Image Processing

Certified Journal

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

MASTER OF SCIENCE
(INFORMATION_TECHNOLOGY)
By
Anjali Rameshwar Nimje

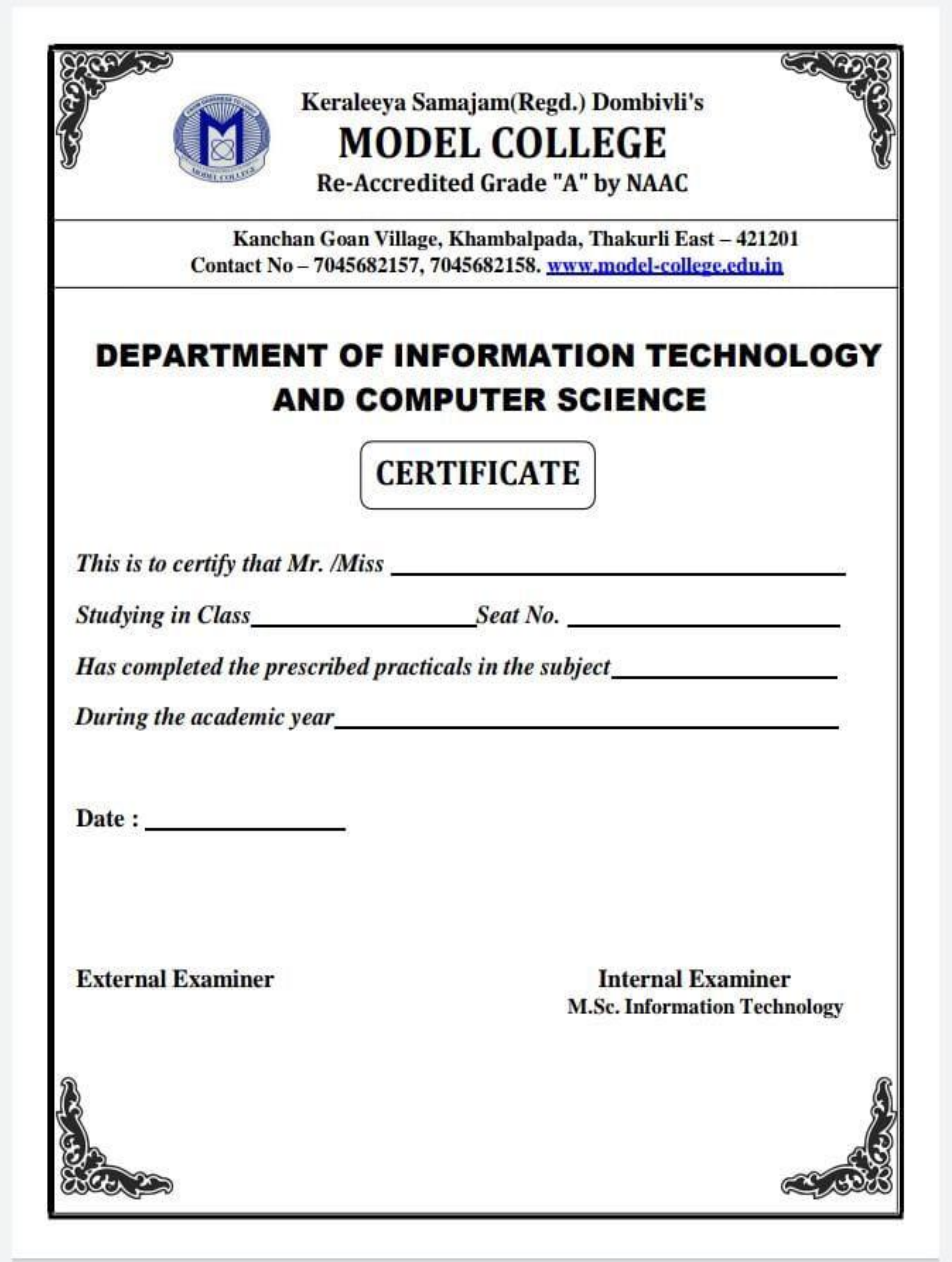
DEPARTMENT OF INFORMATION TECHNOLOGY

KERALEEYA SAMAJAM (REGD.) DOMBIVLI'S
MODEL COLLEGE (AUTONOMOUS)
Re-Accredited 'A' Grade by NAAC

(Affiliated to University of Mumbai)

FOR THE YEAR

(2022-23)



Keraleeya Samajam(Regd.) Dombivli's

MODEL COLLEGE

Re-Accredited Grade "A" by NAAC

Kanchan Goan Village, Khambalpada, Thakurli East – 421201
Contact No – 7045682157, 7045682158. www.model-college.edu.in

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER SCIENCE

CERTIFICATE

This is to certify that Mr. /Miss ______________________________

*Studying in Class*________________*Seat No.* ____________________

*Has completed the prescribed practicals in the subject*________________

*During the academic year*____________________________________

Date : ______________

External Examiner

Internal Examiner
M.Sc. Information Technology

Sr. No.	Name of the Practical	Date	Signature
1	Basics		
1A	Program to calculate number of samples required for an image.	16/04/2023	
1B	Program to study the effects of varying the number of intensity levels in a digital image	25/03/2023	
1C/D	Add and Subtract	30/04/2023	
2	Intensity transformation and Spatial Filtering		
2A	Program to perform Image negation	25/03/2023	
2B	Program to perform threshold on an image.	25/03/2023	
2C	Program to perform Log transformation	25/03/2023	
2D	Power-law transformations	16/04/2023	
2E	Contrast Stretching	16/04/2023	
2F	Gray-level slicing with and without background.	16/04/2023	
2G	Bit-plane slicing	16/04/2023	
2H	Program to plot the histogram of an image	16/04/2023	
2I	Program to apply histogram equalization	16/04/2023	
2J	Write a program to apply smoothing and sharpening filters on grayscale and color images a) Low Pass b) High Pass	16/04/2023	
2K	Write a program to perform convolution and correlation	30/04/2023	
3	Filtering in Frequency Domain		
3A	Program to apply Discrete Fourier Transform on an image	30/04/2023	

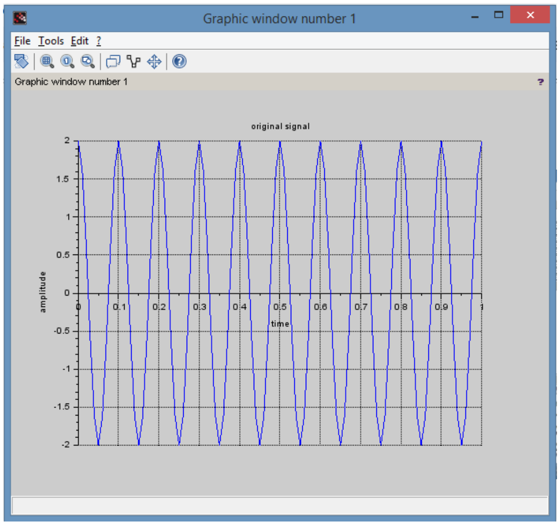
3B	Program to apply Low pass and High pass filters in frequency domain	30/04/2023	
4	Image Denoising		
4A	Program to denoise using spatial mean, median filtering	30/04/2023	
5	Color Image Processing		
5A	Program to read a color image and segment into RGB planes, histogram of color image	30/04/2023	
5B	Program for converting from one color model to another model	30/04/2023	
5C	Program to apply false coloring (pseudo) on a grayscale image	30/04/2023	
6	Fourier Related Transforms		
6A	Program to compute Discrete Cosine Transforms,	30/04/2023	
8	Morphological Image Processing		
8A	Program to apply erosion, dilation, opening, closing	30/04/2023	
9	Image Segmentation		
9A	Program for Edge detection using a. Sobel, Prewitt, Canny and thresholding	30/04/2023	

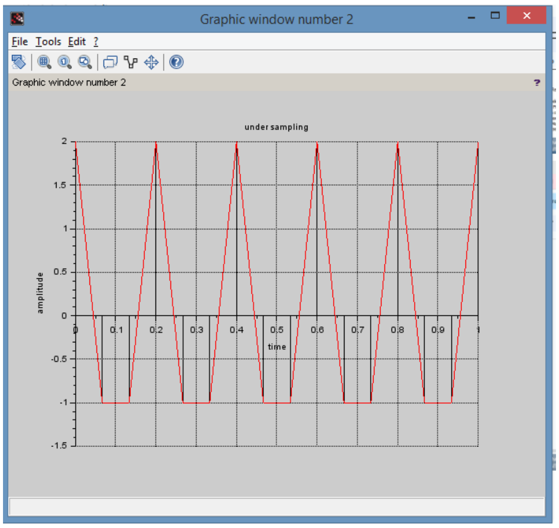
Practical No . 1 Basics

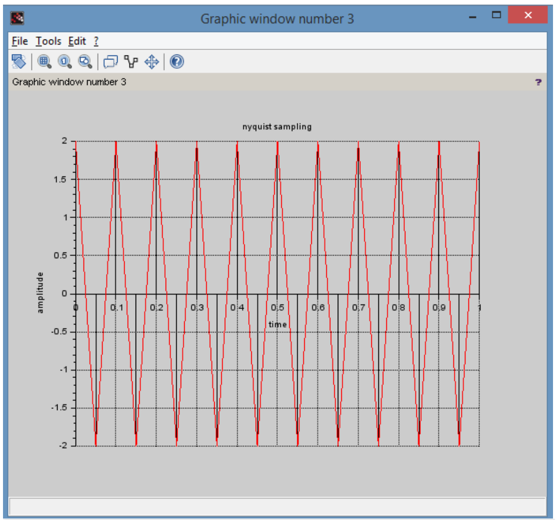
1A. Program to calculate number of samples required for an image.

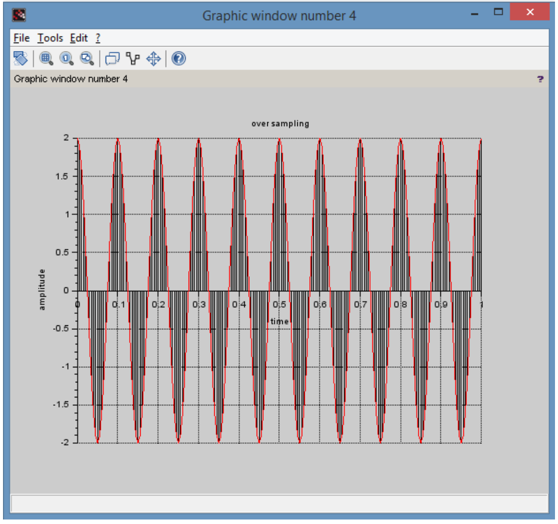
```
clc;
clear;
fm=input("Enter the input signal frequency");
k=input("Enter the number of cycle inputs");
A=input("Enter the amplitude");
tm=0:1/(fm*fm):k/fm;
x=A*cos(2*%pi*fm*tm);
figure(1);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot(tm,x);
title("original signal");
xlabel("time");
ylabel("amplitude");
xgrid(1);
fnyq=2*fm;
fs=(3/4)*fnyq;
n=0:1/fs:k/fm;
xn=A*cos(2*%pi*fm*n);
figure(2);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3('gnn',n,xn);
plot(n,xn,"r");
title("under sampling");
xlabel("time");
ylabel("amplitude");
xgrid(1);
fs=fnyq;
n=0:1/fs:k/fm;
xn=A*cos(2*%pi*fm*n);
figure(3);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3('gnn',n,xn);
plot(n,xn,"r");
title("nyquist sampling");
xlabel("time");
ylabel("amplitude");
xgrid(1);
fs=10*fnyq;
```

```
n=0:1/fs:k/fm;
xn=A*cos(2*%pi*fm*n);
figure(4);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3('gnn',n,xn);
plot(n,xn,"r");
title("over sampling");
xlabel("time");
ylabel("amplitude");
xgrid(1);
```



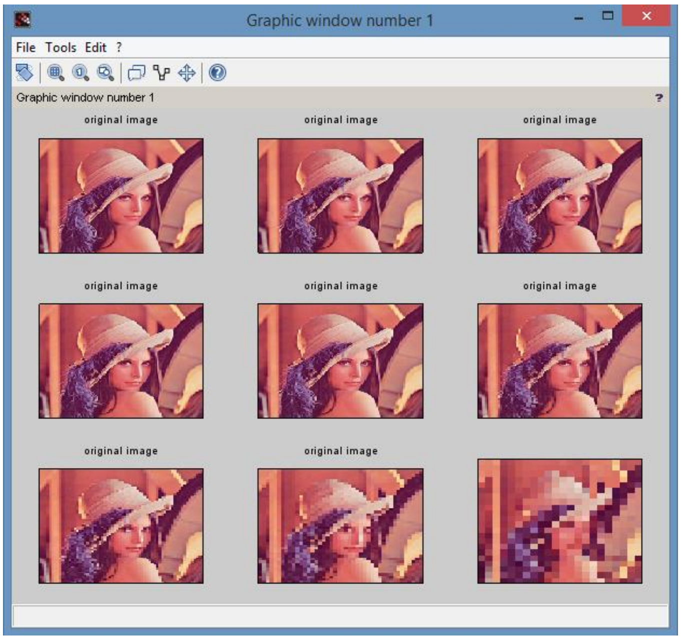



Graphic window number 3
File Tools Edit ?
Graphic window number 3
nyquist sampling
amplitude
time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Graphic window number 4
File Tools Edit ?
Graphic window number 4
over sampling
amplitude
time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

1B Program to study the effects of varying the number of intensity levels in a digital image

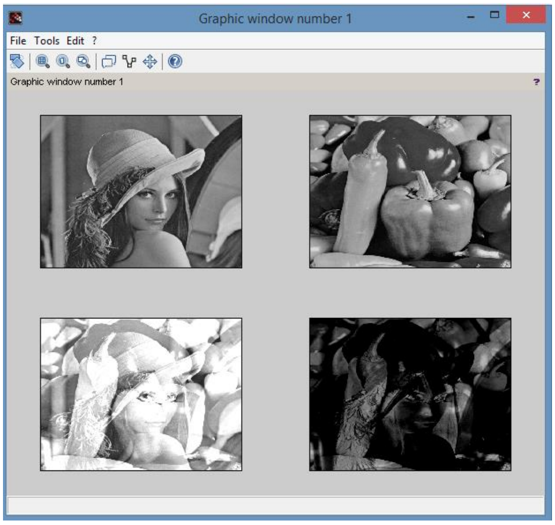
```
clc;
clear all;
figure(1)
//checker board effect sampling
subplot(3,3,1);
i=imread('D:\anjali ip')
imshow(i);
title('original image');
subplot(3,3,2);
j=imresize(i,0.8);
imshow(j);
title('original image');
subplot(3,3,3);
j=imresize(i,0.7);
imshow(j);
title('original image');
subplot(3,3,4);
j=imresize(i,0.6);
imshow(j);
title('original image');
subplot(3,3,5);
j=imresize(i,0.5);
imshow(j);
title('original image');
subplot(3,3,6);
j=imresize(i,0.4);
imshow(j);
title('original image');
subplot(3,3,7);
j=imresize(i,0.3);
imshow(j);
title('original image');
subplot(3,3,8);
j=imresize(i,0.2);
imshow(j);
title('original image');
subplot(3,3,9);
j=imresize(i,0.1);
imshow(j);
```


Graphic window number 1
File Tools Edit ?
Graphic window number 1
?
original image
original image
original image
original image
original image
original image
original image
original image

1C and 1D Add and Subtract

Process for image addition is called image averaging

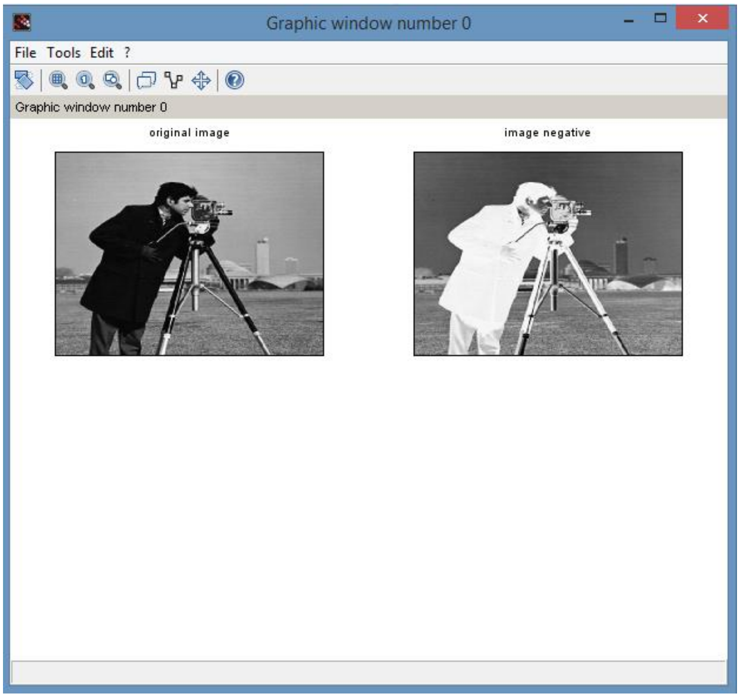
```
clc;
clear all;
A=imread("D:\lena.png");
B=imread("D:\peppers.png");
A=rgb2gray(A);
B=rgb2gray(B);
C=imadd(B,A);
D=imsubtract(B,A);
figure(1);
subplot(2,2,1);
imshow(A);
subplot(2,2,2);
imshow(B);
subplot(2,2,3);
imshow(C);
subplot(2,2,4);
imshow(D);
```



Practical No . 2 Intensity transformation and Spatial Filtering

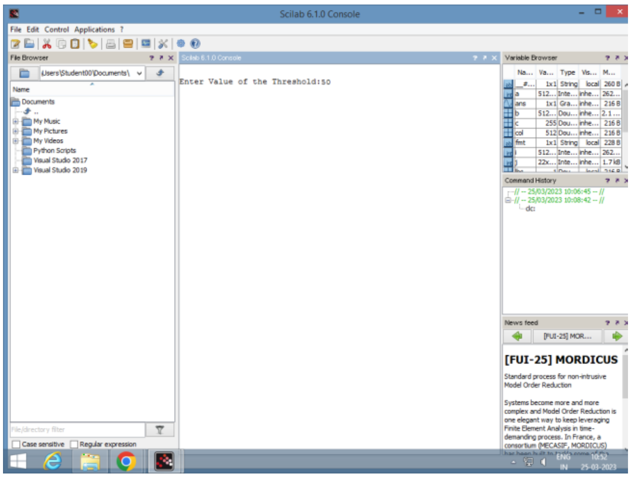
2. A Program to perform Image negation

```
//Image Enhancement in the Spaatial Domain
//Image Negative
clc;
clear all;
i=imread('D:\IPA\cameraman.jpg')
a=double(i);
c=255;
b=c-a;
subplot(2,2,1);
imshow(uint8(a));
title('original image');
subplot(2,2,2);
imshow(uint8(b));
title('image negative');
```



2B Program to perform threshold on an image.

```
//image enhancement in the spatial domain
//thresholding
clc,clear all;
p=imread('D:\IPA\cameraman.jpg');
a=p;
[row col]=size(a);
T=input('Enter Value of the Threshold:'); //Value of threshold 50
for i=1:1:row
   for j=1:1:col
      if(p(i,j)<T)
         a(i,j)=0;
      else
         a(i,j)=255;
      end
   end
end
subplot(1,2,1);
imshow(p);
title('Original Image');
subplot(1,2,2);
imshow(a);
title('Image obtained using threshold');
```



```
Enter Value of the Threshold:50
```

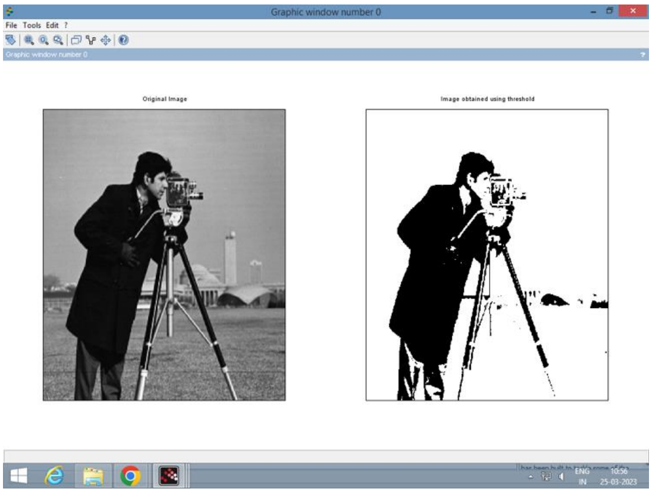
Na...	Va...	Type	Vis...	M...
__#...	1x1	String	local	260 B
a	512...	Inte...	inhe...	262...
ans	1x1	Gra...	inhe...	216 B
b	512...	Dou...	inhe...	2.1 ...
c	255	Dou...	inhe...	216 B
col	512	Dou...	inhe...	216 B
fmt	1x1	String	local	228 B
i	512...	Inte...	inhe...	262...
j	22x...	Inte...	inhe...	1.7 kB

```
// -- 25/03/2023 10:06:45 -- //
// -- 25/03/2023 10:08:42 -- //
   clc
```

[FUI-25] MORDICUS

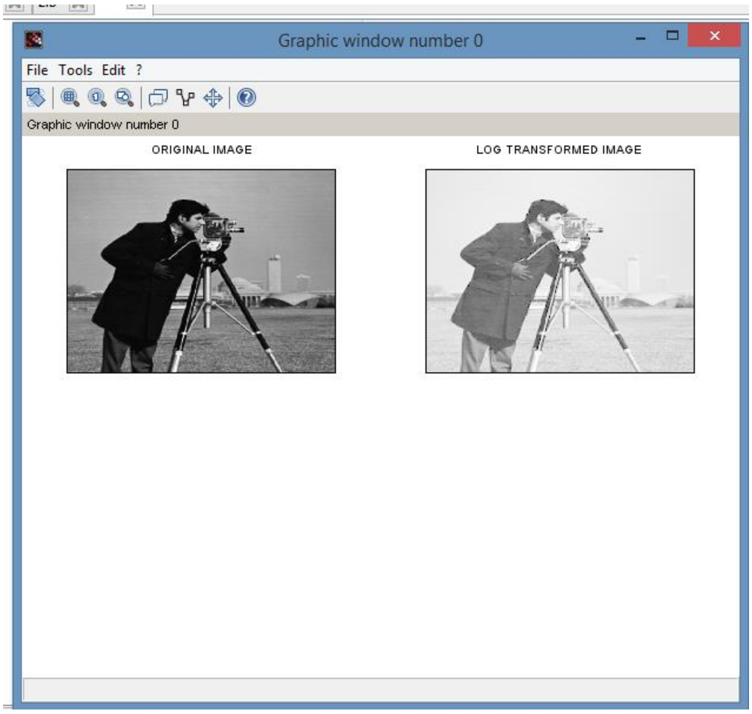
Standard process for non-intrusive Model Order Reduction

Systems become more and more complex and Model Order Reduction is one elegant way to keep leveraging Finite Element Analysis in time-demanding process. In France, a consortium (MECASIF, MORDICUS)


Graphic window number 0
File
Tools
Edit
?
Graphic window number 0
Original Image
Image obtained using threshold
ENG
IN
10:56
25-03-2023

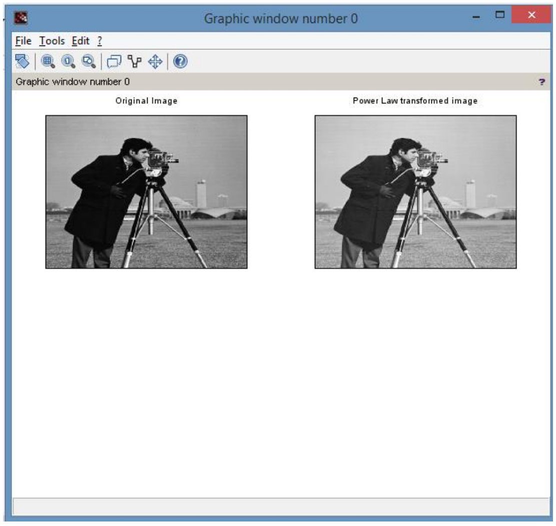
2C Program to perform Log transformation

```
//Log transformation
clc;
clear all;
Img2=imread('D:\IPA\cameraman.jpg');
L=255;
C=L/log(1+L);
//display C
S=C*log(1+double(Img2));
O=uint8(S);
subplot(2,2,1);
imshow(Img2);
title('ORIGINAL IMAGE');
subplot(2,2,2);
imshow(O);
title('LOG TRANSFORMED IMAGE');
```



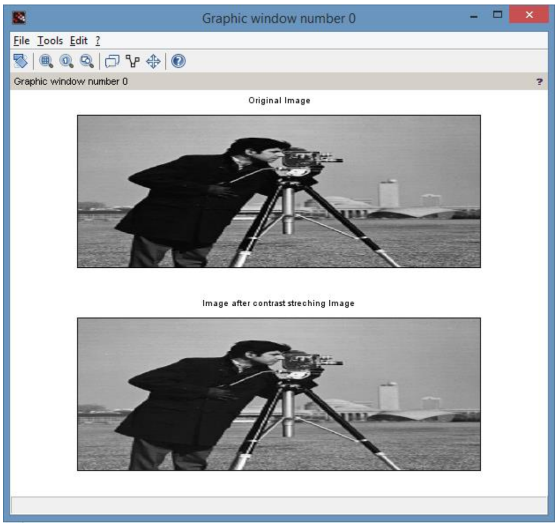
2D Power-law transformations

```
clear all;
clc;
itemp=imread('D:\cameraman.jpg');
r=double(itemp)/255;
c=1;
gamma=0.6;  // dark image >1 , light image <1
s=c*(r).^gamma;
subplot(2,2,1),imshow(uint8(itemp)),title('Original Image');
subplot(2,2,2),imshow(s),title('Power Law transformed image');
```



2E Contrast Stretching

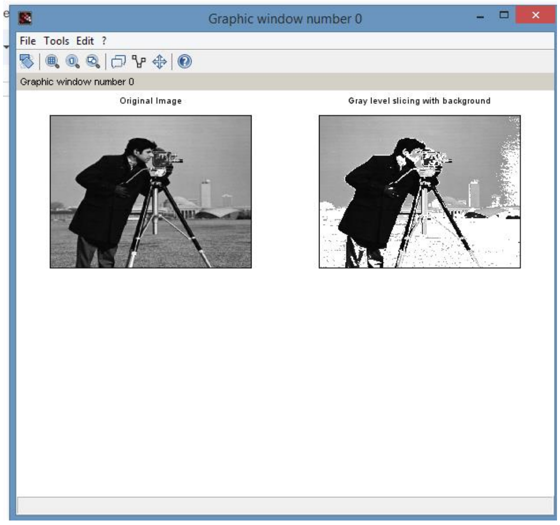
```
clc;
clear all;
a=imread('D:\cameraman.jpg');
a=double(a);
[row col]=size(a);
r1=input("Enter the value of R1");
s1=input("Enter the value of S1");
r2=input("Enter the value of R2");
s2=input("Enter the value of S2");
m1=s1/r1;
m2=(s2-s1)/(r2-r1);
m3=(255-s2)/(255-r2);
for x=1:1:row
    for y=1:1:col
        if a(x,y)<r1
            b(x,y)=m1*a(x,y);
        else if a(x,y)>=r1 && a(x,y)<=2
                b(x,y)=m2*(a(x,y)-r1)+s1;
            else b(x,y)=m3*(a(x,y)-r2)+s2;
            end
        end
     end
end
subplot(2,1,1);
imshow(uint8(a));
title("Original Image");
subplot(2,1,2);
imshow(uint8(b));
title("Image after contrast streching Image");
```



2F Gray-level slicing with and without background.

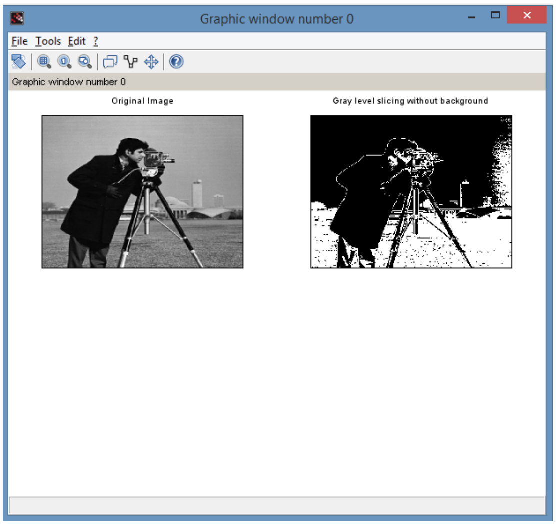
1. With background

```
clc;
clear all;
p=imread('D:\cameraman.jpg');
z=double(p);
[row col]=size(p);
for i=1:1:row
for j=1:1:col
if(z(i,j)>50)&&(z(i,j)<150)
z(i,j)=255;
else
z(i,j)=p(i,j);
end
end
end
subplot(2,2,1);
imshow(p);
title("Original Image");
subplot(2,2,2);
imshow(uint8(z));
title("Gray level slicing with background");
```



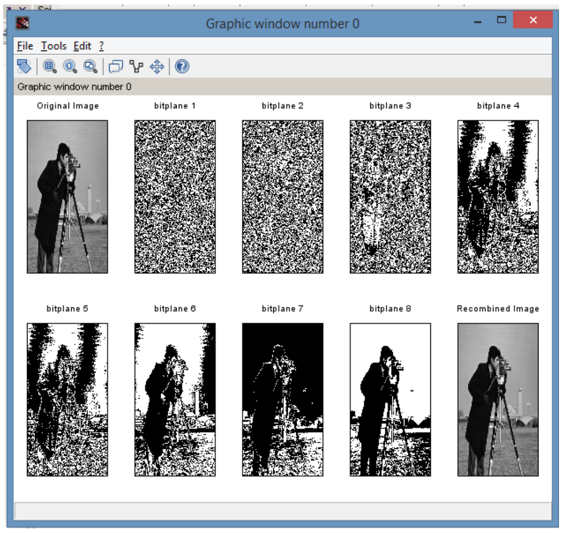
2. Without background

```
clc;
clear all;
p=imread('D:\cameraman.jpg');
z=double(p);
[row col]=size(p);
for i=1:1:row
for j=1:1:col
if(z(i,j)>50)&&(z(i,j)<150)
z(i,j)=255;
else
z(i,j)=0;
end
end
end
subplot(2,2,1);
imshow(p);
title("Original Image");
subplot(2,2,2);
imshow(uint8(z));
title("Gray level slicing without  background");
```



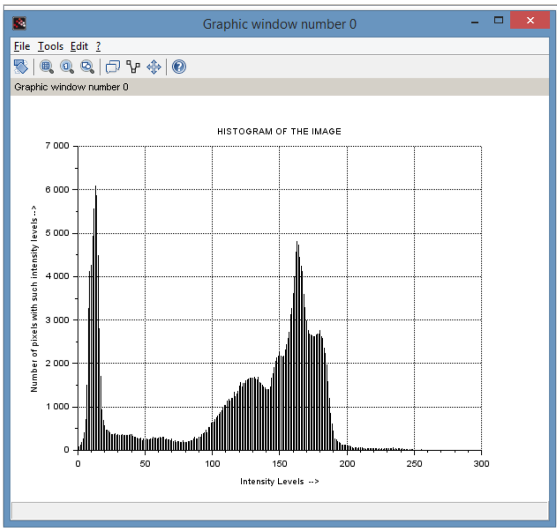
2G Bit-plane slicing

```
clc;
clear all;
c=imread('D:\cameraman.jpg');
cd=double(c);
c1=modulo(cd,2);
c2=modulo(floor(cd/2),2);
c3=modulo(floor(cd/4),2);
c4=modulo(floor(cd/8),2);
c5=modulo(floor(cd/16),2);
c6=modulo(floor(cd/32),2);
c7=modulo(floor(cd/64),2);
c8=modulo(floor(cd/128),2);
cc=(2*(2*(2*(2*(2*(2*(2*c8+c7)+c6)+c5)+c4)+c3)+c2)+c1);
subplot(2,5,1);
imshow(c);
title("Original Image");
subplot(2,5,2);
imshow(c1);
title("bitplane 1");
subplot(2,5,3);
imshow(c2);
title("bitplane 2");
subplot(2,5,4);
imshow(c3);
title("bitplane 3");
subplot(2,5,5);
imshow(c5);
title("bitplane 4");
subplot(2,5,6);
imshow(c5);
title("bitplane 5");
subplot(2,5,7);
imshow(c6);
title("bitplane 6");
subplot(2,5,8);
imshow(c7);
title("bitplane 7");
subplot(2,5,9);
imshow(c8);
title("bitplane 8");
subplot(2,5,10);
imshow(uint8(cc));
title("Recombined Image");
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original Image
bitplane 1
bitplane 2
bitplane 3
bitplane 4
bitplane 5
bitplane 6
bitplane 7
bitplane 8
Recombined Image

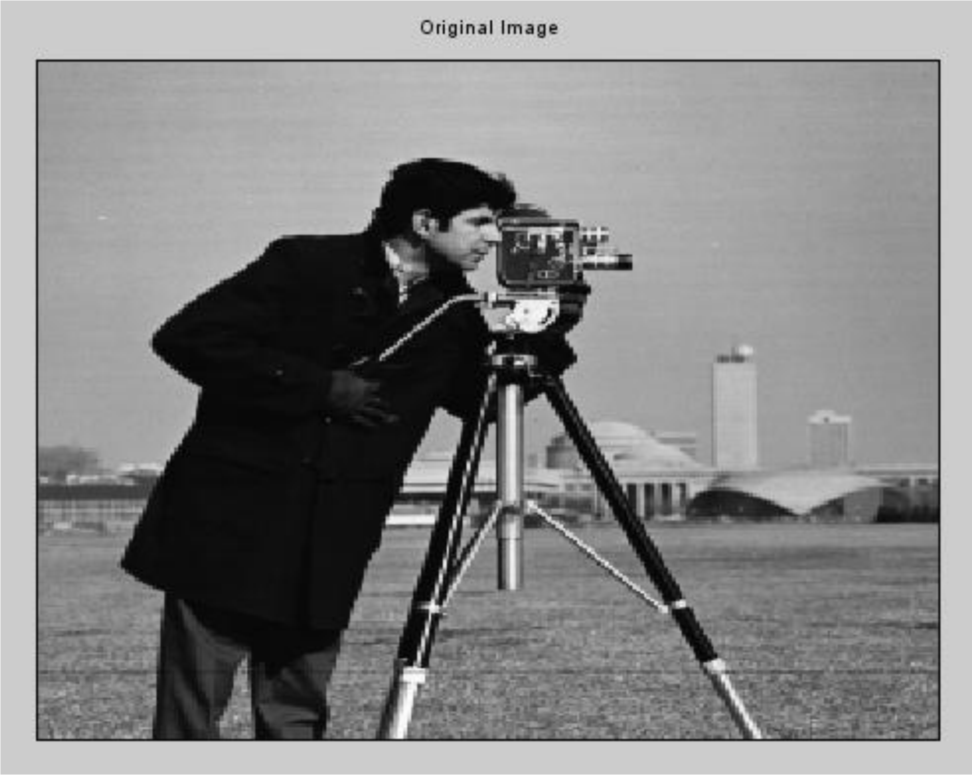
2H Program to plot the histogram of an image

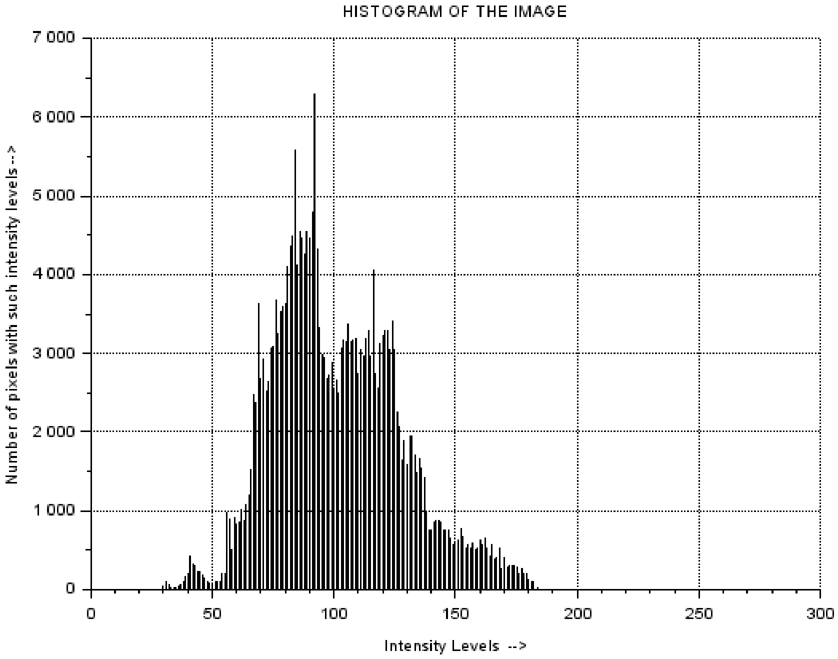
```
//histogram
clc;
clear all;
//Read source image file
a=imread('D:\cameraman.jpg');
//Convert image to grayscale image
//img=rgb2gray(img);
//get the dimension of the image
[row col]=size(a);
//Create a frequency array of size 256
frequency = 1 : 256;
count = 0;
//Iterate over grayscale image matrix
//for every possible intensity value
//and count them
   for i = 1 : 256
   for j = 1 : row
      for k = 1 : col

         //if image pixel value at location (j, k) is i-1
         //then increment count
         if a(j,k) == i-1
               count = count + 1;
         end
      end
   end
   //update ith position of frequency array with count
   frequency(i) = count;

   // reset count
   count = 0;
end
n = 0 : 255;
//Display Histogram
plot2d3(n, frequency);
xgrid(1);
ylabel('Number of pixels with such intensity levels -->');
xlabel('Intensity Levels  -->');
title('HISTOGRAM OF THE IMAGE');
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
HISTOGRAM OF THE IMAGE
Number of pixels with such intensity levels -->
Intensity Levels -->
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000
0
50
100
150
200
250
300

2I Program to apply histogram equalization

```
//histogram equalization
GIm=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tif');
numofpixels=size(GIm,1)*size(GIm,2);
figure,imshow(GIm);
title('Original Image');
HIm=uint8(zeros(size(GIm,1),size(GIm,2)));
freq=zeros(256,1);
probf=zeros(256,1);
probc=zeros(256,1);
cum=zeros(256,1);
output=zeros(256,1);
//freq counts the occurrence of each pixel value.
//The probability of each occurrence is calculated by probf.
for i=1:size(GIm,1)
    for j=1:size(GIm,2)
        value=GIm(i,j);
        freq(value+1)=freq(value+1)+1;
        probf(value+1)=freq(value+1)/numofpixels;
    end
end
sum=0;
no_bins=255;
//The cumulative distribution probability is calculated.
for i=1:size(probf)
   sum=sum+freq(i);
   cum(i)=sum;
   probc(i)=cum(i)/numofpixels;
   output(i)=round(probc(i)*no_bins);
end
for i=1:size(GIm,1)
    for j=1:size(GIm,2)
            HIm(i,j)=output(GIm(i,j)+1);
    end
end
figure,imshow(HIm);
title('Histogram equalization');
```


Original Image


HISTOGRAM OF THE IMAGE
Number of pixels with such intensity levels -->
7 000
6 000
5 000
4 000
3 000
2 000
1 000
0
0
50
100
150
200
250
300
Intensity Levels -->

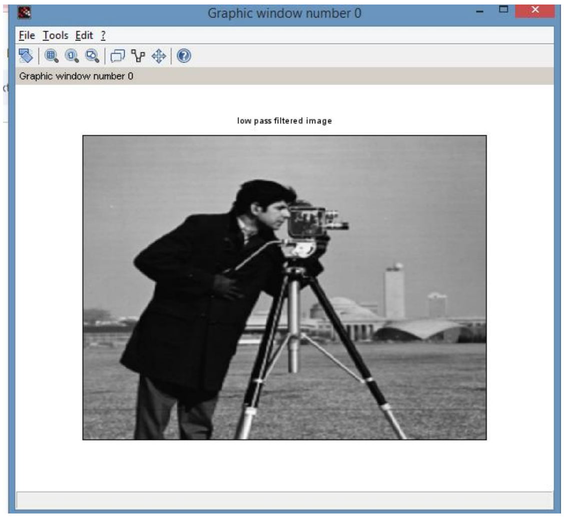
2j Write a program to apply smoothing and sharpening filters on grayscale and color images

a) Low Pass

b) High Pass

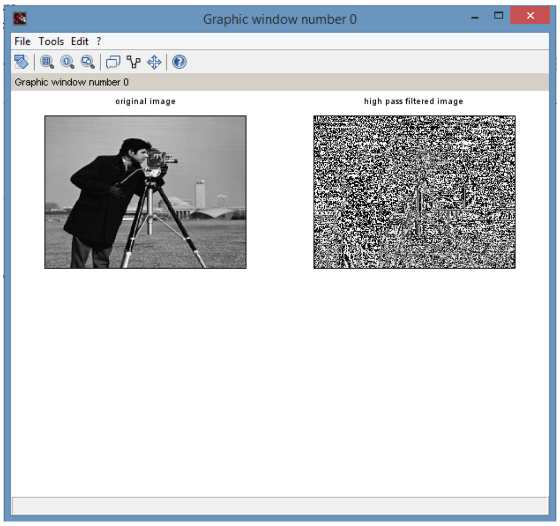
LOW

```
clc;
clear all;
a1=imread('D:\cameraman.jpg');
a=double(a1);
//subplot(2,2,1);
imshow(double(a));
title('original image');
[row col]=size(a);
w = (1/9) * (ones (3,3));
for x=1:row
    for y=1:col
        new (x,y) = a(x,y);
    end
end
for x=2:1:row-1
 for y = 2:1:col-1
        new (x,y)=(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));
    end
end
//subplot (2,2,2);
imshow(uint8(new));
title("low pass filtered image");
```



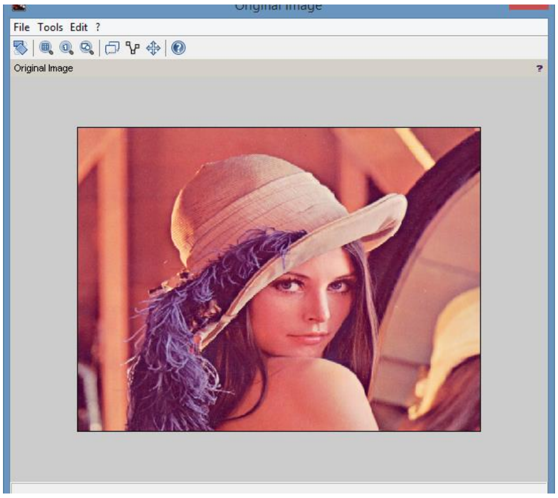
HIGH

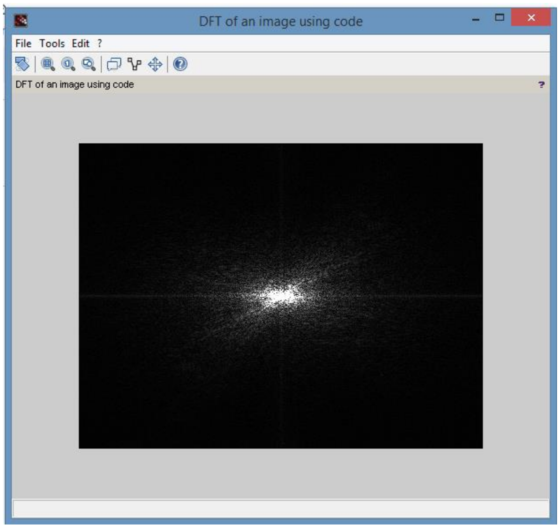
```
//high pass filter
clc;
clear all;
a=imread('D:\cameraman.jpg');
//a=double(a1);
subplot(2,2,1);
imshow(a);
title('original image');
[row col]=size(a);
w = [-1,-1,-1; -1,8,-1; -1,-1,-1];
for x=2:1:row-1
for y = 2:1:col-1
new (x,y)=(w(1)*a(x-1,y-1))+ (w(2)*a(x-1,y))+(w(3)*a(x-1,y+1))+(w(4)*a(x,y-1))+(w(5)*a(x,y))+(w(6)*a(x,y+1))+(w(7)*a(x+1,y-1))+(w(8)*a(x+1,y))+(w(9)*a(x+1,y+1));
end
end
subplot (2,2,2);
imshow(uint8(new));
title("high pass filtered image");
```

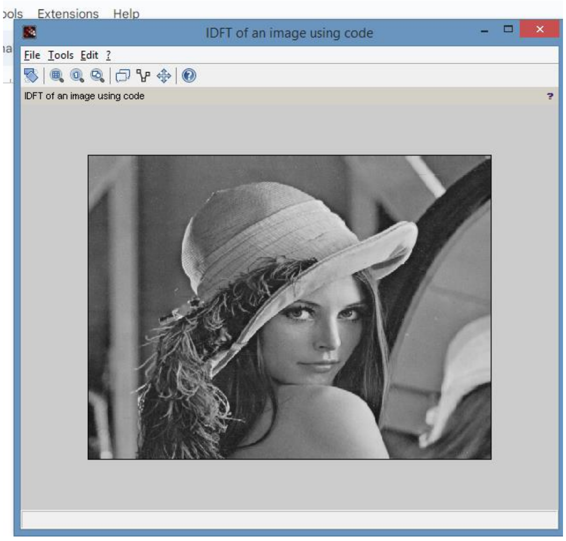


Practical No . 3 Filtering in Frequency Domain
3A. Program to apply Discrete Fourier Transform on an image

```
clear;
clc;
clear all;
close;
img=imread("D:\lena.png");
figure ();
xname("Original Image");
imshow(img);
img_gray=rgb2gray(img);
img_double=im2double(img_gray);
[m,n]=size(img_gray);
for x=1:m
for y=1:n
c(x,y)=exp((-2*%i*%pi*((x-1)*(y-1)))/m);
end
end
dft=c*img_double*inv(c);
res=dft;
dft=fftshift(dft);
dft=abs(dft);
figure();
xname('DFT of an image using code');
imshow(dft);
idfv=inv(c)*res*c;
res_idft=abs(idfv);
figure();
xname("IDFT of an image using code");
imshow(res_idft);
```




DFT of an image using code
File Tools Edit ?
DFT of an image using code
?


ools Extensions Help
IDFT of an image using code
File Tools Edit ?
IDFT of an image using code
?

3B Program to apply Low pass and High pass filters in frequency domain

```
clear ;
clc ;
clear all;
close ;
img = imread ('D:\lena.png'); // input image −−> l e n a .jpg
img_gray = rgb2gray ( img );
img_gray = imresize ( img_gray , [256 , 256]) ;
figure ();
xname ('Gray image');
imshow ( img_gray );
// Cr e a t i n g LPF mask LPF
mask_LPF = ones (3 ,3) /9;
disp ( mask_LPF );
img_LPF = conv2 ( double ( img_gray ), mask_LPF );
img_LPF = uint8 ( img_LPF )
figure ();
xname ('Image a f t e r LPF');
imshow ( uint8 ( img_LPF ));
// Cr e a t i n g HPF mask HPF
mask_HPF = ones (3 ,3);
mask_HPF = mask_HPF * -1;
mask_HPF (2 ,2) = mask_HPF (2 ,2) + 9
disp ( mask_HPF );
mask_HPF = mask_HPF /9
disp ( mask_HPF );
img_HPF = conv2 ( double ( img_gray ), mask_HPF );
figure ();
xname ('Image a f t e r HPF');
// To make n e g a t i v e numbers z e r o s
img_HPF = (abs( img_HPF ) + img_HPF )/2;
img_HPF = uint8 ( img_HPF )
imshow ( uint8 ( img_HPF ));
//High Boos t F i l t e r
// Cr e a t e HBF mask
mask_HBF = ones (3 ,3);
mask_HBF = mask_HBF * -1;
A = 5;
mask_HBF (2 ,2) = 8 + A
disp ( mask_HBF );
mask_HBF = mask_HBF /9
disp ( mask_HBF );
[m,n] = size ( img_gray )
padded_img = zeros (m+2,n +2) ;
// c r e a t e image wi th z e r o s padded at the b o u n d a r i e s
u =2;
v =2;
for x=1: m
for y =1: n
```

```
padded_img (u,v) = img_gray (x,y);
v = v+1;
end
u = u+1;
v = 2;
end
hbf = zeros (m+2,n +2) ;
// applying the HBF mask on the image
u =1; v=1;
for x=2: m+1
for y =2: n+1
hbf (x,y) = padded_img (x -1,y -1) * mask_HBF (1 ,1) + padded_img (x -1,y)* mask_HBF (1
,2) + padded_img (x -1,y+1) * mask_HBF (1 ,3) + padded_img (x,y -1)* mask_HBF (2 ,1) +
padded_img (x,y)* mask_HBF (2 ,2) + padded_img (x,y+1)* mask_HBF (2 ,3) +
padded_img (x+1,y-1) * mask_HBF (3 ,1) + padded_img ( +1,y)* mask_HBF (3 ,2) +
padded_img (x+1,y+1)* mask_HBF (3 ,3) ;
v=v+1;
end
u=u+1;
end
// remove padded zeros
for x=2: m+1
for y =2: n+1
hbf_img (x -1,y -1) = hbf (x,y);
end
end
// convert all negative values to zeros
hbf_img = (abs( hbf_img )+ hbf_img )/2;
// Display HBF image
figure ();
xname ('HBF image');
imshow ( uint8 ( hbf_img ));
// Gaussian Filtering
N = 3
sigma = 1
ind = -floor (N /2) : floor (N /2) ;
disp (ind)
[X Y] = meshgrid (ind , ind )
// create gaussian Mask
mask_gaussian = (1/(2* %pi * sigma ))* exp (-(X .^2 + Y .^2) / (2* sigma * sigma ));
mask_gaussian = [[1 , 2 , 1];[2 ,4 ,2];[1 ,2 ,1]];
disp ( mask_gaussian )
// Normalize so that total area ( sum of all weights )is 1
mask_gaussian = mask_gaussian / sum ( mask_gaussian (:));
disp ( mask_gaussian )
img_gaussian = conv2 ( double ( img_gray ), mask_gaussian);
figure ();
xname ('Image after Gaussian Filter ');
imshow ( uint8 ( img_gaussian ));
imwrite ( uint8 ( img_gaussian ), 'noisefiltered img gaussian . jpg' );
```

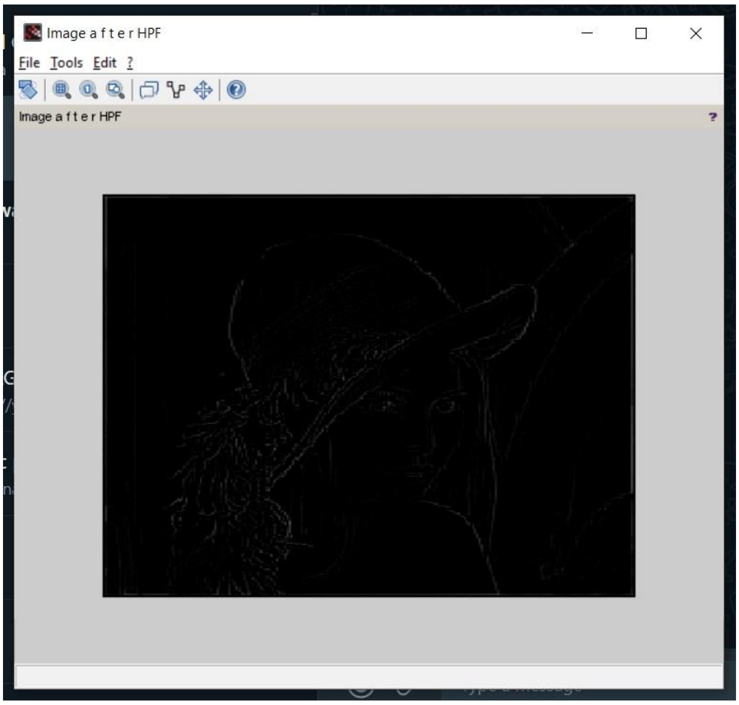


Image a f t e r HPF
File Tools Edit ?
Image a f t e r HPF

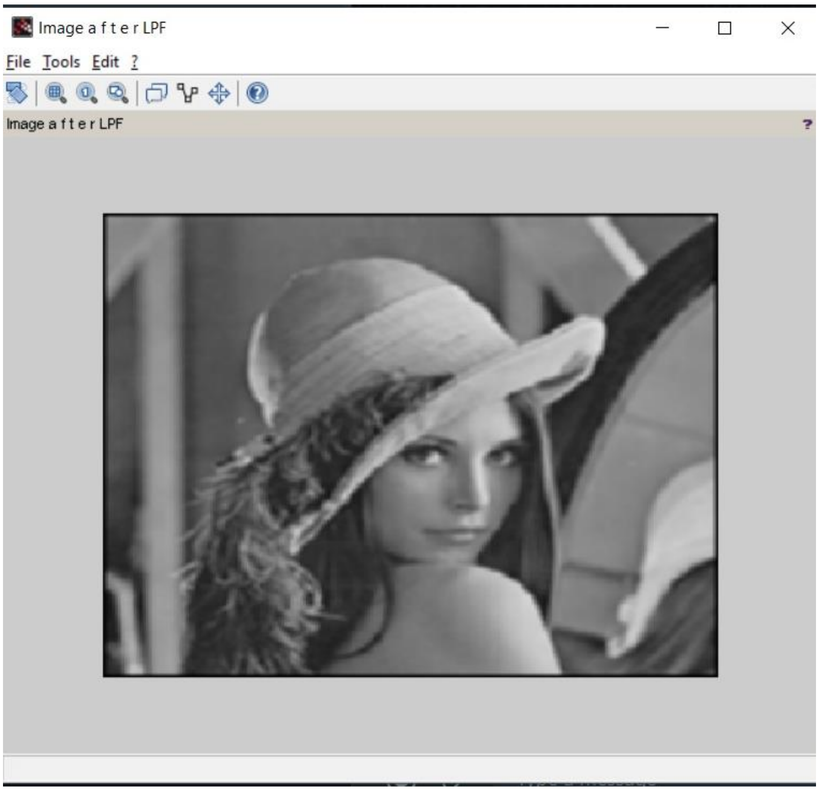
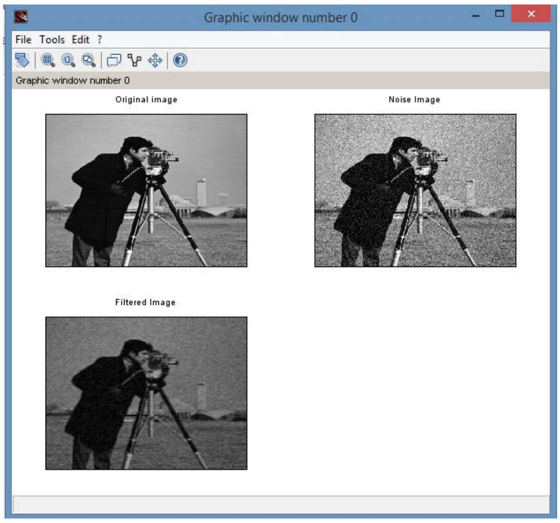


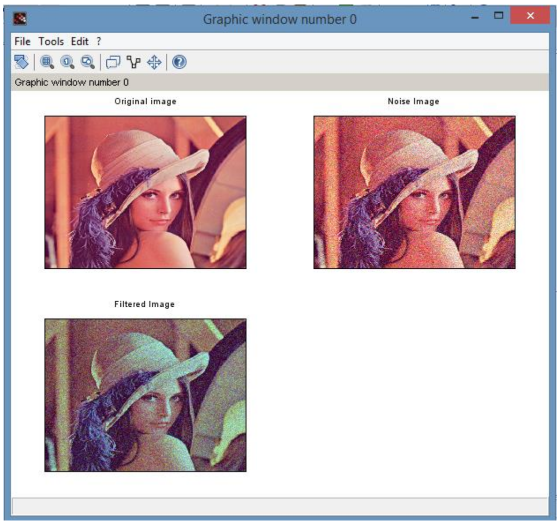
Image a f t e r LPF
File Tools Edit ?
Image a f t e r LPF

Practical No .4 Image Denoising

4A. Program to denoise using spatial mean, median filtering

```
a=imread("D:\cameraman.tif");
b1=double(a);
c=imnoise(a,'gaussian');
d=double(c);
b=d;
m=(1/9)*(ones(3,3));
[r1,c1]=size(a);
subplot(2,2,1);
imshow(a);
title('Original image');
subplot(2,2,2);
imshow(c);
title('Noise Image');
for i=2:r1-1
for j=2:c1-1
a1=d(i-1,j-1)+d(i-1,j)+d(i-1,j+1)+d(i,j-1)+d(i,j)+d(i,j+1),+d(i+1,j-1)+d(i+1,j)+d(i+1,j+1);
b(i,j)=a1*(1/9);
end
end
subplot(2,2,3);
imshow(uint8(b));
title('Filtered Image');
```



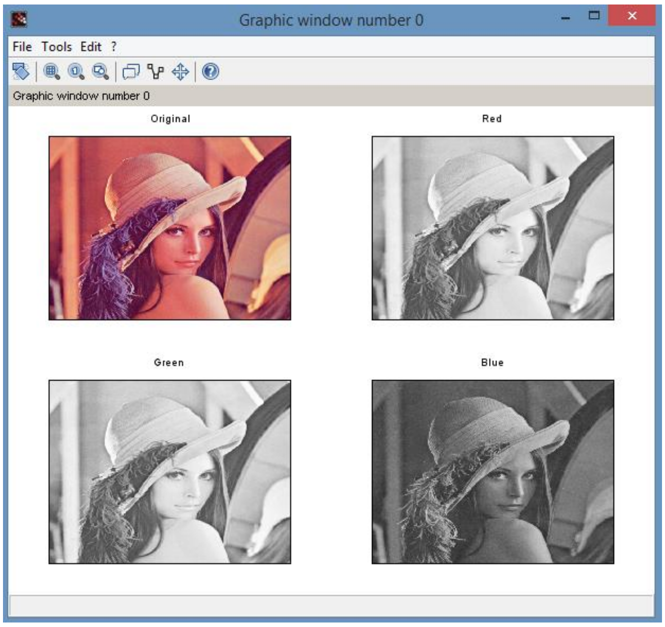


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original image
Noise Image
Filtered Image

Practical No . 5 Color Image Processing

5A. Program to read a color image and segment into RGB planes, histogram of color image

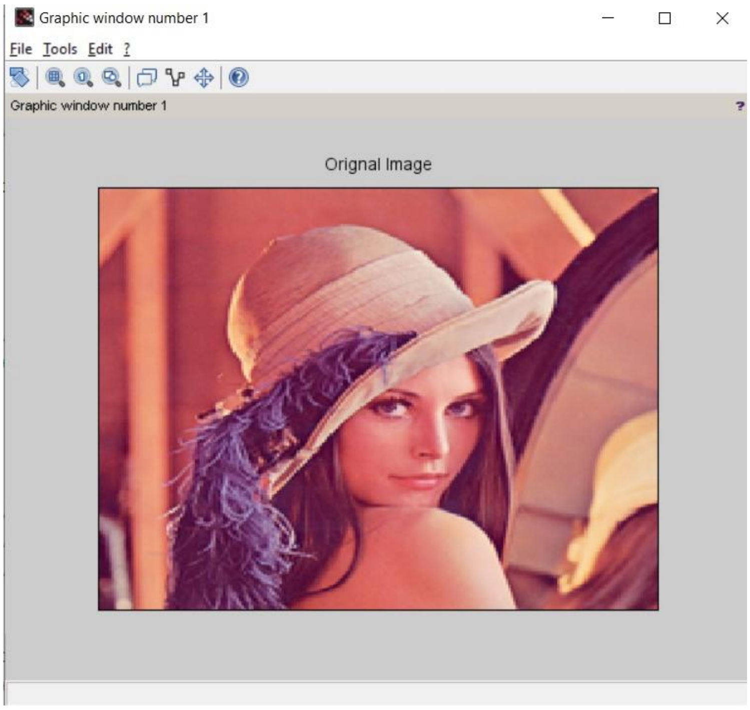
```
//program to show RGB planes
original=imread('D:\lena.png');
im_red=original(:,:,1);
im_green=original(:,:,1);
im_blue=original(:,:,3);
subplot(2,2,1),imshow(original),title('Original');
subplot(2,2,2),imshow(im_red),title('Red');
subplot(2,2,3),imshow(im_green),title('Green');
subplot(2,2,4),imshow(im_blue),title('Blue');
```

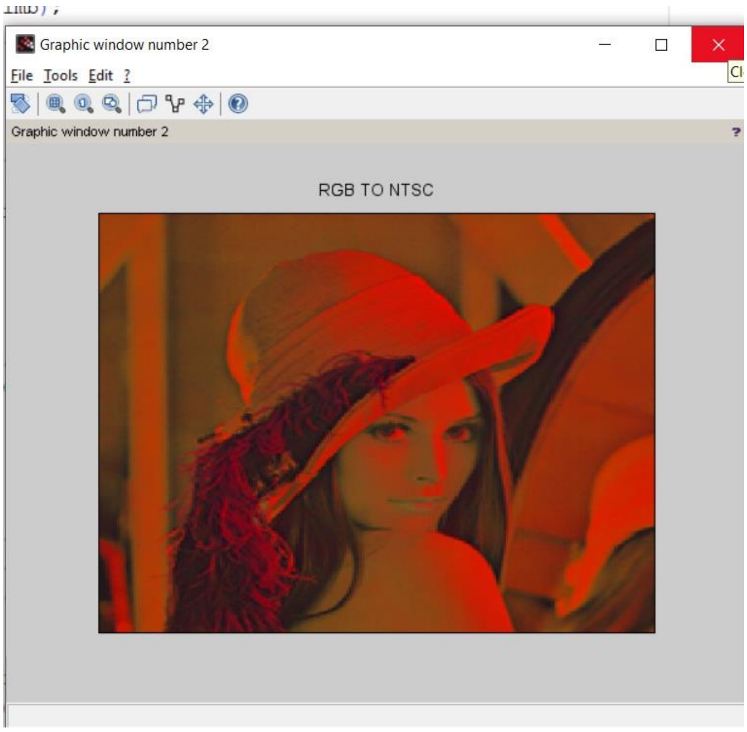


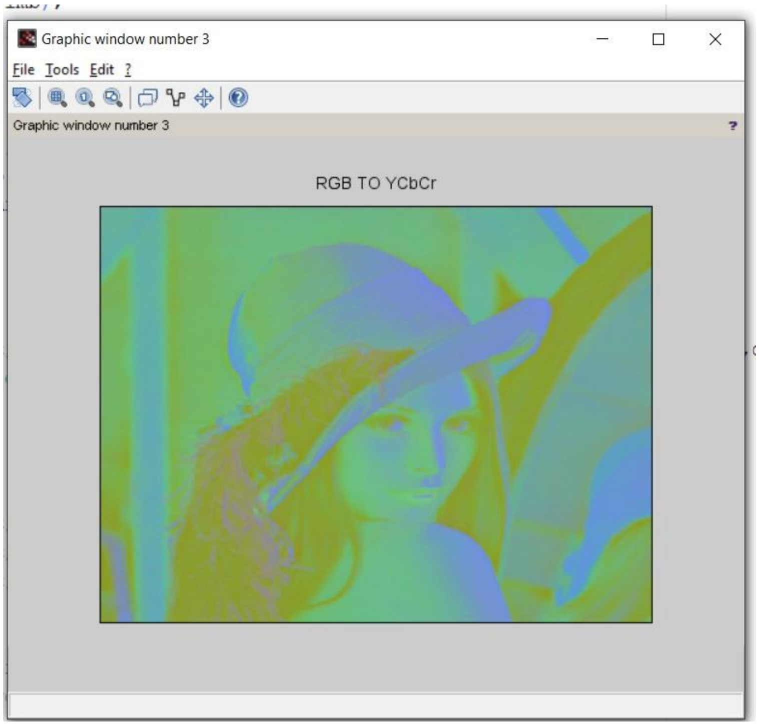
5B. Program for converting from one color model to another mode

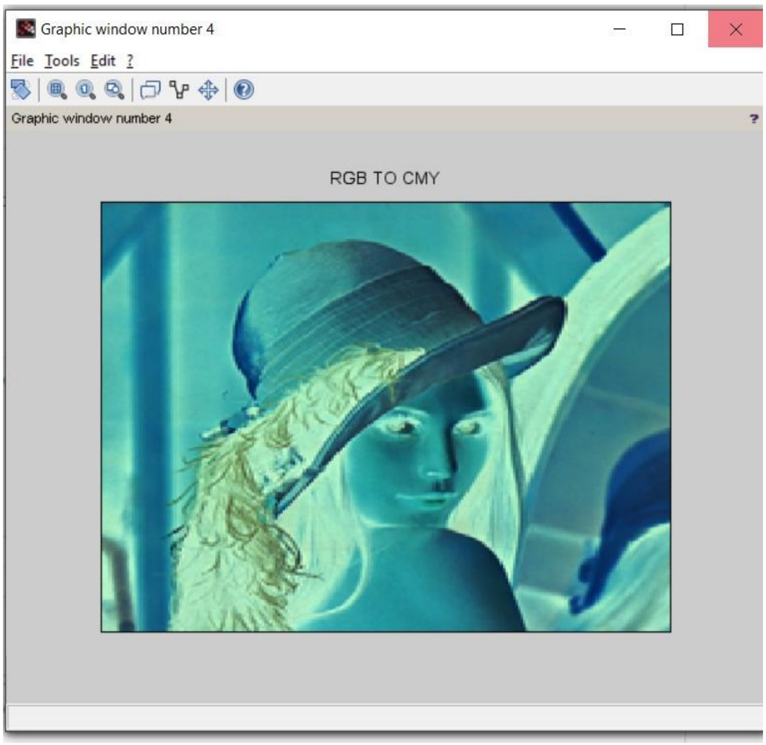
//WAP to convert RGB to NTSC, RGB to YCbCr and RGB to CMY

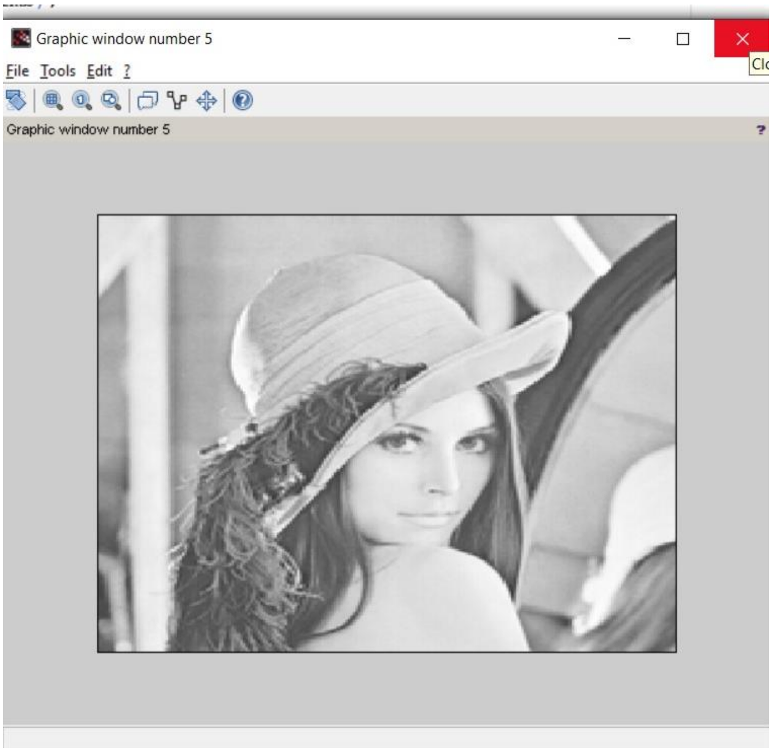
```
clc;
clear all;
close all;
a = imread('D:\lena.png');
figure(1),imshow(a);
title('Orignal Image');
k=rgb2ntsc(a);
figure(2),imshow(k);
title('RGB TO NTSC');
l=rgb2ycbcr(a);
figure(3),imshow(l);
title('RGB TO YCbCr');
m=imcomplement(a);
figure(4),imshow(m);
title('RGB TO CMY');
imr=a(:,:,1);
img=a(:,:,2);
imb=a(:,:,3);
figure(5),imshow(imr);
figure(6),imshow(img);
figure(7),imshow(imb);
I=(imr+img+imb)/3;
[m,n]=size(imr);
for c=1:m
for d=1:n
min1=min(imr(c,d),img(c,d));
min2=min(min1,imb(c,d));
S(c,d) = 1-(3/(imr(c,d)+img(c,d)+imb(c,d)))*min2;
end
end
for c=1:m
for d=1:n
temp= (0.5*(imr(c,d)-img(c,d))+(imr(c,d)-imb(c,d)))/sqrt(double(imr(c,d)*imr(c,d)+(imr(c,d)-imb(c,d))*(img(c,d)-imb(c,d))));
H(c,d)=acos(double(temp));
end
end
for c=1:m
for d=1:n
finali(c,d,1)=I(c,d);
finali(c,d,2)=S(c,d);
finali(c,d,3)=H(c,d);
end
end
figure(8),imshow(finali);
title('Final image');
```

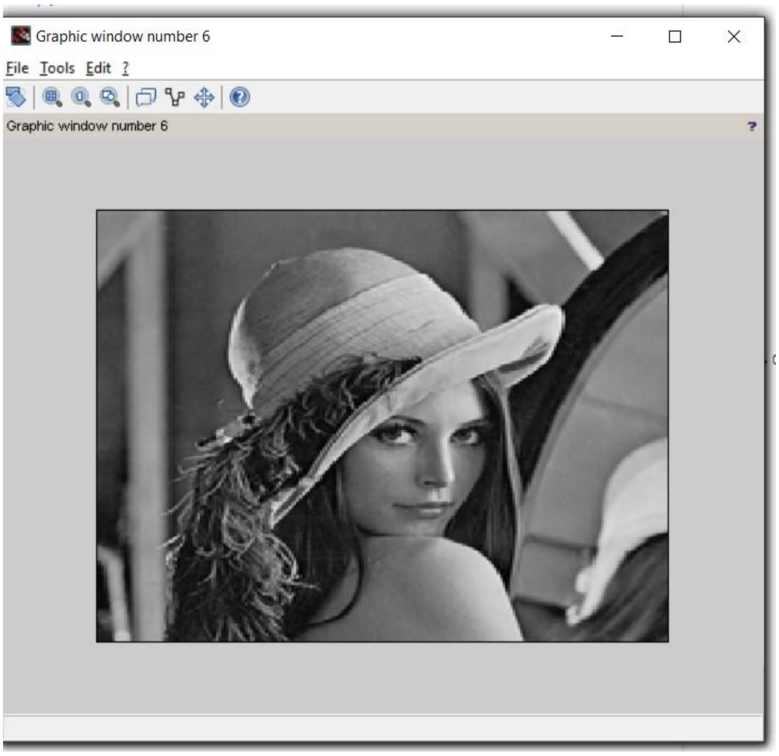

Graphic window number 1
File Tools Edit ?
Graphic window number 1
Orignal Image

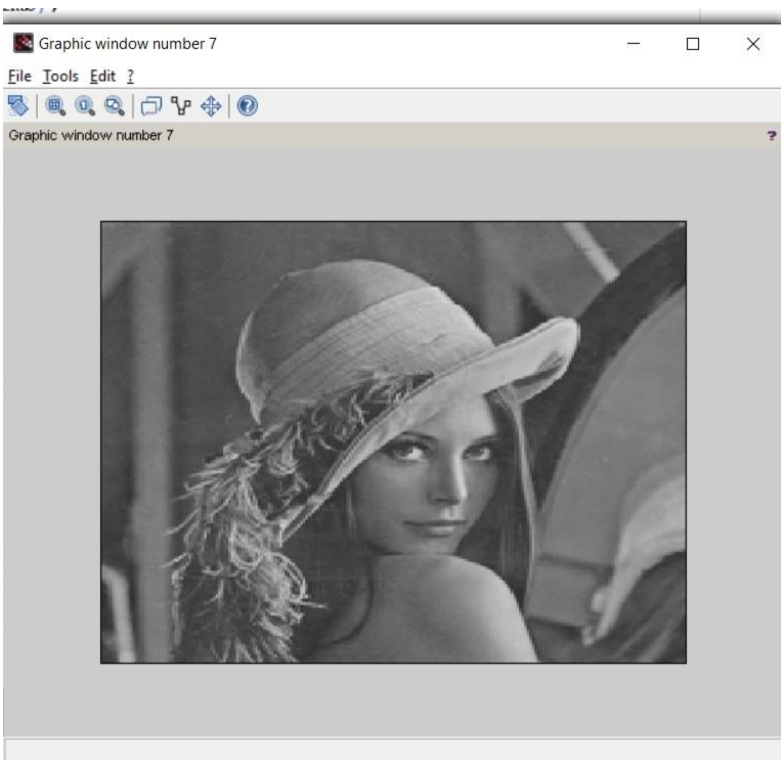

Graphic window number 2
File Tools Edit ?
Graphic window number 2
RGB TO NTSC


Graphic window number 3
File Tools Edit ?
Graphic window number 3
RGB TO YCbCr


Graphic window number 4
File Tools Edit ?
Graphic window number 4
RGB TO CMY

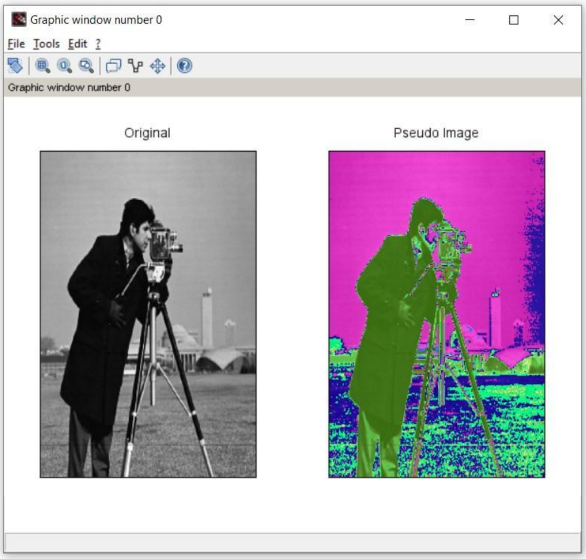

Graphic window number 5
File Tools Edit ?
Graphic window number 5
?
Clo


Graphic window number 6
File Tools Edit ?
Graphic window number 6
?


Graphic window number 7
File Tools Edit ?
Graphic window number 7
?

5C. Program to apply false coloring (pseudo) on a grayscale image

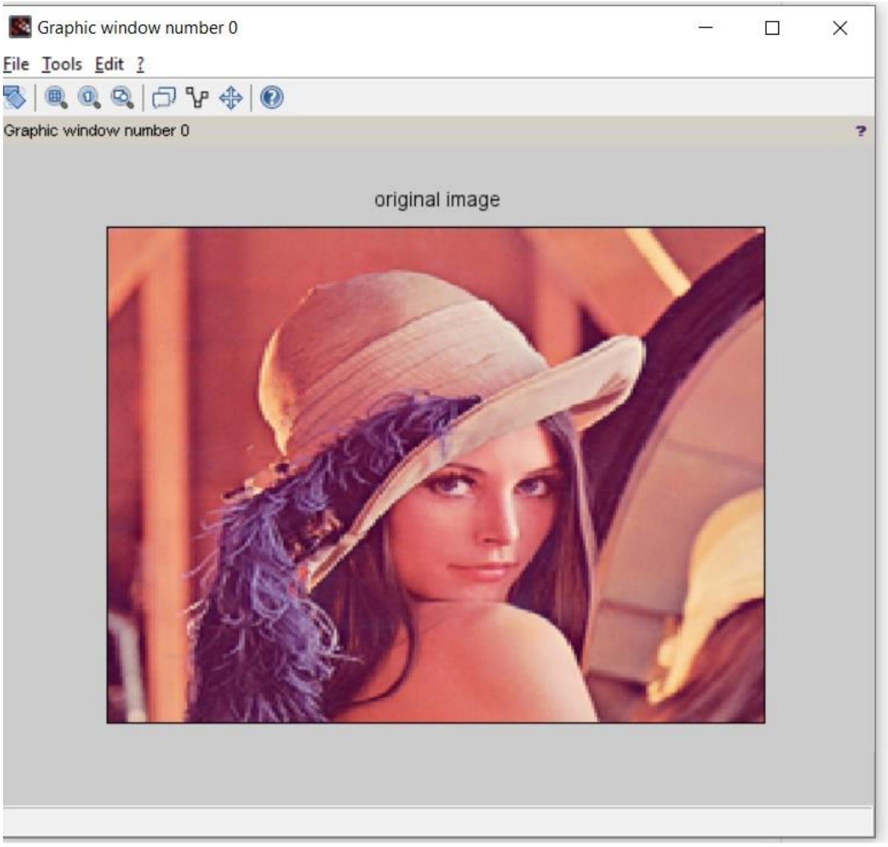
```
//Write a program to achieve Pseudo coloring.
clc;
clear all;
a=imread('D:\lena.png');
[l,m,n]=size(a);
for i=1:l
for j=1:m
for k=1:n
if a(i,j)>=0 & a(i,j) < 50
b(i,j,1)=a(i,j,1)+50;
b(i,j,2)=a(i,j,1)+100;
b(i,j,3)=a(i,j,1)+10;
end
if a(i,j)>=50 & a(i,j) < 100
b(i,j,1)=a(i,j,1)+35;
b(i,j,2)=a(i,j,1)+128;
b(i,j,3)=a(i,j,1)+10;
end
if a(i,j)>=100 & a(i,j) < 150
b(i,j,1)=a(i,j,1)+152;
b(i,j,2)=a(i,j,1)+130;
b(i,j,3)=a(i,j,1)+15;
end
if a(i,j)>=150 & a(i,j) < 200
b(i,j,1)=a(i,j,1)+50;
b(i,j,2)=a(i,j,1)+140;
b(i,j,3)=a(i,j,1)+25;
end
if a(i,j)>=200 & a(i,j) < 256
b(i,j,1)=a(i,j,1)+120;
b(i,j,2)=a(i,j,1)+160;
b(i,j,3)=a(i,j,1)+45;
end
end
end
end
subplot(1,2,1),imshow(a),title('Original');
subplot(1,2,2),imshow(b),title('Pseudo Image');
```

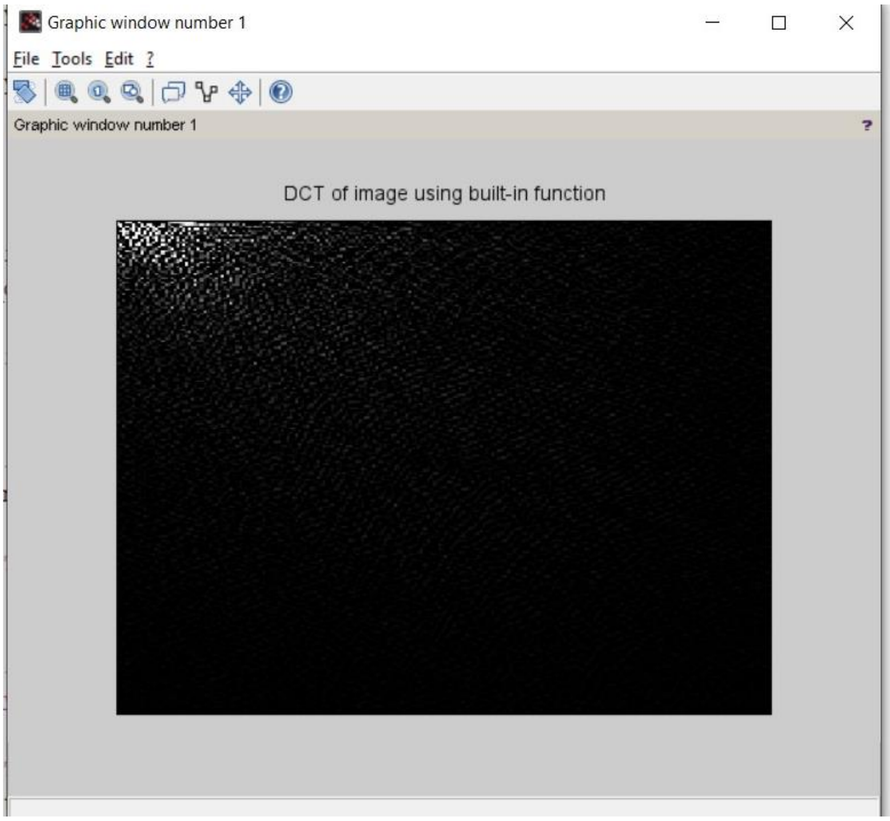

Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original
Pseudo Image

Practical No. 6 Fourier Related Transforms

6A. Program to compute Discrete Cosine Transforms

```
clear ;
clc ;
clear all;
close ;
img = imread ('D:\lena.png');
figure ();
xname ('original image');
imshow (img);
img_gray = rgb2gray ( img );
img_double = im2double ( img_gray );
// DCT of image using scilab function
img_dct = dct( img_double );
figure ();
xname ('DCT of image using buitin function');
imshow ( img_dct );
// Creating the Twiddle Factor Matrix c
[m,n]= size ( img_gray );
for x=1: m
for y =1: n
if x ==1 // for row number one
c(1,y)= sqrt (1/ m);
else
c(x,y) = sqrt (2/ m)*cos (( %pi *(2* y +1) *x)/(2* m));
end
end
end
// DCT of image using code
result = c * img_double * c' ;
figure ();
xname ('DCT of an image using code');
imshow ( result );
// Inverse DCT of image using scilab function
img_idct = idct ( img_dct );
figure ();
xname ('inverse DCT using buitin function');
imshow ( img_idct );
// Inverse DCT of image using code
result_idct = inv (c) * result * inv (c');
figure ();
xname ('inverse DCT using code');
imshow ( result_idct );
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
?
original image

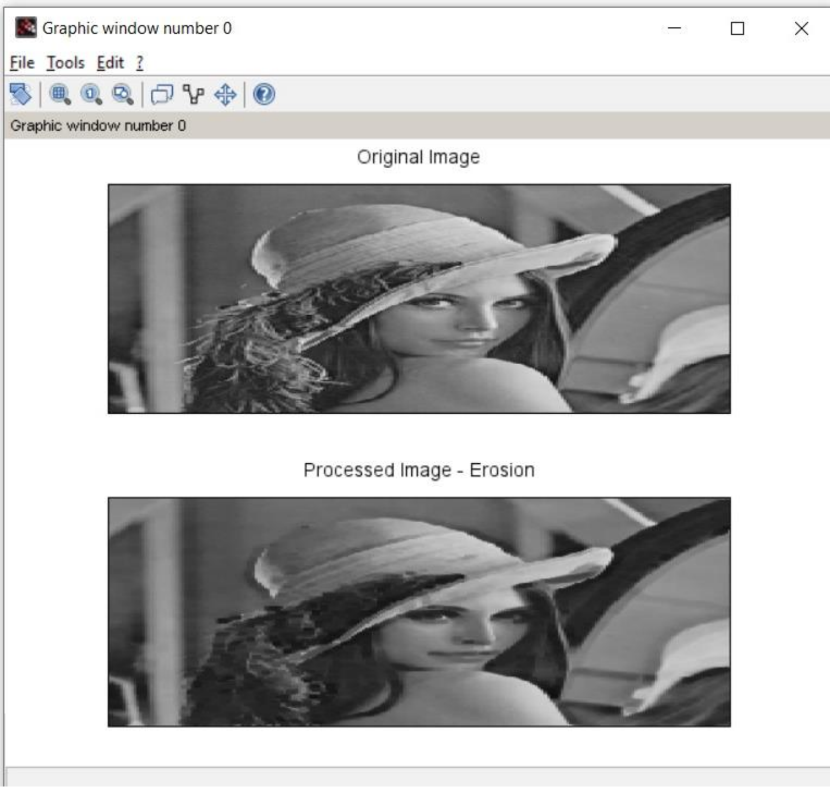

Graphic window number 1
File Tools Edit ?
Graphic window number 1
?
DCT of image using built-in function

Practical No . 7 Morphological Image Processing

7A. Program to apply erosion, dilation, opening, closing

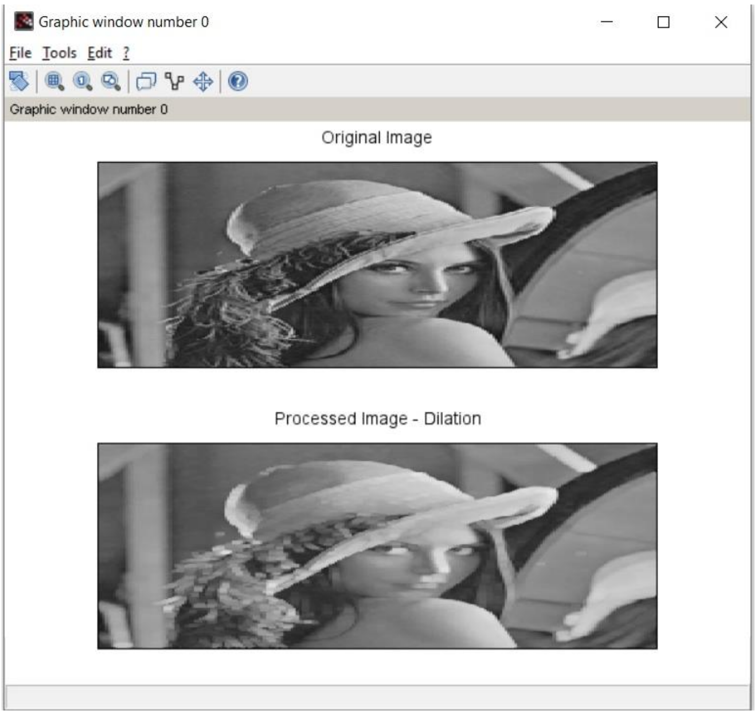
Erosion

```
clc;
clear all;
a=imread('D:\lena.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
A1=a;
d=a;
[r,c]=size(d);
m=[1 1 1;1 1 1;1 1 1];
// m=ones(5,5);
for i=2:1:r-1
for j=2:1:c-1
   new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1)) (m(5)*d(i,j))
(m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
A1(i,j)=min(new);
end
subplot(2,1,2);
title('org img');imshow(A1);title('Processed Image - Erosion');
end
```



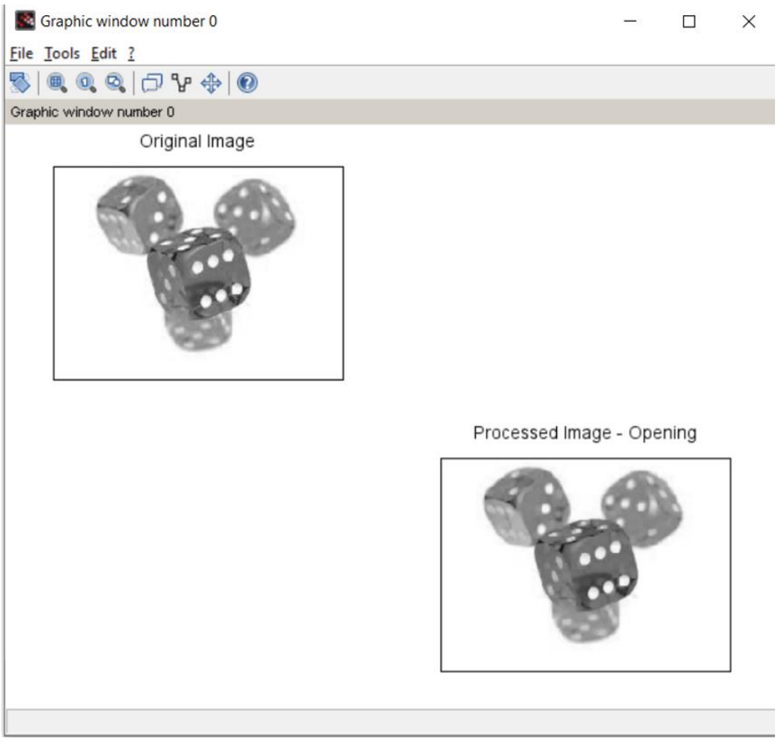
Dilation

```
clc;
clear all;
a=imread('D:\lena.png');
a=rgb2gray(a);
d=a;
A1=a;
[r,c]=size(d);
subplot(2,1,1);
imshow(a);
title('org img');
m=[1 1 1;1 1 1;1 1 1];
// m=ones(5,5);
for i=2:1:r-1
for j=2:1:c-1
new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
A1(i,j)=max(new);
end
subplot(2,1,2);
imshow(A1);title('Processed Image - dilation');
end
```



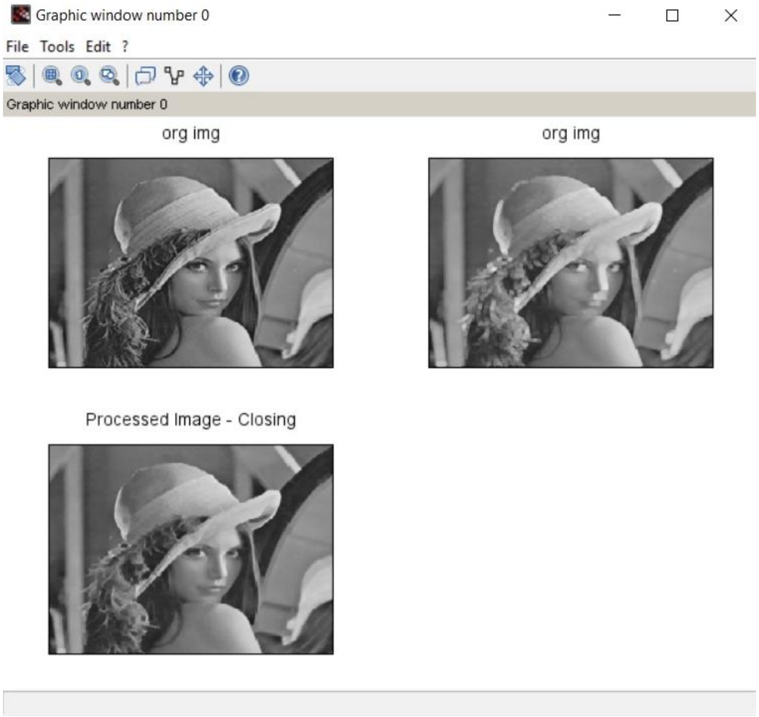
Opening

```
clc;
clear all;
a=imread("C:\Users\Dell\Downloads\dice.png");
a=rgb2gray(a);
d=a;
A2=d;
A1=d;
subplot(2,2,1);
imshow(a);
title('org img');
[r,c]=size(d);
m=[1 1 1;1 1 1;1 1 1];
for i=2:1:r-1
        for j=2:1:c-1
        new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1))
        (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1))
        (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
        A2(i,j)=min(new);
        end
end
subplot(2,2,2);
imshow(A2);
title('org img');
d = A2;
A1=A2;
[r,c]=size(d);
for i=2:1:r-1
        for j=2:1:c-1
        new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1))
        (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1))
        (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
        A1(i,j)=max(new);
        end
end
subplot(2,2,4);
imshow(A1);
title('Processed Image - Opening');
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original Image
Processed Image - Opening

Closing

```
clc;
clear all;
a=imread('D:\lena.jpeg');
a=rgb2gray(a);
d=a;
A2=d;
A1=d;
subplot(2,2,1);
imshow(a);
title('org img');
[r,c]=size(d);
m=[1 1 1;1 1 1;1 1 1];
for i=2:1:r-1
    for j=2:1:c-1
        new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1))
            (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1))
            (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
        A2(i,j)=max(new);
    end
end
subplot(2,2,2);
imshow(A2);
title('org img');

d = A2;
A1=A2;
[r,c]=size(d);
for i=2:1:r-1
    for j=2:1:c-1
        new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1))
            (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1))
            (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
        A1(i,j)=min(new);
    end
end
subplot(2,2,3);
imshow(A1);
title('Processed Image - Closing');
```


Graphic window number 0
File Tools Edit ?
Graphic window number 0
org img
org img
Processed Image - Closing

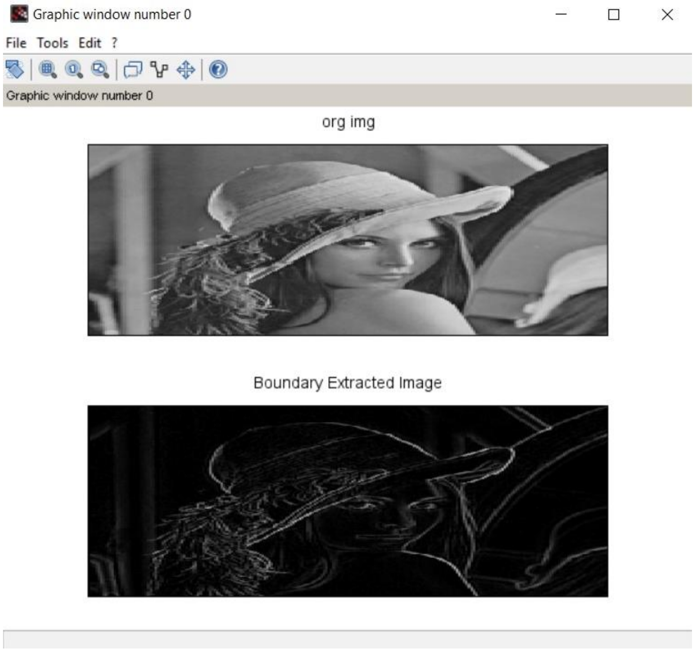
Practical No . 8 Image Segmentation

8A. Program for Edge detection using

a. Sobel, Prewitt, Canny and thresholding

BOUNDARY

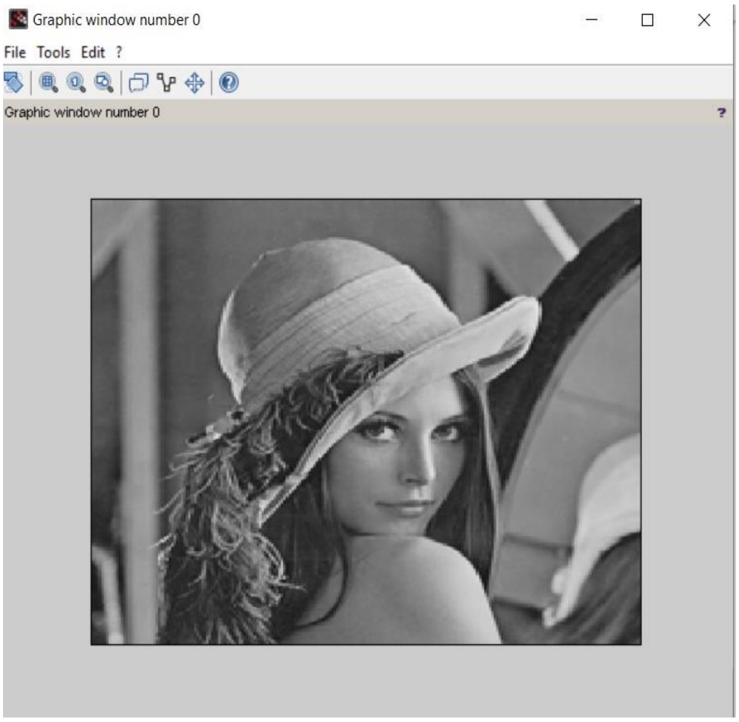
```
clc;
clear all;
a=imread('D:\lena.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
d=a;
[r,c]=size(d);
m=[1 1 1;1 1 1;1 1 1];
A2 = zeros(r, c);  % Initialize A2
for i=2:1:r-1
    for j=2:1:c-1
        new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1))
            (m(4)*d(i,j-1)) (m(5)*d(i,j)) (m(6)*d(i,j+1))
            (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
        A2(i,j)=min(new);
        aa(i,j)=d(i,j)-A2(i,j);
    end
end
subplot(2,1,2);
imshow(aa);
title('Boundary Extracted Image');
```

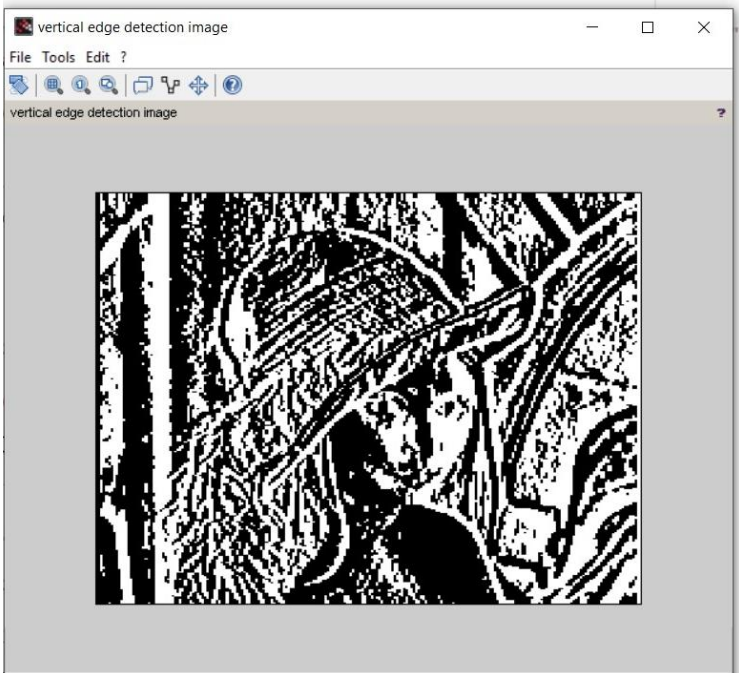


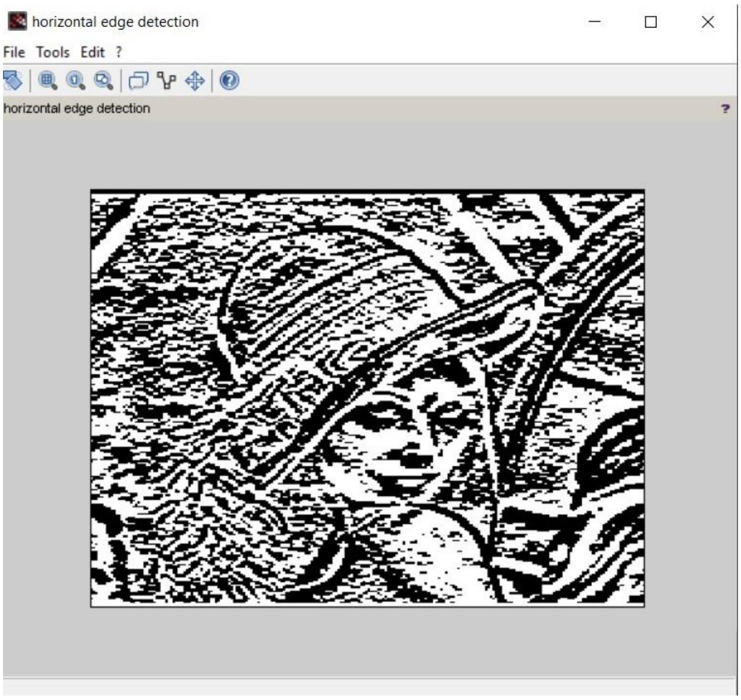
EDGE

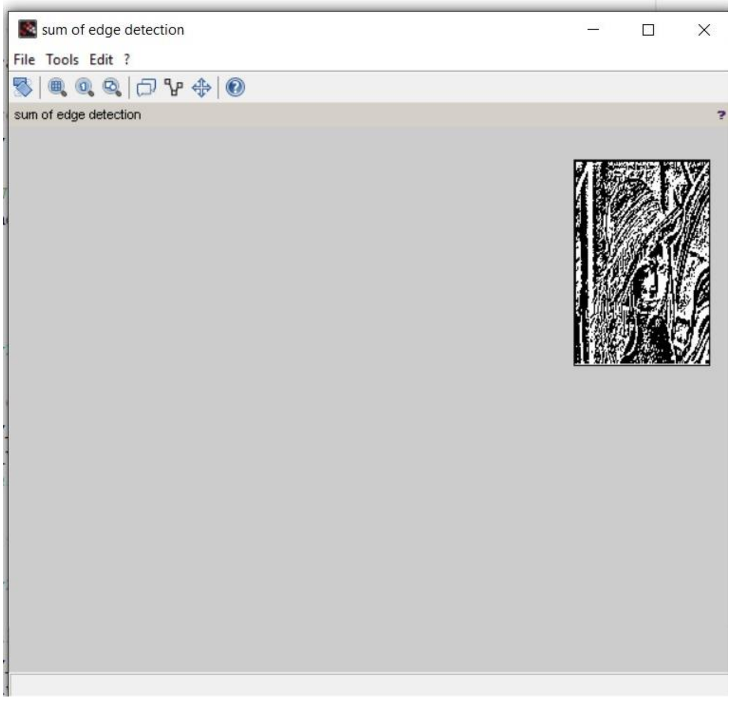
```
clc ;
clear all;
close ;

img = imread ('D:\lena.png');
img_gray = rgb2gray ( img );
figure ();
plot ('Input image 1');
imshow ( img_gray );

// EDGE DETECTION
[v,h] = size ( img_gray );

v_sobel = [-1, 0, 1; -2 ,0 ,2; -1 ,0 ,1];
disp ( v_sobel );

img_gray_v = conv2 ( double ( img_gray ), v_sobel );
figure ();
xname ('vertical edge detection image');
imshow ( img_gray_v );
imwrite(img_gray_v, 'vertical_edge_detection.jpg');
//imshow (vertical_edge_detection.jpg);

h_sobel = [-1, -2, -1; 0 ,0 ,0; 1 ,2 ,1];
disp ( h_sobel );
img_gray_h = conv2 ( double ( img_gray ), h_sobel );
figure ();
xname ('horizontal edge detection');
imshow ( img_gray_h );
```

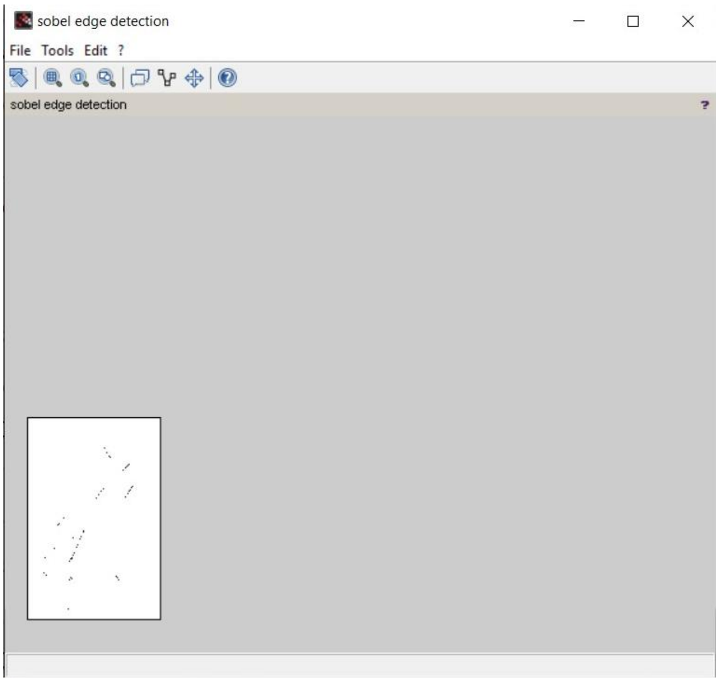
```
imwrite ( img_gray_h, 'horizontal_edge_dection.jpg');
//clf;
//imshow (horizontal_edge_dection.jpg);

img_res = img_gray_h + img_gray_v ;
figure ();
xname ('sum of edge detection');
subplot (2,4,4);
imshow ( img_res );
imwrite (img_res, 'sum_of_edge_detection.jpg');
//clf;
//imshow (sum_of_edge_detection.jpg);

// Edge De t e c t i o n u s i n g in−b u i l t f u n c t i o n s
E = edge (img_gray, 'sobel');
figure ();
xname ('sobel edge detection');
subplot (2,4,5);
imshow (E);
imwrite (E,'sobel.jpg');
//clf;
//imshow(sobel.jpg)

E2 = edge ( img_gray , 'canny' , [0.06 , 0.2]) ;
figure ();
xname ('Canny edge d e t e c t i o n');
subplot (2,4,6);
imshow (E2);
imwrite (E2,'canny.jpg')
//clf;
//imshow (canny.jpg)

E3 = edge ( img_gray , 'prewitt' );
figure ();
xname ('Pr ewi t t edge d e t e c t i o n');
imshow (E3);
imwrite (E3,'prewitt.jpg')
//clf;
//imshow(prewitt.jpg)

//THRESHOLDING
img_thresh = int ( img_gray /128) *255;
figure ();
plot ('Gl oba l Th r e s h o l d i n g');
imshow ( img_thresh );
imwrite ( img_thresh, 'threshhold.jpg');
//clf;
//imshow(threshhold.jpg);
```

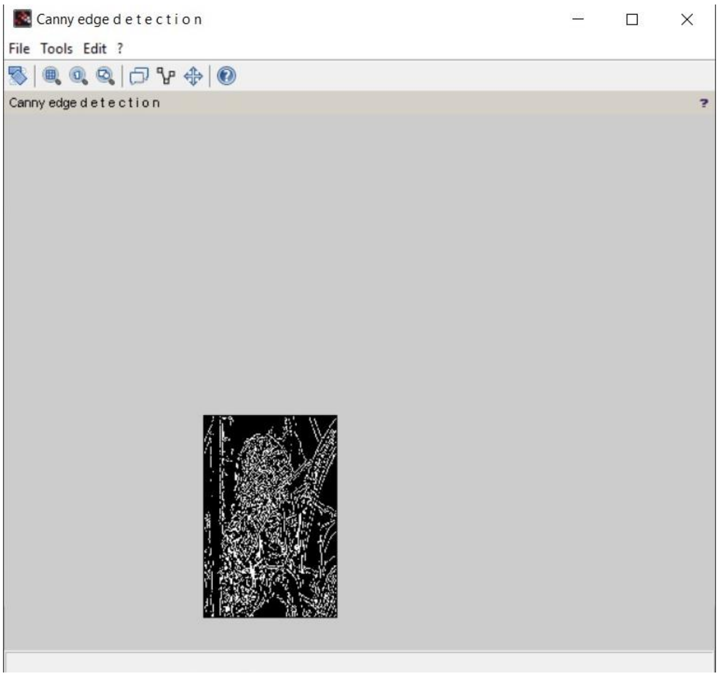

Graphic window number 0
File Tools Edit ?
Graphic window number 0
?

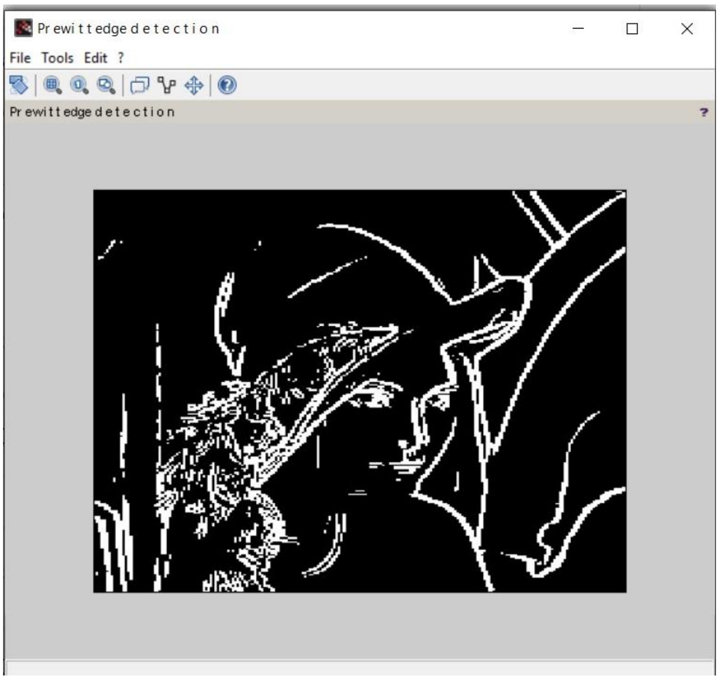

vertical edge detection image
File Tools Edit ?
vertical edge detection image
?

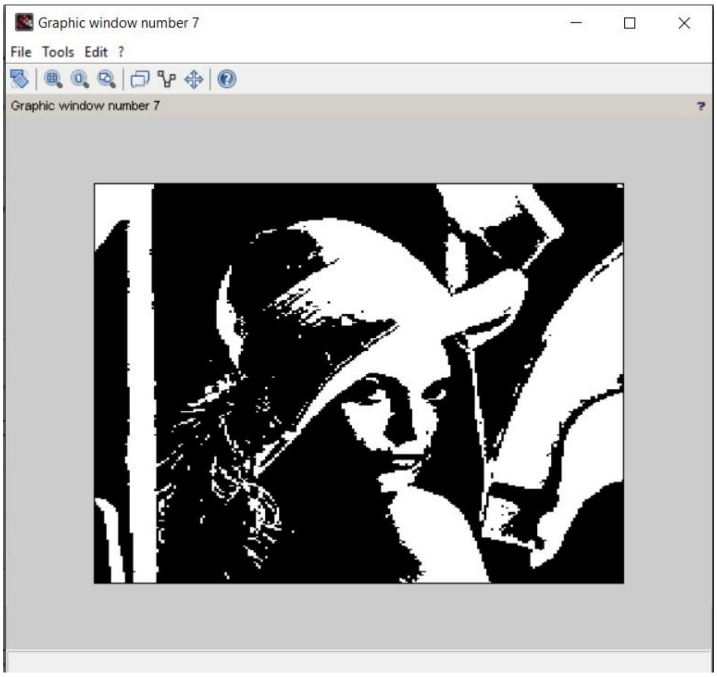

horizontal edge detection
File Tools Edit ?
horizontal edge detection
?


sum of edge detection
File Tools Edit ?
sum of edge detection
?


sobel edge detection
File Tools Edit ?
sobel edge detection
?


Canny edge detection
File Tools Edit ?
Canny edge detection
?


Prewitt edge detection
File Tools Edit ?
Prewitt edge detection
?


Graphic window number 7
File Tools Edit ?
Graphic window number 7
?