

Digital  
Image  
Process  
Report #6



imfilter  
&  
Median Filter

로봇기계공학과  
21721160  
최병희

## Lab 4-2 MATLAB: imfilter

- 문제 - imfilter 커스텀 함수 구현

imfilter 내장함수는 이미지에 필터를 correlation 방식으로 적용하는 함수입니다. 이미지 첫 번째 픽셀부터 마지막 번째 픽셀까지  $m \times n$  사이즈의 필터의 값을 각 픽셀에 요소별 연산을 적용하여 그 합을 나타냅니다. 이 경우, 필터의 사이즈 및 stride에 의해 출력 이미지의 사이즈에 차이가 있으므로, 필터를 적용할 때 padding 방식을 정하게 됩니다. 내장함수에서 제공하는 padding 방식으로는 Zero(default), symmetric, replicate, circular 방식이 있습니다. 본 과제에서는 주어진 조건에 따라 Zero padding 방식을 구현하여 사용합니다.

- Source code

myimfilter.m

```
function f_map = myimfilter(image, kernel)
[m,n] = size(image);
[a,b] = size(kernel);
pad_img = zeros(m+floor(a/2)*2,n+floor(b/2)*2);
pad_img(floor(a/2)+1:m+floor(a/2),floor(b/2)+1:n+floor(b/2)) = image;

f_map = uint8(zeros(m,n));
for i = floor(a/2)+1:m+floor(a/2)
    for j = floor(b/2)+1:n+floor(b/2)
        corr =
pad_img(i-floor(a/2):i+floor(a/2),j-floor(b/2):j+floor(b/2)).*kernel;
        kernel_sum = sum(kernel, 'all');
        f_map(i-floor(a/2), j-floor(b/2)) = sum(corr, 'all')/kernel_sum;
    end
end
end
```

myimfilter의 구현방식은 Zero padding 부분과 correlation 연산 부분으로 나누어져 있습니다. Zero padding은 먼저 원본 이미지의 사이즈에 필터의 사이즈/2만큼을 더한 사이즈의 영행렬을 생성합니다. 그 후 필터 사이즈/2+1 좌표부터 원본 이미지를 저장해 줌으로서 이미지의 테두리에 zero행렬을 생성한 효과를 주고 있습니다.

correlation 연산은 먼저 uint8 타입의 원본 이미지 사이즈의 영행렬 f\_map을 생성한 후, 필터 사이즈/2+1 좌표부터 시작한 연산의 결과를 f\_map에 저장합니다. 변환 결과는 f\_map에 저장되어 반환됩니다.

필터의 사이즈와 관계없이 필터링 할 수 있도록 하기 위해 사이즈에 따른 변환 공식을 사용하였습니다.

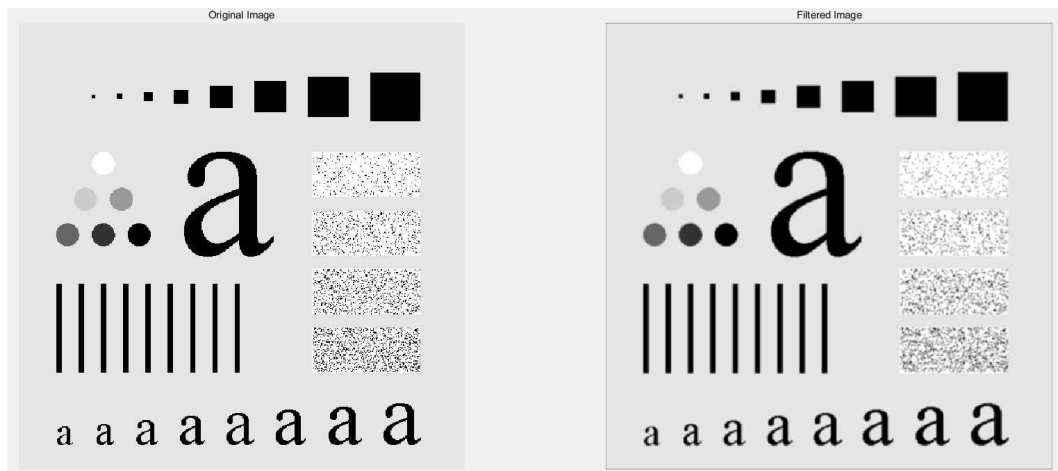
DIP\_6\_imfilter.m

```
clc;clear all;close all;
img = imread('data/pattern.tif');
kernel = 1/16*[1 2 1; 2 4 2; 1 2 1];
f_map = myimfilter(img, kernel);

subplot(1,2,1);
imshow(img);
title('Original Image');
subplot(1,2,2);
imshow(f_map);
title('Filtered Image');
```

myimfilter 함수의 결과를 출력하기 위한 코드입니다.

- 결과 비교 및 배운 점



원본 이미지 pattern.tif 파일에 가우시안 형태의 Smoothing Averaging Filter를 적용한 결과입니다. average filter의 특징인 이미지가 조금 흐려 보이는 결과를 얻을 수 있었습니다.

## Lab 4-3 Conduct median filter

- 문제 - median filter 내장함수 실습

이번 Lab의 목적은 median filter를 실습해 보는 것입니다. median filter는 중앙값 필터로,  $m \times n$  사이즈의 필터를 적용하여 그 중앙값을 반환하는 필터입니다. correlation과 다른 점은 correlation 연산은 필터를 곱한 후 그 합을 반환하지만, median filter는 중앙값을 반환한다는 점입니다. 내장함수를 사용할 수 있으므로, imnoise 함수를 사용하여 salt & pepper 노이즈를 적용하고, medfilt2 함수를 사용하여 노이즈를 제거한 결과를 출력했습니다.

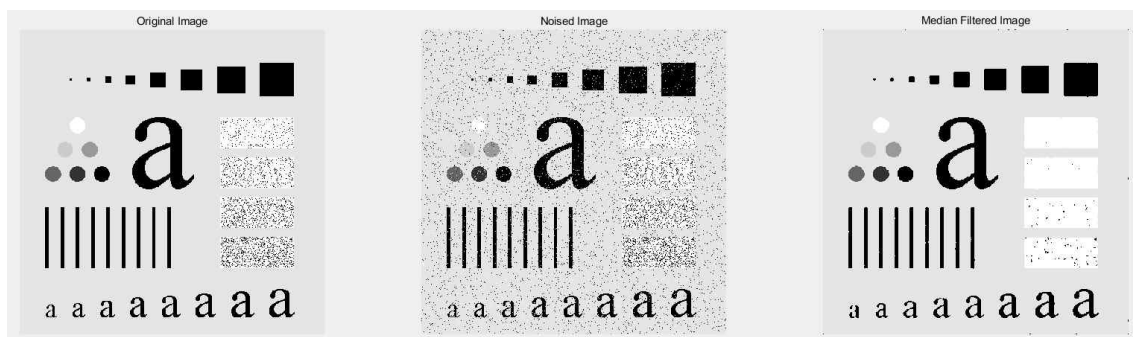
- Source code

```
DIP_6_medfilter.m

clc;clear all;close all;
img = imread('data/pattern.tif');
noise_img = imnoise(img, 'salt & pepper');
med_img = medfilt2(noise_img);
subplot(1,3,1);
imshow(img);
title('Original Image');
subplot(1,3,2);
imshow(noise_img);
title('Noised Image');
subplot(1,3,3);
imshow(med_img);
title('Median Filtered Image');
```

커스텀 함수를 제작하지 않고 내장함수만을 사용하였습니다. 위의 설명처럼 `imnoise` 함수로 salt&pepper noise를 적용하고, `medfilt2` 함수로 noise를 제거하는 과정을 거쳤습니다.

- 결과 비교 및 배운 점



노이즈를 적용한 이미지와 중앙값 필터를 적용한 결과는 다음과 같습니다. 중앙값 필터를 적용하여 대부분의 노이즈가 사라졌지만, 몇몇 노이즈가 남아 있는 것을 확인할 수 있었습니다. 범위 내의 중앙값을 확인하는 필터의 특성상, 노이즈의 완벽한 제거는 할 수 없으므로 추측됩니다.