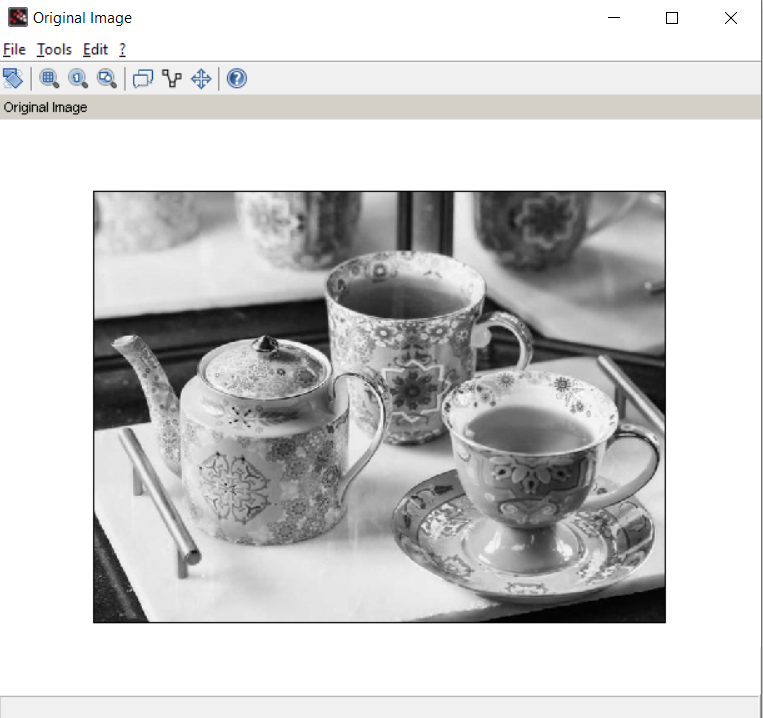
imshow(LogicalImage);

**After finding the threshold value from the histogram, the image is inverted then segmented according to the value found.**

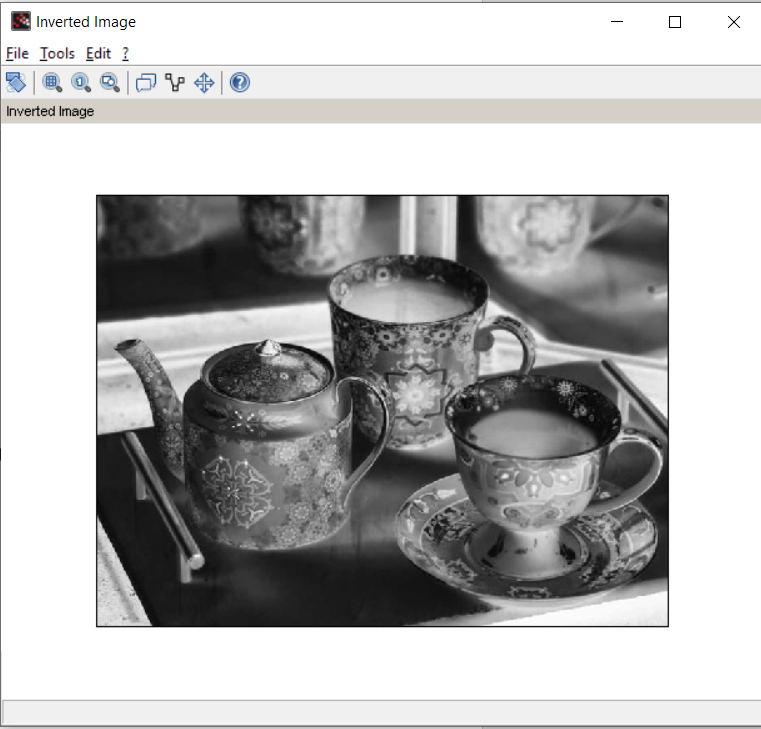
**Output:**



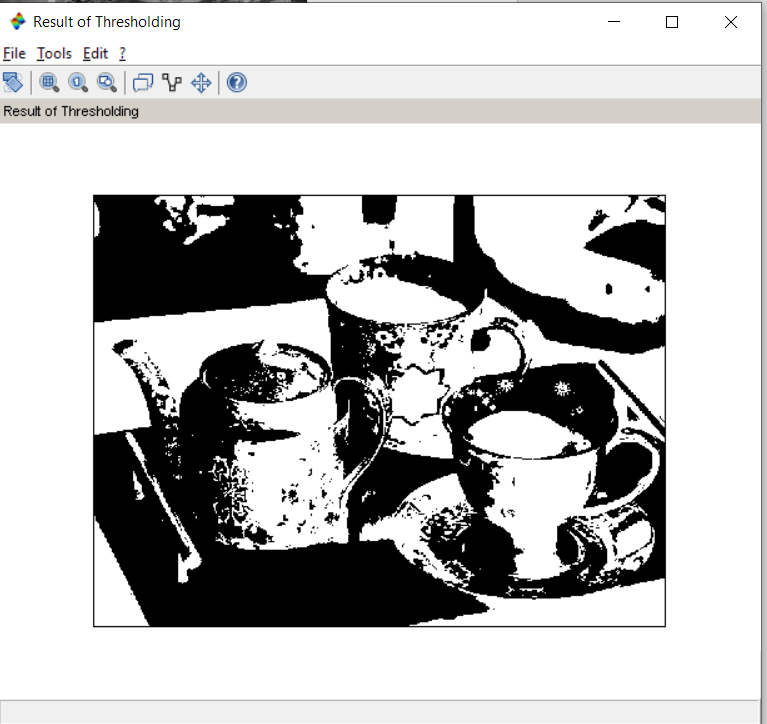
**Original image:**



**Inverted image:**



**Segmented image:**



**Result:** Image was segmented according to threshold value.

|  |  |
| --- | --- |
| **Ex.No:** 8 | Edge detection |
| **Date: 25/3/2022** |

**Aim:** To apply edge detection on an image for segmentation.

**Program:**

close ;

clc ;

a = imread('D:\ImageProcessing\sunset.jfif');

a = rgb2gray (a);

c = edge (a, 'sobel' );

d = edge (a, 'prewitt');

e = edge (a, 'log' );

f = edge (a, 'canny' );

imshow(a)

title ('Original Image' )

figure

imshow(c)

title ( 'Sobel' )

figure

imshow(d)

title ( 'Prewitt' )

figure

imshow(e)

title ( ' Log ' )

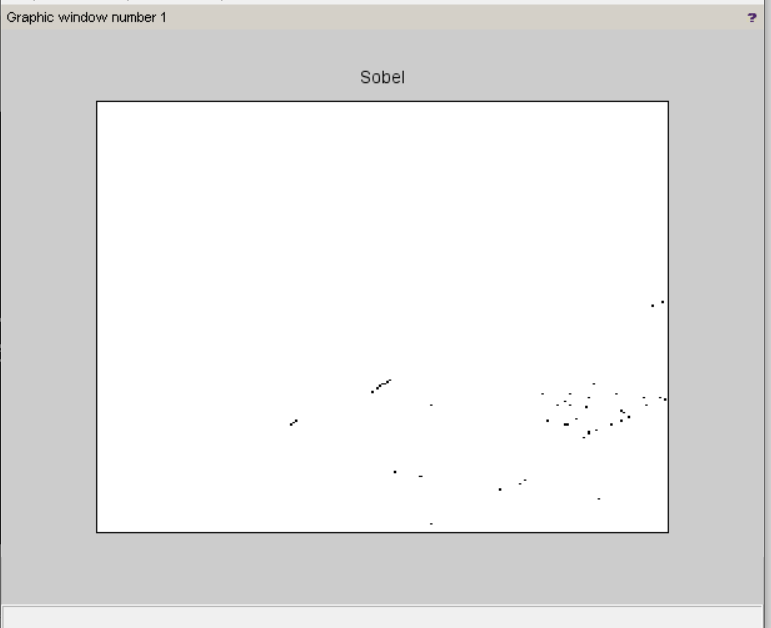
figure

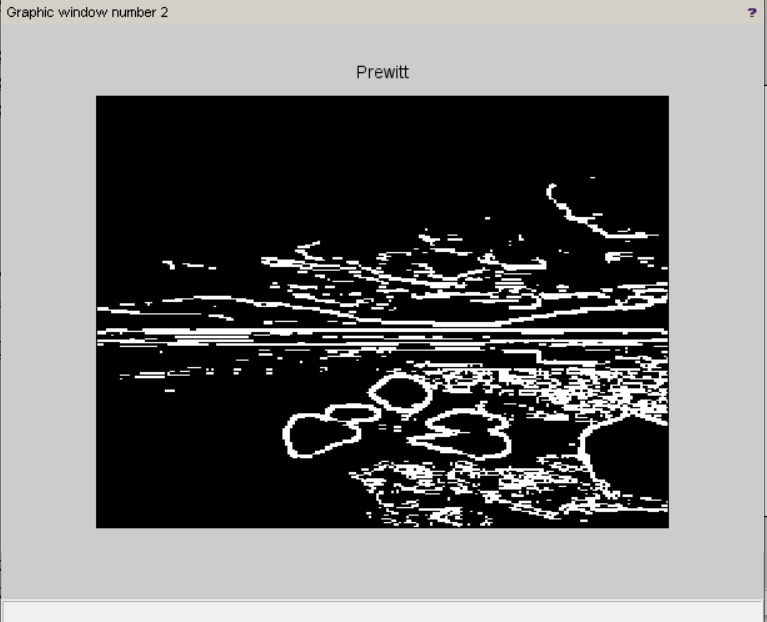
imshow(f)

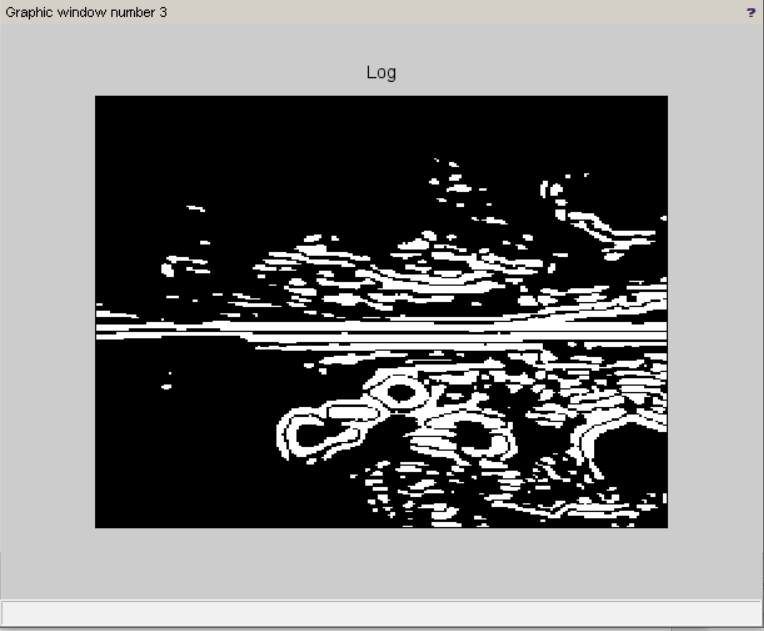
title ( 'Canny ' )

**Output:**









**Result:** Edge detection has been performed on the given image for segmentation

|  |  |
| --- | --- |
| **Ex.No:** 9 | Hough transform |
| **Date: 1/4/2022** |

**Aim:** To perform Hough transform on a given image

The image is first converted into binary form and then put into Hough transform, and the region with the highest frequency is marked on the graph.

**Program:**

I = imread('C:\SRM\SEM6\Image Processing\LabExercises\image6.png');

imshow(I)

title('original image')

figure

I =im2bw(double(I),0.5);

[y,x]=find(I);

[sy,sx]=size(I);

imshow(I);

title('binary image')

figure

totalpix = length(x);

HM = zeros(sy,sx);

R = 34;

R2 = R^2;

b = 1:sy;

for cnt = 1:totalpix

a = (round(x(cnt) - sqrt(R2 - (y(cnt) - [1:sy]).^2)));

for cnt2 =1:sy

if isreal(a(cnt2),0) & real(a(cnt2))>0

HM(cnt2,real(a(cnt2))) = HM(cnt2,real(a(cnt2))) + 1;

end

end

end

[maxval, maxind] = max(max(HM));

[B,A] = find(HM==maxval);

imshow(double(I));

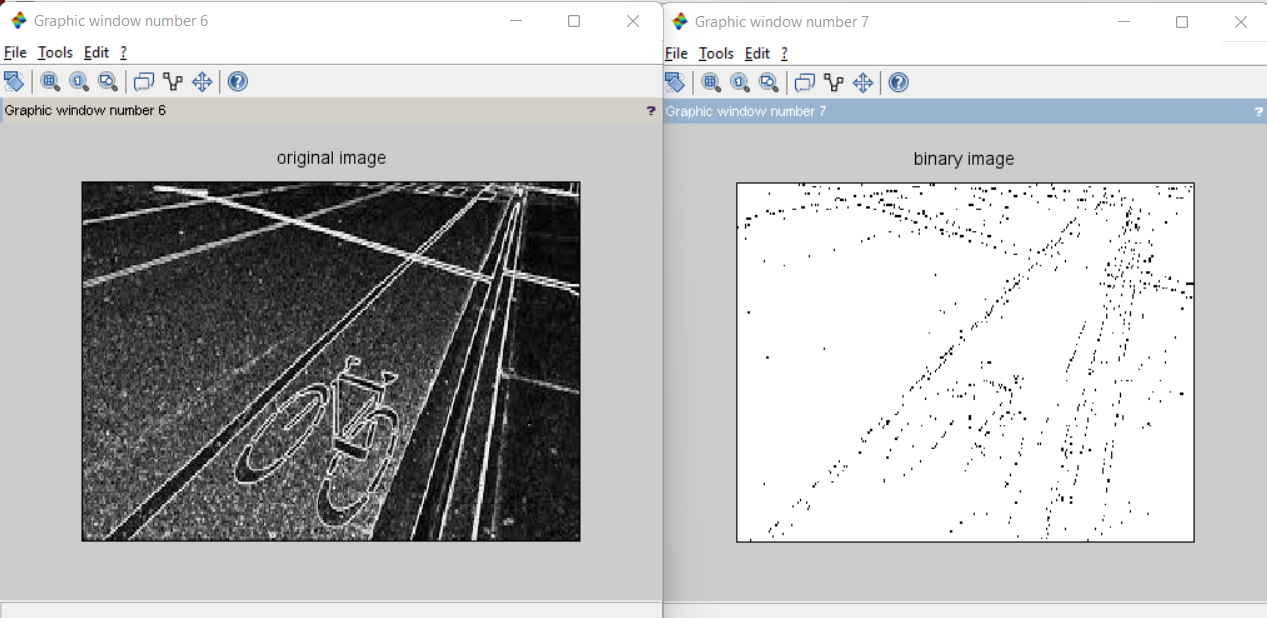
title('Hough transform')

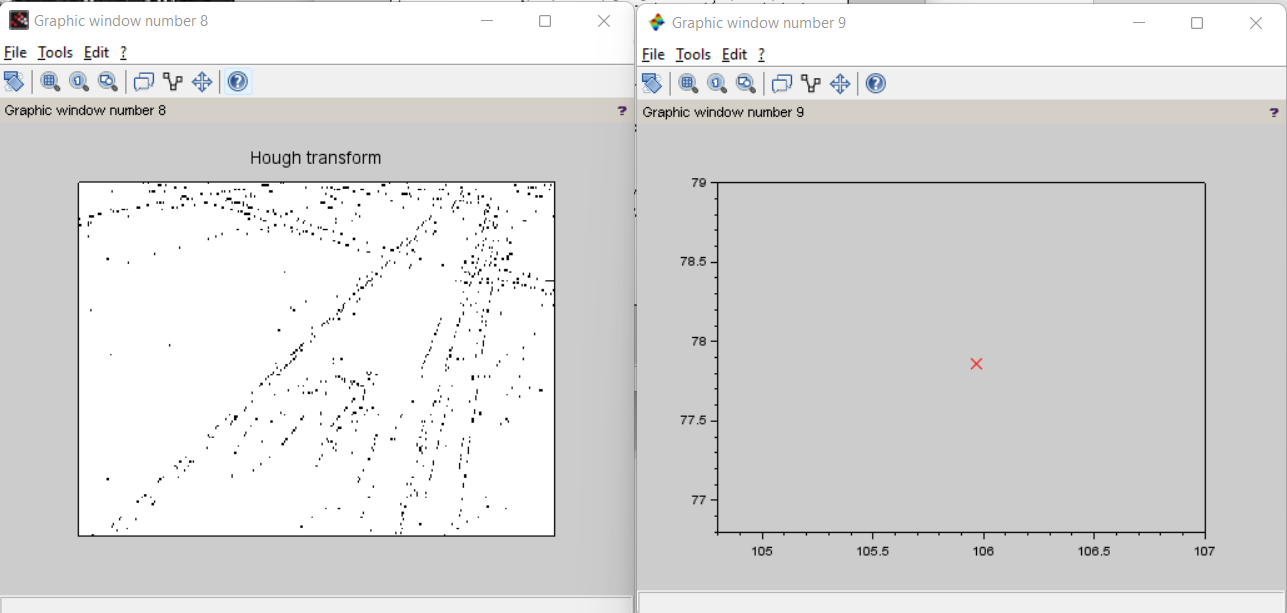
figure

mtlb\_hold on;

plot(mean(A),sy-mean(B),'rx');

**Output:**





**Result:** Hough transform has been successfully implemented for the given image.

|  |  |
| --- | --- |
| **Ex.No:** 10 | Feature extraction |
| **Date: 8/4/2022** |

**Aim:** To apply feature extractions operations on an image

**Source code:**

clc

clear;

close;

S= imread ("D:\ImageProcessing\nift.jpg");

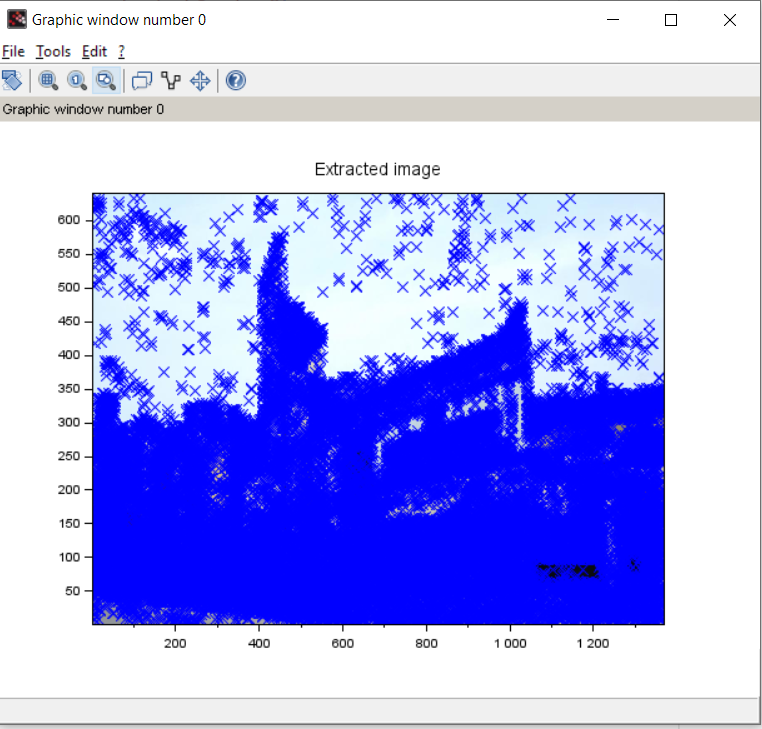
fobj = imdetect\_FAST(S);

title("Extracted image");

imshow(S);

plotfeature( fobj );

**Output:**



**Result:** Feature extraction has been performed on the given image.

|  |  |
| --- | --- |
| **Ex.No:** 11 | Dilation and erosion operators |
| **Date: 22/4/2022** |

**Aim:** To apply dilation and erosion operations on an image

**Program:**

clc

clear;

close;

*//Importing the image*

I = imread("D:\ImageProcessing\sunset.jfif");

subplot(2, 3, 1),

imshow(I);

title("Original image");

*//Dilated Image*

se = imcreatese('ellipse',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*

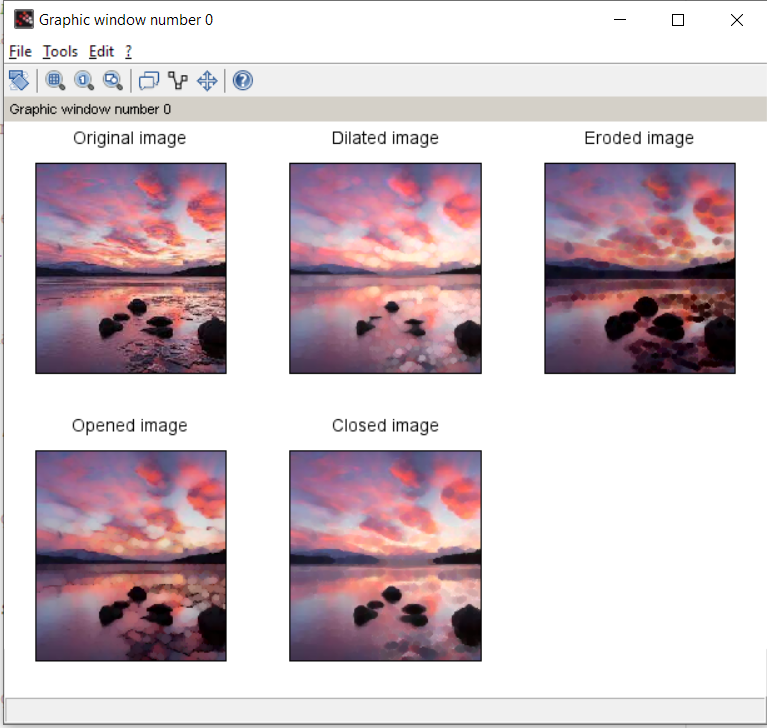
Close = imclose(I, se);

subplot(2, 3, 5),

imshow(Close);

title("Closed image");

**Output:**



**Result:** Dilation and erosion have been applied on the given image with the other morphological operations.