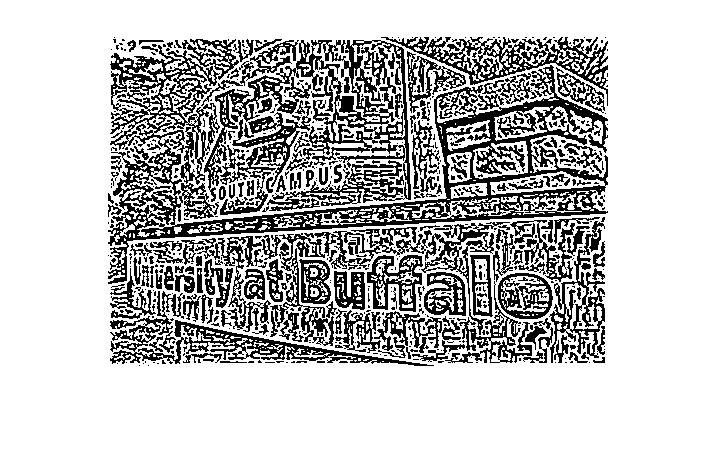
State University of New York at Buffalo CSE 473/573 Fall 2016 Homework Set #4 Date:

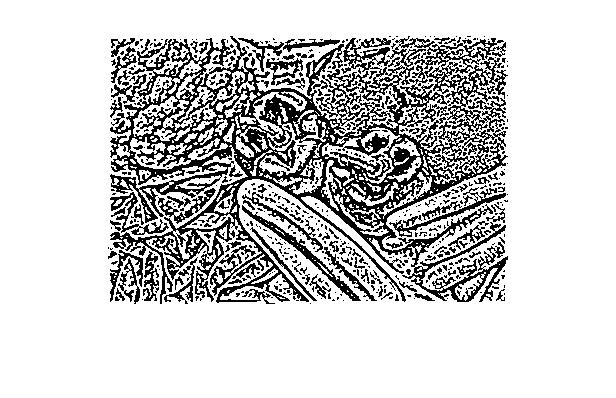
Name- Biplab Kumar Das

50170315

1. A)

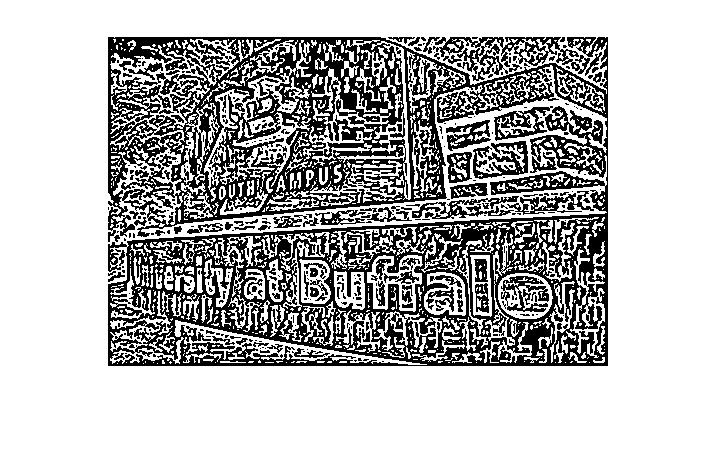
DOG Image

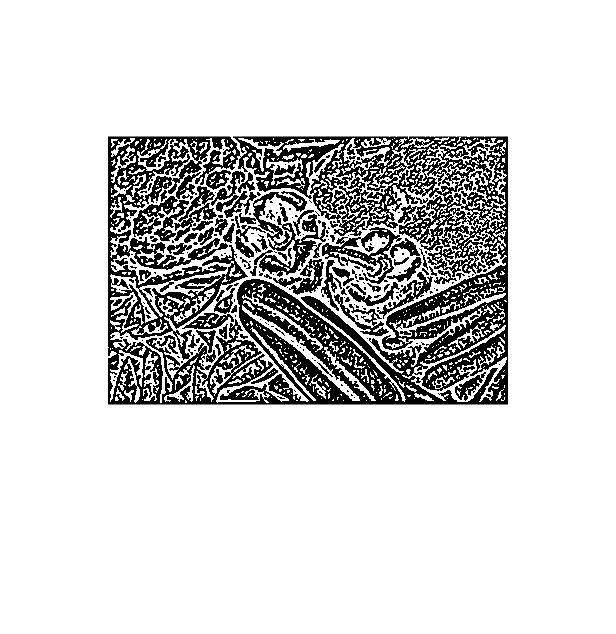




B)

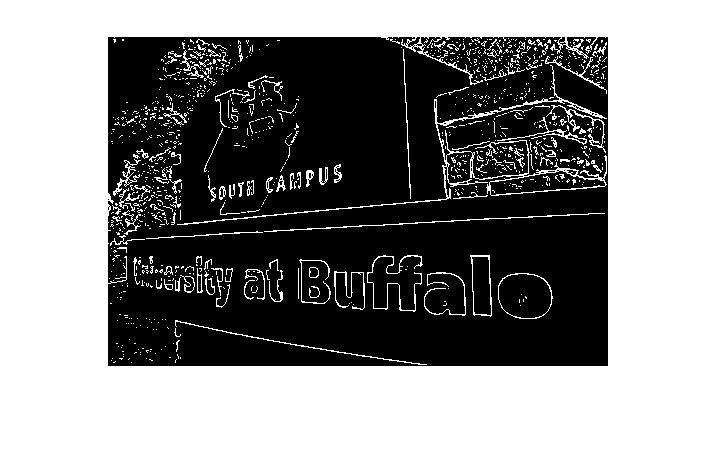
DOG Zero Crossing

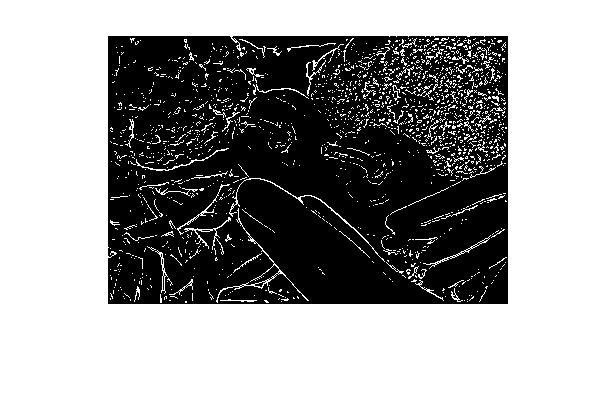




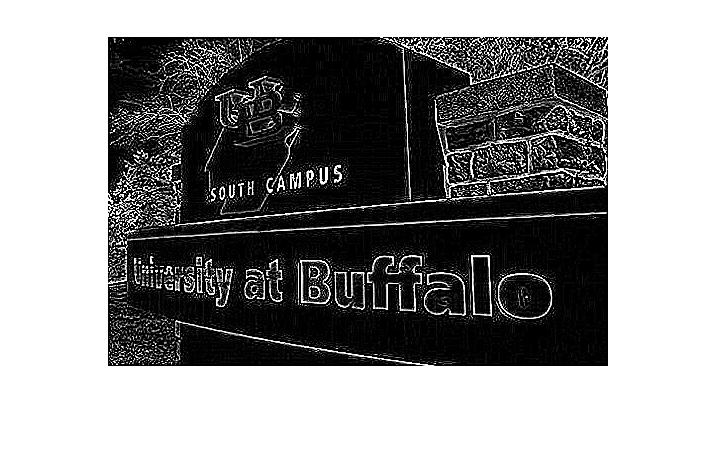
C)

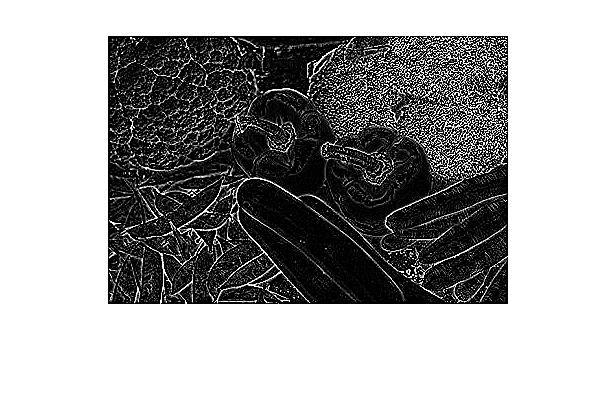
Strong Edges removing the weak edges



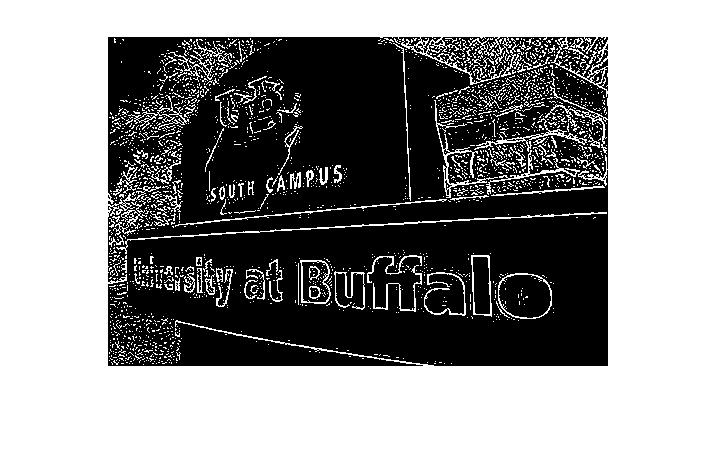


D)

LOG Image



LOG Zero Crossing





E)

Difference

The major difference is that there are large number of minute features which are getting detected in the log which may be due to noise or faint details.

Another difference is we are taking just the strong edges to display that means that the weak edges are suppressed or the minute details are suppressed, hence there is a difference between the 2 images. But if we can threshold the LOG output image to an appropriate threshold we can get the expected output as shown in c.

%%%%%% Code %%%%%%%%%%%

%% homework 4

clc;

I = imread('MixedVegetables.jpg');

I2 = imread('MixedVegetables.jpg');

I3 = imread('MixedVegetables.jpg');

im = im2uint8(rgb2gray(I));

%im = im2double(im);

[r, c] = size(im);

G=[ 0   0  -1  -1  -1  0   0;

    0  -2  -3  -3  -3 -2   0

   -1  -3   5   5   5 -3  -1

   -1  -3   5  16   5 -3  -1

   -1  -3   5   5   5 -3  -1

    0  -2  -3  -3  -3 -2   0

    0   0  -1  -1  -1  0   0];

im\_pad = PadZero(im, 7);

out = conv2d(G,im\_pad, im);

%-----------output part a -------------

figure

imshow(out);

%======================================

%% for checking zero crossing, checking the up, down, left and right for change of sign i.e 4 neighbours

out = PadZero(out, 3);

zc\_dog = zeros(r, c);

for i = 2:1:r

    for j = 2:1:c

        if(out(i,j-1) > 0 && out(i, j) < 0 || out(i,j) < 0 && out(i,j + 1) < 0 ||out(i,j) < 0 && out(i+1,j) < 0)

            zc\_dog(i - 1, j - 1) = 1;

        else

            zc\_dog(i - 1, j - 1) = 0;

        end

    end

end

% %-----------output part b -------------%

figure

imshow(zc\_dog);

%======================================%

im2 = im2uint8(rgb2gray(I2));

im2 = im2double(im2);

% Application of sobel Filter

h1 = [1  2  1

      0  0  0

     -1 -2 -1];

h3 = [-1 0 1

    -2 0 2

    -1 0 1];

g\_pad = PadZero(im2, 3);

Hx = fspecial('sobel');

Hy = Hx';

Gx = imfilter(im2, h1, 'conv');

Gy = imfilter(im2, h3, 'conv');

G = sqrt(Gx.^2 + Gy.^2);

% thresholding G with an appropriate value.

thrsh = graythresh(G);

S = im2bw(G,thrsh);

ZC = (zc\_dog);

%% doing logical and operation between both the matrices

cout = S & ZC;

%-----------output part c -------------%

figure

imshow(cout)

%=======================================

%%%%%%%%%%%%%%%%%%%%%%%% LOG part %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

im3 = im2uint8(rgb2gray(I3));

im3 = im2double(im3);

Lmsk = [0   0   1   0   0

       0   1   2   1   0

       1   2  -16  2   1

       0   1   2   1   0

       0   0   1   0   0];

lpad = PadZero(im3, 5);

LOG = conv2d(Lmsk, lpad, im3);

%-----------output part d -------------%

figure

imshow(LOG);

thrsh = graythresh(LOG);

SA = im2bw(LOG,thrsh);

figure

imshow(SA);

%=======================================

2)

%% region merging segmentation

I = imread('UBcampus.jpg');

im = im2uint8(rgb2gray(I));

im = im2double(im);

[r, c]=size(im);

newim = zeros(r+r, c+c);

[r1, c1] = size(newim);

tmp1 = 0;

tmp2 = 0;

for i = 1:1:r

    for j = 1:1:c

        newim(i + tmp1,j + tmp2) = im(i , j);

        tmp2 = tmp2 + 1;

    end

    tmp1 =tmp1 + 1;

    tmp2 =0;

end

%% for crack edges input the difference of the adjacent image pixels in alternate places

for i = 1:2:r1

    for j = 2:2:c1

        if(j == c1)

            newim(i,j) = ((newim(i,1) - newim(i,j - 1) ));

        else

            newim(i,j) =( ( newim(i,j + 1)- newim(i,j - 1) ));

        end

    end

end

for i = 2:2:r1

    for j = 1:1:c1

       if(i == r1 || j == c1)

            newim(i,j) =( ( newim(1,1) - newim(i - 1,j)));

       else

            newim(i,j) =( (newim(i + 1,j + 1) - newim(i - 1,j) ));

       end

    end

end

%===========crack image with double the size of image===================

% figure

% imshow(newim)

% setting a threshold of the image

T1 = graythresh(newim);

%

% making the values of the image to zero if it is less than the

% threshold only considering the crack image pixels

for i = 1:2:r1

    for j = 2:2:c1

        if(newim(i, j) < T1)

            newim(i, j) = 0;

        else

            newim(i, j) = 1;

        end

    end

end

for i = 2:2:r1

    for j = 1:1:c1

        if(newim(i, j) < T1)

            newim(i, j) = 0;

        else

            newim(i, j) = 1;

        end

    end

end

figure

imshow(newim)

% region labelling and segmentation

inpt = PadZero(newim, 3);

reg = zeros(size(newim));

reg(1,1) = 1;

for i = 1:2:r1

    for j = 1:2:c1

        if(i == 1 && j == 1) % 1st element top left

            if(inpt(i + 1, j) == 0)

                reg(i + 1 ,j) = 1;

            elseif(inpt(i, j + 1) == 0)

                reg(i  ,j + 1) = 1;

            elseif(inpt(i + 1, j) ~=0 ||inpt(i, j + 1) == 0)

                reg(i + 1 ,j) = 1;

            end

        elseif( i == 1 && j > 1)

            if(inpt(i, j -1) == 0) %left

                label = 1;

            elseif(inpt(i + 1, j ) == 0) % down

                label = 1;

            elseif(inpt(i , j + 1) == 0) % right

                label = 1;

            end

        elseif(i == r1 && j == 1 ) %bottom left

            if(inpt(i - 1, j) == 0)

                label = 1;

            elseif(inpt(i, j + 1) == 0)

                label = 1;

            end

        elseif(i == r1 && j == c1) %% bottom right

             if(inpt(i - 1, j) == 0)

                label = 1;

            elseif(inpt(i, j - 1) == 0)

                label = 1;

             end

        elseif(i == 1 && j == c1)%% top right

            if(inpt(i, j - 1) == 0)

                label = 1;

            elseif(inpt(i + 1, j) == 0)

                label = 1;

            end

        elseif(i == r1 && j > 1)

            if(inpt(i, j - 1) == 0) %left

                label = 1;

            elseif(inpt(i - 1, j ) == 0) % up

                label = 1;

            elseif(inpt(i, j + 1) == 0) % right

                label = 1;

            end

        else %% if r1 >1 and c1 >1

            if(inpt(i - 1, j) == 0 )

            elseif(inpt(i + 1, j) == 0)

            elseif(inpt(i, j - 1) == 0)

            elseif(inpt(i, j + 1) == 0)

            end

        end

    end

end

function y = marklabel(x, y ,reg)

end