

프로그래밍 과제 1 – Image Filtering

- Denoising the salt-and-pepper noisy image through filtering
- 목표 : 필터링 연산을 통해 salt-and-pepper 노이즈가 있는 이미지에서 노이즈를 제거한다.
- 제출 : 10월 16일 강의 23:59까지 Blackboard로 제출
 - 스크립트 파일과 Input 이미지를 Project1_학번.zip으로 압축하여 제출
 - MATLAB 스크립트 파일 실행 시 발표자료 page 2의 4가지 이미지가 한 창에 출력 되어야 함 (2 페이지 그림과 같이 각 이미지의 이름 필수적으로 표기)
 - 코드 실행 시 2페이지의 그래프와 Output 이미지가 저장되어야 함.

출력 그래프 (사진은 예시)

noisy image



Gaussian Filtered



Average Filtered



Median Filtered



함수 작성

```
function img_avg = Average_3x3(img)
    %%%%%%%%%%% input : noisy img, output : 3x3 average filtered image %%%%%%%%%%%

end

function img_gaussian = Gaussian_3x3(img)
    %%%%%%%%%%% input : noisy img, output : 3x3 gaussian filtered image %%%%%%%%%%%

end

function img_med = Median3x3(img)
    %%%%%%%%%%% input : noisy img, output : 3x3 median filtered image %%%%%%%%%%%

end
```

$$(1/16) \times$$

1	2	1
2	4	2
1	2	1

3x3 Gaussian Filter의 Weight

- Image filtering을 위한 3가지 함수 작성.
- 각 함수의 몸체를 채워넣으면 됨.
- 세 함수 모두 noisy image가 Input, Filterd image가 Output, kernel size는 3 x 3.
- Average_3x3, Gaussian_3x3 함수 구현 시에는 MATLAB의 conv2 함수 사용.
- Gaussian Filter의 Weight는 오른쪽 그림과 같이 지정.
- Median Filtering 시 Padding은 Replicate Padding 진행. (5 page 참조)

Main script 작성

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
main function start %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

img = imread('165052.jpg'); % 읽어들이 파일명 입력.
img = rgb2gray(img);

img_noisy = imnoise(img, 'salt & pepper', 0.005);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Start From here %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

img_avg = Average_3x3(img_noisy); % average filtered image
img_gau = Gaussian_3x3(img_noisy); % gaussian filtered image
img_med = Median3x3(img_noisy); % median filtered image

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plot 4 images %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

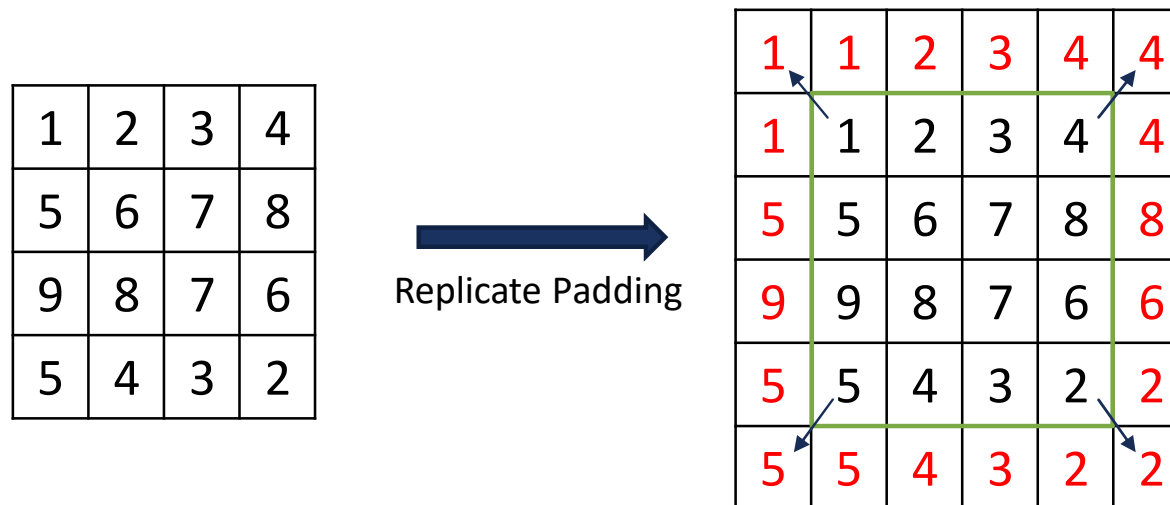
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Save Output images%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% main function end %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

- 앞 페이지에서 작성한 함수를 이용해 Main 함수에서 이미지 필터링을 진행.
- Main 함수와 같은 역할을 하는 부분의 빈칸을 채워넣으면 됨.
- 빨간색으로 표시된 이미지 파일 이름은 본인이 찍은 사진으로 교체 (파일 이름은 학번.jpg로)
- Main 함수에서는 파란색으로 표시된 곳들에 4가지 이미지를 한번에 Plot하는 부분, 결과 영상을 저장하는 부분 코드 작성.
- 여러 이미지를 동시에 Plot 하기 위해서는 subplot 함수 사용.
- 이미지 Plot 시 순서, 각 이미지의 이름은 2페이지의 사진과 동일하게 작성.
- 결과 영상은 noisy, average filtered, gaussian filtered, median filtered 영상이 각각 **학번_noisy.jpg**, **학번_avg.jpg**, **학번_gau.jpg**, **학번_med.jpg** 로 코드 실행시 저장되게 작성.

REPLICATE PADDING



- Median Filter 구현 시 가장자리 처리를 위해 이미지의 가장자리 부분에 상하좌우로 1 Pixel 씩 Padding 진행.
- 빨간색으로 표시된 원래 비어있던 부분에는 원본 이미지에서 가장 가까운 Pixel 값이 대입됨.

참고 사항

- 코드 표절 시 0점
 - **지각 제출 불가능**
 - 스크립트 파일과 이미지 파일은 “**Project1_학번.zip**” 으로 압축 저장 (아니면 감점)
-
- 기타 문의 및 제출은 아래 조교 메일로 하시길 바랍니다.
 - minje234@korea.ac.kr