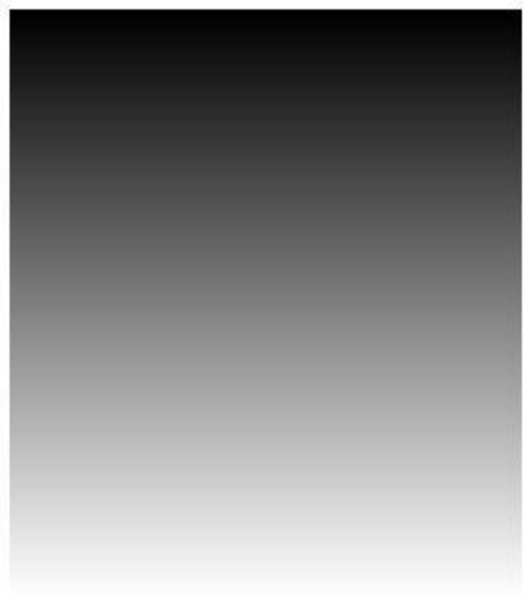
Experiment No: 1

GRADIENT

Aim: To display a 256x256 image matrix where the intensity level of each line changes from 0 to 255.

Program:

```
A = ones(1,256);

B = [0:255];

C = (uint8(A'*B))';

imshow(C);
```



Result: The script was successfully executed.

Experiment No: 2

IMAGE QUANTIZATION AND SAMPING

Aim: To perform image quantization and sampling at different rates.

Program:

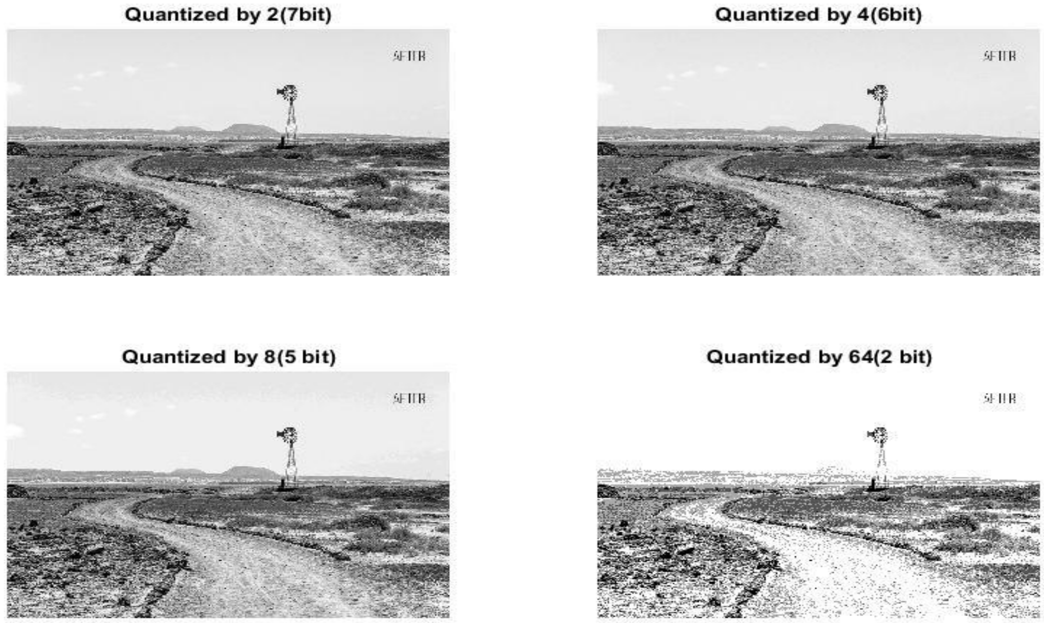
```
%Quantization

A=imread('expt 2.jpg');
B=rgb2gray(A);
imshow(B);
title('Original Image(8 bit)');

Q1=B/2;
subplot(2,2,1);
imshow(Q1,[0,127]);
title('Quantized by 2(7bit)');

Q2=B/4;
subplot(2,2,2);
imshow(Q2,[0,63]);
title('Quantized by 4(6bit)');

Q2=B/8;
subplot(2,2,3);
imshow(Q2,[0,31]);
title('Quantized by 8(5 bit)');

Q3=B/64;
subplot(2,2,4);
imshow(Q3,[0,3]);
title('Quantized by 64(2 bit)');

%Sampling

S1=imresize(B,[480 640]/4);
subplot(2,2,1);
imshow(S1);
title('Sampled by 2');

S2=imresize(B,[480 640]/4);
subplot(2,2,2);
imshow(S2);
```

```
title('Sampled by 4');

S3=imresize(B,[480 640]/8);
subplot(2,2,3);
imshow(S3);
title('Sampled by 8');

S4=imresize(C,[480 640]/64);
subplot(2,2,4);
imshow(S4);
title('Sampled by 64');
```

Observation:

Original Image

After Quantization



Quantized by 2(7bit)

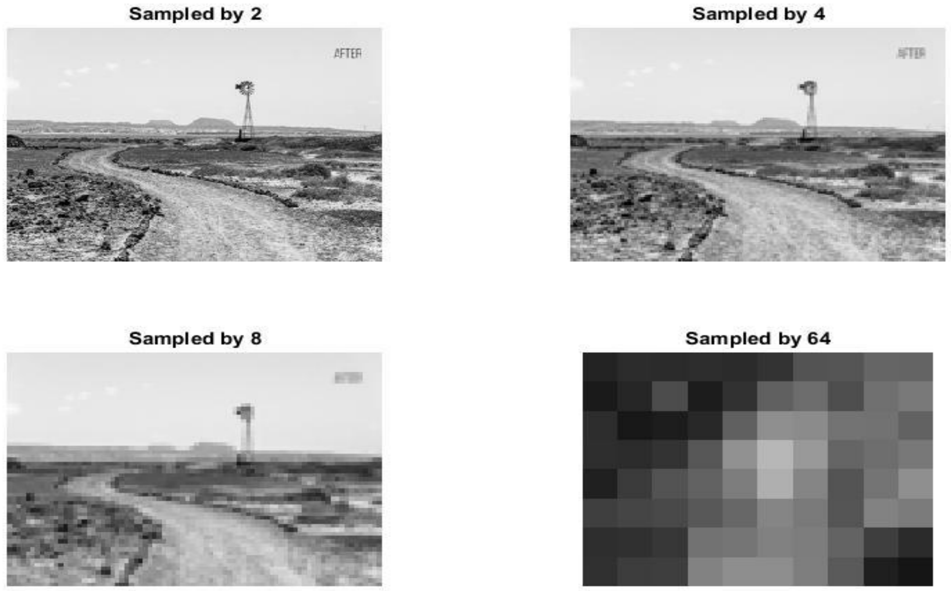
Quantized by 4(6bit)

Quantized by 8(5 bit)

Quantized by 64(2 bit)

It was observed that if quantization value is increased the amount of grey level available to represent the image decreases.

After Sampling



Sampled by 2

Sampled by 4

Sampled by 8

Sampled by 64

It was observed that as the sampling value is increased the image becomes blurred

Experiment No: 3

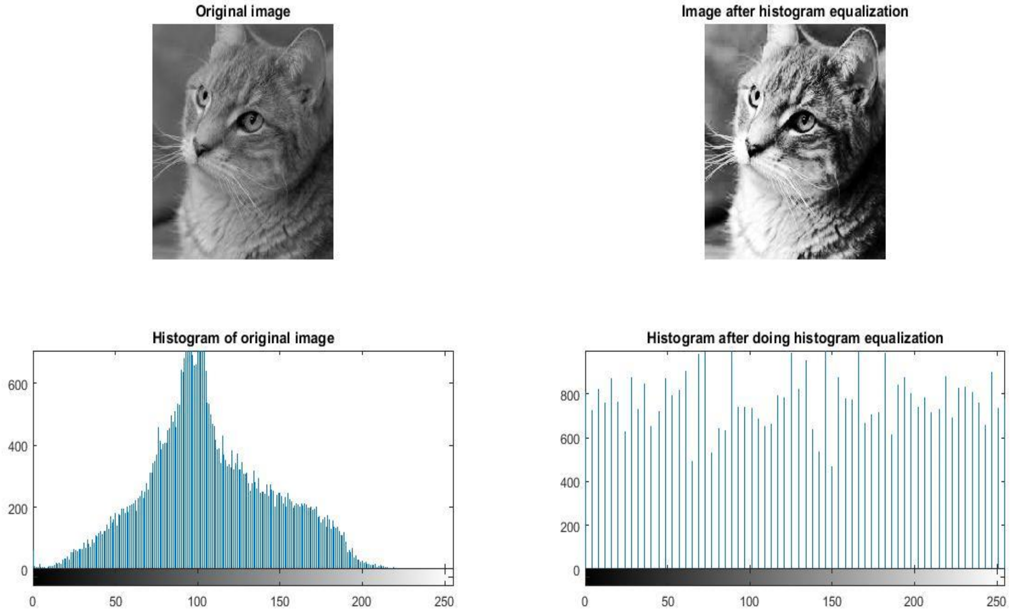
HISTOGRAM EQUALISATION

Aim: To perform Histogram Equalisation on a low contrast image.

Program:

```
A=imread('hist.jpg');
B=rgb2gray(A);
C=histeq(B);
subplot(2,2,1);
imshow(B);
title('Original image');
subplot(2,2,2);
imshow(C);
title('Image after histogram equalization');
subplot(2,2,3);
imhist(B);
title('Histogram of original image');
subplot(2,2,4);
imhist(C);
title('Histogram after doing histogram equalization');
```

Observation:



Result: Histogram equalization was performed on the image and it was observed by comparing the histograms that the pixel values after equalization get uniformly distributed, thus increasing the dynamic range of the initially washed out image.

Experiment No: 4

CONTRAST STRETCHING

Aim: To perform contrast stretching on washed out (low contrast) image.

Program:

```
A=imread('contrastlow.jpg');
B=rgb2gray(A);
subplot(2,2,1);
imshow(B);
title('Original Image');
subplot(2,2,3);
imhist(B);
title('Histogram of Original Image');
m=min(min(B));
n=max(max(B));
for i=1:size(B,1)
for j=1:size(B,2)
X(i,j)=uint8(255*(double(B(i,j)-m)/double(n-m)));
end
end
 subplot(2,2,2);
imshow(X);
title('Image after contrast stretching');
subplot(2,2,4);
imhist(X);
title('Histogram after contrast stretching');
```

Observations:

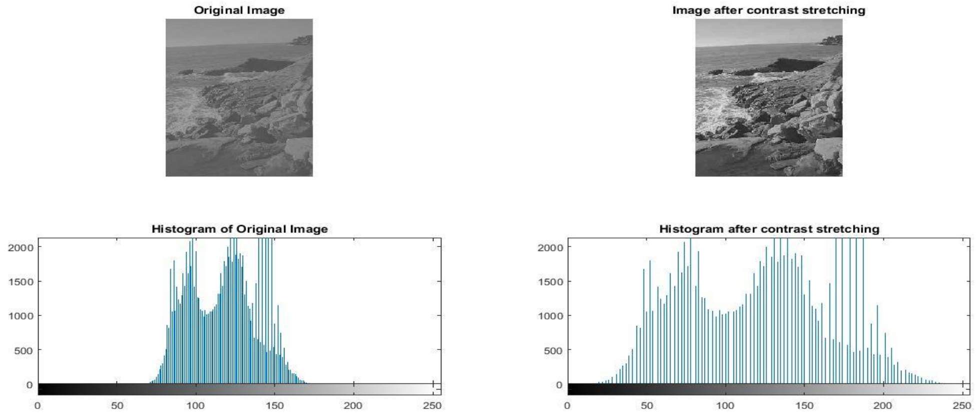

Original Image

Image after contrast stretching

Histogram of Original Image

Histogram after contrast stretching

It was observed that performing contrast stretching increases the dynamic range of the image.

Experiment No: 5:

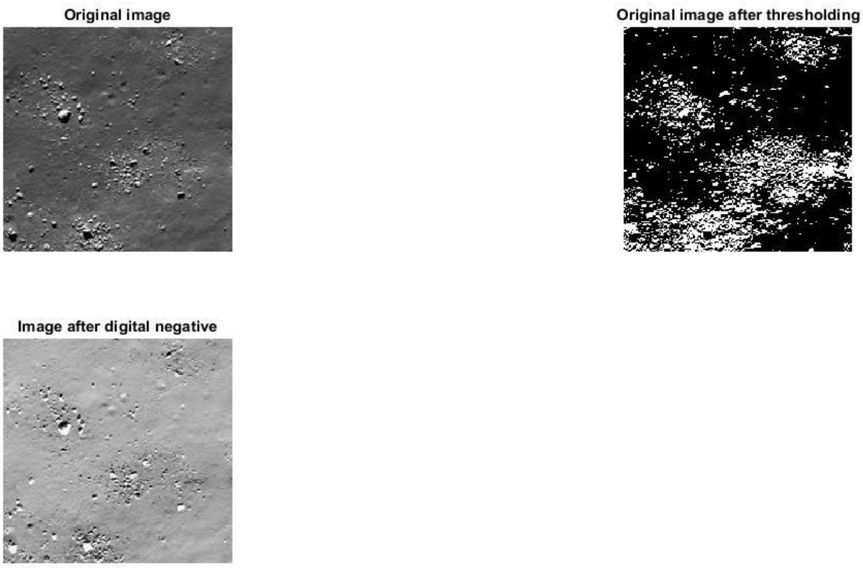
THRESHOLDING AND DIGITAL NEGATIVE

Aim: To perform thresholding on a high contrast image.

Program:

```
A=imread('thresholding.jpg');
B=rgb2gray(A);
subplot(2,2,1);
imshow(B);
title('Original image');
K=(B>100);
subplot(2,2,2);
imshow(K);
title('Original image after thresholding');

for i=1:size(B,1)
for j=1:size(B,2)
N(i,j)=uint8(abs(255-B(i,j)));
end
end
subplot(2,2,3);
imshow(N);
title('Image after digital negative');
```

Observations:



Original image

Original image after thresholding

Image after digital negative

Experiment No: 6

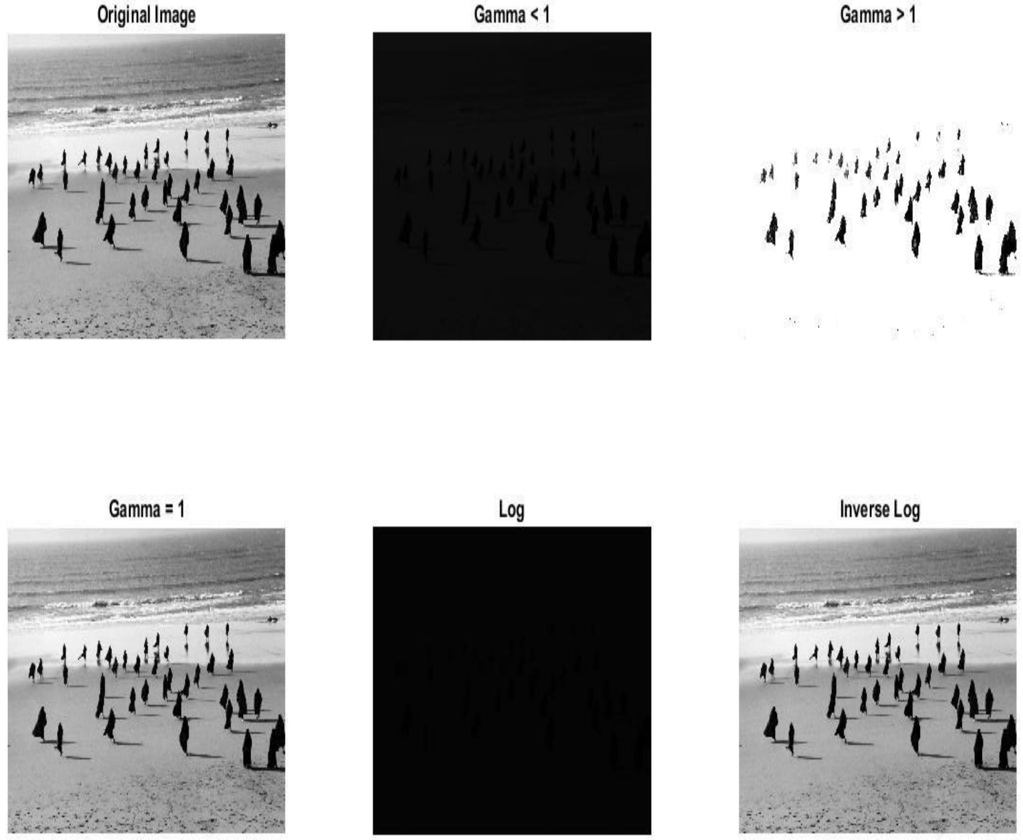
Gamma, Log and Inverse

Aim: To perform log, gamma and inverse log on the image.

Program:

```
B=imread('restoration.jpg');
subplot(2,3,1);
imshow(B);
title('Original Image');

A=double(imread('restoration.jpg'));
X=power(A,0.5);
subplot(2,3,2);
imshow(uint8(X));
title('Gamma < 1');

k=power(A,1.7);
subplot(2,3,3);
imshow(uint8(k));
title('Gamma > 1');

M=power(A,1);
subplot(2,3,4);
imshow(uint8(M));
title('Gamma = 1');

L=log(1+I);
subplot(2,3,5);
imshow(uint8(L));
title('Log');

iL=exp(L)-1;
subplot(2,3,6);
imshow(uint8(iL));
title(' Inverse Log');
```

Observations:



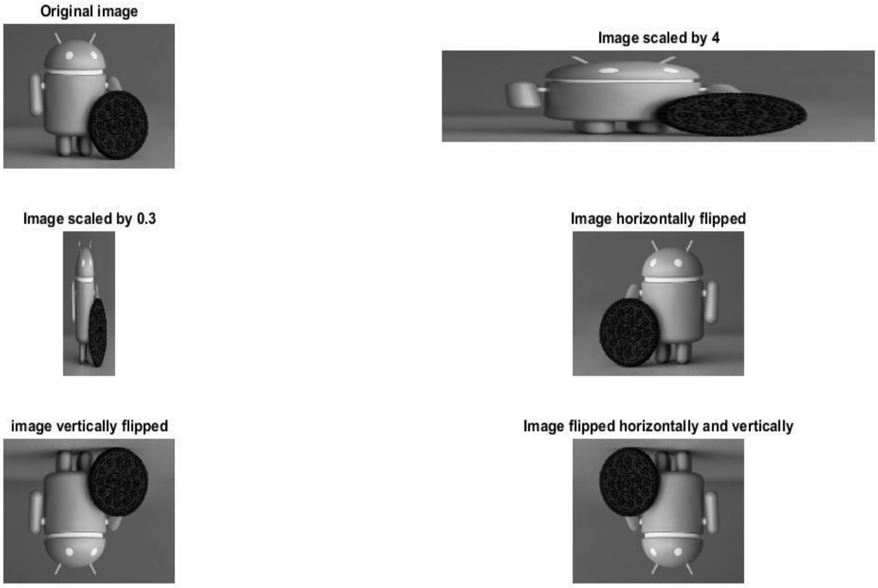
Experiment No: 7

GEOMETRIC OPERATIONS

Aim: To perform various geometric operations such as scaling, flipping, rotation and shear on the given image.

Geometric operations

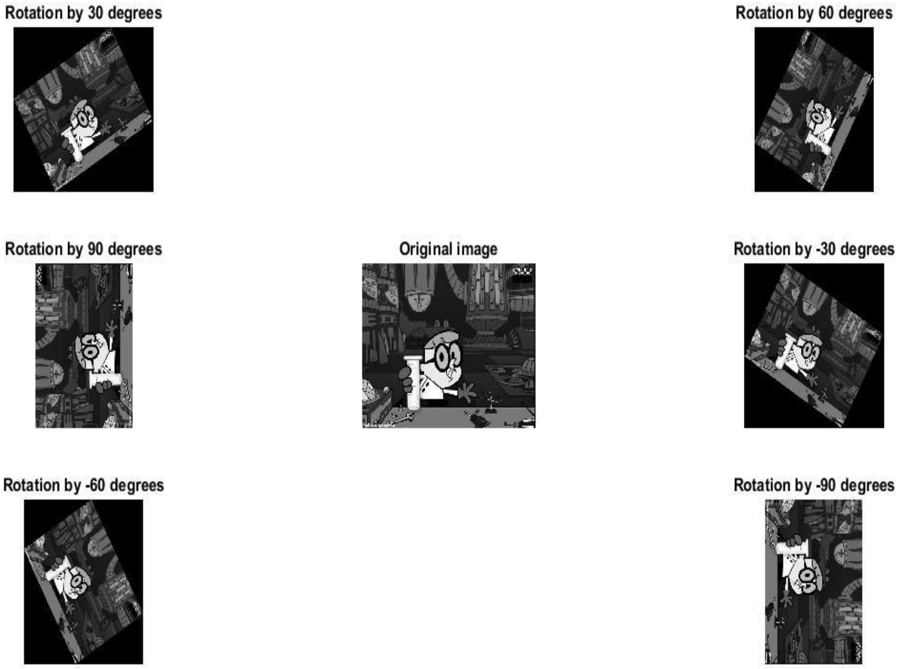
```
Image=rgb2gray(imread('android.jpg'));
imshow(Image);
img1=affine2d([4 0 0;0 1 0;0 0 1]); %scaling
img2=affine2d([0.3 0 0;0 1 0;0 0 1]); %scaling
img3=affine2d([-1 0 0;0 1 0;0 0 1]);%flipping
img4=affine2d([1 0 0;0 -1 0;0 0 1]);%flipping
img5=affine2d([-1 0 0;0 -1 0;0 0 1]);%flipping
C1=imwarp(Image,img1);
C2=imwarp(Image,img2);
C3=imwarp(Image,img3);
C4=imwarp(Image,img4);
C5=imwarp(Image,img5);
subplot(3,2,1);
imshow(Image);
title('Original image');
subplot(3,2,2);
imshow(C1);
title('Image scaled by 4');
subplot(3,2,3);
imshow(C2);
title('Image scaled by 0.3');
subplot(3,2,4);
imshow(C3);
title('Image horizontally flipped');
subplot(3,2,5);
imshow(C4);
title('image vertically flipped');
subplot(3,2,6);
imshow(C5);
title('Image flipped horizontally and vertically');
```



Rotate:

```
Image=rgb2gray(imread('dexter.jpg'));
subplot(3,3,5);
imshow(Image);
title('Original image');


img1=affine2d([cosd(30) -sind(30) 0;sind(30) cosd(30) 0;0 0
1]);%30
img2=affine2d([cosd(60) -sind(60) 0;sind(60) cosd(60) 0;0 0
1]);%60
img3=affine2d([cosd(90) -sind(90) 0;sind(90) cosd(90) 0;0 0
1]);%90
img4=affine2d([cosd(-30) -sind(-30) 0;sind(-30) cosd(-30) 0;0
0 1]);%-30
img5=affine2d([cosd(-60) -sind(-60) 0;sind(-60) cosd(-60) 0;0
0 1]);%-60
img6=affine2d([cosd(-90) -sind(-90) 0;sind(-90) cosd(-90) 0;0
0 1]);%-90
C1=imwarp(Image,img1);
C2=imwarp(Image,img2);
C3=imwarp(Image,img3);
C4=imwarp(Image,img4);
C5=imwarp(Image,img5);
C6=imwarp(Image,img6);
subplot(3,3,1);

imshow(C1);
title('Rotation by 30 degrees');
```

```
subplot(3,3,3);
imshow(C2);
title('Rotation by 60 degrees');
subplot(3,3,4);
imshow(C3);
title('Rotation by 90 degrees');
subplot(3,3,6);
imshow(C4);
title('Rotation by -30 degrees');
subplot(3,3,7);
imshow(C5);
title('Rotation by -60 degrees');
subplot(3,3,9);
imshow(C6);
title('Rotation by -90 degrees');
```



Rotation by 30 degrees

Rotation by 60 degrees

Rotation by 90 degrees

Original image

Rotation by -30 degrees

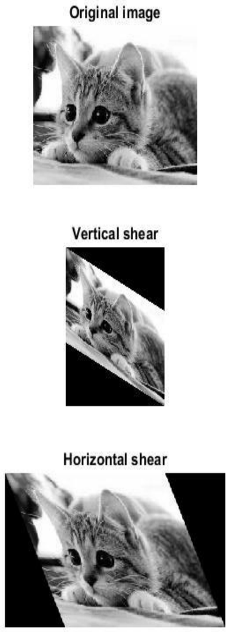
Rotation by -60 degrees

Rotation by -90 degrees

Shear

```
Image=rgb2gray(imread('cat.jpg'));
img1=affine2d([1 0.5 0;0 1 0;0 0 1]);%vertical
img2=affine2d([1 0 0;0.5 1 0;0 0 1]);%horizontal
C1=imwarp(Image,img1);
C2=imwarp(Image,img2);
subplot(3,1,1);
imshow(Image);
title('Original image');
subplot(3,1,2);
```

```
imshow(C1);
title('Vertical shear');
subplot(3,1,3);
imshow(C2);
title('Horizontal shear');
```



Results: The specified geometric operations were performed on the image and effects were observed.

Experiment No: 8

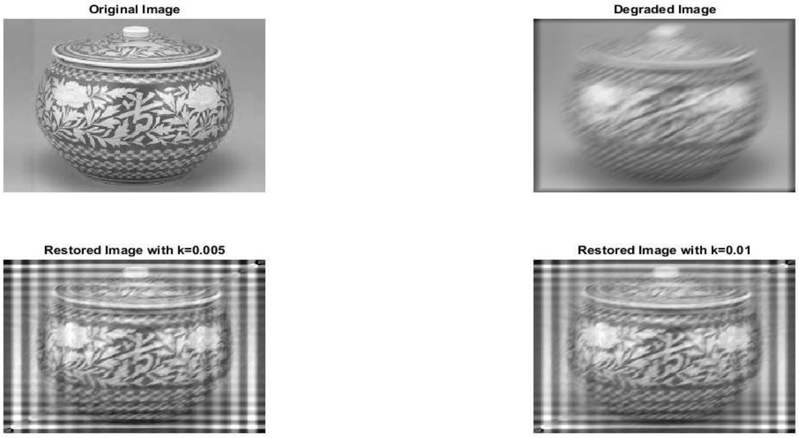
Image Restoration

Aim: To restore a motion degraded image using wiener filter for different values of k.

Program:

```
Imgage=imread('pot.jpg');
Imgage=rgb2gray(Imgage);
subplot(2,2,1);
imshow(Imgage);
title('Original Image')
[N,M]=size(Imgage);
k1=0.005;
k2=0.01;
LEN = 20;
THETA = 30;
Deg = fspecial('motion',LEN,THETA);
image2 = imfilter(Imgage,Deg);
subplot(2,2,2);
imshow(image2)
title('Degraded Image')
Img3=deconvwnr(image2, Deg, k1);
subplot(2,2,3);
imshow(Img3)
title('Restored Image with k=0.005')
Img4=deconvwnr(image2, Deg, k2);
subplot(2,2,4);
imshow(Img4);
title('Restored Image with k=0.01');
```

Observations:



Original Image

Degraded Image

Restored Image with k=0.005

Restored Image with k=0.01

Experiment No: 6

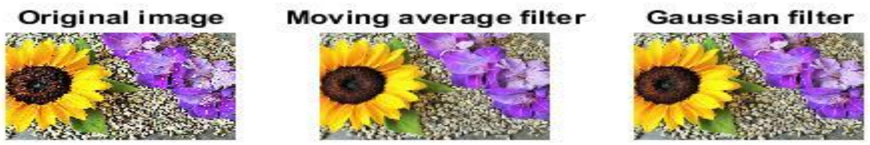
SPATIAL DOMAIN FILTERS

Aim: To implement Spatial domain Low pass and High pass filters.

Program:

Low Pass Filter

```
Image=imread('flower.jpg');
avg=ones(3,3)/9;
Image2=imfilter(Image,avg);
subplot(1,3,1);
imshow(Image);
title('Original image');
subplot(1,3,2);
imshow(Image2);
title('Moving average filter');
gaussian=[1 2 1;2 4 2;1 2 1]/16;
Image3=imfilter(Image,gaussian);
subplot(1,3,3);
imshow(Image3);
title('Gaussian filter');
```



High Pass Filter

```
Image=rgb2gray(imread('coins.jpg'));
laplaceT=[0 1 0;1 -4 1;0 1 0];
ident=[0 0 0;0 1 0;0 0 0];
Image2=imfilter(Image,laplaceT);
subplot(3,2,1);
imshow(Image);title('Original image');
subplot(3,2,2);
imshow(Image2);title('Image after laplacian masking');
sharp=ident-laplaceT;
Image3=imfilter(Image,sharp);
```

```
subplot(3,2,[3,4]);
imshow(Image3);title('Sharpened image');
gau=[1 2 1;2 4 2;1 2 1]/16;
highPass=ident-gau;
unMask=ident+(0.1*highPass);
Image4=imfilter(Image,unMask);

subplot(3,2,5);
imshow(Image4);title('Unsharped image');
boost=ident+(7*highPass);
E=imfilter(Image,boost);
subplot(3,2,6);
imshow(E);
title('High boost filtering');
```