Image and Video Processing

Programming Assignment 5

Smoothing Filters

Submitted by
Madhu Krishnan A P
(Student ID: 24100488)
M.Tech VLSI and Embedded Systems
Cochin University of Science and Technology
Cochin - 22

1 Question 1

Implement the following spatial filters for different types of noises - Salt & pepper noise, Gaussian noise, Speckle noise

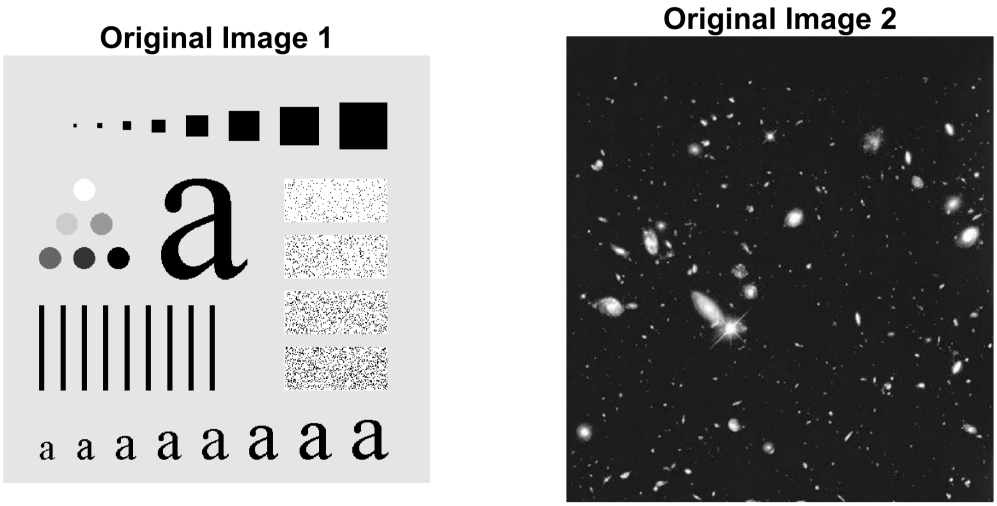
- Implement a Low Pass Box Filter
- Implement weighted average LPF
- Implement a median filter

Repeat the same for mask sizes of 5x5, and 7x7. What is your inference?

Low Pass Box Filter

Define Image

```
i1 = imread('Fig0333(a)(test_pattern_blurring_orig).tif');
i2 = imread('Fig0334(a)(hubble-original).tif');
figure;
subplot(1, 2, 1), imshow(i1), title('Original Image 1');
subplot(1, 2, 2), imshow(i2), title('Original Image 2');
```



Define and display filters

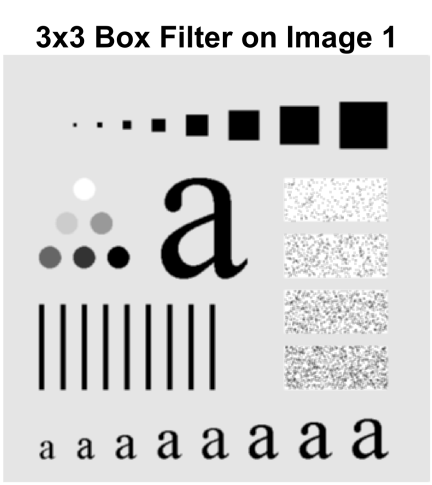
```
box3 = ones(3,3) / 9;
box5 = ones(5,5) / 25;
box7 = ones(7,7) / 49;
disp('Box Filters Defined');
```

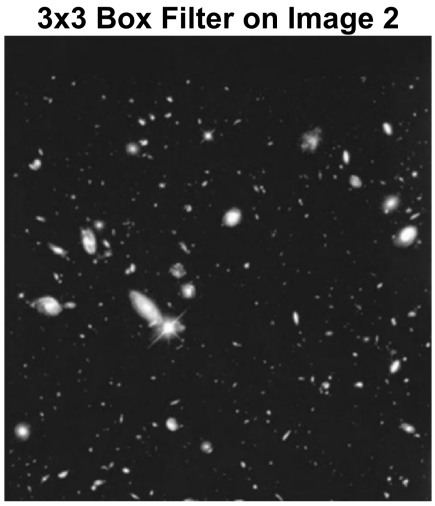
```
Box Filters Defined
```

Apply 3x3 filters

```
img3_1 = imfilter(i1, box3, 'replicate');
img3_2 = imfilter(i2, box3, 'replicate');
figure;
subplot(1, 2, 1), imshow(img3_1), title('3x3 Box Filter on Image 1');
```

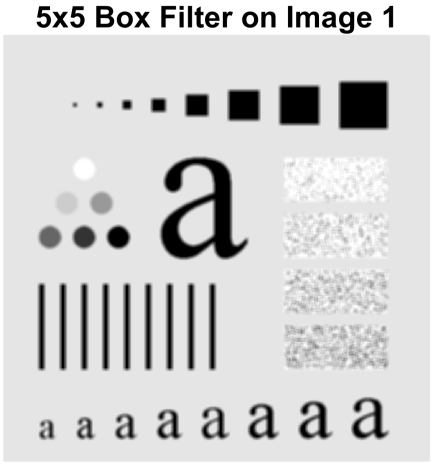
```
subplot(1, 2, 2), imshow(img3_2), title('3x3 Box Filter on Image 2');
```

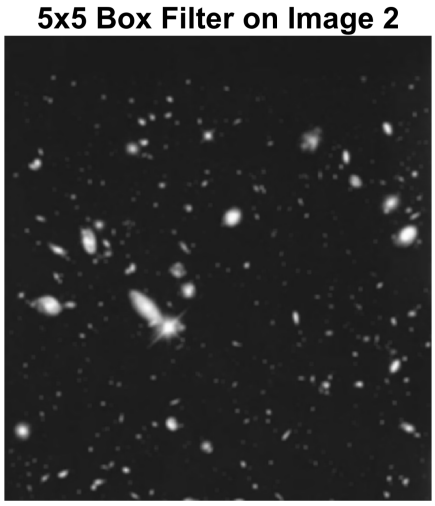




Apply 5x5 filter

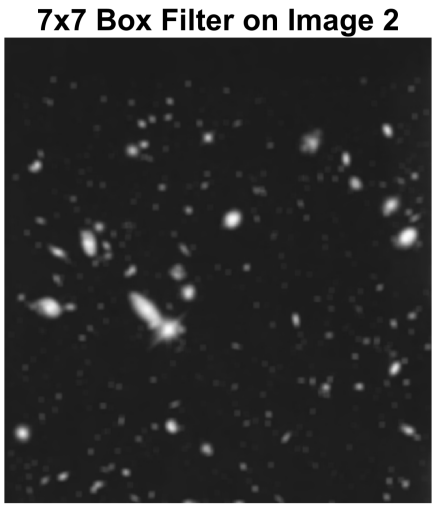
```
img5_1 = imfilter(i1, box5, 'replicate');
img5_2 = imfilter(i2, box5, 'replicate');
figure;
subplot(1, 2, 1), imshow(img5_1), title('5x5 Box Filter on Image 1');
subplot(1, 2, 2), imshow(img5_2), title('5x5 Box Filter on Image 2');
```





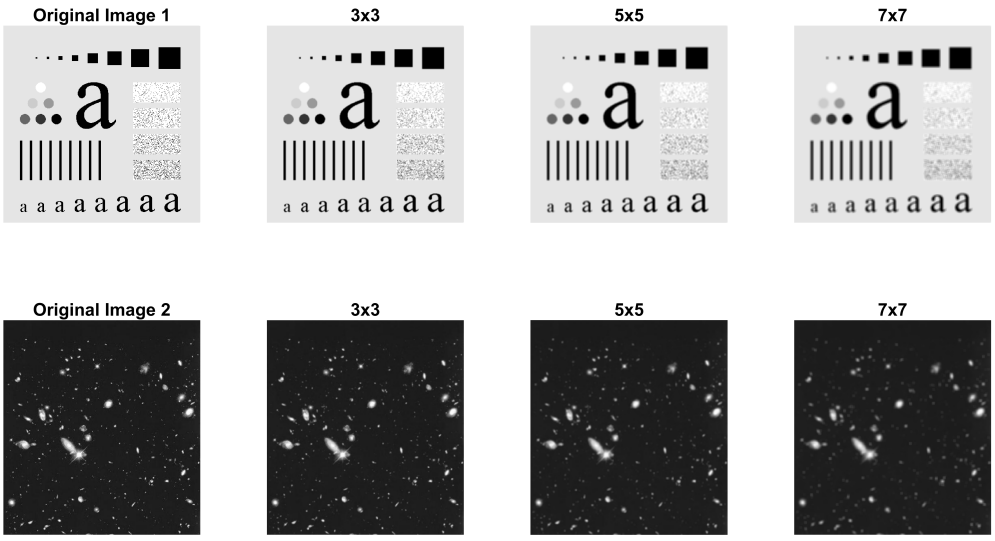
Apply 7x7 filters

```
img7_1 = imfilter(i1, box7, 'replicate');
img7_2 = imfilter(i2, box7, 'replicate');
figure;
subplot(1, 2, 1), imshow(img7_1), title('7x7 Box Filter on Image 1');
subplot(1, 2, 2), imshow(img7_2), title('7x7 Box Filter on Image 2');
```



Result

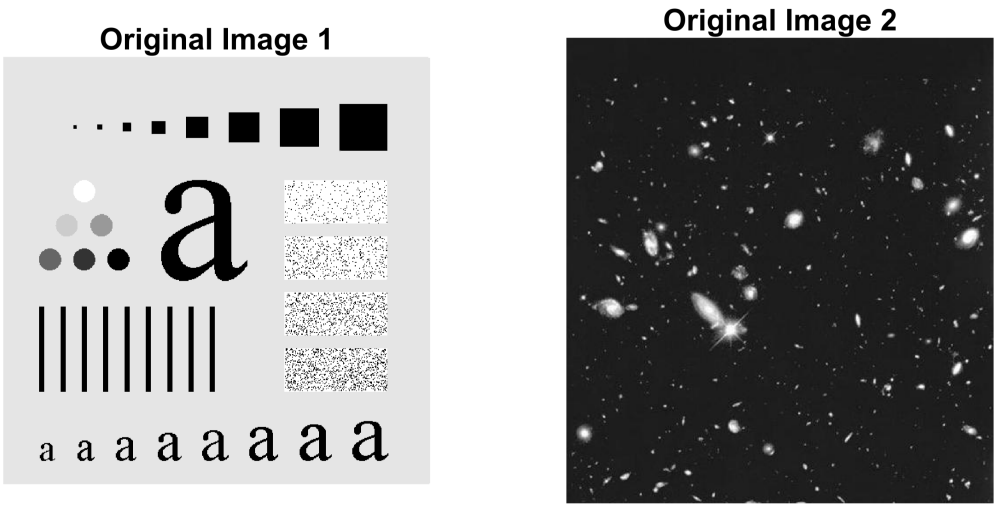
```
figure('Position', [100, 100, 1400, 700]);

subplot(2, 4, 1), imshow(i1), title('Original Image 1');
subplot(2, 4, 2), imshow(img3_1), title('3x3');
subplot(2, 4, 3), imshow(img5_1), title('5x5');
subplot(2, 4, 4), imshow(img7_1), title('7x7');

subplot(2, 4, 5), imshow(i2), title('Original Image 2');
subplot(2, 4, 6), imshow(img3_2), title('3x3');
subplot(2, 4, 7), imshow(img5_2), title('5x5');
subplot(2, 4, 8), imshow(img7_2), title('7x7');
```



Weighted Average Filters (Low-Pass Filter)

Define Image

```
i1 = imread('Fig0333(a)(test_pattern_blurring_orig).tif');
i2 = imread('Fig0334(a)(hubble-original).tif');
figure;
subplot(1, 2, 1), imshow(i1), title('Original Image 1');
subplot(1, 2, 2), imshow(i2), title('Original Image 2');
```



Define Weighted Average Filters (Low-Pass Filter)

3x3 weighted average filter

```
w3 = [1 2 1; 2 4 2; 1 2 1] / 16;
```

5x5 weighted average filter

```
w5 = [1 4 6 4 1; 4 16 24 16 4; 6 24 36 24 6; 4 16 24 16 4; 1 4 6 4 1] / 256;
```

7x7 weighted average filter

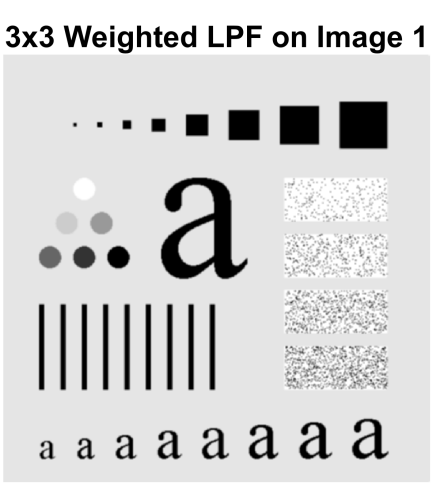
```
w7 = [1 6 15 20 15 6 1; 6 36 90 120 90 36 6; 15 90 225 300 225 90 15;
      20 120 300 400 300 120 20; 15 90 225 300 225 90 15;
      6 36 90 120 90 36 6; 1 6 15 20 15 6 1] / 784;

disp('Weighted Average LPF Filters Defined');
```

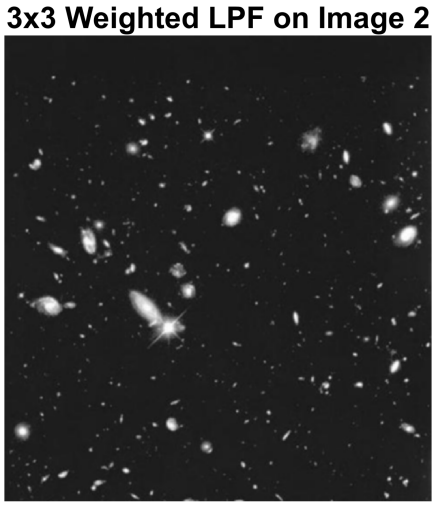
```
Weighted Average LPF Filters Defined
```

Apply 3x3 Weighted Filter

```
imgw3_1 = imfilter(i1, w3, 'replicate');
imgw3_2 = imfilter(i2, w3, 'replicate');
figure;
```

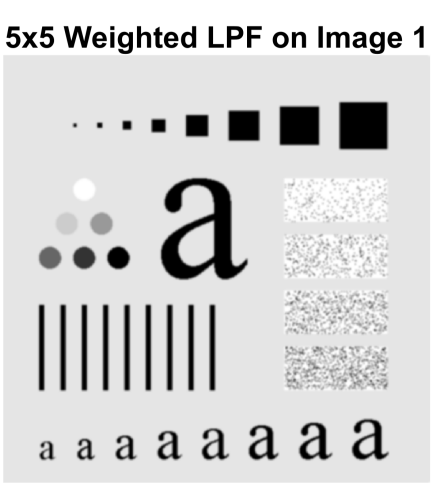
```
subplot(1, 2, 1), imshow(imgw3_1), title('3x3 Weighted LPF on Image 1');
subplot(1, 2, 2), imshow(imgw3_2), title('3x3 Weighted LPF on Image 2');
```

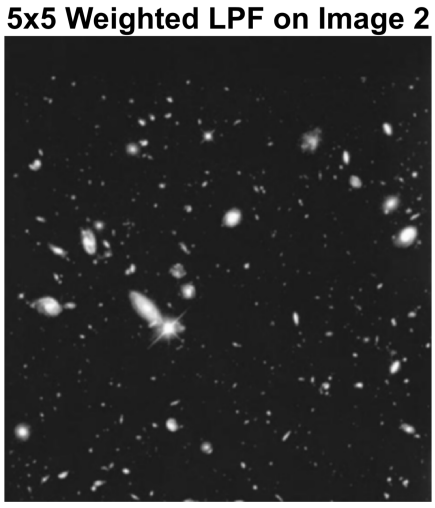




Apply 5x5 Weighted Filter

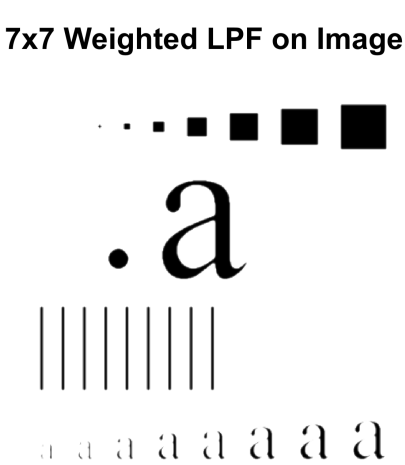
```
imgw5_1 = imfilter(i1, w5, 'replicate');
imgw5_2 = imfilter(i2, w5, 'replicate');
figure;
subplot(1, 2, 1), imshow(imgw5_1), title('5x5 Weighted LPF on Image 1');
subplot(1, 2, 2), imshow(imgw5_2), title('5x5 Weighted LPF on Image 2');
```

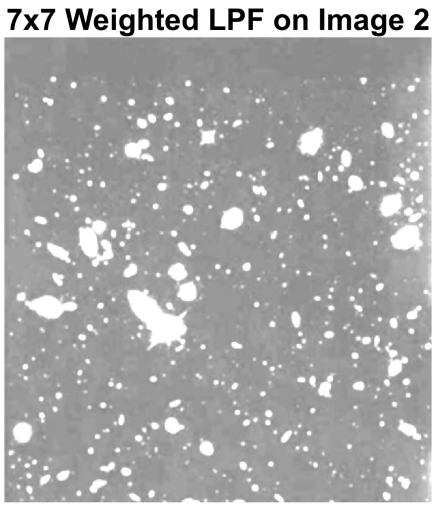




Apply 7x7 Weighted Filter

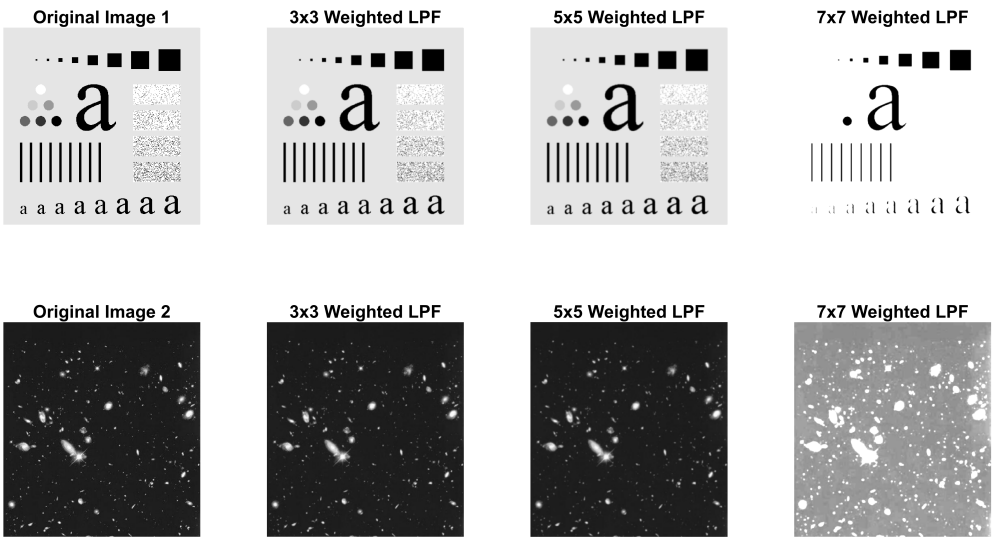
```
imgw7_1 = imfilter(i1, w7, 'replicate');
imgw7_2 = imfilter(i2, w7, 'replicate');
figure;
subplot(1, 2, 1), imshow(imgw7_1), title('7x7 Weighted LPF on Image 1');
subplot(1, 2, 2), imshow(imgw7_2), title('7x7 Weighted LPF on Image 2');
```





Display results for Weighted Filters

```
figure('Position', [100, 100, 1400, 700]);
subplot(2, 4, 1), imshow(i1), title('Original Image 1');
subplot(2, 4, 2), imshow(imgw3_1), title('3x3 Weighted LPF');
subplot(2, 4, 3), imshow(imgw5_1), title('5x5 Weighted LPF');
subplot(2, 4, 4), imshow(imgw7_1), title('7x7 Weighted LPF');
subplot(2, 4, 5), imshow(i2), title('Original Image 2');
subplot(2, 4, 6), imshow(imgw3_2), title('3x3 Weighted LPF');
subplot(2, 4, 7), imshow(imgw5_2), title('5x5 Weighted LPF');
subplot(2, 4, 8), imshow(imgw7_2), title('7x7 Weighted LPF');
```



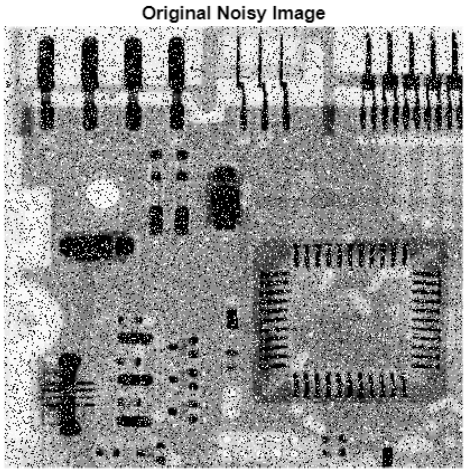
Median Filter

Read noisy image

```
i_noisy = imread('Fig0335(a)(ckt_board_saltpep_prob_pt05).tif');
```

Display original noisy image

```
figure;
imshow(i_noisy);
title('Original Noisy Image');
```

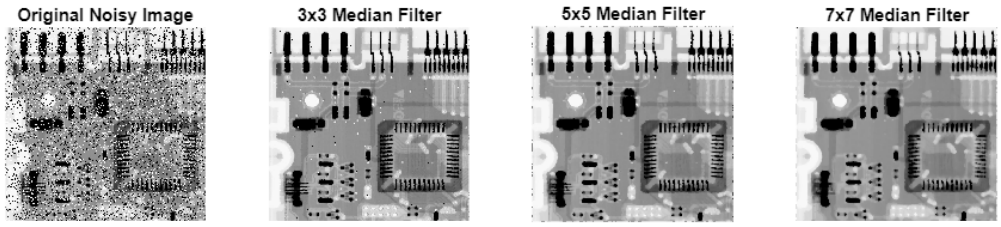


Apply Median Filters

```
img_med3 = medfilt2(i_noisy, [3 3]);  % 3x3 median filter
img_med5 = medfilt2(i_noisy, [5 5]);  % 5x5 median filter
img_med7 = medfilt2(i_noisy, [7 7]);  % 7x7 median filter
```

Display Results

```
figure('Position', [100, 100, 1400, 400]);
subplot(1, 4, 1), imshow(i_noisy), title('Original Noisy Image');
subplot(1, 4, 2), imshow(img_med3), title('3x3 Median Filter');
subplot(1, 4, 3), imshow(img_med5), title('5x5 Median Filter');
subplot(1, 4, 4), imshow(img_med7), title('7x7 Median Filter');
```



Inference

Low pass box filter, weighted average low pass filter, and median filters of sizes 3x3, 5x5, and 7x7 were applied to different images using Matlab. It was observed that the performance of spatial filters in removing noise varies significantly depending on the type of noise and the size of the filter mask applied.

The 3x3 Median Filter is the most effective method for removing salt & pepper noise. It replaces noisy pixel values with the median of neighbouring pixels, removing isolated black and white pixels without blurring edges. The Low Pass Box Filter and Weighted Average Filter also help reduce salt & pepper noise, but their effectiveness is limited. Larger mask sizes provide better noise reduction, but they tend to blur edges and smooth details.

For Gaussian noise, Low-Pass Filters (Box and Weighted Average) are more effective. These filters average pixel values, which helps in reducing the noise caused by random variations in pixel intensity. Larger masks provide more effective noise reduction but may cause significant blurring of image details. The Median Filter is less effective for Gaussian noise compared to salt & pepper noise.

Speckle noise is multiplicative, and the Low-Pass Filters help to reduce this noise by averaging pixel values. However, they also lead to a loss of fine details due to blurring, especially with larger mask sizes. The Median Filter offers a reasonable balance in noise reduction for speckle noise, but its performance is still not as robust as for salt & pepper noise.

Increasing the mask size generally improves the noise removal effectiveness across all filter types. Larger masks provide a broader neighbourhood for averaging or median computation, which helps in removing noise more effectively. However, this comes at the cost of blurring edges and fine details. Smaller masks tend to preserve image details better but are less effective at eliminating noise, particularly for Gaussian and speckle noise.

Conclusion

The Median Filter is particularly suitable for removing salt & pepper noise, while Low-Pass Filters (Box and Weighted) are more effective for Gaussian and Speckle noise. However, larger masks tend to blur fine image details. The choice of filter and mask size should be carefully selected based on the type of noise present and the level of detail preservation required for the application.