

01 Matlab Code

1. Diagonal Reverse Function

설명 : 사선 대칭 함수

```
function y = Conv_rev(x) %Function that makes mask reversed vertically and
horizontally
[m, n] = size(x);
for i = 1:m
    for j = 1:n
        y(i, j) = x(m-i+1, n-j+1);
    end
end
end
```

2. Zero Padding Function 1 (Used in HW2)

설명 : 제로 패딩 함수 방법 1 - 크기가 큰 zeros 행렬에 이미지 덧붙이기

```
function y = Zero_Pad(x1, x2) %Zero Padding, x1 is original image matrix,
x2 is mask matrix
[m1, n1] = size(x1);
[m2, n2] = size(x2);
m3 = m1+2*(m2-1);
n3 = n1+2*(n2-1);
y = zeros(m3, n3);
for i = 1:m1
    for j = 1:n1
        y(i+m2-1, j+n2-1) = x1(i, j);
    end
end
end
```

3. Zero Padding Function 2

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
```

4. Convolution Function

설명 : 컨볼루션(공간 필터링)을 실행하는 함수

```
function y = Conv_IM(x1, x2) %Convolution filtering function, x1 is padded
image matrix, x2 is mask matrix
[m1, n1] = size(x1);
[m2, n2] = size(x2);

for i = 1:m1-m2+1
    for j = 1:n1-n2+1
        y(i, j) = sum(sum(x1(i:i+m2-1, j:j+n2-1).*x2));
    end
end
y = abs(y);
end
```

5. Scaling Function (Used in HW2)

설명 : 스케일 변환을 도와주는 함수

```
% Scaling transformation function
function y = Scale(x, max_value, min_value) %x is image matrix
y = (255/(max_value-min_value)*(x-min_value));
end
```

6. Main Code

```

%%% Robot Vision%%%
%% Dept. of Electronic Engineering
%% 201314651 Lee Wonjai

IM_Rose = imread('C:\Users\user\OneDrive\1ÙÁÁ È-é\2019 È°°, 4ÇÐ³â\4ÇÐ³â
2ÇÐ±â\·î¿°ñÀü\Original Images\dipum_images_ch02\Fig0206(a) (rose-
original).tif'); % read the targeted image
Gy = [-1 0 1; -2 0 2; -1 0 1];
Gx = [-1 -2 -1; 0 0 0; 1 2 1];

%Make the mask ready to do convolution
Conv_Gy = Conv_rev(Gy);
Conv_Gx = Conv_rev(Gx);

[my, ny] = size(Conv_Gy);
[mx, nx] = size(Conv_Gx);

%Padding the original image
Pd_Rose_Gy = double(Zero_Pad1(IM_Rose, Conv_Gy));
Pd_Rose_Gx = double(Zero_Pad1(IM_Rose, Conv_Gx));

%Calculate the convolution
B1Conv_Rose_Gy = Conv_IM(Pd_Rose_Gy, Conv_Gy);
B1Conv_Rose_Gx = Conv_IM(Pd_Rose_Gx, Conv_Gx);

[my1, ny1] = size(B1Conv_Rose_Gy);
[mx1, nx1] = size(B1Conv_Rose_Gx);

my2 = fix(my/2);
ny2 = fix(ny/2);
mx2 = fix(mx/2);
nx2 = fix(nx/2);

%Cut the image as original size
B2Conv_Rose_Gy = B1Conv_Rose_Gy(1+my2:my1-my2, 1+ny2:ny1-ny2);
B2Conv_Rose_Gx = B1Conv_Rose_Gx(1+mx2:mx1-mx2, 1+nx2:nx1-nx2);

%Scaling Process
Min1 = min(min(B2Conv_Rose_Gy));
Max1 = max(max(B2Conv_Rose_Gy));

Min2 = min(min(B2Conv_Rose_Gx));
Max2 = max(max(B2Conv_Rose_Gx));

Conv_Rose_Gy = Scale(B2Conv_Rose_Gy, Max1, Min1);
Conv_Rose_Gx = Scale(B2Conv_Rose_Gx, Max2, Min2);

%Combine the result of vertical and horizontal convolution
Gy1 = Conv_Rose_Gy;
Gx1 = Conv_Rose_Gx;

Conv_Rose = uint8(sqrt((Gy1).^2+(Gx1).^2));

```

```
%Show the Images
figure, imshow(IM_Rose)
figure, imshow(uint8(Conv_Rose_Gx))
figure, imshow(uint8(Conv_Rose_Gy))
figure, imshow(Conv_Rose)
```

02 Result 1

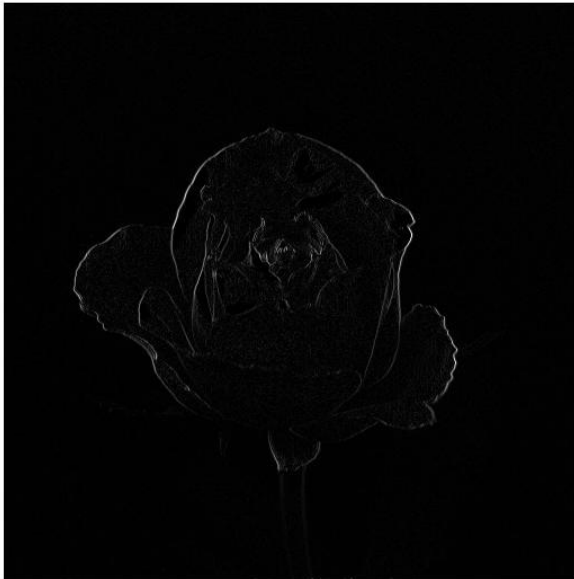
1. Original Image



2. Horizontal Edge Detector



3. Vertical Edge Detector



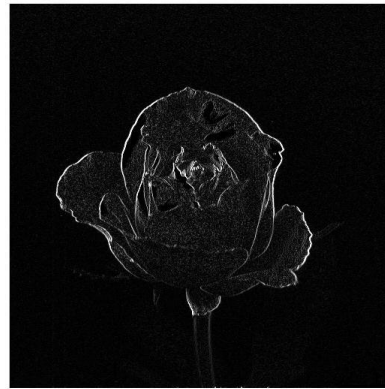
4. Combined Edge Detector



Conclusion

1. Matlab

- Zero Padding에 대한 견해 : 기존의 zero padding 함수의 방법으로는 다른 padding 함수를 만드는 것에 대하여 한계가 있었으나, 새로 만든 zero padding 함수는 다른 패딩 함수 적용에 용이하고, 코드도 간단하다.
- Diagonal Reverse Function의 경우 Convolution과 Correlation을 구분하기 위해 사용한 것이나 딱히 큰 차이는 없다.
- Scale 변환 함수를 쓴 이유 (scale 변환 안 했을 시)



이유 : 컨볼루션을 하고 나면 이미지가 255값이 넘어가는 Threshold 현상이 발생하기 때문이다.

따라서 이를 해결하기 위해 가장 큰 값을 찾아낸 후 비례적으로 scale을 조정하는 scale 변환함수를 사용하였다.

- Combine the Vertical and Horizontal Edge Detected Image

$$G = \sqrt{(G_x)^2 + (G_y)^2}$$

위의 식을 이용하여 총 합친 edge detected image를 만들 수 있다.