# IMAGE PROCESSING

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR. NO.** | **PR.**  **NO.** | **NAME OF PRACTICAL** | **PAGE NO.** | **SIGN** |
| **1.** | 1A) | Program to calculate number of samples required for an image. | 147 |  |
| 1B) | Program to study the effects of reducing the spatial resolution of a digital image. | 148 |  |
| **2.** | 2A) | Basic Intensity Transformation functions :  Program to perform Image negation | 149 |  |
| 2B) | Program to perform threshold on an image. | 150 |  |
| 2C) | Program to perform Log transformation | 151 |  |
| 2D) | Power-law transformations | 152 |  |
| 2E) | Piece-wise linear transformatioN | 153 |  |
| **3.** | 3) | Program to plot the histogram of an image and categorise | 154 |  |
| **4.** | 4) | Color Image Processing :  Program to read a color image and segment into RGB planes , histogram of color image | 155 |  |
| **5.** | 5) | Fourier Related Transforms :  Program to apply Discrete Fourier Transform on an image | 156 |  |
| **6.** | 6A) | Write a program to apply following morphological operations on the image:  Opening | 157 |  |
| 6B) | Closing | 158 |  |
| 6C) | Morphological Gradient | 159 |  |
| 6D) | Top-hat transformation | 160 |  |

**PRATICAL 1A):** **Program to calculate number of samples required for an image.**

**Code:**

clc;

close;

m =4;

n=6;

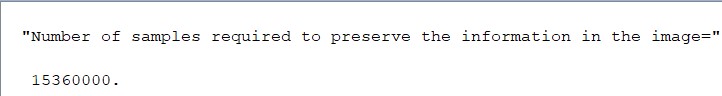
N=400;

N2=2\*N;

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

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

**Output:**



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

clc;

clear all;

n = input ('Enter the input sample');

img=imread("C:\Users\bnnco\abc.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

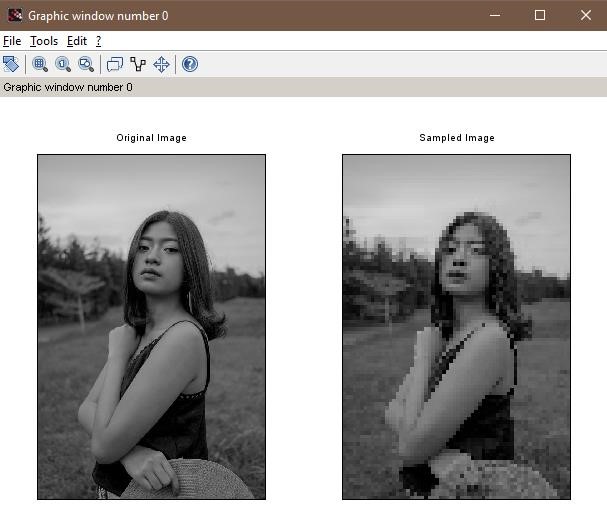
subplot(1,2,1);

imshow(uint8(img));title('original image');

subplot(1,2,2);

imshow(uint8(im));title('sampled image');

**Output:**



**PRATICAL 2: Basic Intensity Transformation functions**

**A) Program to perform Image negation**

clc;

clear all;

a=imread("C:\Users\bnnco\xyz1.jpg");

subplot(1,2,1);

imshow(a)

title('original image')

[m,n]=size(a);

for i=1:m

for j=1:n

c(i,j)=255-a(i,j)

end

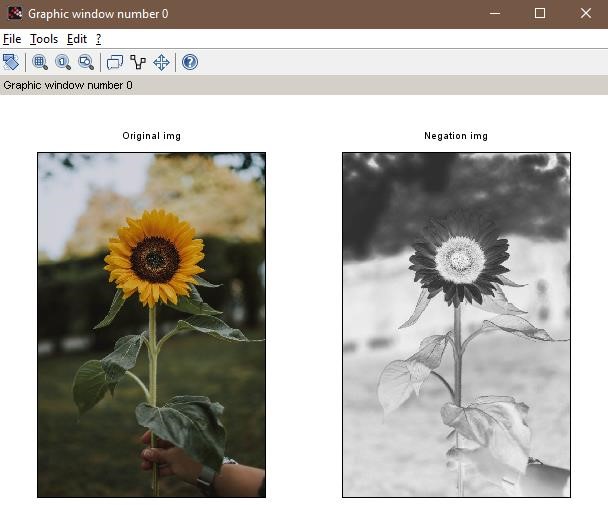
end

subplot(1,2,2);

imshow(c)

title('negation img')

**Output:**



**B) Program to perform threshold on an image.**

clc;

clear all;

a=imread("C:\Users\bnnco\xyz2.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

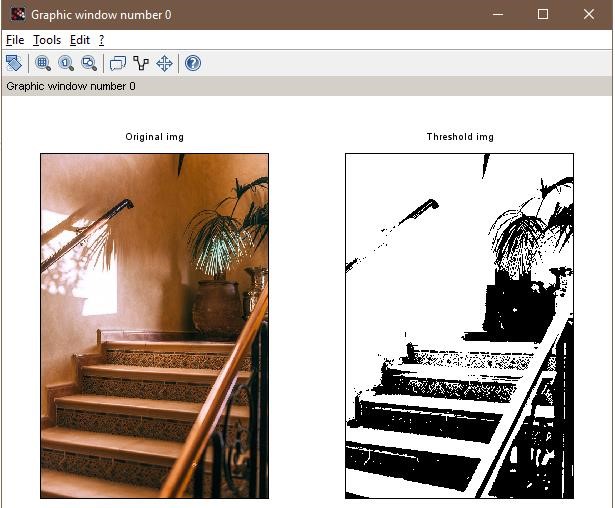
end

subplot(1,2,2);

imshow(c);

title('threshold img');

**Output:**



**C) Program to perform Log transformation**

clc;

clear all;

a=imread("C:\Users\bnnco\xyz3.jpg");

b=double(a)

subplot(1,2,1);

imshow(a);

title('original img');

t=10;

[m,n]=size(b);

for i=1:m

for j=1:n

c(i,j)=t\*log(1+b(i,j))

end

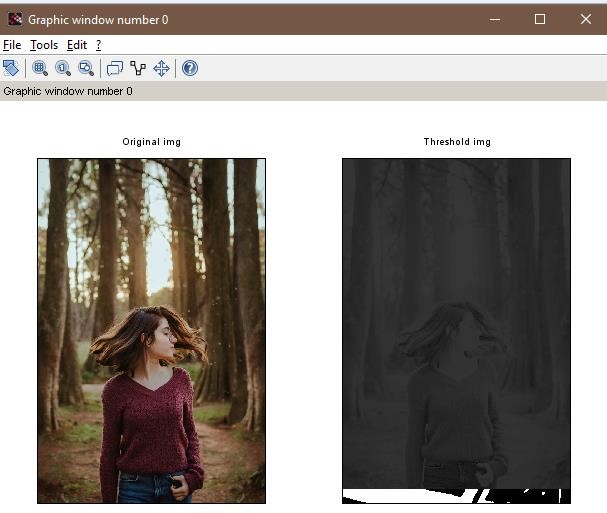
end

subplot(1,2,2);

imshow(uint8(c));

title('threshold img');

**Output:**



**D) Power-law transformations**

clc;

clear all;

a=imread("C:\Users\bnnco\xyz3.jpg");

b=double(a)

subplot(1,2,1);

imshow(a);

title('original img');

k=1;

gamma=1;

[m,n]=size(b);

for i=1:m

for j=1:n

c(i,j)=k\*(b(i,j)^gamma);

end

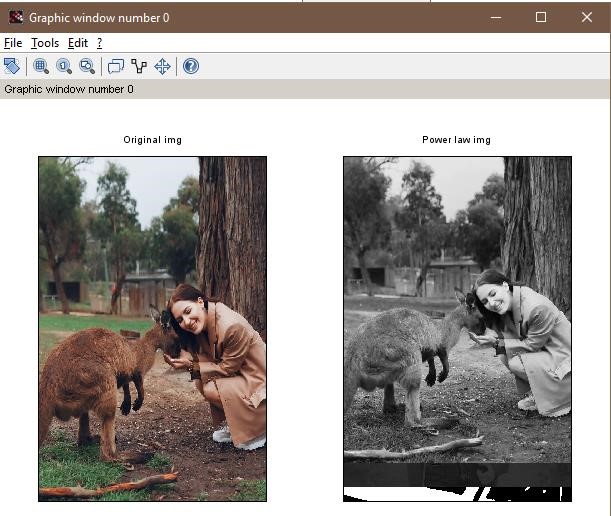
end

subplot(1,2,2);

imshow(uint8(c));

title('power law img');

**Output:**



**E) Piecewise linear transformations**

clc;

clear all;

a=imread("C:\Users\bnnco\xyz.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');

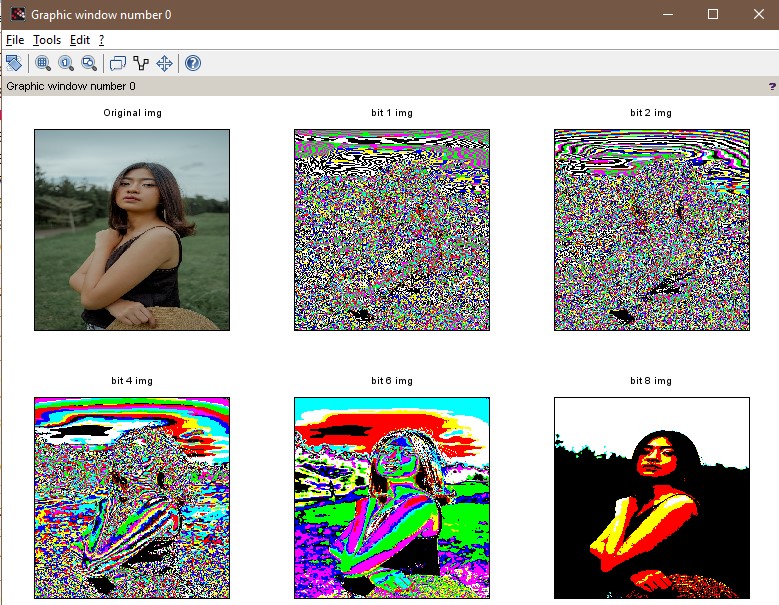
f5=bitget(b,8);

subplot(2,3,6);

imshow(f5);

title('bit 8 img');

**Output:**



**PRATICAL 3: Program to plot the histogram of an image and categorise**

a = imread("cm.tif");

a =double(a);

big=256;

[row ,col ] = size(a);

c=row\*col;

h=zeros(1,300);

z = zeros(1,300);

for n = 1:1:row

for m = 1:1:col

if a(n,m)==0

a(n,m) = 1;

end

end

end

for n = 1:1:row

for m = 1:1:col

t= a(n,m);

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

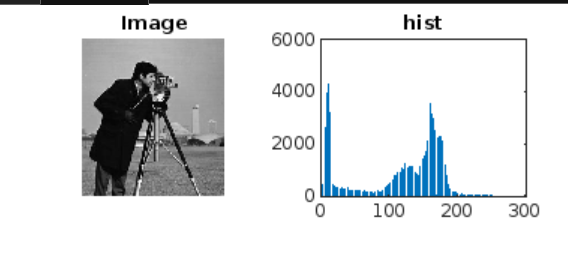
end

end

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

subplot(1,2,2);bar(h);title('hist');

**Output:**



**PRATICAL 4: Program to read a color image and segment into RGB planes , histogram of color image**

a = imread('vege.jpg');

figure(3),subplot(2,2,1),imshow(a);

title('Orignal Image');

k=rgb2ntsc(a);

figure(3),subplot(2,2,2),imshow(k);

title('RGB TO NTSC');

l=rgb2ycbcr(a);

figure(3),subplot(2,2,3),imshow(l);

title('RGB TO YCbCr');

m=imcomplement(a);

figure(3),subplot(2,2,4),imshow(m);

title('RGB TO CMY');

imr=a(:,:,1);

img=a(:,:,2);

imb=a(:,:,3);

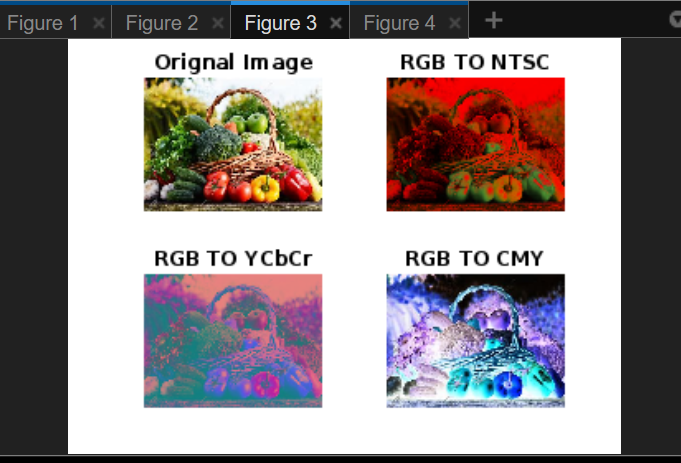
figure(4),subplot(2,2,1),imshow(a),title('original image');

figure(4),subplot(2,2,2),imshow(imr),title('Red');

figure(4),subplot(2,2,3),imshow(img),title('Green');

figure(4),subplot(2,2,4),imshow(imb),title('Blue');

**Output:**

****



**PRATICAL 5: Program to apply Discrete Fourier Transform on an image**

I=imread("vege.jpg");

subplot(2,2,1)

imshow(I)

title("original")

I=double(I);

J=fft2(I);

subplot(2,2,2)

imshow(J)

title("fourier transform image")

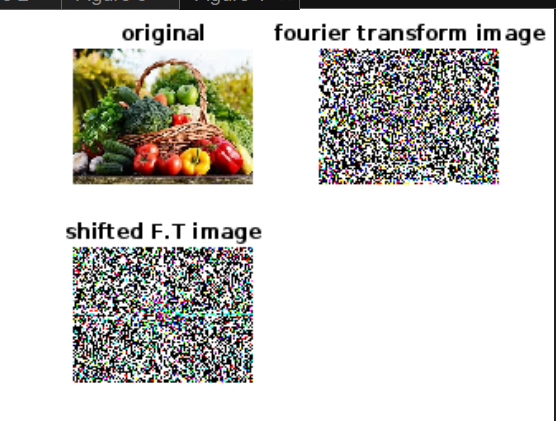
L=fftshift(real(J));

subplot(2,2,3)

imshow(L)

title("shifted F.T image")

**Output:**



**PRATICAL 6 : Write a program to apply following morphological operations on the image**

**A) OPENING:-**

img=imread('cameraman.tif');

se1 = strel('square',11);

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');

**Output:**



**B) CLOSING:-**

aa=imread('cameraman.tif');

se1=strel('square',11);

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:**



**C) MORPHOLOGICAL GRADIENT:-**

img=imread('cameraman.tif');

se1=strel('square',12);

im1=imdilate(img,se1);

im2=imerode(im1,se1);

g=im1-im2;

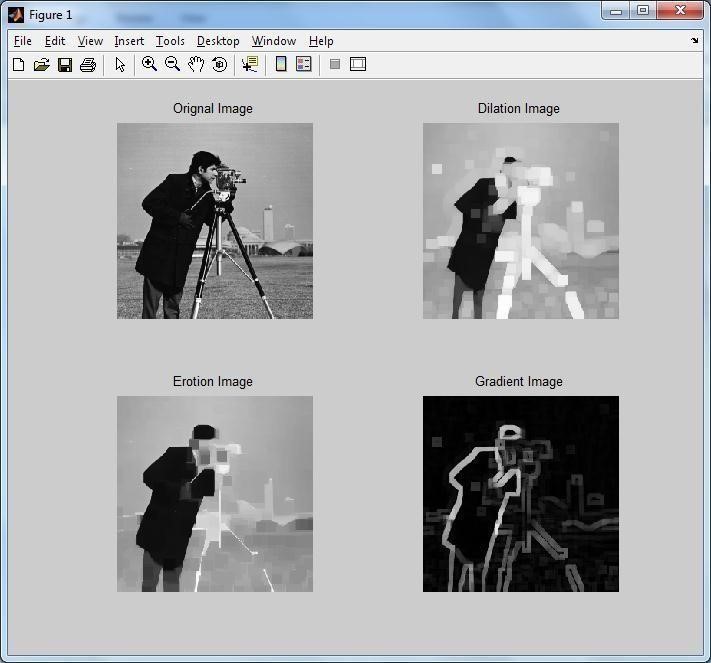
subplot(2,2,1),imshow(img),title('Orignal Image');

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:**



**D) TOP-HAT TRANSFORMATION:-**

i=imread('cameraman.tif');

se1=strel('square',22);

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:**

