**01 Matlab Code**

1. **Zero Padding Function 2 (Used in HW3)**

설명 : 위,아래, 양 옆 append 시킨 패딩법

function y = Zero\_Pad1(x1, x2) %Zero Padding, x1 is original image matrix, x2 is mask matrix

[m1, n1] = size(x1);

[m2, n2] = size(x2);

A = [zeros(m1,n2-1) x1 zeros(m1,n2-1)];

y = [zeros(m2-1,n1+2\*(n2-1)); A; zeros(m2-1,n1+2\*(n2-1))];

end

1. **Median Filtering Function**

설명 : Median Filtering을 실행시키는 함수

function y = Median(x, n) %Median Filtering, x is original image matrix, n is the unit number want to calculate

n\_mid = fix(n/2);

%padding the image

x1 = Zero\_Pad2(x,n\_mid);

[m1, n1] = size(x1);

for i = 1:m1-n+1

for j = 1:n1-n+1

x2 = x1(i:i+n-1, j:j+n-1);

x3 = sort(x2(:));

y(i, j) = x3((1+n^2)/2);

end

end

end

1. **Main Code**

%%% Robot Vision%%%

%%% Dept. of Electronic Engineering

%%% 201314651 Lee Wonjai

% read the targeted image

IM\_Fruits = imread('C:\Users\user\OneDrive\¹ÙÅÁ È­¸é\2019 Æ¯º°, 4ÇÐ³â\4ÇÐ³â 2ÇÐ±â\·Îº¿ºñÀü\hw4\pepper\_noisy\_30.png');

%Median Filtering

Median\_Friuits3 = Median(IM\_Fruits,3);

Median\_Friuits5 = Median(IM\_Fruits,5);

Median\_Friuits7 = Median(IM\_Fruits,7);

figure, imshow(IM\_Fruits)

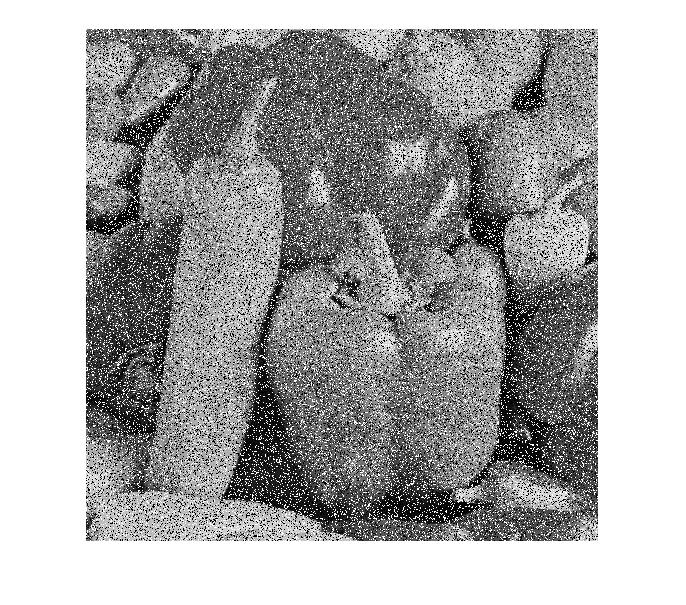
figure, imshow(Median\_Friuits3)

figure, imshow(Median\_Friuits5)

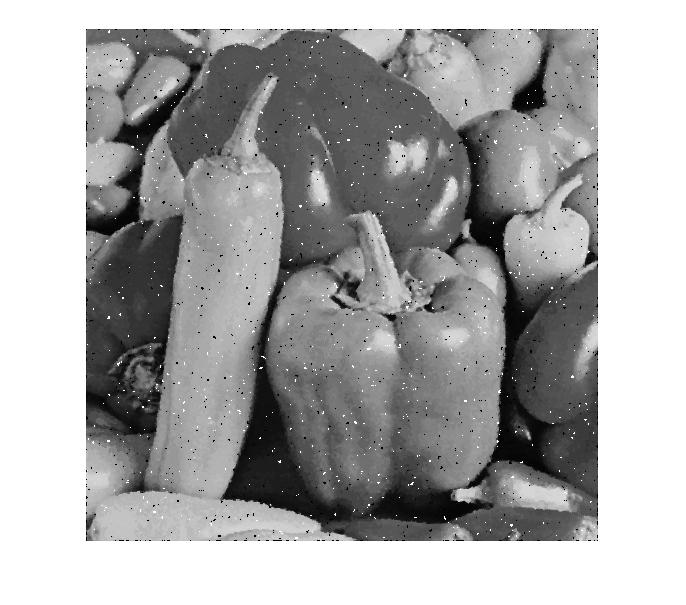
figure, imshow(Median\_Friuits7)

**02 Result 1**

1. **Original Image**



1. **3X3 Size Median Filter**



1. **5X5 Size Median Filter**



1. **7X7 Size Median Filter**



**Conclusion**

1. **Matlab**

* **Matrix(:)** – nxn Matrix 를 1xn2 Matrix로 바꿔준다.

사용 이유 : 정렬의 편의성을 위해

**Why?** 그냥 Matrix에 **sort 시키면 열단위로만 정렬**이 된다(물론 sort 두 번 시키면 해결), **sort를 한 번** 시켜 보기 좋은 정렬 모양을 나타내기 위해 사용했다.

* Median Filter의 크기에 따른 결과

1. Median Filter의 역할 : Image 내 **noise를 제거**하고, **blur 현상**을 일으킴
2. Filter 크기 **단위가 커질**수록 noise 제거 및 blur 현상의 **정도가 커짐**

**Why?** Filter의 크기 내에서 중간 값을 찾아 Image에 치환하기 때문에 **average** 와 **gaussian**과 **비슷한 효과**를 얻는 것인데 Filter크기(정확히는 중간 값을 찾는 범위)가 커질수록 **더 많은 부위가 중간 값에 영향**을 받기 때문이다.