**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

1. **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

1. **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

1. **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

1. **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

1. **Main Code**

%%% Robot Vision%%%

%%% Dept. of Electronic Engineering

%%% 201314651 Lee Wonjai

IM\_Rose = imread('C:\Users\user\OneDrive\¹ÙÅÁ È­¸é\2019 Æ¯º°, 4ÇÐ³â\4ÇÐ³â 2ÇÐ±â\·Îº¿ºñÀü\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**

****

1. **Horizontal Edge Detector**



1. **Vertical Edge Detector**



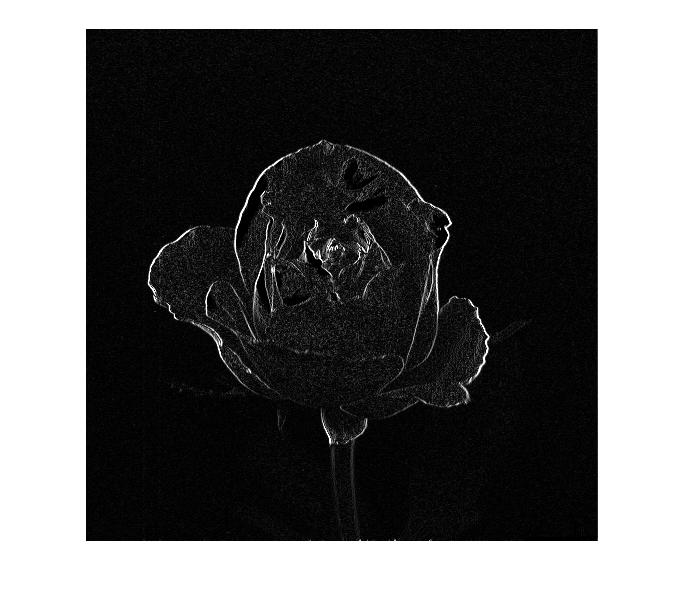
1. **Combined Edge Detector**



**Conclusion**

1. **Matlab**

* **Zero Padding**에 대한 견해 : **기존**의 zero padding 함수의 방법으로는 **다른 padding 함수를 만드는 것에 대하여 한계**가 있었으나, **새로 만든 zero padding 함수**는 다른 패딩 함수 **적용에 용이**하고, **코드도 간단**하다.
* 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 현상**이 발생하기 때문이다.

따라서 이를 해결하기 위해 **가장 큰 값을 찾아**낸 후 **비례적으로 scale을 조정**하는 **scale 변환함수**를 사용하였다.

* **Combine** the Vertical and Horizontal **Edge Detected Image**

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