

# Digital Image Processing

## Lab #4. imfilter imlementation



학과 : 전자공학과

학번 : 21611648

이름 : 유준상

담당교수 : 김성호

## **Table of Contents**

### **1. Introduction**

### **2. Main body**

### **3. Conclusion**

## Introduction

Spatial filtering은 관심 픽셀의 이웃 픽셀과 사전 정의된 연산자에 의해서 이루어진다. 크게 Linear / Non-Linear로 나누어진다.

연산자는 correlation 방법과 convolution 방법이 있다. 두 방법의 가장 큰 차이점은 filter mask를 180도 회전함 하지 않음이다.

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

Spatial Correlation

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$

Spatial Convolution

이 차이점 때문에 연산 후 결과 이미지도 달라진다.

Cropped correlation result	Cropped <u>convolution</u> result
0 0 0 0 0	0 0 0 0 0
0 9 8 7 0	0 1 2 3 0
0 6 5 4 0	0 4 5 6 0
0 3 2 1 0	0 7 8 9 0
0 0 0 0 0	0 0 0 0 0

Weighted averaging filter를 나타내는 식이다.

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

where  $m = 2a + 1$ ,  $n = 2b + 1$ .

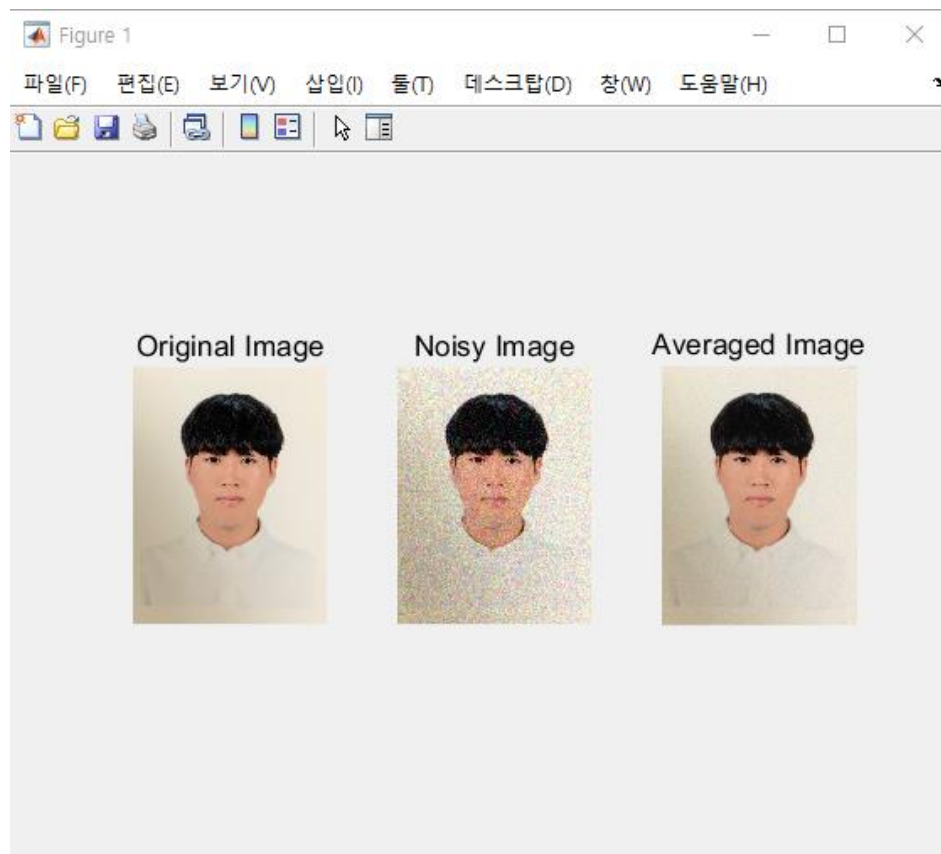
Padding은 입력 이미지의 외곽에 특정 숫자나 방법으로 가장자리를 만들어 주는 것을 말한다. 0을 padding하는 zero padding과 가장자리 픽셀과 같은 값으로 padding하는 replicate가 대표적이다. 이외에도 edge 경계에서 거울처럼 반대로 보여주는 Symmetric, 영상이 반복된다고 생각하고 일정 부분 더 포함하는 방법인 Circular가 있다. padding 후 crop하여 원본 이미지와 같은 크기로 출력 이미지를 생성하는 것이 일반적이다.

## 1. Lab 4-1

1) source code

```
편집기 - C:\WIP\HW\lab4_1.m
lab4_1.m x lab4_3.m +
1   %% 21611648 jsyoo
2   %% linear filter : averaging filter
3   I=imread('jsyoo.png'); figure(1);
4   subplot(1,3,1); imshow(I); title('Original Image','fontsize',13);
5   % add Gaussian noise
6   n=randn(size(I))+20; % normalize
7   In=uint8(double(I)+n);
8   %Ist=find(In<0); In(Ist)=0;
9   %Ist=find(In>255); In(Ist)=255;
10  subplot(1,3,2); imshow(In); title('Noisy Image','fontsize',13);
11  % averaging filter
12  w=ones(5,5)/25; % 5x5크기 weight 만들기
13  Ir=imfilter(In,w,'same','replicate');
14  % parameter : Input image, weight, output size, padding
15  subplot(1,3,3); imshow(Ir); title('Averaged Image','fontsize',13);
```

2) result figure



### 3) discussions

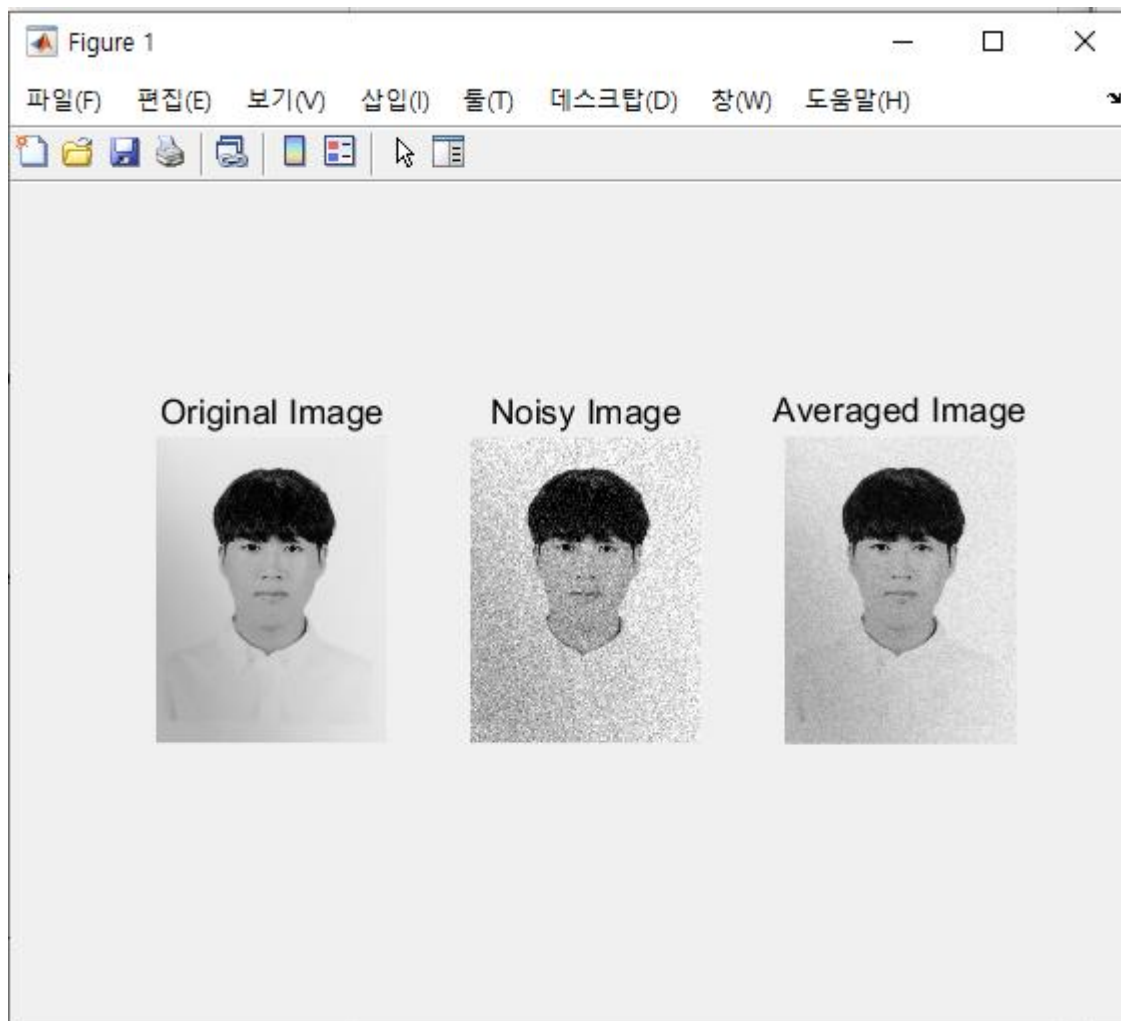
제 사진으로 실습하고 싶어서, imread로 작업하는 폴더에서 제 사진을 읽어오고 figure를 만든 후, imshow로 읽어온 이미지를 디스플레이 했습니다. 그 후, randn을 이용해서 표준편차가 20인 Gaussian noise를 생성 후 입력 이미지에 더해서 노이즈가 낀 이미지를 생성합니다. 5x5 크기의 weight 행렬을 만들고, imfilter 함수를 이용해서 출력 이미지 크기와 padding을 정해준 뒤 filtering을 진행하였습니다. 노이즈가 제거된 이미지를 확인했습니다.

## 2. Lab 4-2

### 1) source code

```
편집기 - C:\DIP\HW\lab4_2.m
lab4_2.m x +
1 - I=imread('jyoo.png'); Ig=rgb2gray(I);
2 - figure(1); subplot(1,3,1); imshow(Ig);
3 - title('Original Image','fontsize',13);
4 - % add Gaussian noise
5 - n=randn(size(Ig))*20; % normalize
6 - In=uint8(double(Ig)+n);
7 - subplot(1,3,2); imshow(In); title('Noisy Image','fontsize',13);
8 - % averaging filter
9 - w=ones(3,3)/9; % 3x3크기 filter 만들기
10 - [M,N]=size(In);
11 - [m,n]=size(w);
12 - % 3x3 filter를 사용하기 위해 상하좌우 3라인씩 zero padding
13 - % zero padding까지한 크기의 0 행렬 만들기
14 - padded = zeros(M+6,N+6);
15 - % zero padding 제외한 부분에 In데이터 할당
16 - padded(4:M+3,4:N+3) = In;
17 - [p,q] = size(padded);
18 - % filter overlay를 위해 (2,2)부터 시작해서 (p-1,q-1)까지 적용
19 - for i=2:p-1
20 -     for j=2:q-1
21 -         % (1,1)부터 (3,3)까지의 filter 크기 범위의 이미지 값에
22 -         % filter weight 곱하기
23 -         temp = padded(i-1:i+1,j-1:j+1).*w;
24 -         % overlay한 범위의 합을 출력 이미지의 (1,1)에 저장
25 -         Iout(i-1,j-1)=sum(temp(:));
26 -     end
27 - end
28 - % 원래 크기의 이미지를 얻기위해 zero padding된 부분을 crop
29 - Iout = Iout(4:M+3,4:N+3);
30 - Iout = uint8(Iout);
31 - subplot(1,3,3); imshow(Iout); title('Averaged Image','fontsize',13);
```

2) result figure



3) discussions

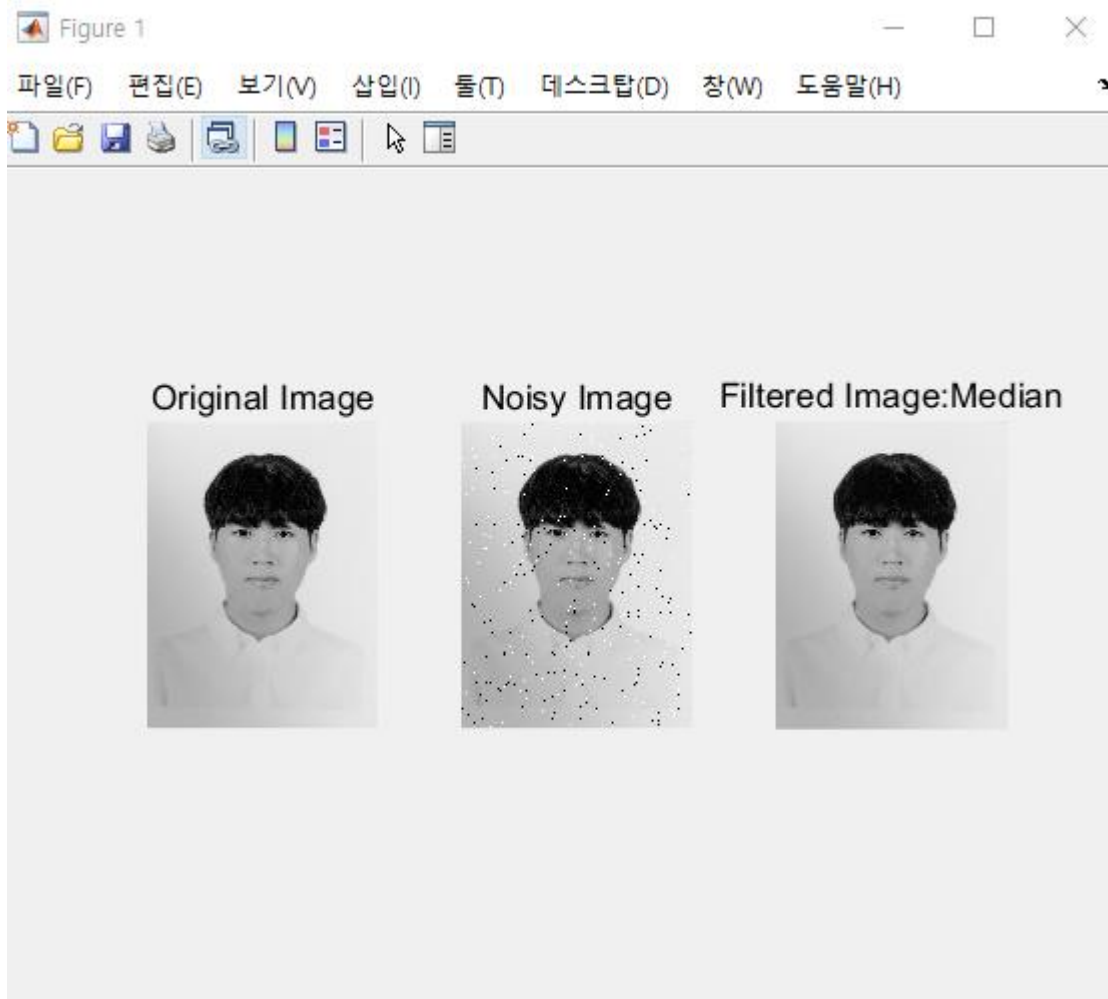
### 3. Lab 4-3

1) source code



```
편집기 - C:\WDIP\WWW\lab4_3.m
lab4_1.m x lab4_3.m x +
1 %% 21611648 jsyoo
2 %% Median filter
3 - I=imread('jsyoo.png');
4 - Ig=rgb2gray(I);
5 - figure(1); subplot(1,3,1); imshow(Ig);
6 - title('Original Image','fontsize',13);
7 - J=imnoise(Ig,'salt & pepper',0.02);
8 - figure(1); subplot(1,3,2); imshow(J);
9 - title('Noisy Image','fontsize',13);
10 - Imed=medfilt2(J,[3 3]);
11 - figure(1); subplot(1,3,3); imshow(Imed);
12 - title('Filtered Image:Median','fontsize',13);
```

2) result figure



3) discussions

원본 이미지인 제 사진은 2d 이미지가 아니어서 `rgb2gray` 를 사용하여 2d 이미지로 먼저 변환했습니다. 그 후, `imnoise` 를 사용하여 원본 이미지에 salt&pepper 즉 흰 점과 까만 점을 넣어 노이즈가 낀 이미지를 생성하였습니다. 노이즈 이미지를 median filtering 을 통해 결과를 보기 위해 `medfilt2` built-in func 를 이용하였습니다. 결과에서는 원본 이미지와 같게 노이즈가 하나도 없는 것을 확인하였습니다.

## Conclusion

실습을 해보면서 노이즈가 많이 낀 이미지에서 노이즈만 골라서 없애고, 결과 이미지가 연산 과정, 필터링 과정을 거쳤는데 품질이 원본과 같아서 신기했다. 다른 최신의 필터에 대해서도 알고 싶어졌다.