VALIA C.I. COLLEGE OF COMMERCE & VALIA LC

COLLEGE OF ARTS CES ROAD D.N NAGAR

(Affiliated to University Of Mumbai)

Mumbai-Maharashtra-400053.

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Journal entitled **IMAGE PROCESSING** is bonafied work of **GOVIND SAINI** bearing Seat No **: 07** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Date: **Internal Guide :**

INDEX

Practical No : 1

Aim: 1A) Program to calculate number of samples required for image.

Description :

The program will only calculate the number of samples required for an image.

Code :

*prac1.sce P2.sce

```
clc;
close;
//dimension of the image in inches
m = 4;
n = 6;
N = 400; //number of dots per inch in each direct
N2 = 2*N; //number of dots per inch in both horiz
Fs = m*N2*n*N2;
disp('Number of samples required to preserve the
' information in the image=',Fs)
```

Output:

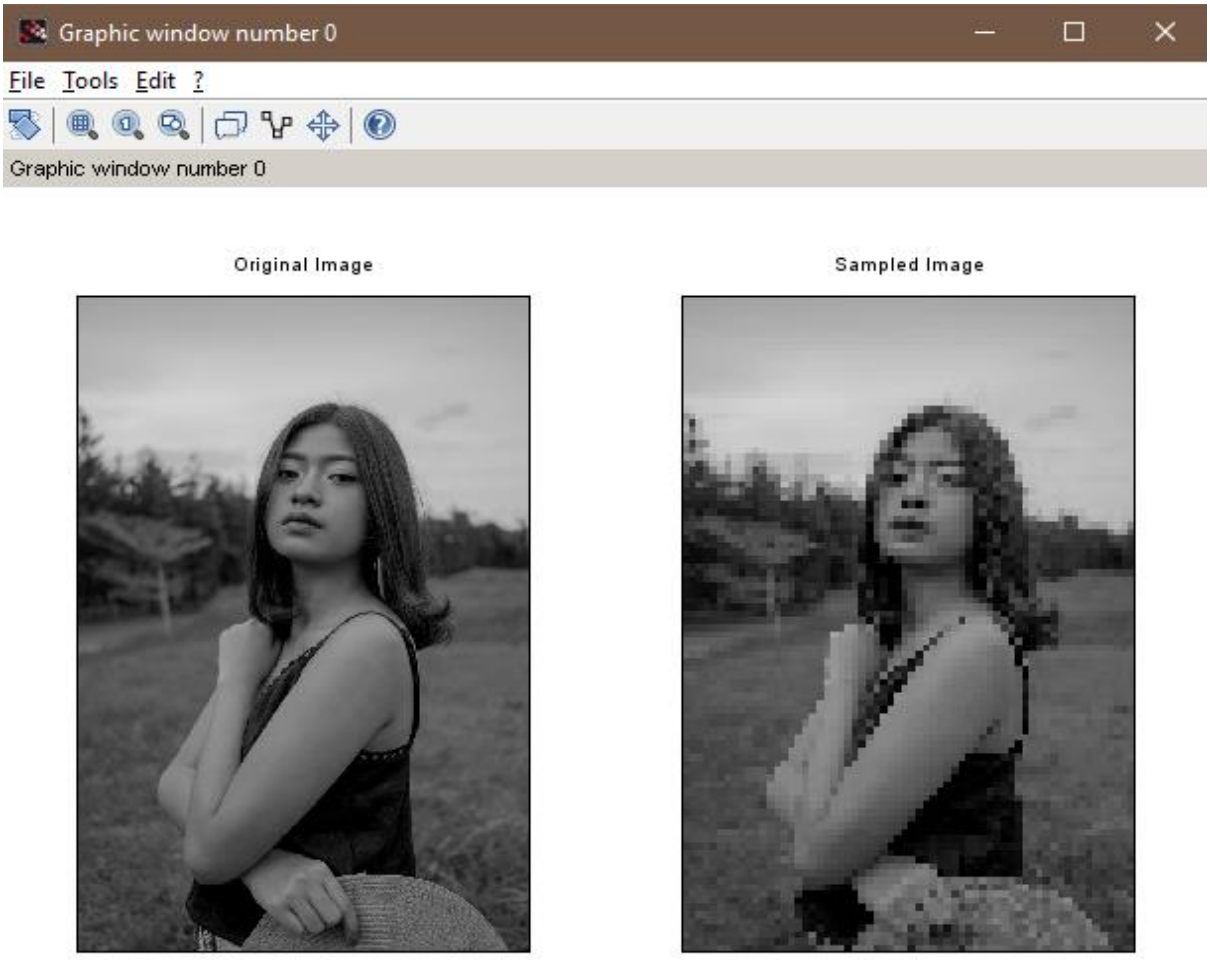
```
"Number of samples required to preserve the information in the image="

 15360000.
```

1 B) Program to study the effects of reducing the spatial resolution of a digital image

Code :

```
clc;
clear all;
//n=16; or 32 , 64 etc.
n = input ('Enter the input samples') ;
img = rgb2gray(imread('C:\govind\images\g1.jpg'));
a=size(img);
w=a(2);
h=a(1);
im=zeros(100);
for i=1:n:h
for j=1:n:w
for k=0:n-1
for l=0:n-1
    im(i+k,j+l)=img(i,j);
end
end
end
end
subplot(1,2,1);
imshow(uint8(img));title('Original Image');
subplot(1,2,2);
imshow(uint8(im));title('Sampled Image');
```

Output:


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

Practical No : 2

Aim: Basic Intensity Transformation functions

2.a) Program to perform Image negation

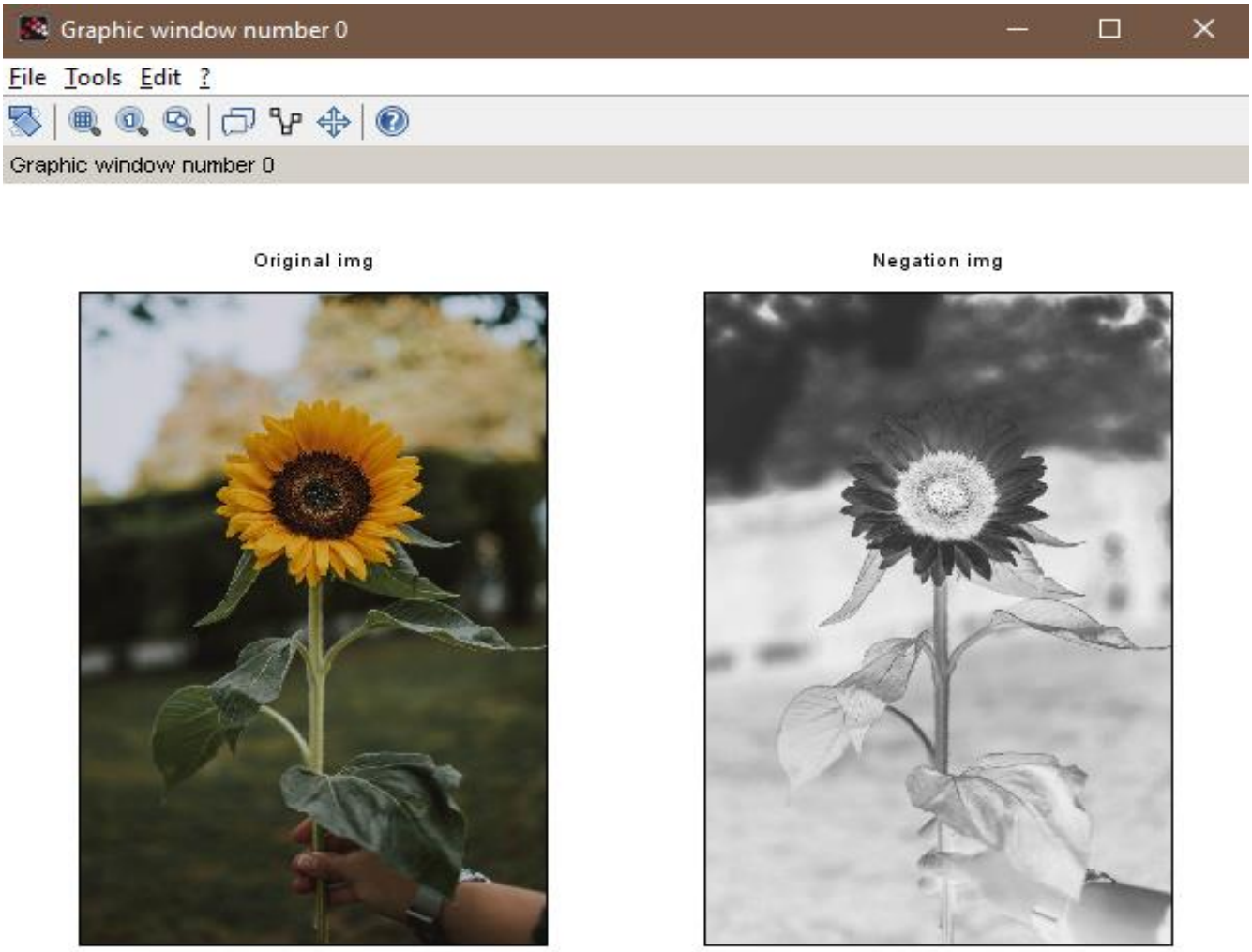
Description:

when you are working with gray-scale images, sometimes you want to modify the intensity values. For instance, you may want to reverse black and the white intensities or you may want to make the darks darker and the lights lighter. An application of intensity transformations is to increase the contrast between certain intensity values so that you can pick out things in an image. For instance, the following two images show an image before and after an intensity transformation.

Code :

```
clc;
clear all;
a=imread('C:\govind\images\g2.jpg');
subplot(1,2,1);
imshow(a)
title('Original img')
[m,n]=size(a);
for i=1:m
for j=1:n
c(i,j)=255-a(i,j)
end
end
subplot(1,2,2);
imshow(c)
title('Negation img')
```

Output:

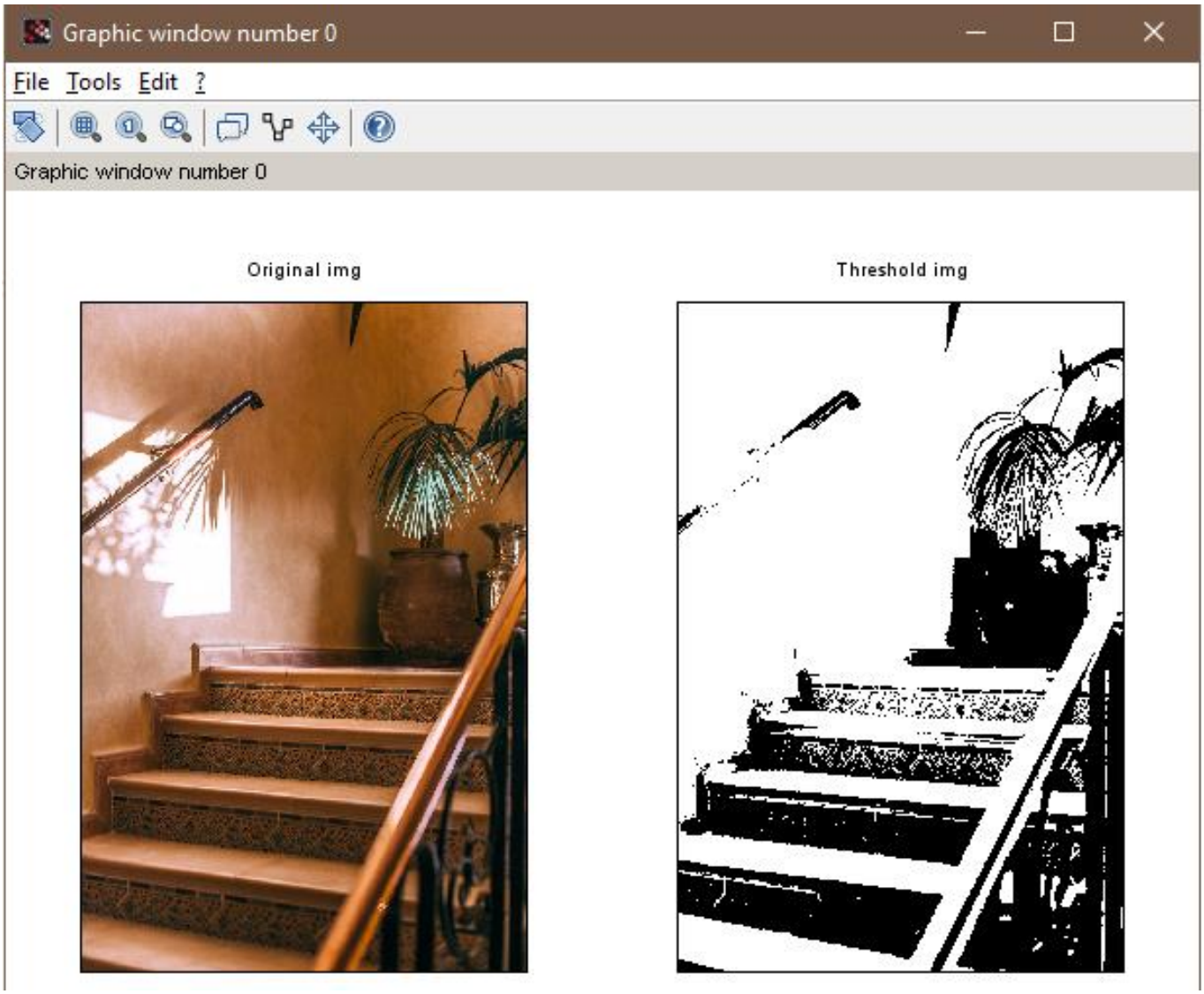

Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original img
Negation img

2.b) Program to perform threshold on an image

Code :

```
clc;
clear all;
a=imread('C:\govind\images\g3.jpg');
b=double(a)
subplot(1,2,1);
imshow(a);
title('Original img');
t=100;
[m,n]=size(b);
for i=1:m
for j=1:n
if (b(i,j)<t)
c(i,j)=0;
else
c(i,j)=255;
end
end
end
subplot(1,2,2);
imshow(c);
title('Threshold img');
```

Output:

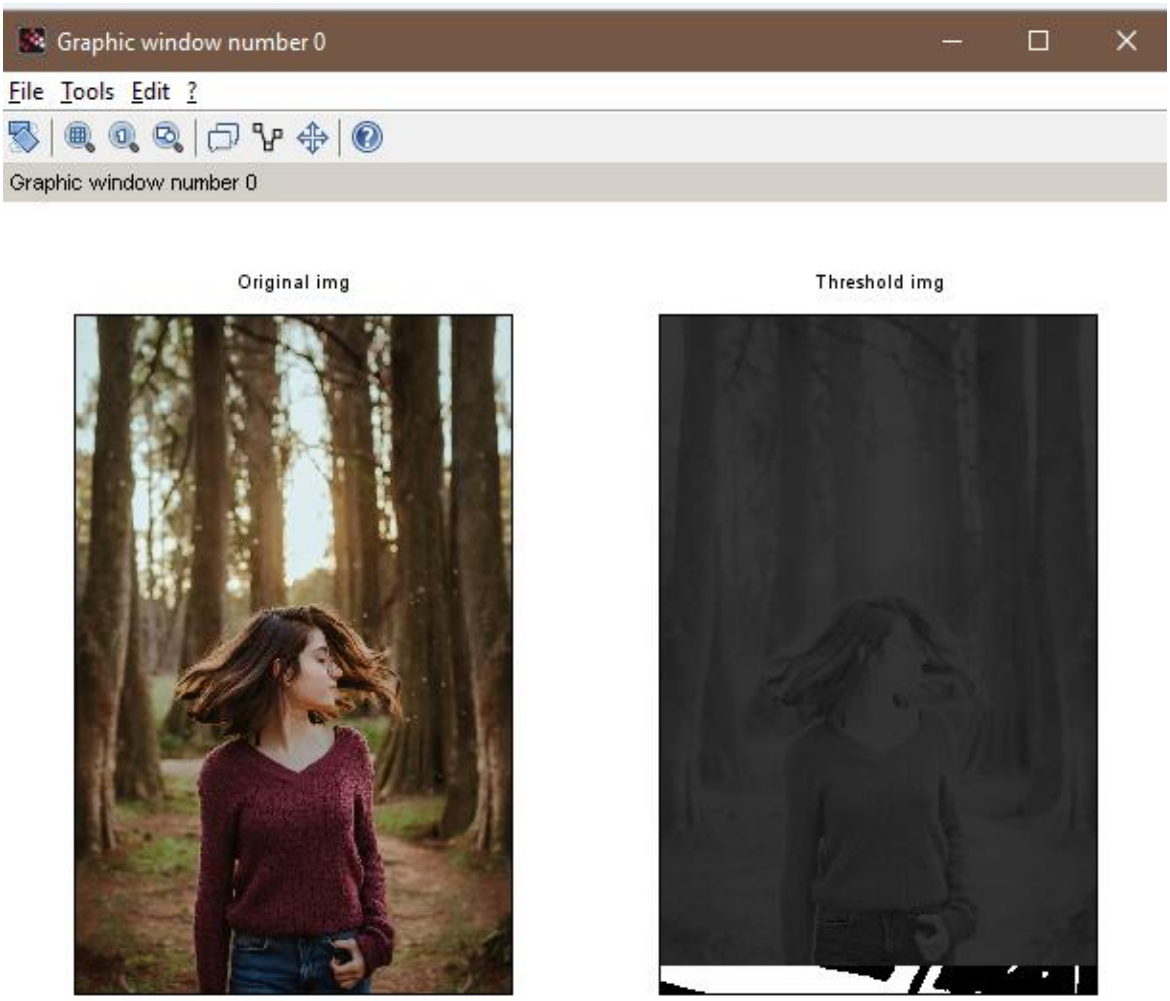

Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original img
Threshold img

2.c) Program to perform Log transformation

Code :

```
clc;
clear all;
a=imread('C:\govind\images\g4.jpg');
b=double(a)
subplot(1,2,1);
imshow(a);
title('Original img');
t=10;//constant value
[m,n]=size(b);
for i=1:m
for j=1:n
c(i,j)=t*log(1+ b(i,j))
end
end
subplot(1,2,2);
imshow(uint8(c));
title('Threshold img');
```

Output:

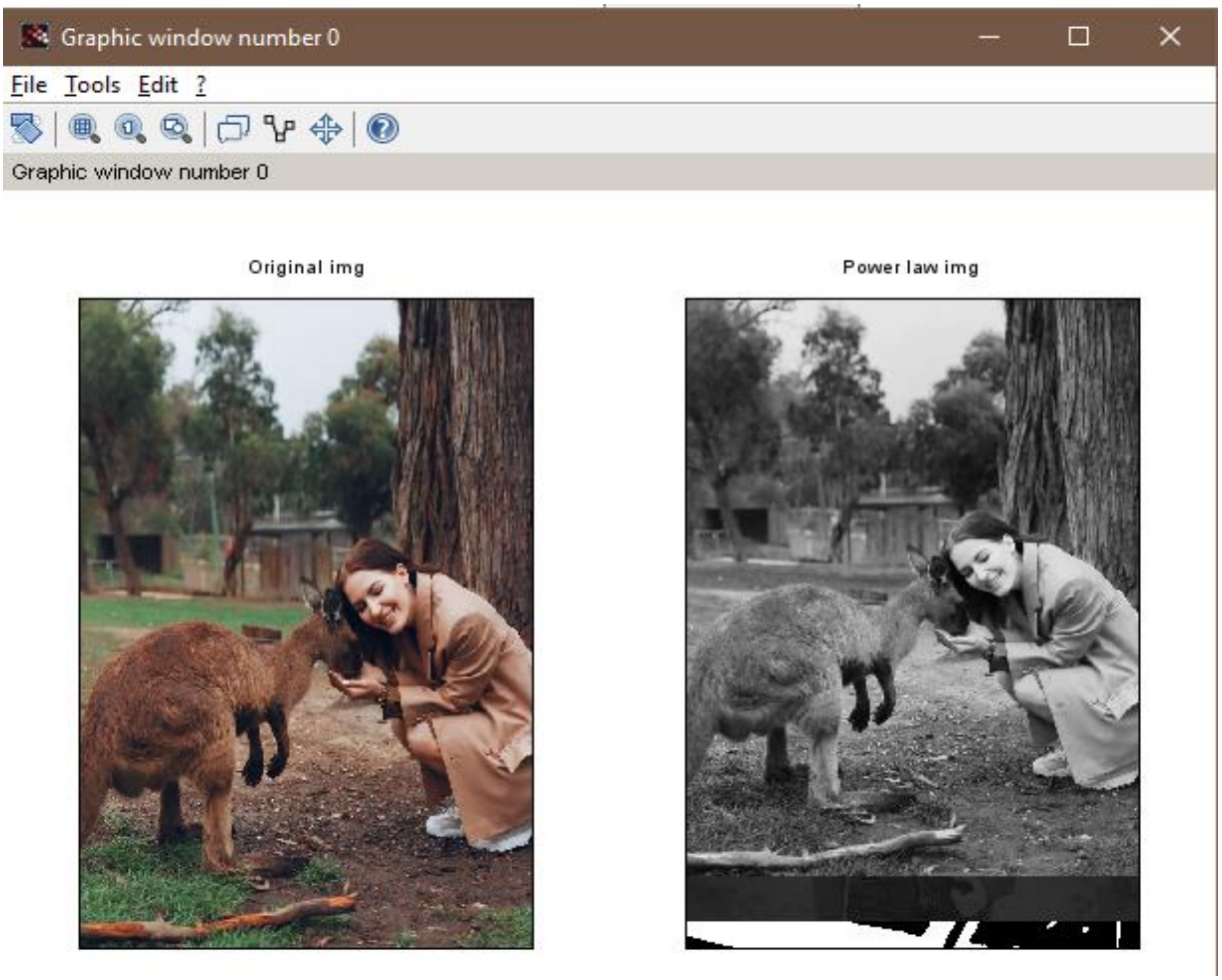

Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original img
Threshold img

2.d) Power-law transformations

Code :

```
//Power law transformation
clc;
clear all;
a=imread('C:\govind\images\g5.jpg');
b=double(a)
subplot(1,2,1);
imshow(a);
title('Original img');
k=1;
gamma=1;//gamma value
[m,n]=size(b);
for i=1:m
for j=1:n
c(i,j)= k*(b(i,j)^gamma);
end
end
subplot(1,2,2);
imshow(uint8(c));
title('Power law img');
```

Output:

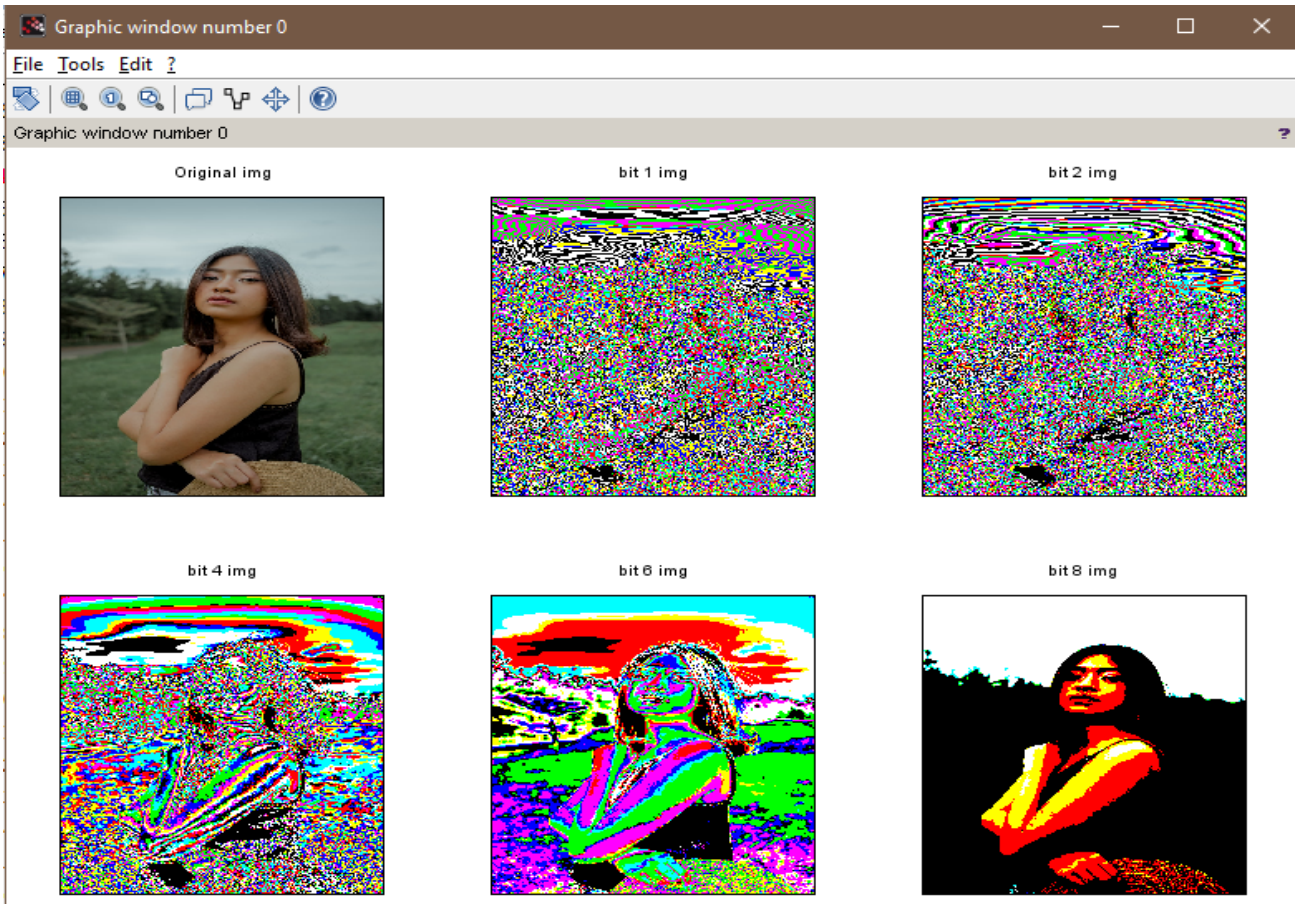

Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original img
Power law img

2.e) Piecewise linear transformations

Code :

```
clc;
clear all;
a=imread('C:\govind\images\g4.jpg');
b=double(a)
subplot(2,3,1);
imshow(a);
title('Original img');
f1=bitget(b,1),
subplot(2,3,2);
imshow(f1);
title('bit 1 img');
f2=bitget(b,2),
subplot(2,3,3);
imshow(f2);
title('bit 2 img');
f3=bitget(b,4),
subplot(2,3,4);
imshow(f3);
title('bit 4 img');
f4=bitget(b,6),
subplot(2,3,5);
imshow(f4);
title('bit 6 img');
f5=bitget(b,8),
subplot(2,3,6);
imshow(f5);
title('bit 8 img');
```

Output:


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Original img
bit 1 img
bit 2 img
bit 4 img
bit 6 img
bit 8 img

Practical No : 3

Aim: Program to plot the histogram of an image and categorise

Description:

An **image histogram** is a type of **histogram** that acts as a graphical representation of the tonal distribution in a digital **image**. It plots the number of pixels for each tonal value. By looking at the **histogram** for a specific **image** a viewer will be able to judge the entire tonal distribution at a glance.

Code :

```
clc ;
clear ;
close ;
I = imread ( 'C:\govind\images\g10.jpg');
[ count , cells ]= imhist(I);
ShowImage (I,'C:\govind\images\g10.sci');
scf (1) ;
plot2d3 ( ' gnn ' , cells , count )
title ( 'Histogram Plot of OriginalImage')
exec('C:\histeq.sci' ) ;
Iheq = histeq ( I ) ;
[ count , cells ]= imhist ( Iheq );
scf (2)
ShowImage ( Iheq , ' Histogram Equalized Image g10.png ' )
scf (3)
plot2d3 ('gnn', cells , count )
title ( 'Histogram of Histogram Equalized Image')
```

Output:

Practical No : 4

Aim: Program to read a color image and segment into RGB planes , histogram of color image

Description :

Color image processing is divided into two major areas: full-**color** and pseudo-**color processing**. In the first category, the **images** in question typically are acquired with a full-**color** sensor, such as a **color** TV camera or **color** scanner

Code :

```
clc ;
clear ;
close ;
RGB = imread ( 'C:\govind\images\g10.jpg');
figure
ShowColorImage ( RGB ,'RGB ColOr Image ' )
YIQ = rgb2ntsc ( RGB ) ;
figure
ShowColorImage ( YIQ , 'NTSC image YIQ ' )
RGB = ntsc2rgb ( YIQ ) ;
YCC = rgb2ycbcr ( RGB ) ;
figure
ShowColorImage ( YCC ,'equivalent HSV image YCbCr' )
RGB = ycbcr2rgb ( YCC ) ;
HSV = rgb2hsv ( RGB ) ;
figure
ShowColorImage ( HSV , 'equivaLent HSV image' )
RGB = hsv2rgb ( HSV ) ;
R = RGB (: ,: ,1) ;
G = RGB (: ,: ,2) ;
B = RGB (: ,: ,3) ;
figure
ShowImage (R , 'Red Matrix' )
figure
ShowImage (G , 'Green Matrix' )
figure
ShowImage (B , 'Blue Matrix ' )
```

Output:

Practical No : 5

Aim: Program to apply Discrete Fourier Transform on an image

Description:

In mathematics, the **discrete Fourier transform** (**DFT**) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the **discrete**-time **Fourier transform** (DTFT), which is a complex-valued function of frequency.

Code :

```
2
3  clc ;
4  clear ;
5   close ;
6  I = imread ( `C:\Users\Desktop\
7  Gautam PAL Lab\DIP Lab2\lenna.jpg ` ) ;
8  exec ( `C:\Users\Desktop\
9  Gautam PAL Lab\DIP Lab2\fft2d.sci`) ;
10 exec ( `C:\Users\Desktop\
11 Gautam PAL Lab\DIP Lab2\fft2d.sci` ) ;
12
13
14 I = double ( I ) ;
15 J = fft2d ( I ) ;
16 K = real( ifft2d ( J )) ;
17 figure
18 ShowImage (I , `Original Lenna Image `)
19 figure
20 ShowImage ( abs (J ) , `2DDFT (spectrum) of Lenna Image`)
21 figure
22 ShowImage (K , `2dIDFT of Lenna Image ` )
23
24 L = fftshift ( J ) ;
25 M = fftshift ( L ) ;
26 figure
27 ShowImage ( abs (L ) , `fftshited spectrum of Lenna Image`)
28 figure
29 ShowImage ( abs (M ) , `two times fftshifted ` )
```

Practical No : 6

Aim: Program to apply erosion, dilation, opening, closing

Code :

```
clc ;
clear ;
close ;
Image = imread('C:\govind\images\g7.jpg');

StructureElement = CreateStructureElement('square',3);
ResultImage1 = ErodeImage(Image,StructureElement);
ResultImage2 = DilateImage(Image,StructureElement);
ResultImage3 = BottomHat(Image,StructureElement);
ResultImage4 = TopHat(Image,StructureElement);
ShowImage(Image,'Original Image');
figure
ShowImage(ResultImage1,'Eroded Image');
figure
ShowImage(ResultImage2,'Dilated Image');
figure
ShowImag(ResultImage3,'bottom hat filtered image');
figure
ShowImage(ResultImage,'top hat filtered image');
ResultImage5 = imadd(ResultImage3,ResultImage4 );
figure
ShowImage(ResultImage4,'top hat fitltered image + bottom hat filtered image');
```

Output: