ImAGE   
PROCESSING

A PRACTICAL REPORT

ON

IMAGE PROCESSING

SUBMITTED BY  
Mr. **[FARHAN SAYED]**

Roll No: **[IT21015]**

UNDER THE GUIDANCE OF

PROF. AQSA ABBASI

Submitted in fulfillment of the requirements for qualifying

MSc. IT Part I Semester - II Examination 2021-2022

University of Mumbai

Department of Information Technology

R.D. & S.H National College of Arts, Commerce & W.A. Science College Bandra (West), Mumbai – 400 050



R. D. & S.H National & W. A. Science College

**Bandra (W), Mumbai – 400050*.***

**Department of Information Technology**

**M.Sc. (IT)**

**Certificate**

*This is to certify that* ***Image processing*** *performed at* ***R.D & S.H National & W.A. Science College*** *by Mr.* ***Farhan Sayed*** *holding Seat No.* ***IT21015*** *studying Master of Science in Information Technology Semester – II has been satisfactorily completed as prescribed by the University of Mumbai, during the year 2021 – 2022.*

|  |  |  |
| --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Lecturer In charge** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **External Examiner** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Head of Department** |

**College Stamp**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Date** | **Practical** | **Page No.** | **Sign** |
| 1 |  | **Practical 1**   1. Program to calculate number of samples required for an image. 2. Program to study the effects of reducing the spatial resolution of a digital image. 3. Program to study the effects of varying the number of intensity levels in a digital image |  |  |
| 2 |  | **Practical 2**  A1) Program to perform image negation  A2) Perform threshold on an image  A3) Program to perform Log Transformation.  A4) Program to perform Power Law Transformation / Gamma Correction  A5.1) Programs to perform Piecewise Linear transformations A5.1 Contrast Stretching  A5.2a) Gray Level Slicing With Background Input File : Cameraman.tiff  A5.2b) Gray Level Slicing Without Background Input File : Baloons.png  A5.3) Bit plane Slicing Input File : coins.jpg  B1) Program to create Histogram of an Image Input File : Cameraman.tiff  B2) Histogram Equalization Input File : Cameraman.tiff  2 C) Program for Convolution, Convolution 2D, Correlation  2 D) Write a program to apply smoothing and sharpening filters on grayscale and color images 1. Low Pass Average Filter Input file : coins.jpg  2.2) Low Pass Median Filter  2.3) High Pass Filtering |  |  |
| 3 |  | **Practical 3**    3a) Filtering in Frequency Domain Program to apply Low pass and High pass filters in frequency domain Ideal Low Pass Filter  3b) Gaussian Low Pass Filter  3c) High Pass Filter  3d) Program for high frequency emphasis filtering, high boost and homomorphic filtering |  |  |
| 4 |  | **Practical 4**  4a) Image Denoising Program to denoise using spatial mean, median filtering  4b) Median filter  4c) Program for Image deblurring using inverse, Weiner filters |  |  |
| 5 |  | **Practical 5**  5a) Color Image Processing Program to read a color image and segment into RGB planes , histogram of color image  5b) Program for converting from one color model to another model RGB to CMY  5c) Program to apply false colouring(pseudo) on a gray scale image |  |  |
| 6 |  | **Practical 8**  8a) Practical 8 Morphological Image Processing 8a Program to apply erosion, dilation, and morphological gradient on an image  8b) Program for detecting boundary of an image  8c) Program to apply opening, closing Opening  8d) Program for detecting boundary of an image |  |  |
| 7 |  | **Practical 9**  9a) Practical 9 Image Segmentation Program for Edge detection using Sobel  9b) Prewitt  9c) Canny |  |  |

**Practical No. 1**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 1A**

**Aim:-** Program to calculate number of samples required for an image.

**Code Implementation:- (Include steps)**

clc; close;

//dimension of the image in inches

m = 4;

n = 6;

N = 400; //number of dots per inch in each direction

N2 = 2\*N; //number of dots per inch in both horizontal & vertical

Fs = m\*N2\*n\*N2;

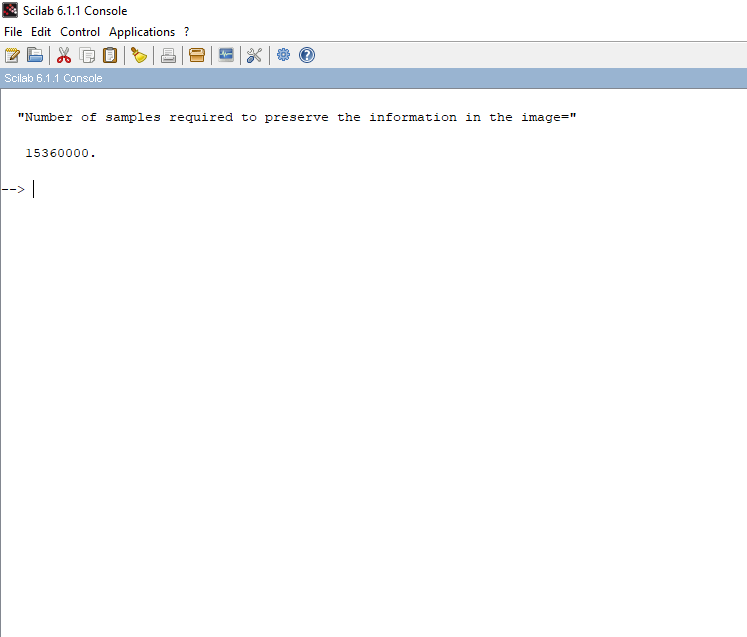
disp('Number of samples required to preserve the information in the image=',Fs)

//Result

//Number of samples required to preserve the information in the image=

//15360000.

**Output:-**



**Practical No. 1b**

**Aim:-** Program to study the effects of reducing the spatial resolution of a digital image.

**Code Implementation:- (Include steps)**

//Program to study the effects of reducing the spatial resolution of a digital image

clc; clear all;

figure(1);

//checker board effect sampling

subplot(3,3,1);

i=imread('C:\Program Files\scilab-6.1.0\IPCV\images\lena.png'); imshow(i);

title('og image');

subplot(3,3,2) j=imresize(i,0.8); imshow(j); title('resized image0.8')

subplot(3,3,3) j=imresize(i,0.7); imshow(j); title('resized image0.7')

subplot(3,3,4) j=imresize(i,0.6); imshow(j);

title('resize image 0.6')

subplot(3,3,5) j=imresize(i,0.5); imshow(j);

title('resize image 0.5')

subplot(3,3,6) j=imresize(i,0.4); imshow(j);

title('resize image 0.4')

subplot(3,3,7) j=imresize(i,0.3); imshow(j);

title('resize image 0.3')

subplot(3,3,8) j=imresize(i,0.2); imshow(j);

title('resize image 0.2')

OR

//Program to study the effects of reducing the spatial resolution of a digital image

clc; clear all;

Img = imread( 'C:\Program Files\scilab-6.1.0\IPCV\images\lena.png' ); subplot (2,2,1), imshow(Img), title( 'ORGINAL IMAGE 512 x 512' ); Samp = zeros (256);

for i = 1 : 1 : 512

for j = 1 : 1 : 512

if modulo(i,2) = = 0 m = i / 2;

if modulo ( j ,2 ) = = 0 n = j / 2 ;

Samp ( i-m , j-n ) = Img( i, j ); else

n = 0;

end else

m = 0;

end;

end

end

SampImg256 = mat2gray ( Samp );

subplot(2,2,2), imshow(SampImg256), title('SAMPLED IMAGE 256 x 256'); Samp = zeros ( 128 );

for i = 1 : 1 : 512

for j = 1 : 1 : 512

if modulo ( i , 4) = = 0 m = i / 4 \* 3;

if modulo ( j , 4 ) = = 0 n = j / 4 \* 3 ;

Samp( i-m, j-n ) = Img ( i , j ) ; else

n = 0 ;

end else

m = 0;

end

end

end

SampImg128 = mat2gray ( Samp );

subplot(2,2,3), imshow(SampImg128), title('SAMPLED IMAGE 128 x 128'); Samp = zeros( 64 );

for i = 1 : 1 : 512

for j = 1 : 1 : 512

if modulo( i , 8 ) = = 0 m = i / 8 \* 7 ;

if modulo ( j ,8 ) = = 0 n = j / 8 \* 7 ;

Samp ( i-m , j-n ) = Img ( i , j ) ; else

n = 0;

end; else

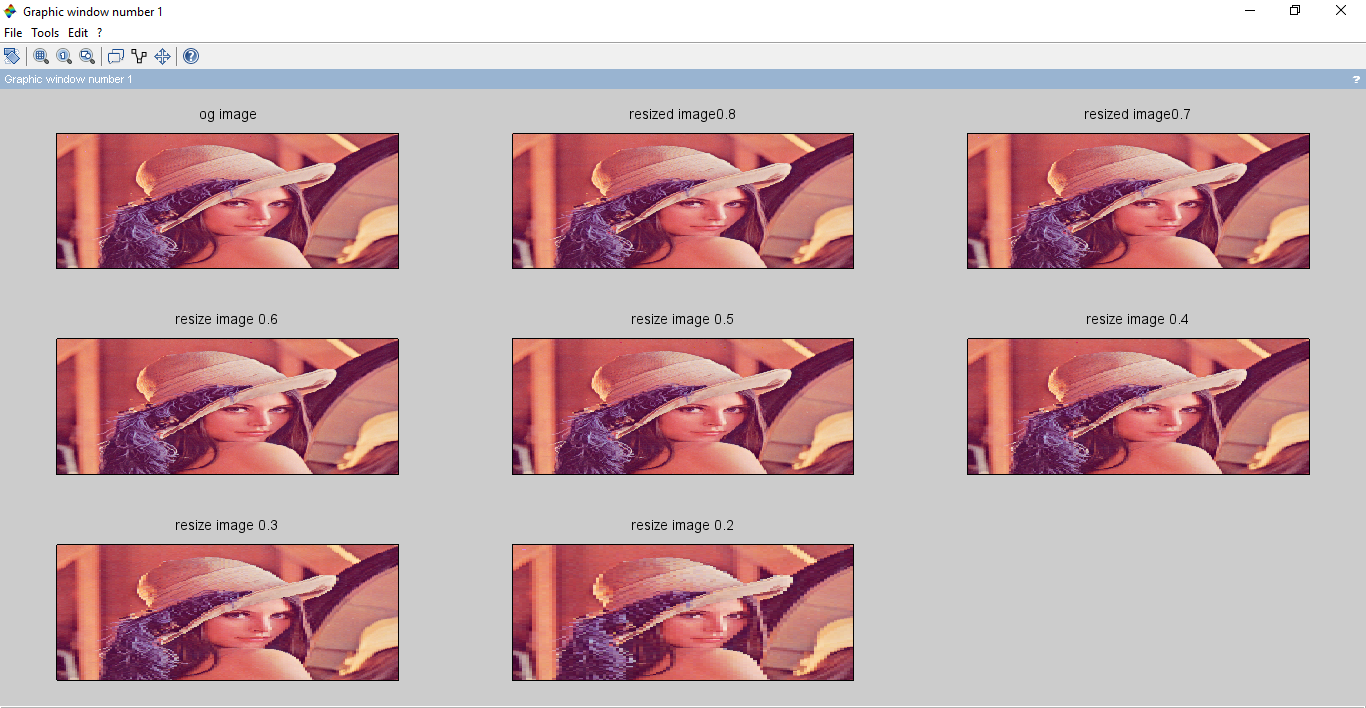
m = 0;

end end end

SampImg64 = mat2gray ( Samp );

subplot(2,2,4), imshow(SampImg64), title('SAMPLED IMAGE 64 x 64');

**Output:-**



**Practical No. 1c**

**Aim:-** Program to study the effects of varying the number of intensity levels in a digital image

**Code Implementation:- (Include steps)**

//Quantization

clc; clear all;

subplot(3,3,1);

i=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png'); i=double(i);

imshow(uint8(i)); title('og image')

k=floor(i\*255)/128; subplot(3,3,2); imshow(uint8(k)); title('quantized image 128')

k=floor(i\*255)/64; subplot(3,3,3); imshow(uint8(k)); title('quantized image 64')

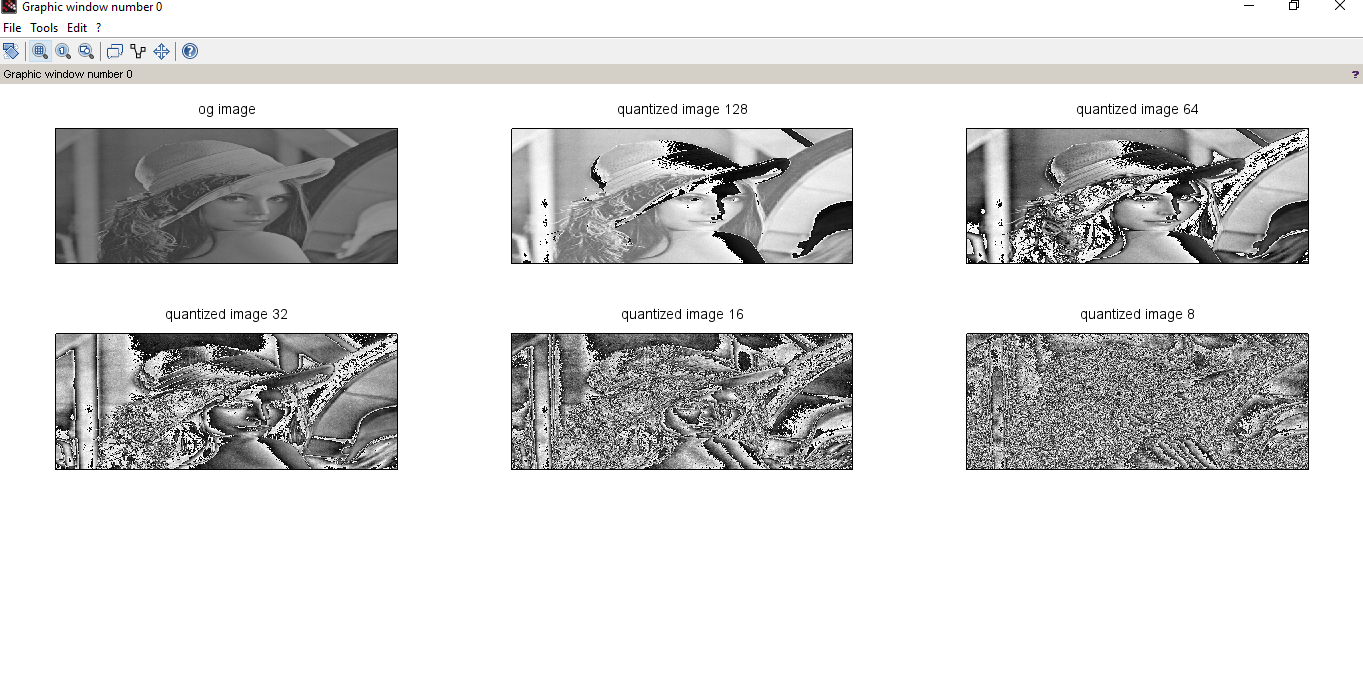
k=floor(i\*255)/32; subplot(3,3,4); imshow(uint8(k)); title('quantized image 32')

k=floor(i\*255)/16; subplot(3,3,5); imshow(uint8(k)); title('quantized image 16')

k=floor(i\*255)/8;

subplot(3,3,6); imshow(uint8(k)); title('quantized image 8')

**Output:-**



**Practical No. 2**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 2A1**

**Aim:-** Program to perform image negation

**Code Implementation:- (Include steps)**

Input file: cameraman.tiff

Code :

// Image Enhancement in the Spatial Domain

//Image Negative

clc; clear all;

i=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

disp("Practical 2A : 1 Image Negation")

a=double(i);

c=255;

b=c-a;

subplot(1,2,1);

imshow(uint8(a));

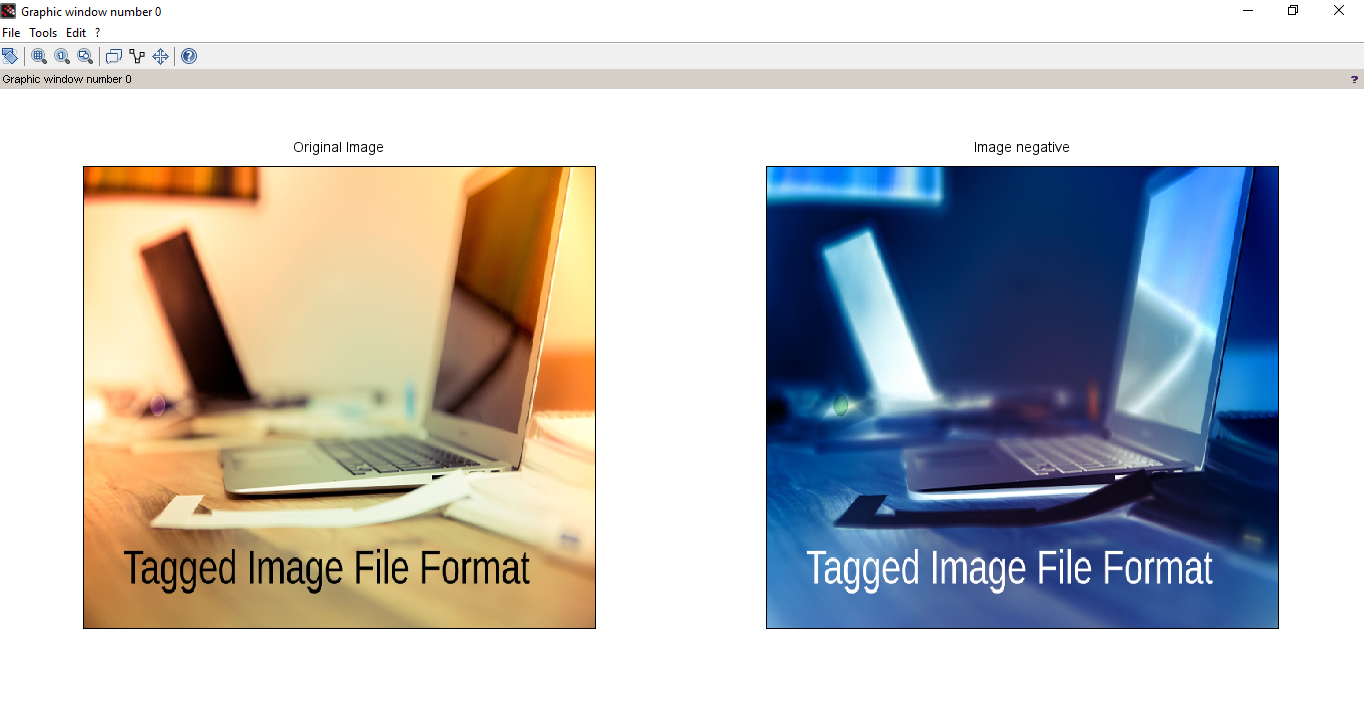
title('Original Image');

subplot(1,2,2);

imshow(uint8(b));

title('Image negative');

**Output:-**



**Practical No. 2A2**

**Aim:-** Perform threshold on an image

**Code Implementation:- (Include steps)**

Input file: cameraman.tiff

Code :

// Image Enhancement in the Spatial Domain

// Thresholding

clc, clear all;

p=imread('C:\Program Files\scilab-6.1.0\IPCV\images\balloons.png);

printf "Practical 2A : 2 Image Thresholding"

a=p;

[row col]=size(a);

T=input('Enter value of 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')

**Output:-**



**Practical No. 2A3**

**Aim:-** Program to perform Log Transformation.

**Code Implementation:- (Include steps)**

Input File : Balloons.png

Code :

// Log transformation

clc;

clear all;

Img2 = imread('C:\Program Files\scilab-6.1.0\IPCV\images\balloons.png');

printf "Practical 2A : 3 Log Transformation"

L = 255;

C = L / log ( 1 + L );

//display( C );

S = C \* log ( 1 + double ( Img2 ) );

subplot(1,2,1),

imshow(Img2),

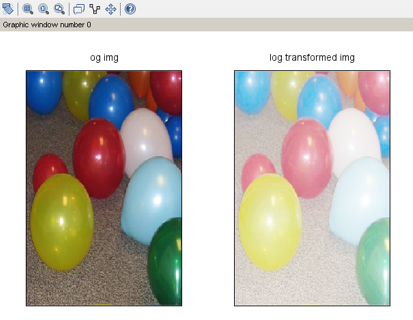
title('ORGINAL IMAGE');

subplot(1,2,2);

imshow(uint8 ( S ));

title('LOG TRANSFORMED IMAGE');

**Output:-**



**Practical No. 2A4**

**Aim:-** Program to perform Power Law Transformation / Gamma Correction

**Code Implementation:- (Include steps)**

Input File : Balloons.png

Code :

//power law transformation

clc;clear;

Img3 = imread('C:\Program Files\scilab-6.1.0\IPCV\images\balloons.png');

printf "Practical 2A : 4 Power Law Transformation"

r=double(Img3)/255;

c = 1;

gamma=1.8;

//S = C \*.^( double(Img3) , 1 );

s =c.\*(r).^gamma;

subplot(1,2,1),

imshow(Img3);

title('ORGINAL IMAGE');

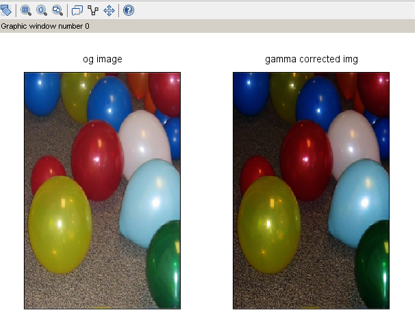
subplot(1,2,2);

imshow(s);

title('GAMMA CORRECTED IMAGE');

// to make image dark take value of gamma > 1, to make image bright take vlue of gamma < 1

**Output:-**



**Practical No. 2A5.1**

**Aim:-** Programs to perform Piecewise Linear transformations A5.1 Contrast Stretching

**Code Implementation:- (Include steps)**

Input File : girl.jpg

Code :

// Image Enhancement in the Spatial Domain

// Contrast Stretching

clc; clear;

disp("Contrast Stretching")

a=uigetfile('\*.\*','Select the Image:-'); a=imread(a);

a1=min(min(a)); b=max(max(a));

l=0.5; n=0.5; m=3; v=l\*a1; w=v+(m\*(b-a1));

[r,c]=size(a);

for i=1:r

for j=1:c

if (a(i,j)<a1)

new(i,j)=l\*a(i,j);

elseif (a(i,j)>=a1 & a(i,j)<b)

new(i,j)=(m\*(a(i,j)-a1))+v;

else

new(i,j)=(n\*(a(i,j)-b))+w;

end

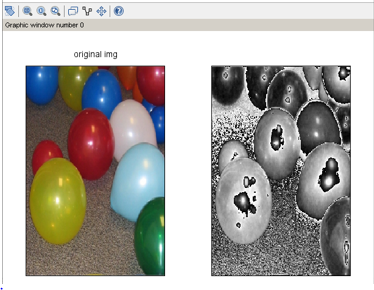
end

end

subplot(1,2,1);imshow(a);title("Original Image");

subplot(1,2,2);new=double(new);imshow(uint8(new));title("After contrast stretching");

**Output:-**



**Practical No. 2A5.2a**

**Aim:-** Gray Level Slicing With Background Input File : Cameraman.tiff

**Code Implementation:- (Include steps)**

Code :

// Image Enhancement in the Spatial Domain

// Grey level slicing with background

clc, clear all;

printf "Practical 2A : 5 .2 Gray Level Slicing With Background"

p=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

z=double(p);

[row col]=size(z);

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(1,2,1); imshow(p); title('Original Image');

subplot(1,2,2); imshow(uint8(z)); title('Grey level slicing with background');

**Output:-**



**Practical No. 2A5.2b**

**Aim:-** Gray Level Slicing Without Background Input File : Baloons.png

**Code Implementation:- (Include steps)**

Code :

// Image Enhancement in the Spatial Domain Gray level slicing without background

clc, clear all;

printf "Practical 2A : 5 . Gray Level Slicing Without Background"

p=imread('C:\Program Files\scilab-6.1.0\IPCV\images\balloons.png);

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

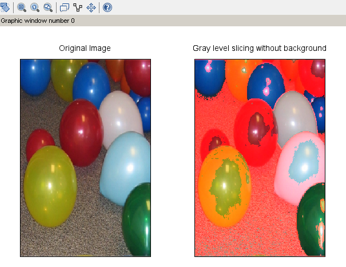
end

end

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

subplot(1,2,2); imshow(uint8(z));title('Gray level slicing without background');

**Output:-**



**Practical No. 2A5.3**

**Aim:-** Bit plane Slicing Input File : coins.jpg

**Code Implementation:- (Include steps)**

Code :

//Bit Plane Slicing

clc; clear; B=imread('C:\Program Files\scilab-6.1.0\IPCV\images\coins.png');

[r,c]=size(double(B)); disp("Practical 2A 5C")

for i=1:r

for j=1:c

MSB(i,j)=bitand(B(i,j),bin2dec('10000000'));

LSB(i,j)=bitand(B(i,j),bin2dec('00000001'));

Second(i,j)=bitand(B(i,j),bin2dec('01000000'));

Third(i,j)=bitand(B(i,j),bin2dec('00100000'));

Fourth(i,j)=bitand(B(i,j),bin2dec('00010000'));

Fifth(i,j)=bitand(B(i,j),bin2dec('00001000'));

Sixth(i,j)=bitand(B(i,j),bin2dec('00000100'));

Seventh(i,j)=bitand(B(i,j),bin2dec('00000010'));

end

end

subplot(4,4,1); imshow(MSB); title("Bit Plane 7");

subplot(4,4,2); imshow(Second); title("Bit Plane 6");

subplot(4,4,3); imshow(Third); title("Bit Plane 5");

subplot(4,4,4); imshow(Fourth); title("Bit Plane 4");

subplot(4,4,5); imshow(Fifth); title("Bit Plane 3");

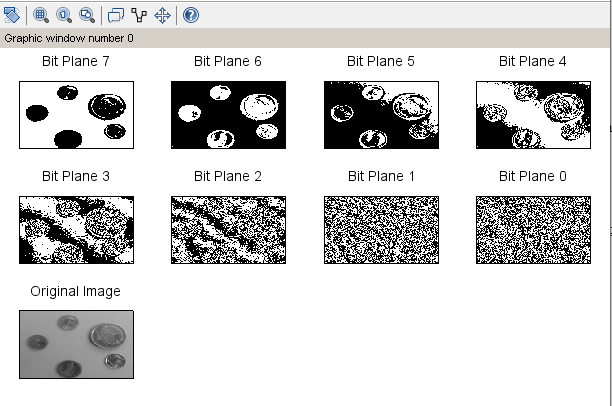
subplot(4,4,6); imshow(Sixth); title("Bit Plane 2");

subplot(4,4,7); imshow(Seventh); title("Bit Plane 1");

subplot(4,4,8); imshow(LSB); title("Bit Plane 0");

subplot(4,4,9); imshow(B); title("Original Image");

**Output:-**



**Practical No. 2B.1**

**Aim:-** Program to create Histogram of an Image Input File : Cameraman.tiff

**Code Implementation:- (Include steps)**

Code :

clc; clear;

a=imread("C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff");

disp("Practical 2B : 1 Histogram of an Image")

subplot(1,2,1); imshow(a); title('Image');

r=size(a,1); c=size(a,2); h=zeros(1,256);

for i=1:r

for j=1:c

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

a(i,j)=1;

end

k=a(i,j);

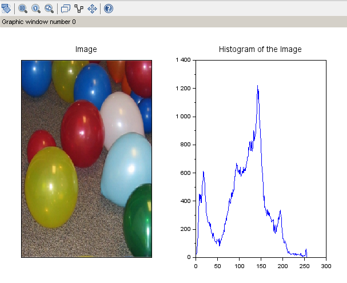
h(k)=h(k)+1;

end

end

subplot(1,2,2); plot(h); title('Histogram of the Image');

**Output:-**



**Practical No. 2B.2**

**Aim:-** Histogram Equalization Input File : Cameraman.tiff

**Code Implementation:- (Include steps)**

Code :

clc;clear;

a=imread("C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff");

subplot(2,2,1);

imshow(a)

title("Original Image")

r=size(a,1);

c=size(a,2);

h=zeros(1,256);

for i=1:r

for j=1:c

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

a(i,j)=1;

end

k=a(i,j);

h(k)=h(k)+1;

end

end

subplot(2,2,2);

plot(h);

title("Histogram of Original Image")

disp("Practical 2B : 2 Histogram Equalization")

a=double(a);

big=max(max(a));

[r c]=size(a);

tot = r\*c; h=zeros(1,256); // to store the histogram values

z=zeros(1,256);

for i=1:1:r

for j=1:1:c

if a(i,j)==0

a(i,j)=1;

end

end

end

for i=1:1:r

for j=1:1:c

t=a(i,j);

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

end

end

pdf=h/tot;

cdf(1)=pdf(1);

for i=2:1:big

cdf(i)=pdf(i)+cdf(i-1);

end

new=round(cdf\*big);

new=new+1;

for i=1:1:r

for j=1:1:c

temp=a(i,j);

b(i,j)=new(temp);

t=b(i,j);

z(t)=z(t)+1;

end

end

subplot(2,2,4);

plot(z);

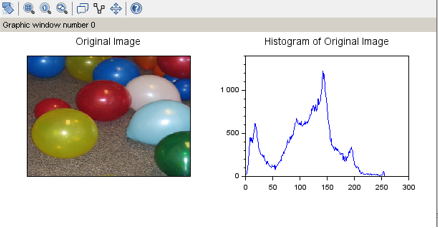
title('Equalized Histogram');

subplot(2,2,3)

imshow(uint8(z))

title('Equalized image');

**Output:-**



**Practical No. 2C**

**Aim:-** Program for Convolution, Convolution 2D, Correlation

**Code Implementation:- (Include steps)**

clc; disp("Practical 2C 1 Convolution")

x =[4,5,6;7,8,9]; h = [1;1;1]; y= conv2(x,h); disp(y);

disp("Practical 2C 2DConvolution")

a =[1,2,3;4,5,6;7,8,9]; b = [1,1;1,1;1,1]; c = convol2d(a,b); disp(c);

disp("Practical 2C Correlation")

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

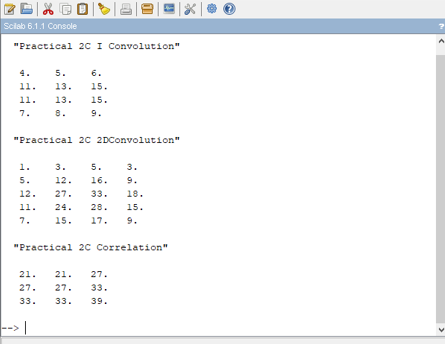
h = [1,1;1,1;1,1];

y = corr2(x,h);

disp(y);­­­­­­

**Output:-**

­­­­­­



**Practical No. 2D**

**Aim:-** Write a program to apply smoothing and sharpening filters on grayscale and color images 1. Low Pass Average Filter Input file : coins.jpg

**Code Implementation:- (Include steps)**

Code :

clc; clear;

disp("Practical 2D A Low Pass Average Filtering")

a=uigetfile('\*.\*','Select the Image:-');

a=imread(a); b=double(a); c=imnoise(a,'salt & pepper',0.2); d=double(c); m=(1/9)\*(ones(3,3));

[r1,c1]=size(a);

for i=1:r1

for j=1:c1

new(i,j)=a(i,j);

end

end

for i=2:1:r1-1

for j=2:1:c1-1

new(i,j)=(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));

end

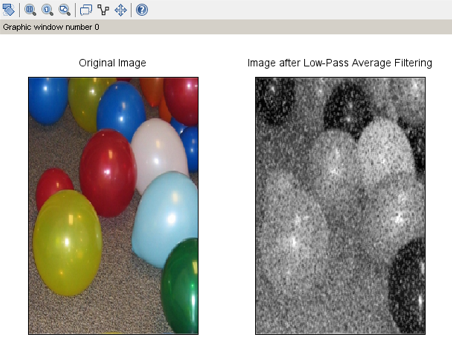
end

subplot(1,2,1); imshow(a); title("Original Image");

subplot(1,2,2); imshow(uint8(new)); title("Image after Low-Pass Average Filtering")

­­­­

**Output:-**



**Practical No. 2.2**

**Aim:-** Low Pass Median Filter

**Code Implementation:- (Include steps)**

Code :

clc;clear;

disp("Practical 2D A Low Pass Median Filtering")

a=uigetfile('\*.\*','Select the Image:-');

a=imread(a); b=double(a); c=imnoise(a,'salt & pepper',0.2); d=double(c); m=(1/9)\*(ones(3,3));

[r1,c1]=size(a);

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)];

a2=gsort(a1);

med=a2(4);

b(i,j)=med;

end

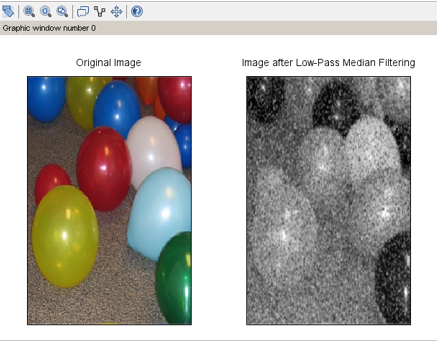
end

subplot(1,2,1); imshow(a); title("Original Image");

subplot(1,2,2); imshow(uint8(b)); title("Image after Low-Pass Median Filtering")

­­

**Output:-**



**Practical No. 2.3**

**Aim:-** High Pass Filtering

**Code Implementation:- (Include steps)**

Code :

clc;clear;

disp("Practical 2D B High Pass Filtering")

a=uigetfile('\*.\*','Select the Image:-');

a=imread(a); d=double(a);

[r1,c1]=size(a);

m=[-1 -1 -1;-1 8 -1;-1 -1 -1];

for i=2:1:r1-1

for j=2:1:c1-1

new(i,j)=(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));

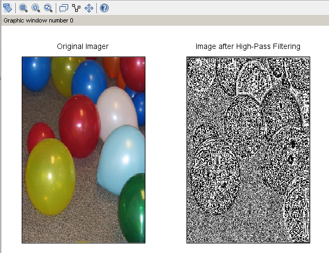
end

end

subplot(1,2,1); imshow(a); title("Original Image");

subplot(1,2,2); imshow(uint8(new)); title("Image after High-Pass Filtering")

**Output:-**



**Practical No. 3**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 3a**

**Aim:-** Filtering in Frequency Domain Program to apply Low pass and High pass filters in frequency domain Ideal Low Pass Filter

**Code Implementation:- (Include steps)**

clear all;

clc;

img=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

a=double(img);

r=size(a,1);

c=size(a,2);

d0=input('Enter the cut-off frequency - (Radius):-');

for u=1:1:r

for v=1:1:c

d=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5;

if d<=d0

h(u,v)=1;

else

h(u,v)=0;

end

end

end

b=fft2(a);

p\_original=(abs(b))^2 + (atan(imag(b),real(b)))^2;

p\_original=sum(sum(p\_original));

c=fftshift(b);

c1=uint16(c);

new=c.\*h;

new2=uint8(new);

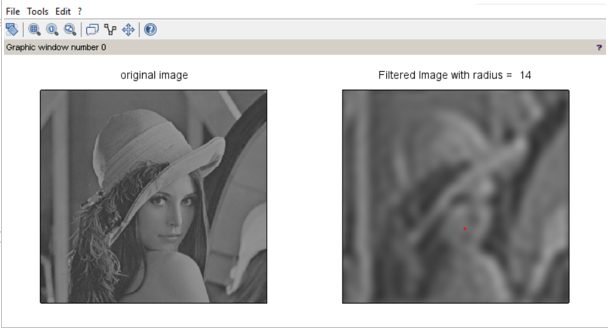
new1=abs(ifft(new));

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

subplot(1,2,2);

imshow(uint8(new1));title(['Filtered Image with radius = ',string(d0)]);

**Output:-**



**Practical No. 3b**

**Aim:-** Gaussian Low Pass Filter

**Code Implementation:- (Include steps)**

clear all;

clc;

img=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

a=double(img);

r=size(a,1);

c=size(a,2);

d0=input('Enter the cut-off frequency - (Radius):-');

for u=1:1:r

for v=1:1:c

d=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5;

dd=d\*d;

h(u,v)=exp(-dd/(2\*d0\*d0));

end

end

b=fft2(double(a));

c=fftshift(b);

c1=uint16(c);

new=c.\*h;

new2=uint16(new);

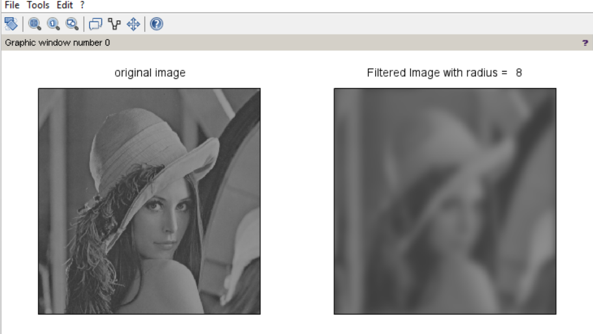
new1=abs(ifft(new));

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

subplot(1,2,2);

imshow(uint8(new1));title(['Filtered Image with radius = ',string(d0)]);

**Output:-**



**Practical No. 3c**

**Aim:-** High Pass Filter

**Code Implementation:- (Include steps)**

clear all;

clc;

a=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

r=size(a,1);

c=size(a,2);

d0=input('Enter the cut-off frequency - (Radius):-');

for u=1:1:r

for v=1:1:c

d=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5;

if d<=d0

h(u,v)=0;

else

h(u,v)=1;

end

end

end

b=fft2(double(a));

c=fftshift(b);

c1=uint16(c);

new=c.\*h;

new2=uint16(new);

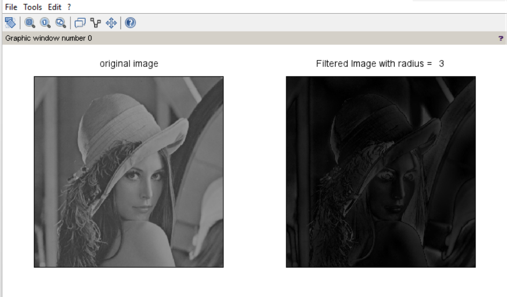
new1=abs(ifft(new));

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

subplot(1,2,2);

imshow(uint8(new1));title(['Filtered Image with radius = ',string(d0)]);

**Output:-**



**Practical No. 3d**

**Aim:-** Program for high frequency emphasis filtering, high boost and homomorphic filtering.

**Code Implementation:- (Include steps)**

clear all;

clc;

I=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

GM=im2double(I);

t = 512;

p = 1/t:1/t:1;

for k = 1:t

l(k,:)=p(:);

end

product = GM.\*l;

log\_GM = log(GM + 1);

log\_product = log(product + 1);

fft\_GM = (fftshift(fft2(log\_GM)));

fft\_product = (fftshift(fft2(log\_product)));

diff\_ill = (fft\_product - fft\_GM);

restored = (fft\_product - diff\_ill);

restored\_im = ifft(restored);

restored\_image = (exp(restored\_im)) - 1;

abs\_restored\_image = abs(restored\_image);

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

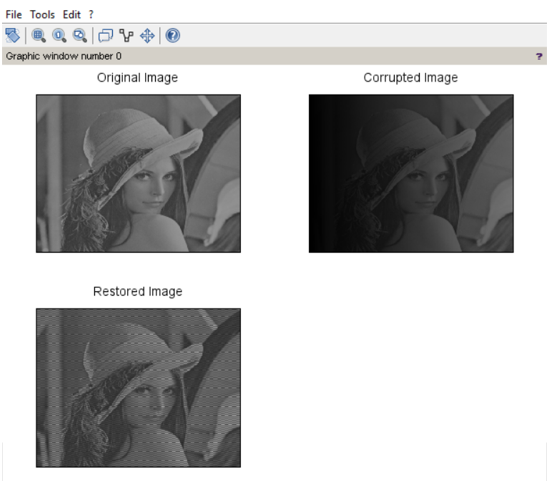
subplot(2,2,2);

imshow(log\_product);title(' Corrupted Image');

subplot(2,2,3);

imshow(abs\_restored\_image);title('Restored Image');

**Output:-**



**Practical No. 4**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 4A**

**Aim:-**  Image Denoising Program to denoise using spatial mean, median filtering

Mean filter

**Code Implementation:- (Include steps)**

clc; clear all;

a=uigetfile('\*.\*','Select the Image:-'); a=imread(a);

b1=double(a); c=imnoise(a,'gaussian'); d=double(c);

b=d; m=(1/9)\*(ones(3,3));

[r1,c1]=size(a); 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

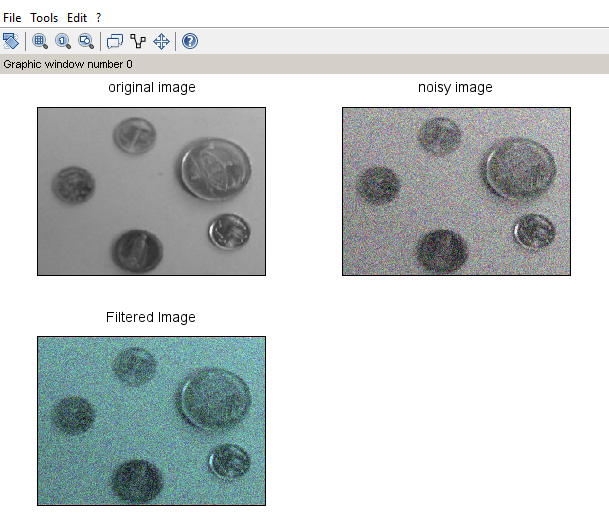
end subplot(2,2,1); imshow(uint8(a));

title('original image')

subplot(2,2,2); imshow(uint8(c)); title('noisy image')

subplot(2,2,3); imshow(uint8(b)); title('Filtered Image');

**Output:-**



**Practical No. 4B**

**Aim:-**  Median filter

**Code Implementation:- (Include steps)**

clc; clear all;

a = uigetfile("\*.\*","Select the Image:-"); a = imread(a);

b1 = double(mtlb\_double(a));

c = imnoise(a,"salt & pepper",0.2);

d = double(mtlb\_double(c)); b = d;

m = (1/9)\*ones(3,3);

[r1,c1] = size(mtlb\_double(a)); 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)]; a2 = gsort(a1,"g","i");

med = a2(5); b(i,j) = med; end;

end; b=uint8(b);

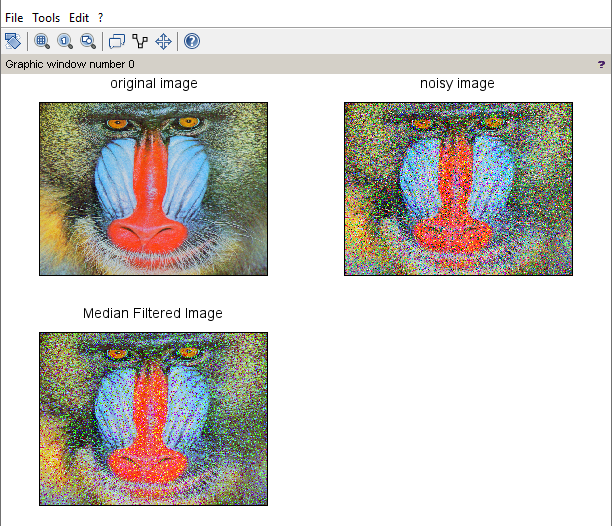
subplot(2,2,1); imshow(uint8(a)); title('original image')

subplot(2,2,2); imshow(uint8(c)); title('noisy image')

subplot(2,2,3);

imshow(b);title("Median Filtered Image");

**Output:-**



**Practical No. 4C**

**Aim:-** Program for Image deblurring using inverse, Weiner filters

**Code Implementation:- (Include steps)**

clc; clear all;

Ioriginal = imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff'); subplot(2,2,1);

imshow(Ioriginal) title('Original Image')

PSF = fspecial('motion',21,11); Idouble = im2double(Ioriginal);

imf = imfilter(Idouble,PSF,'circular','full'); subplot(2,2,2);

imshow(imf) title('Blurred Image')

noise\_mean = 0;

noise\_var = 0.0001;

blurred\_noisy = imnoise(imf, 'gaussian',noise\_mean, noise\_var); subplot(2,2,3);

imshow(blurred\_noisy) title('Simulate Blur and Noise')

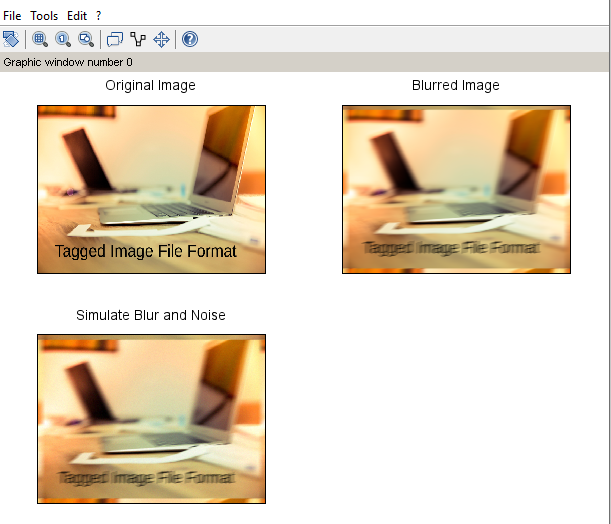
estimated\_nsr = noise\_var;

wnr3 = imdeconvwiener(blurred\_noisy, PSF, estimated\_nsr); subplot(2,2,4);

imshow(wnr3)

title('Restoration of Blurred, Noisy Image Using Estimated NSR');

**Output:-**



**Practical No. 5**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 5a**

**Aim:-**  Color Image Processing Program to read a color image and segment into RGB planes , histogram of color image

**Code Implementation:- (Include steps)**

Code:

clc;

close;

original=imread('C:\Program Files\scilab-6.1.0\IPCV\images\peppers.png');

im\_red=original(:,:,1);

im\_green=original(:,:,1);

im\_blue=original(:,:,3);

subplot(3,3,1),imshow(original),title('Original');

subplot(3,3,2),imshow(im\_red),title('Red');

r=size(im\_red,1); c=size(im\_red,2); h=zeros(1,256);

for i=1:r

for j=1:c

if (im\_red(i,j)==0)

im\_red(i,j)=1;

end

k=im\_red(i,j);

h(k)=h(k)+1;

end

end

subplot(3,3,3); plot(h); title('Histogram of the Red Image');

subplot(3,3,4),imshow(im\_green),title('Green');

subplot(3,3,5),imshow(im\_blue),title('Blue');

a=size(im\_blue,1); b=size(im\_blue,2); x=zeros(1,256);

for i=1:a

for j=1:b

if (im\_blue(i,j)==0)

im\_blue(i,j)=1;

end

y=im\_blue(i,j);

x(y)=x(y)+1;

end

end

subplot(3,3,6); plot(x); title('Histogram of the Blue Image');

**Output:-**



**Practical No. 5b**

**Aim:-**  Program for converting from one color model to another model RGB to CMY

**Code Implementation:- (Include steps)**

Code:

clc;

close;

f=imread('C:\Program Files\scilab-6.1.0\IPCV\images\peppers.png');

f = im2double(f);

r=f(:,:,1);

g=f(:,:,2);

b=f(:,:,3);

c = 1-r;

m = 1-g;

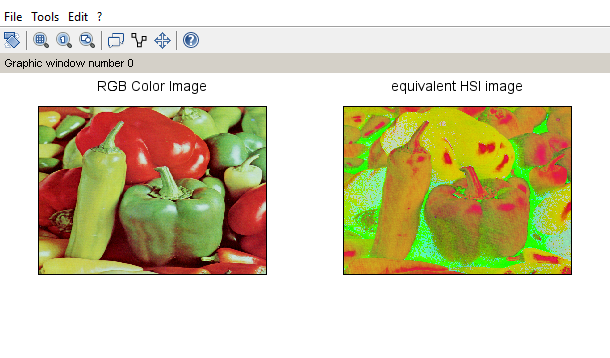
y = 1-b;

CMY = cat(3,c,m,y);

subplot(1,2,1),imshow(f),title('RGB');

subplot(1,2,2),imshow(CMY),title('CMY');

**Output:-**



RGB to HSI

**Code Implementation:- (Include steps)**

clc;

close;

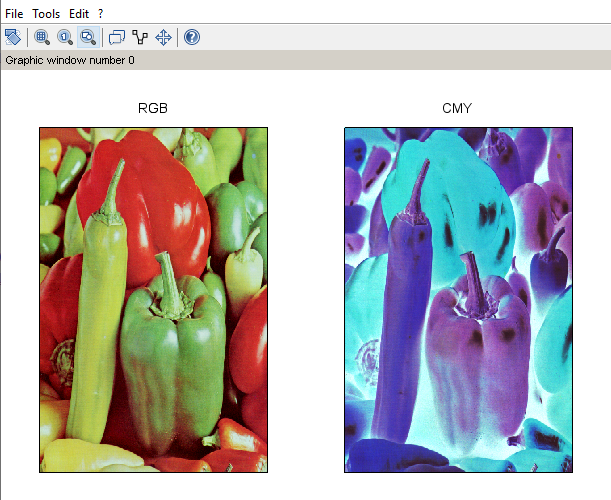
RGB = imread('C:\Program Files\scilab-6.1.0\IPCV\images\peppers.png');

subplot(2,2,1),imshow(RGB),title('RGB Color Image')

HSV = rgb2hsv(RGB);

subplot(2,2,2),imshow(HSV),title('equivalent HSI image')

**Output:-**



**Practical No. 5c**

**Aim:-**  Program to apply false colouring(pseudo) on a gray scale image

**Code Implementation:- (Include steps)**

Code:

clc;

close;

a=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

[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

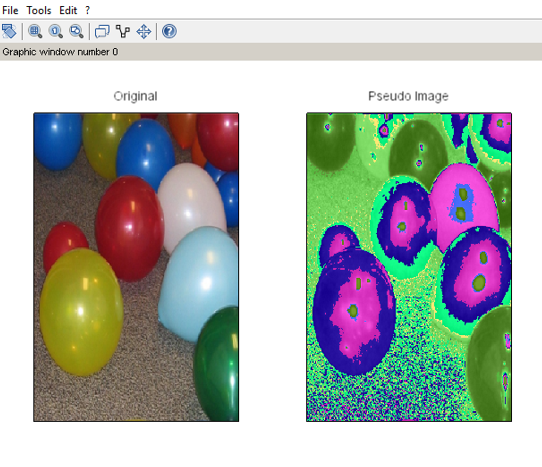
end

end

subplot(1,2,1),imshow(a),title('Original');

subplot(1,2,2),imshow(b),title('Pseudo Image');

**Output:-**



**Practical No. 8**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 8a**

**Aim:-** Practical 8 Morphological Image Processing 8a Program to apply erosion, dilation, and morphological gradient on an image

**Code Implementation:- (Include steps)**

Code:

clc;

close;

img=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

se1 = ones(10,10);

im1=imdilate(img,se1);

im2=imerode(im1,se1);

g=im1-im2;

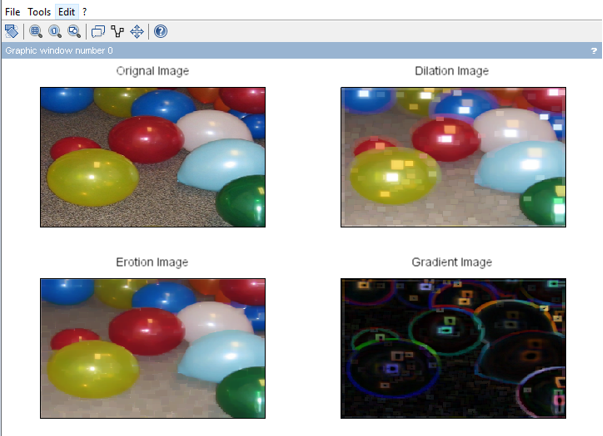
subplot(2,2,1),imshow(img),title('Orignal Image');

subplot(2,2,2),imshow(im1),title('Dilation Image');

subplot(2,2,3),imshow(im2),title('Erotion Image');

subplot(2,2,4),imshow(g),title('Gradient Image');

**Output:-**



**Practical No. 8b**

**Aim:-** Program for detecting boundary of an image

**Code Implementation:- (Include steps)**

Code:

clc;

aa=imread('C:\Program Files\scilab-6.1.0\IPCV\images\moon.jpg');

se1 = imcreatese('rect',3,3);

m1=imerode(aa,se1);

m2=aa-m1;

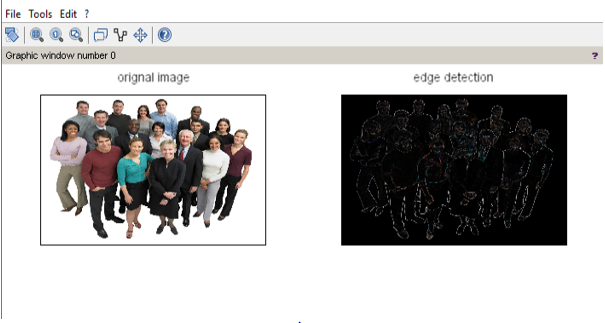
subplot(2,2,1),imshow(aa);

title('orignal image');

subplot(2,2,2),imshow(m2);

title('edge detection');

**Output:-**



**Practical No. 8c**

**Aim:-**  Program to apply opening, closing Opening

**Code Implementation:- (Include steps)**

Code:

clc;

close;

img=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

se1 = ones(10,10);

im2 = imerode(img,se1);

im3 = imdilate(im2,se1);

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

subplot(1,2,2),imshow(im3),title('opening image');

Closing

Code:

clc;

close;

aa=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

se1 = ones(10,10);

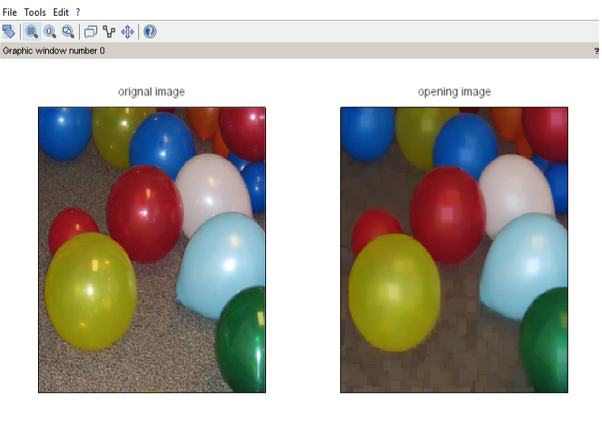
IM2=imdilate(aa,se1);

IM3=imerode(IM2,se1);

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

subplot(1,2,2),imshow(IM3),title('Closed Image');

**Output:-**



**Practical No. 8d**

**Aim:-**  Program to apply Top-Hat/Bottom-hat Transformations

**Code Implementation:- (Include steps)**

Code:

clc;

close;

i=imread('C:\Program Files\scilab-6.1.0\IPCV\images\cameraman.tiff');

se1 = ones(10,10);

im1=imerode(i,se1);

im2=imdilate(im1,se1);

h=i-im2;

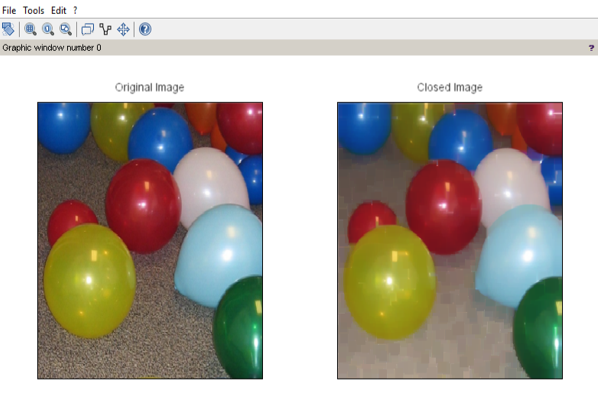
subplot(2,2,1),imshow(i),title('Orignal Image');

subplot(2,2,2),imshow(im1),title('Erotion Image');

subplot(2,2,3),imshow(im2),title('Dilation Image');

subplot(2,2,4),imshow(h),title('Top Hat Transformation Image');

**Output:-**



**Practical No. 9**

**Aim:-**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No. 9a**

**Aim:-**  Practical 9 Image Segmentation Program for Edge detection using Sobel

**Code Implementation:- (Include steps)**

clear all;

clc;

a=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

b=double(a);

d=b;

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

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

[r1,c1]=size(a);

for i=1:r1

for j=1:c1

new(i,j)=a(i,j);

nem(i,j)=a(i,j);

end

end

new=double(new);

nem=double(nem);

for i=2:1:r1-1

for j=2:1:c1-1

new(i,j)=(mh(1)\*d(i-1,j-1))+(mh(2)\*d(i-1,j))+(mh(3)\*d(i-1,j+1))+(mh(4)\*d(i,j-1))+(mh(5)\*d(i,j))+(mh(6)\*d(i,j+1))+(mh(7)\*d(i+1,j-1))+(mh(8)\*d(i+1,j))+(mh(9)\*d(i+1,j+1));

end

end

for i=2:1:r1-1

for j=2:1:c1-1

nem(i,j)=(mv(1)\*d(i-1,j-1))+(mv(2)\*d(i-1,j))+(mv(3)\*d(i-1,j+1))+(mv(4)\*d(i,j-1))+ (mv(5)\*d(i,j))+(mv(6)\*d(i,j+1))+(mv(7)\*d(i+1,j-1))+(mv(8)\*d(i+1,j))+(mv(9)\*d(i+1,j+1));

end

end

new2=new+nem;

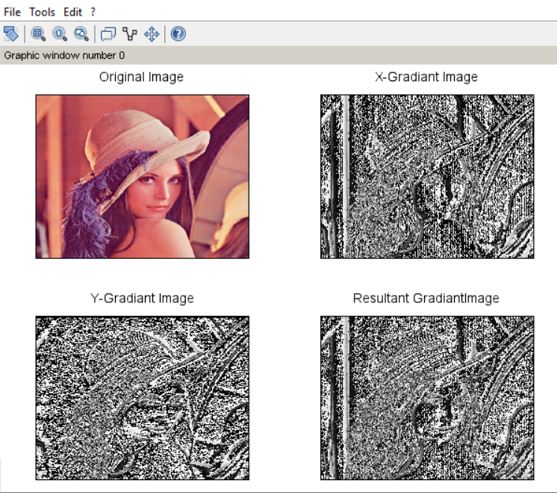
subplot(2,2,1);imshow(uint8(a));title('Original Image');

subplot(2,2,2);imshow(uint8(new));title('X-Gradiant Image');

subplot(2,2,3);imshow(uint8(nem));title('Y-Gradiant Image');

subplot(2,2,4);imshow(uint8(new2));title('Resultant GradiantImage');

**Output:-**



**Practical No. 9b**

**Aim:-**  Prewitt

**Code Implementation:- (Include steps)**

clear all;

clc;

a=uigetfile('\*.\*','Select the Image:-');

a=imread(a);

b=double(a);

d=b;

mh=[-1 -1 -1;0 0 0;1 1 1];

mv=[-1 0 1;-1 0 1;-1 0 1];

[r1,c1]=size(a);

for i=1:r1

for j=1:c1

new(i,j)=a(i,j);

nem(i,j)=a(i,j);

end

end

new=double(new);

nem=double(nem);

for i=2:1:r1-1

for j=2:1:c1-1

new(i,j)=(mh(1)\*d(i-1,j-1))+(mh(2)\*d(i-1,j))+(mh(3)\*d(i-1,j+1))+(mh(4)\*d(i,j-1))+(mh(5)\*d(i,j))+(mh(6)\*d(i,j+1))+(mh(7)\*d(i+1,j-1))+(mh(8)\*d(i+1,j))+(mh(9)\*d(i+1,j+1));

end

end

for i=2:1:r1-1

for j=2:1:c1-1

nem(i,j)=(mv(1)\*d(i-1,j-1))+(mv(2)\*d(i-1,j))+(mv(3)\*d(i-1,j+1))+(mv(4)\*d(i,j-1))+ (mv(5)\*d(i,j))+(mv(6)\*d(i,j+1))+(mv(7)\*d(i+1,j-1))+(mv(8)\*d(i+1,j))+(mv(9)\*d(i+1,j+1));

end

end

new2=new+nem;

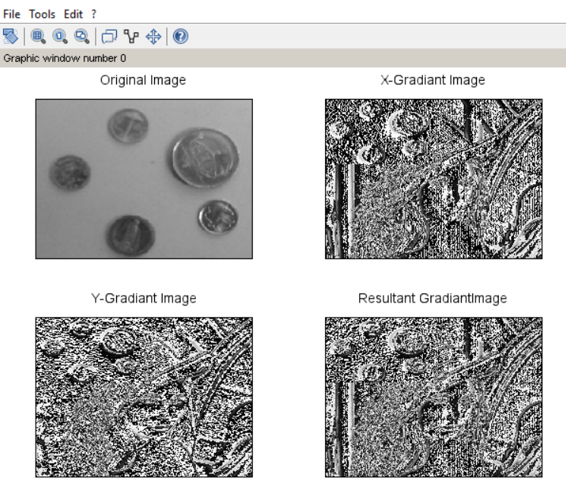
subplot(2,2,1);imshow(uint8(a));title('Original Image');

subplot(2,2,2);imshow(uint8(new));title('X-Gradiant Image');

subplot(2,2,3);imshow(uint8(nem));title('Y-Gradiant Image');

subplot(2,2,4);imshow(uint8(new2));title('Resultant GradiantImage');

**Output:-**



**Practical No. 9c**

**Aim:-**  Canny

**Code Implementation:- (Include steps)**

clear all;

clc;

a=imread('C:\Program Files\scilab-6.1.0\IPCV\images\Lena\_dark.png');

b=double(a);

thresh=0.4;

sigma=3;

E=edge(a, 'canny', thresh, sigma);

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

subplot(2,2,2);imshow(E); title('Gradient Image');

**Output:-**

