**Practical – 3**

**Aim:** Spatial Domain Filters

1. **Box Filtering:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

% appling box filter size of 3\*3

f = [1 1 1; 1 1 1; 1 1 1]/9;

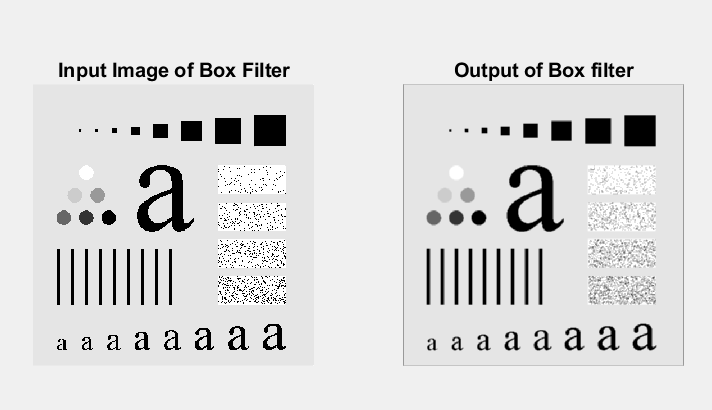
outImg = imfilter(inImg , f);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image of Box Filter');

subplot(1,2,2);imshow(outImg);title('Output of Box filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that in box filtering we use mask with all the value is 1 and use to smoothing of the image.

1. **Weighted Average Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

% appling weighted average filter size of 3\*3

f = [1 2 1; 2 4 2; 1 2 1]/16;

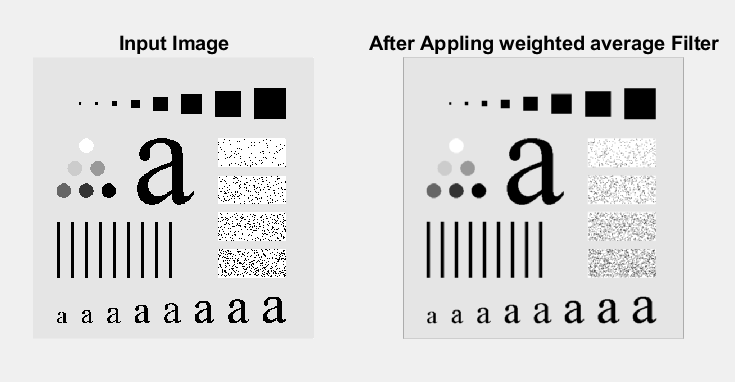
outImg = imfilter(inImg,f);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(outImg);title('After Appling weighted average Filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that in weighted average filter we sum all the neighbouring pixel intensity with multiplication with mask of filter.

1. **Median Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e15.tif');

% appling median Filter of size 3\*3

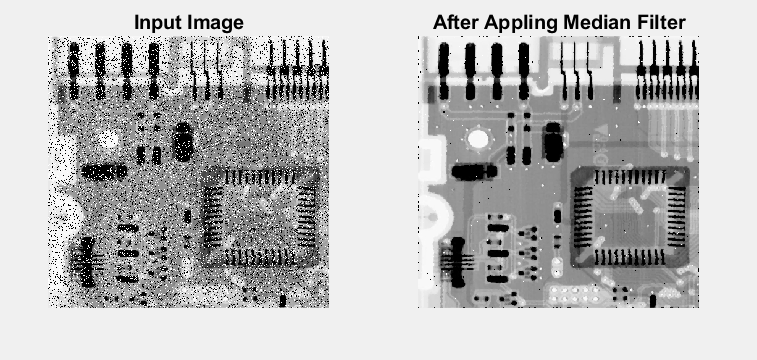
outImg = medfilt2(inImg, [3,3]);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(outImg);title('After Appling Median Filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that in median filter take the median value from all the overlapping pixels in the matrix.

1. **Min Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e15.tif');

% appling min Filter of size 3\*3

f = ones(3,3);

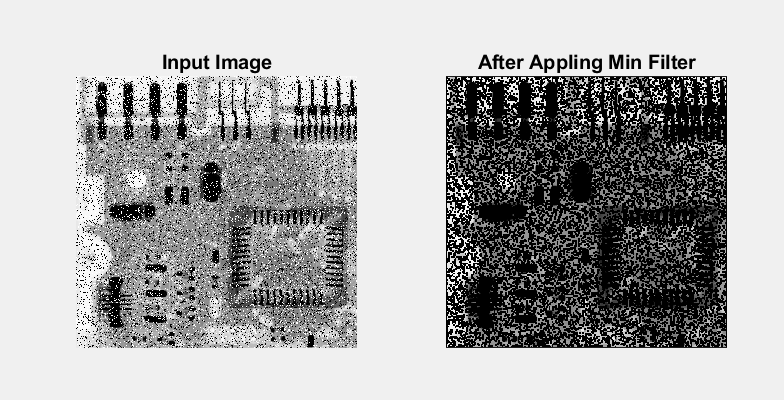
outImg = ordfilt2(inImg,1,f);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(outImg);title('After Appling Min Filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that in min filter take the minimum value from all the overlapping pixels in the matrix.

1. **Max Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e15.tif');

% appling max Filter of size 3\*3

f = ones(3,3);

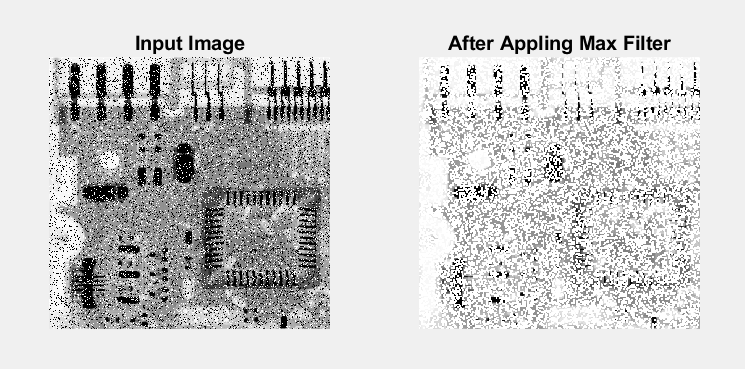
outImg = ordfilt2(inImg,3^2,f);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(outImg);title('After Appling Max Filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that in max filter take the maximum value from all the overlapping pixels in the matrix.

1. **Laplacian Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e16.tif');

% intialize laplasian filter

f = [0 1 0; 1 -4 1; 0 1 0];

% Apply laplacian filter

outImg = imfilter(inImg,f);

% make sharp image of it

sharpImg = imsubtract(inImg,outImg);

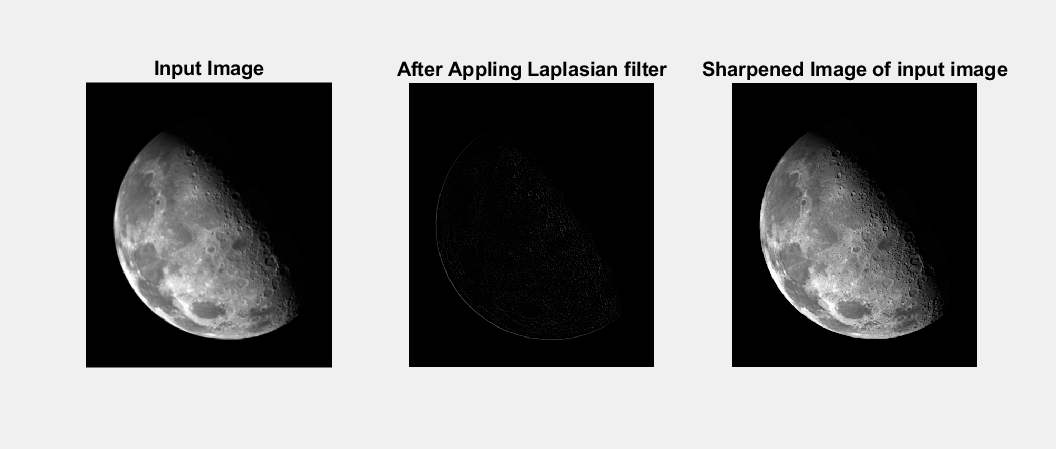
%Display input and output images in plot

subplot(1,3,1);imshow(inImg);title('Input Image');

subplot(1,3,2);imshow(outImg);title('After Appling Laplacian filter');

subplot(1,3,3);imshow(sharpImg);title('Sharpened Image of input image');

**Output:**



**Conclusion:**

By performing this practical we get to know that in Laplacian filter detect edges in the entire image in both dimensions.

1. **Gradient Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

% appling gradient filter

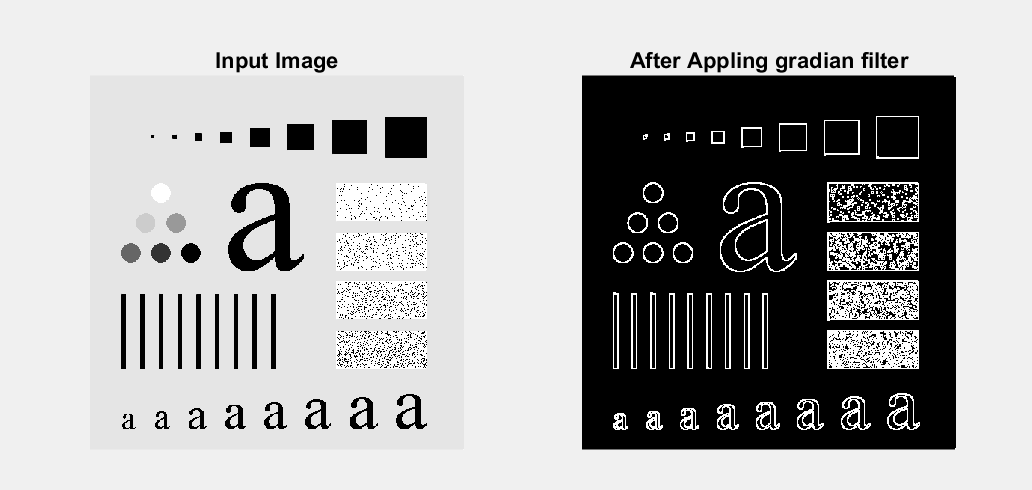
outImg = imgradient(inImg);

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(outImg);title('After Appling gradient filter');

**Output:**



**Conclusion:**

By performing this practical we get to know that Gradient filter is much like Laplacian filter but it uses first order derivative.