Table of Contents

Table of Figures

Introduction to Digital Image Processing Assignment

Student's Name: Nguyen Trong Nghia *Student's ID: SESEIU17023*

A. Basic Definition and Intuition

Kernel (windows) based filtering is a widely used method in digital image processing, which consists of "sliding" an $N \times N$ (N is odd) over every pixel and changing the center pixel value based on the neighboring pixels. In many applications, where spatial information of pixel plays an important role in image interpretation, kernel-based filtering is preferable compared to other methods of processing.

Different type of windows filter is used based on image processing requirement, however, most of these filters can be divided into two main classes: Linear (convolution) and non-linear filters.

I. Convolution

The convolution of f and g is formally defined as: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)dt$

Intuition

As stated in more mathematically focused text: convolution is the integral of the product of two functions after one is reversed and shifted. Personally, I have always found this definition to be unintuitive and vague. Based on my understanding, convolution is better defined as: the function of function.

A function can be though as a mapping from one vector space onto another - as many of us have done - by mapping variable in the "horizontal vector space" (x-axis) to the "vertical vector space" (y-axis) or the "time vector space" to the "distance traveled vector space". In most cases, these are real or complex values vector space, however, there are some cases of vector spaces where the definition of convolution proved to be useful.

Imagine a river in the 2D Cartesian coordinate system, where the water flow vector is defined as a function of x and y coordinate. Then, let a rigid object - with its own shape defined by a function in a 3D Cartesian coordinate system - be dropped into the river, how do we calculate the flow of the object in the water? If the object is a simple point, the flow can be easily calculated, however, as the object is defined by a **continuous function,** it would suggest that the vectors of every point making up the objects should be calculated separately with time shift (as they do not come into contact with the river simultaneously), then, the sum of all these vectors is the translational motion of the whole object. And this is precisely the definition of convolution.

If a function can be defined as a mapping of variables in a Real space onto another Real space by some mathematical operations. It is perfectly acceptable to defined function as a mapping of a function into a Real space by some mathematical operation.

Convolution and System Output

Imagine a system made by a box function started at a and ends at $2a$ with as shown in **Figure 1**. What would happen if a "feed" this system an input signal, which is the same function as the system.

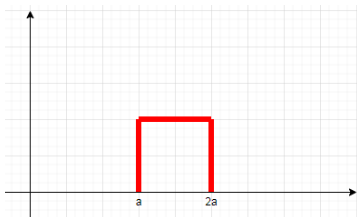


Figure 1

In many Signal&System textbooks, the output is stated to the convolution of the input signal and the system function.

Which would give us the triangular shaped output as shown in **Figure 2.** Closer inspection would suggest that the output begin ramming up at $t = 2a$ and end at $t = 4a$, and this can be explained extremely intuitively.

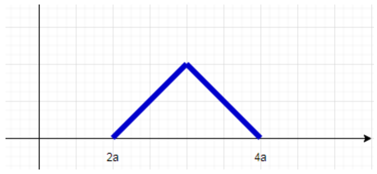


Figure 2

If we feed a signal that starts at a into another function with starts at a. Then the output begins to appear at $t = 2a$. Similarly, why the output ends at $t = 4a$ can also be explained.

Convolution in Digital Image Processing

The mathematics and applicable intuition of Convolution has been developed in the previous sections, however, what does any of this have to with Digital Image Processing?

Image processing consists of manipulating pixel values to achieve desirable image quality or characteristics. Fortunately, pixel value itself is a **real number** stored in computers as a function of row and column (a 2D Cartesian Coordinate), as such, digital image is subject to mathematical operators. Or in another word, convolution can be applied on digital images as: kernel-based filters are essentially functions done on a digital image.

Another insight can be found if we look at digital filters as input and kernel-based filters as system. Then the output – the resultant image - can be obtain be applying a kernel-based filters on a digital image (putting the input through the system).

In conclusion, by discussing the intuition of convolution, it leads naturally to how convolution can be used in Digital Image Processing. Simply due to two simple characteristics:

- Digital Images are stored as real values in a 2D Cartesian Coordinate.
- Many desirable spatial characteristics can be described digitally.

II. Linear Filters

Linear Digital Image filter is defined formally as: $(a * k)(x, y) = \sum_u \sum_v a(u, v)k(x - u, y - v)$

Where instead of time, a spatial shift is in place. This is similar in Signal&System where output also depended on past inputs – not only current input, but in Digital Image, the value of a pixel is dependent on the value of surrounding pixels – denoted by the spatial shift u and v.

And this is the defining characteristic of linear filters – meaning the filter is made up of linear equations, interacting using mathematic **operators**, whereas nonlinear filters use mathematic **functions** (such as max, min or median).

III. Nonlinear Filters

Nonlinear Filters are defined as filers whose output is not a linear function of its input. Or in another word, Nonlinear filters can not be generalize using a set of linear equations (unlike linear filters).

Then, how does one choose the value for a nonlinear filter? By using mathematic **functions**, not mathematic **operators**. For example, the ADC and DAC are both considered Nonlinear filters, and how they work depended entirely on the rounding and truncate **function**, not basic mathematic **operators**.

IV. Remark

Kernel-based filters are widely used in Digital Image Processing where spatial information of each pixel plays an important role in image interpretation. However, there are two types of kernel-based filters: linear and nonlinear.

- Linear filters are based on mathematical **operators –** which are nicely packed inside the definition of convolution.
- Nonlinear Filters are based on mathematical **functions** – whose output is not nicely packed inside any mathematical definition.

As we have discussed both the mathematical ideal and intuition behind kernel-based filters, the next section of this paper will focus entirely on the realization of such filters using Matlab and its effect on a Digital Image.

B. Example Matlab programs for Kernel based filters.

For comparison purposes, all our filters will work on the same image, I will choose the image “pears.png” – shown in **Figure 3 -** since it was provided in our lab course (and I love pears).

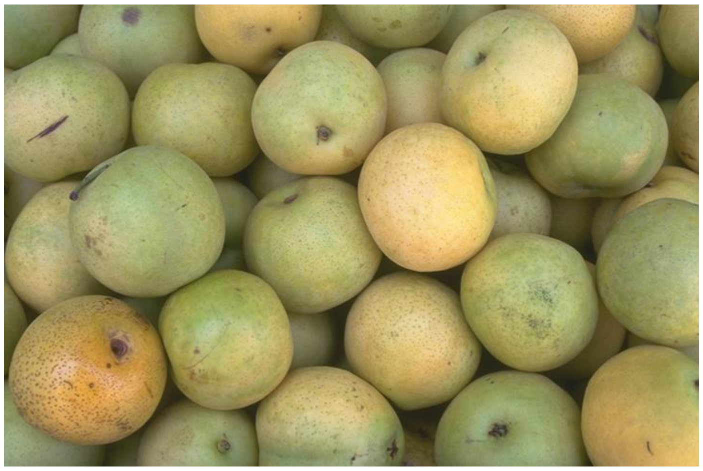


Figure 3

Linear Filters

Linear filters are filters applied through the usage of convolution. This is possible as many desirable spatial characteristics can be described mathematically in a kernel. A possible example is Sobel filter – whose kernel is shown in **Figure 4**.

The result for such an operation is shown in **Figure 5**, which made use of Matlab edge function (code shown in **Figure 6**).

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

Figure 4

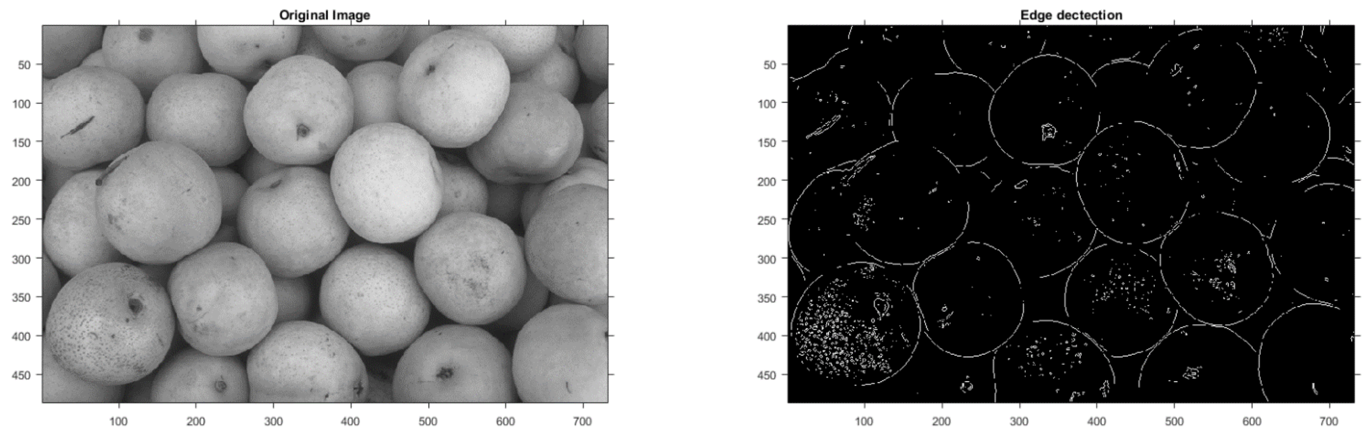
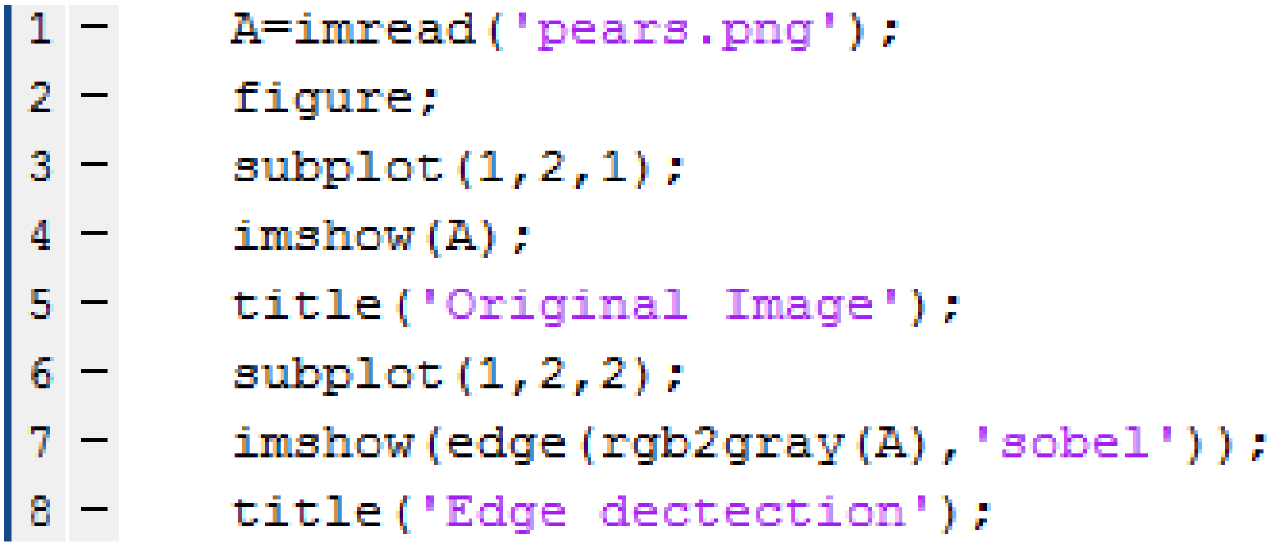


Figure 5



```
A=imread('pears.png');
figure;
subplot(1,2,1);
imshow(A);
title('Original Image');
subplot(1,2,2);
imshow(edge(rgb2gray(A),'sobel'));
title('Edge dectection');
```

Figure 6

In other to show that a kernel can be applied using convolution, I will make use of the conv function in Matlab (shown in **Figure 8**) to obtain **Figure 7**, which is nearly identical to **Figure** 5.

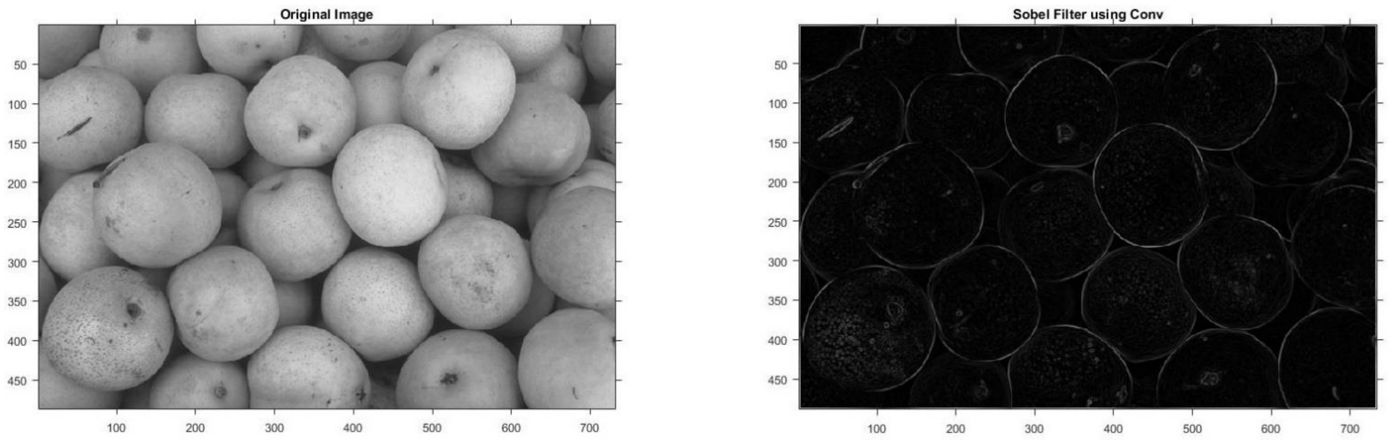
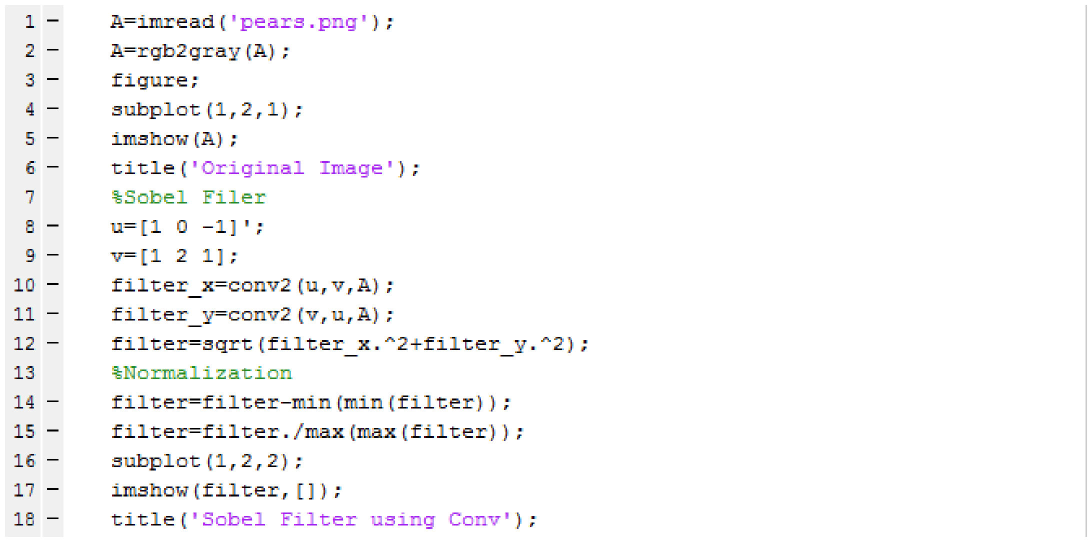


Figure 7



```
A=imread('pears.png');
A=rgb2gray(A);
figure;
subplot(1,2,1);
imshow(A);
title('Original Image');
%Sobel Filer
u=[1 0 -1]';
v=[1 2 1];
filter_x=conv2(u,v,A);
filter_y=conv2(v,u,A);
filter=sqrt(filter_x.^2+filter_y.^2);
%Normalization
filter=filter-min(min(filter));
filter=filter./max(max(filter));
subplot(1,2,2);
imshow(filter,[]);
title('Sobel Filter using Conv');
```

Figure 8

From these examples, it is clear that meaningful spatial feature can be and was expressed in term of mathematical operators. Then, by taking the convolution of these expressed characteristic and a digital image, a desirable image can be convolution obtained.

Nonlinear Filters

Nonlinear filters are hard to be expressed mathematically (sometimes impossible), let look at the example shown **Figure 9** which will be put through t a 3 × 3 median filter.

102	105	115
108	255	130
110	100	122

Figure 9

The median filter will fix the middle value into the median of every value in the windows, which is **110**. The result is shown in **Figure 10**.

Nonlinear filters are hard to describe by word, and even harder to describe using mathematic. Unfortunately, digital images are stored as **numbers**, making nonlinear filters difficult to use and design.

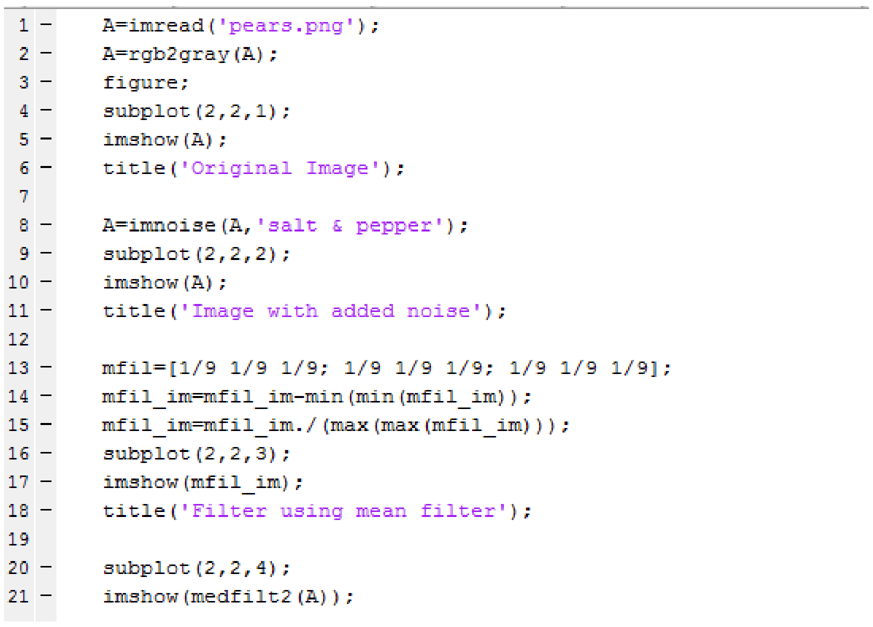
However, nonlinear filter does have a useful property, as it does not depend on mathematic operators, it is better at removing **outliner** data.

102	105	115
108	**110**	130
110	100	122

Figure 10

Looking back at the example in **Figure 9**, it is obvious that 255 is not a meaningful value – as it does not contribute any useful spatial data other than a white point (which is probably a noise). If we apply a 3 × 3 mean filter, the middle pixel would be **127**, meaning the outliner data plays a more prominent role. Whereas, a 3 × 3 median value would give us **110**, meaning the outliner has less effect on our filtered image.

To demonstrate the usefulness of a specific nonlinear filter – the median filter. Let add a salt and pepper noise to our image, then, let compare processed images using mean filter and median filter. The code and resulting image are shown in **Figure 11** and **Figure 12**.



```
A=imread('pears.png');
A=rgb2gray(A);
figure;
subplot(2,2,1);
imshow(A);
title('Original Image');

A=imnoise(A,'salt & pepper');
subplot(2,2,2);
imshow(A);
title('Image with added noise');

mfil=[1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
mfil_im=mfil_im-min(min(mfil_im));
mfil_im=mfil_im./(max(max(mfil_im)));
subplot(2,2,3);
imshow(mfil_im);
title('Filter using mean filter');

subplot(2,2,4);
imshow(medfilt2(A));
```

Figure 11

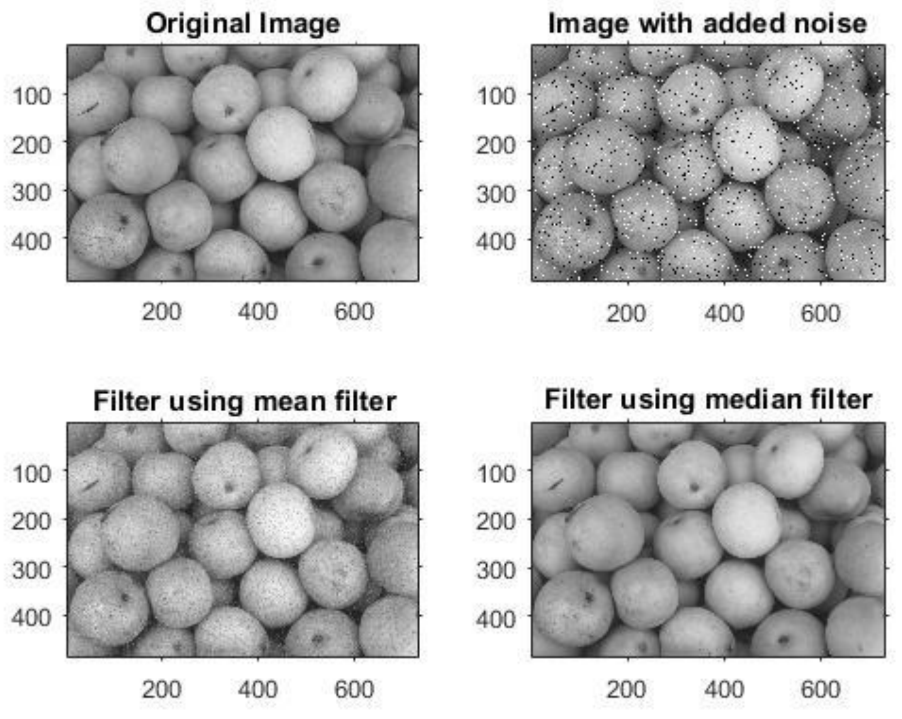


Figure 12

As shown in this example, nonlinear filters do have a niche use in Digital Image Processing, even though they are hard to design, nonlinear filters play an important role when filtering images with nonlinear characteristic.

C. Linear and Nonlinear filter comparison

I think you made a mistake in the assignment requirement, because I am very certain that convolution and linear filters are the same thing. So this section I'm focus on the differences between linear and nonlinear filter.

In the **A** section of this paper, we have discussed the differences between linear and nonlinear filters in a mathematical sense. Whereas **B** section has shown their difference in a practical example, in conclusion:

- Linear filters are based on **operators**, whereas nonlinear filters are based on **functions**.
- Linear filter can be described mathematically, whereas nonlinear filters are hard to define.
- Linear filters play a more "generalizing" role, whereas nonlinear filter play a more niche and supporting role in Digital Image Processing.