Palawan State University
College of Sciences
Computer Studies Department

DIGITAL IMAGE PROCESSING LABORATORY EXERCISE #8

Implementation of Image Smoothening Filters

(Mean and Median filtering of an Image)

The implementation of image smoothing filters involves applying techniques such as mean and median filtering to reduce noise and improve image quality. Here's an overview:

1. **Mean Filtering:**
    - Mean filtering involves replacing each pixel value in an image with the average value of its neighboring pixels.
    - A square or rectangular kernel (also known as a window) of a specified size is centered on each pixel, and the average of all pixel values within the kernel is computed.
    - This process effectively blurs the image, reducing high-frequency noise and sharp transitions between pixel values.
2. **Median Filtering:**
    - Median filtering replaces each pixel value with the median value of its neighboring pixels within a specified kernel size.
    - Unlike mean filtering, median filtering preserves edges and fine details in the image while effectively removing salt-and-pepper noise.
    - The median operation sorts the pixel values within the kernel and selects the middle value as the new pixel intensity.

The steps for implementing mean and median filtering of an image typically involve the following:

1. Define the size of the kernel/window for filtering.
2. Iterate through each pixel in the image.
3. For each pixel, extract the neighborhood defined by the kernel.
4. Compute the mean or median value of the pixel intensities within the neighborhood.
5. Replace the pixel value with the computed mean or median value.
6. Repeat this process for all pixels in the image.

Both mean and median filtering are widely used in image processing applications to enhance images for better visualization or prepare them for subsequent analysis tasks. They are particularly useful in scenarios where images are corrupted by noise, such as in medical imaging, remote sensing, and photography.

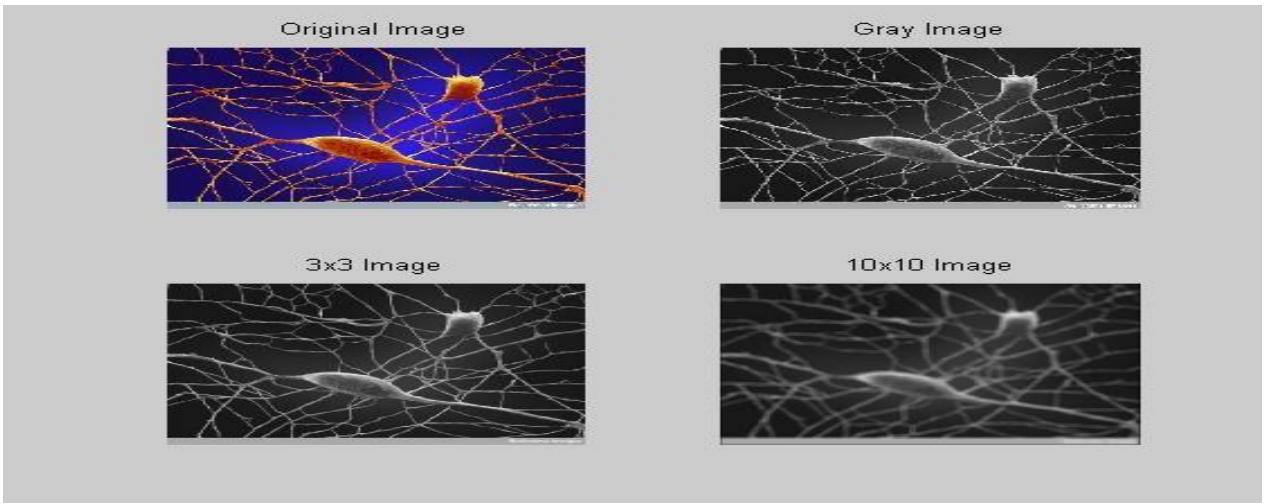
```
I=imread('nuron.jpg');
 K = rgb2gray(I);
 J= imnoise(K ,'salt & pepper',0.05);
 f= medfilt2(J,[3,3]);
 f1=medfilt2(J,[10,10]);

 subplot(3,2,1); imshow(I); title('Original Image');  subplot(3,2,2); imshow(K); title('Gray Image');  subplot(3,2,3); imshow(J); title('Noise added Image');  subplot(3,2,4); imshow(f); title('3x3 Image');  subplot(3,2,5); imshow(f1); title('10x10 Image');
```

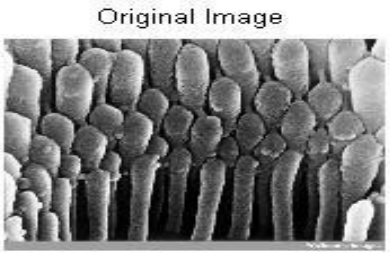
%Mean Filter and Average Filter figure; i=imread('nuron.jpg'); g=rgb2gray(i); g1=fspecial('average',[3 3]); b1 = imfilter(g,g1); subplot(2,2,1); imshow(i); title('Original Image'); subplot(2,2,2); imshow(g); title('Gray Image'); subplot(2,2,3); imshow(b1); title('3x3 Image'); g2= fspecial('average',[10 10]); b2=imfilter(g,g2); subplot(2,2,4); imshow(b2); title('10x10 Image');

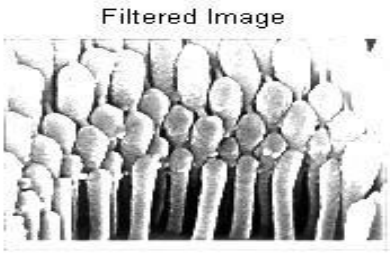
%Implementation of filter using Convolution figure;
I= imread('earcell.jpg');
I=I(:,:,1); subplot(2,2,1); imshow(I); title('Original Image');

a=[0.001 0.001 0.001; 0.001 0.001 0.001; 0.001 0.001 0.001]; R=conv2(a,I);
subplot(2,2,2); imshow(R); title('Filtered Image');

b=[0.005 0.005 0.005; 0.005 0.005 0.005; 0.005 0.005 0.005]; R1=conv2(b,I);
subplot(2,2,3); imshow(R1); title('Filtered Image 2');







Exercise #8

Implementation of Image Smoothening Filters (Mean and Median filtering of an Image)

Name:
Year/Block:
Application/Software:

1. Codes
2. Output
3. Answer the following questions:

   A. How do mean and median filtering techniques differ in noise reduction and image feature preservation? Describe their implementation processes, considering the impact of kernel/window size. Additionally, compare their computational complexity and discuss scenarios favoring one over the other for efficiency and noise reduction.

   B. What are common real-world applications for mean and median filtering, and how do their choices affect outcomes? How effective are they against various types of noise? Lastly, what challenges arise in practical implementations, and how can they be addressed for optimal results?

   C. How can mean and median filtering algorithms be optimized for large-scale or real-time applications, leveraging parallelization and hardware acceleration? Additionally, what alternative approaches aim to balance noise reduction and feature preservation, and how do they compare in complexity and effectiveness? Lastly, discuss the importance of parameter tuning and provide guidelines for selecting kernel size and parameters in different scenarios.