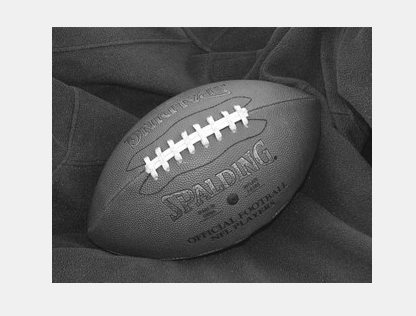
Lab 1

1. **Convert RGB Image To Grayscale Image**

im=imread('football.jpg');

I=rgb2gray(im);

imshow(I);

****

**Output :**

1. **Separate R,G And B From an image and show histograms**

‘

im=imread('football.jpg');

imR=im(:,:,1);

imG=im(:,:,2);

imB=im(:,:,3);

figure(1),imshow(imR);

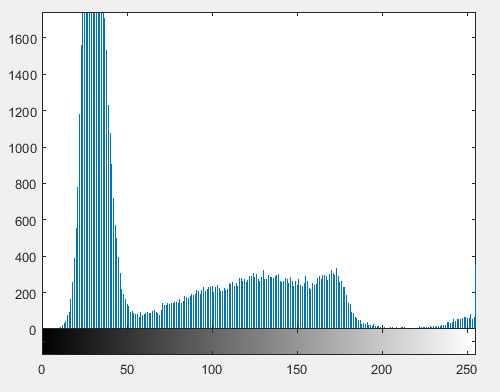
figure(2),imshow(imG);

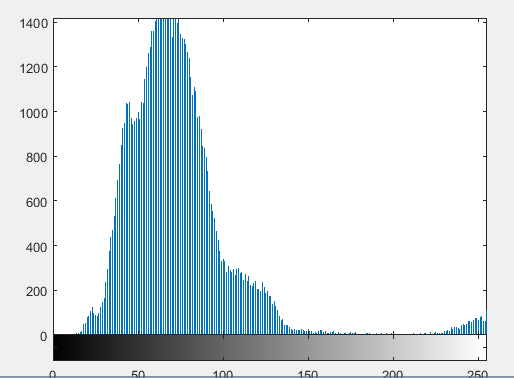
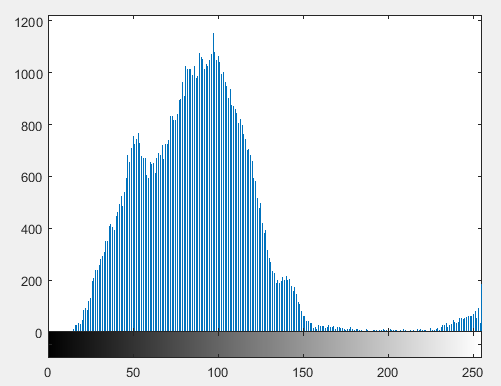
figure(3),imshow(imB);

figure(4),imhist(imR);

figure(5),imhist(imG);

figure(6),imhist(imB);





**4. Create a binary image from a grayscale image**

im=imread('football.jpg');

imR=im(:,:,1);

for i=1:256

for j=1:320

if(imR(i,j)>60)

im2(i,j)=255;

else

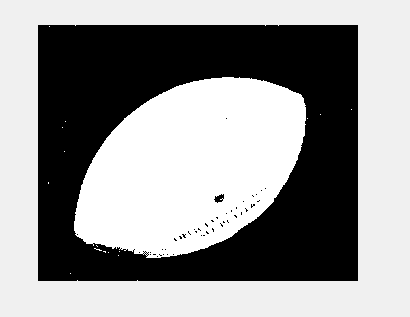
im2(i,j)=0;

end

end

end

figure(1),imshow(im2);



**5. Create a custom histogram from an image**

im=imread('football.jpg');

imR=im(:,:,1);

im3=1:1:256;

for k=1:256

im3(k)=0;

end

for i=1:256

for j=1:320

im3(imR(i,j))=im3(imR(i,j))+1;

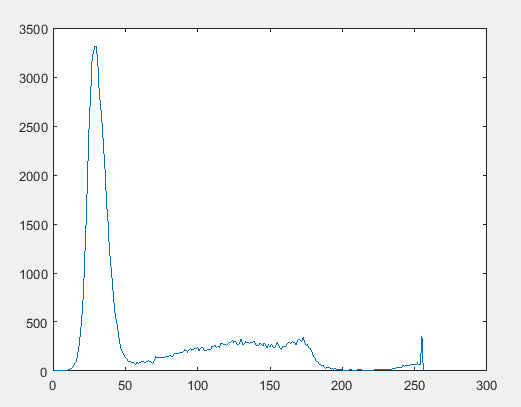
end

end

X=1:1:256;

Y=im3(X);

figure(2),plot(X,Y);



**6. Create a negative image from a grayscale image**

imc=imread('cameraman.tif');

[m,n]=size(imc);

for i=1:m

for j=1:n

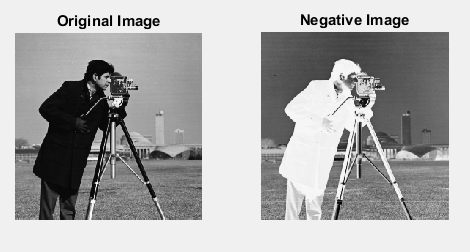
s(i,j)=256-1-imc(i,j); % s=l-1-r

end

end

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

subplot(1,2,2);imshow(s);title('Negative Image');



**7. Log Transform of an image**

A=input('Enter the value of constant A=');

a=imread('cameraman.tif');

[m,n]=size(a);

b=im2double(a);

for i=1:m

for j=1:n

r=b(i,j);

s(i,j)=A\*log(r+1);

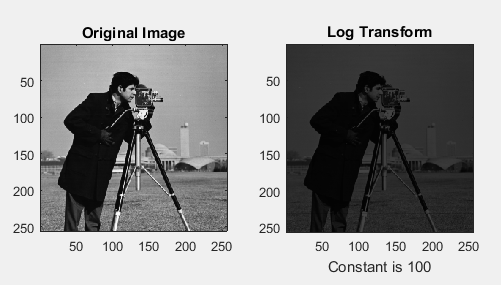
end

end

figure,

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

subplot(1,2,2);subimage(uint8(s));title('Log Transform');xlabel(sprintf('Constant is %g',A));



**8. Power law transform of an image**

A=input('Enter Constant Value=');

x=input('Enter gamma value=');

a=imread('cameraman.tif');

[m,n]=size(a);

for i=1:m

for j=1:n

s(i,j)=A\*(a(i,j)^x);

end

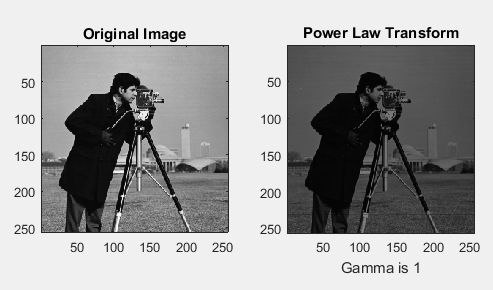
end

figure,

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

subplot(1,2,2);subimage(s);title('Power Law Transform');xlabel(sprintf('Constant is %g',A));

xlabel(sprintf('Gamma is %g',x));



**9. Contrast Stretching Of an Image**

x=input('Enter factor value=');

a=imread('cameraman.tif');

[m,n]=size(a);

for i=1:m

for j=1:n

s(i,j)=x\*(a(i,j));

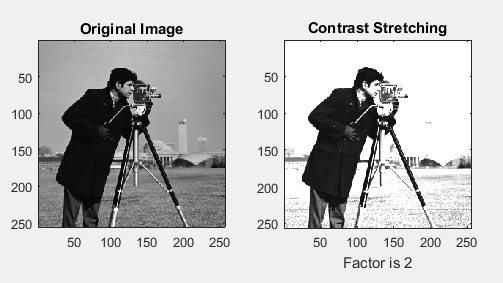
end

end

figure,

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

subplot(1,2,2);subimage(s);title('Contrast Stretching');xlabel(sprintf('Factor is %g',x));



Lab 2

1. **Gaussian Noise**

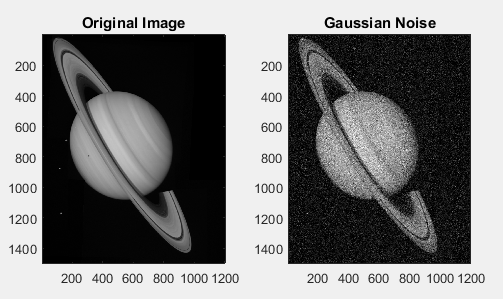
a=imread('saturn.png');

gr=rgb2gray(a);

n=imnoise(gr,'gaussian',0,0.025);

subplot(1,2,1);subimage(gr);title('Original Image');

subplot(1,2,2);subimage(n);title('Gaussian Noise');



1. **Salt Pepper Noise**

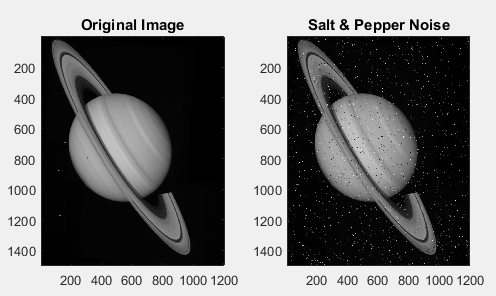
a=imread('saturn.png');

gr=rgb2gray(a);

n=imnoise(gr,'salt & pepper',0.02);

subplot(1,2,1);subimage(gr);title('Original Image');

subplot(1,2,2);subimage(n);title('Salt & Pepper Noise');



1. **Applying Mask And Median Filter To Remove Noise**

a=imread('saturn.png');

gr=rgb2gray(a);

gr=im2double(gr);

n1=imnoise(gr,'gaussian',0,0.025);

n2=imnoise(gr,'salt & pepper',0.02);

mask=ones(3,3)/9;

r1=imfilter(n1,mask);

r2=imfilter(n2,mask);

r3=medfilt2(n1);

r4=medfilt2(n2);

figure(1);

subplot(2,2,1);imshow(n1);title('Gaussian Noise');

subplot(2,2,2);imshow(r1);title('After Removing Noise');

subplot(2,2,3);imshow(n2);title('Salt & Pepper Noise');

subplot(2,2,4);imshow(r2);title('After Removing Noise');

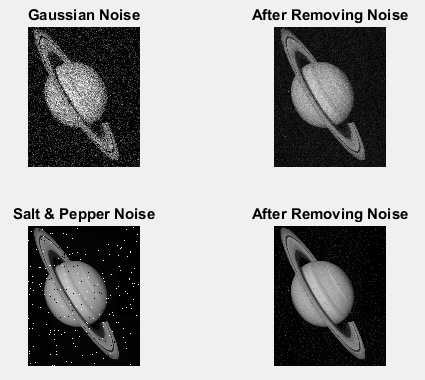
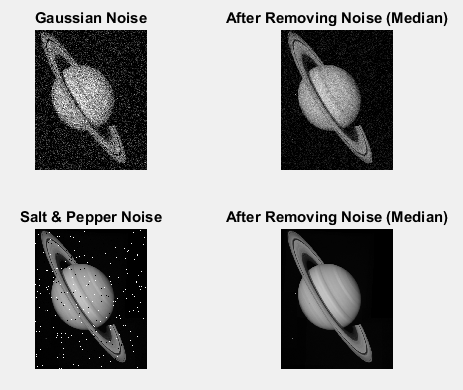
figure(2);

subplot(2,2,1);imshow(n1);title('Gaussian Noise');

subplot(2,2,2);imshow(r3);title('After Removing Noise (Median)');

subplot(2,2,3);imshow(n2);title('Salt & Pepper Noise');

subplot(2,2,4);imshow(r4);title('After Removing Noise (Median)');



1. **Edge Detection**

A=imread('peppers.png');

B=rgb2gray(A);

C=double(B);

for i=1:size(C,1)-2

for j=1:size(C,2)-2

%Sobel mask for x-direction:

Gx=((2\*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2\*C(i,j+1)+C(i,j)+C(i,j+2)));

%Sobel mask for y-direction:

Gy=((2\*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2\*C(i+1,j)+C(i,j)+C(i+2,j)));

%The gradient of the image

%B(i,j)=abs(Gx)+abs(Gy);

B(i,j)=sqrt(Gx.^2+Gy.^2);

end

end

figure,imshow(B); title('Sobel gradient');

[m,n]=size(B);

for i=1:m

for j=1:n

if(B(i,j)>100)

im2(i,j)=0;

else

im2(i,j)=255;

end

end

end

figure(1),imshow(im2);



1. **InterPolation**

im=imread('football.jpg');

a=imresize(im,0.5,'nearest');

b=imresize(im,0.5,'bilinear');

c=imresize(im,0.5,'bicubic');

figure;

subplot(1,3,1);imshow(a);title('Nearest');

subplot(1,3,2);imshow(b);title('Bilinear');

subplot(1,3,3);imshow(c);title('Bicubic');

