CSCI-631: Foundations of Computer Vision

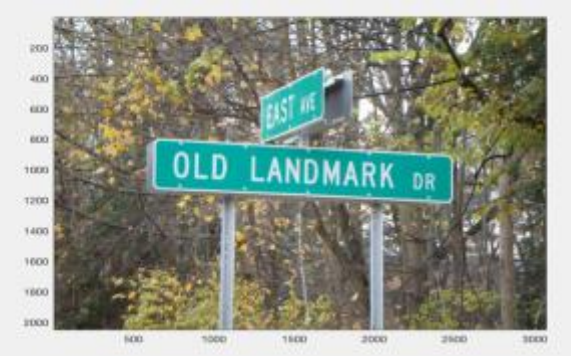
Submitted by: - Yash Jain (yj8359)

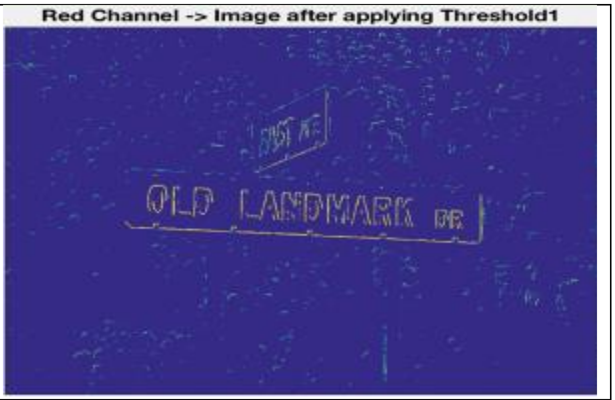
HOMEWORK – 7B

NOTE:- Though I am putting screenshots of output image, I am writing it to file using imwrite command. The reason I am putting screenshots because I can put title on image using title() command to distinguish between images. But I am also saving it to file using imwrite().

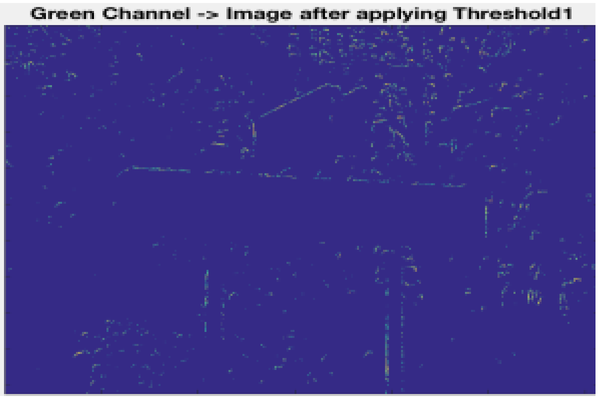
I am asked to work on '***TBK_OLD_LANDMARK_Dr.jpg*'.**

Original Image

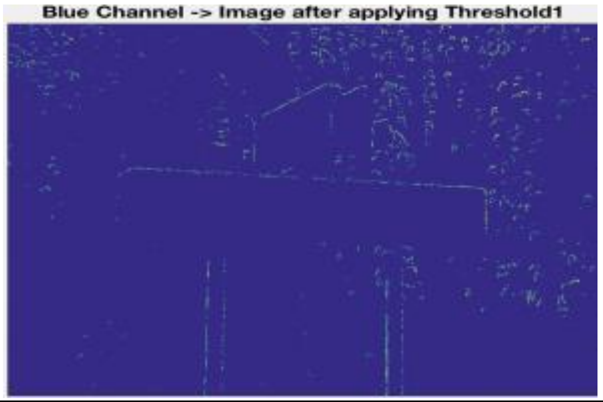




Red Channel -> Image after applying Threshold1



Green Channel -> Image after applying Threshold1



Blue Channel -> Image after applying Threshold1

NOTE: - The above images are generated using value of standard deviation = 20 and threshold1 = 0.03. Discussion about effect of changing these values is below.

Also, all the above 3 output images are **not binary** as they have some edge strength (not just 0 or 1).

The above images show that **red channel** is giving the best outcome after applying threshold1 that is non-maximal suppression.

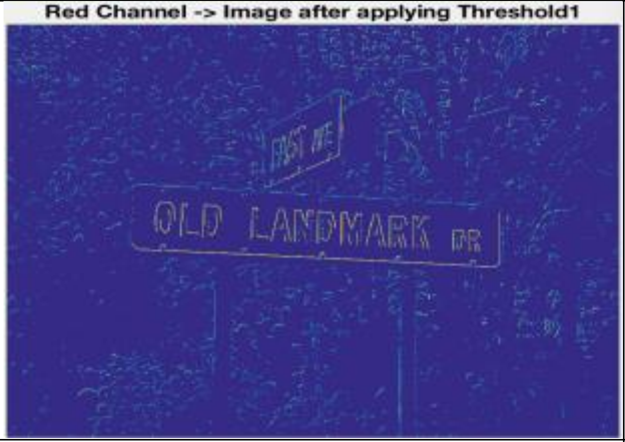
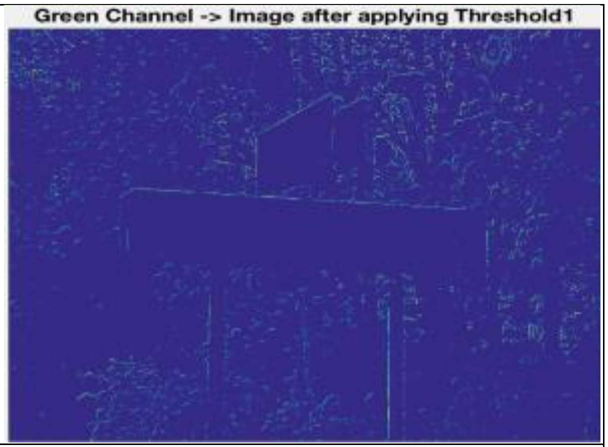
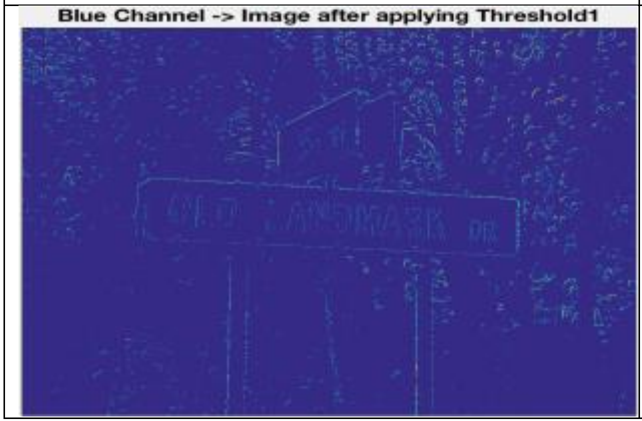
I worked with all the 3 color channels of input image and applied canny edge algorithm on all 3 channels. I can see better output on Red. Green is not a good choice because the background of sign board is green in color.

Effect of changing Sigma values:

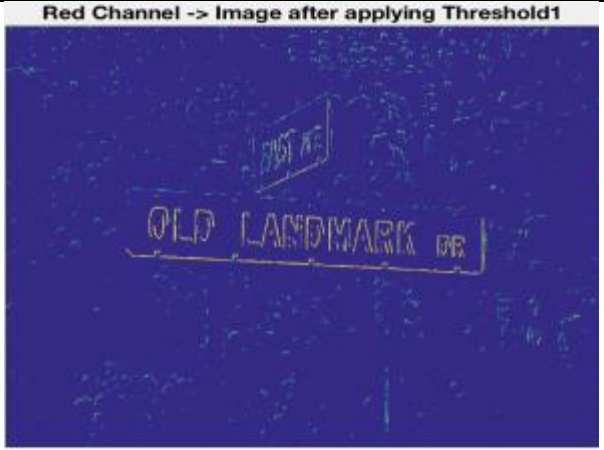
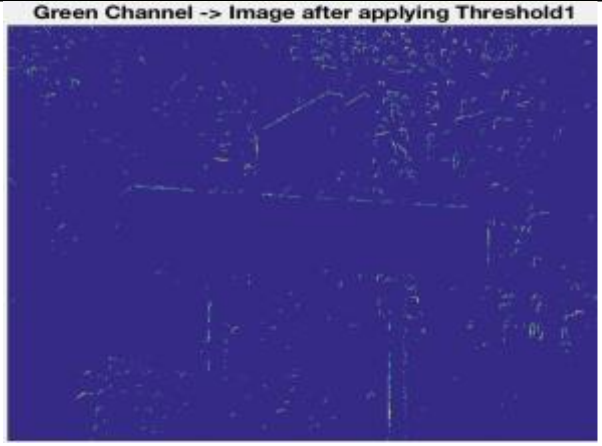
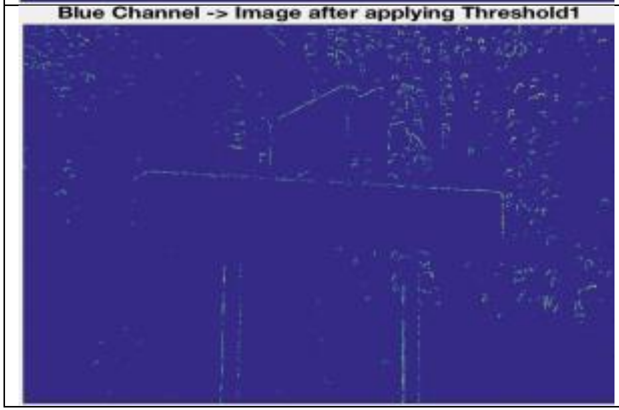
If we increase the sigma value while applying the Gaussian filter then it will give more smoothened image as output. So, if we apply canny edge detection on such image, then we will tend to lose edges as there will be lesser edges in a smooth image. So, with increase in sigma values, we will lose more edges. Various output with different sigma values are given below.

Final output obtained by changing the values is below:

For sigma (standard deviation) = 5

 Red Channel -> Image after applying Threshold1	 Green Channel -> Image after applying Threshold1
 Blue Channel -> Image after applying Threshold1	

For sigma (standard deviation) = 20

 Red Channel -> Image after applying Threshold1	 Green Channel -> Image after applying Threshold1
 Blue Channel -> Image after applying Threshold1	

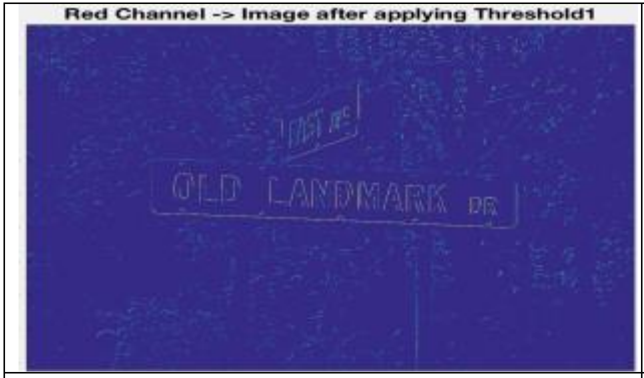
Observation:-

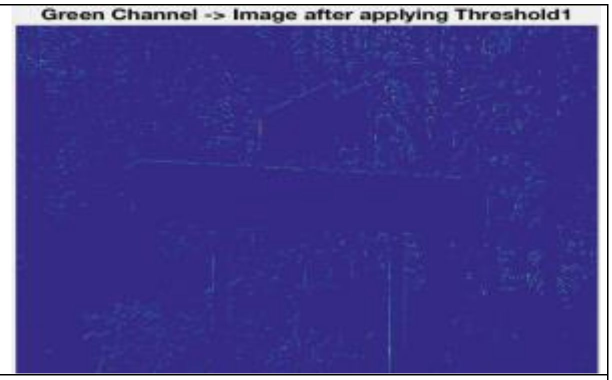
- The above picture shows that for higher values of sigma, we will start losing edge details.
- This will remove unwanted noise edges that are coming in for lower values of sigma.
- To achieve goal, that is to retrieve text from the sign board, I found sigma = 20 works great on Red Channel.

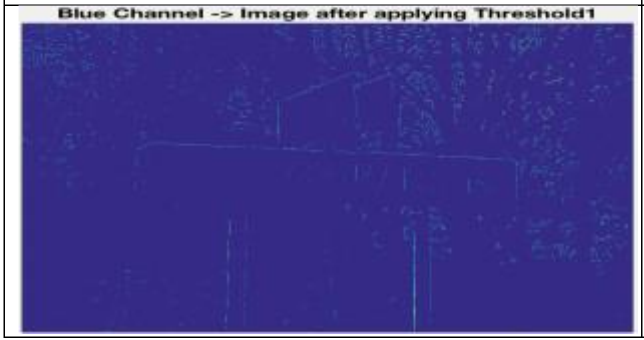
<u>Effect of changing Threshold1 values:</u>

We need to choose the threshold1 value wisely to get the desired output. So, choosing threshold value along with sigma is equally important. Threshold1 should have lower value so that important edges are not getting lost.

Below image is for standard deviation = 2, threshold1 = 0.05

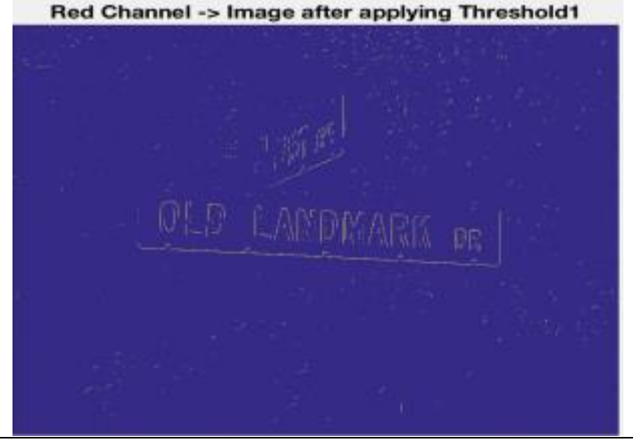


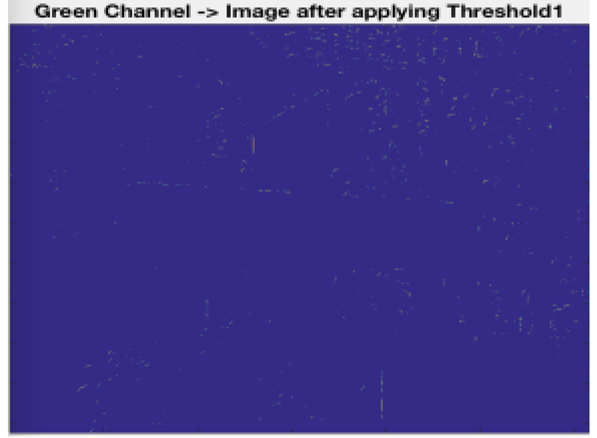


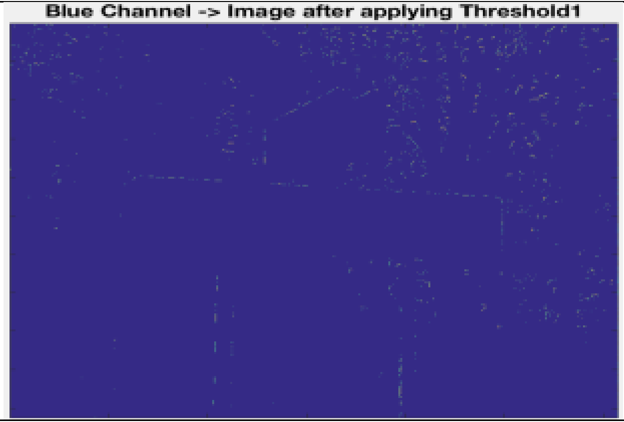


We are getting respectively good output with sigma =2 and threshold1 = 0.05

Below image is for standard deviation = 2.5, threshold1 = 0.07

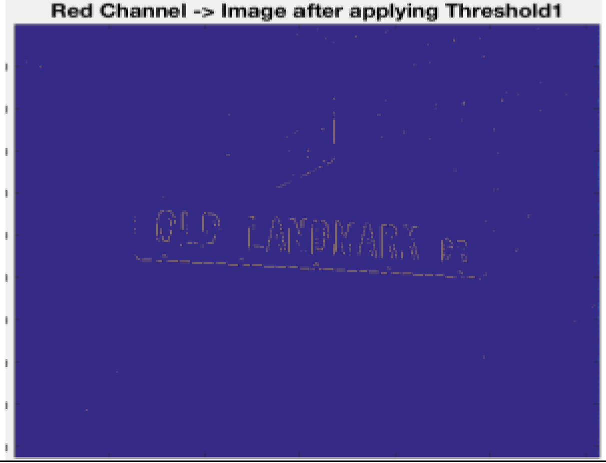
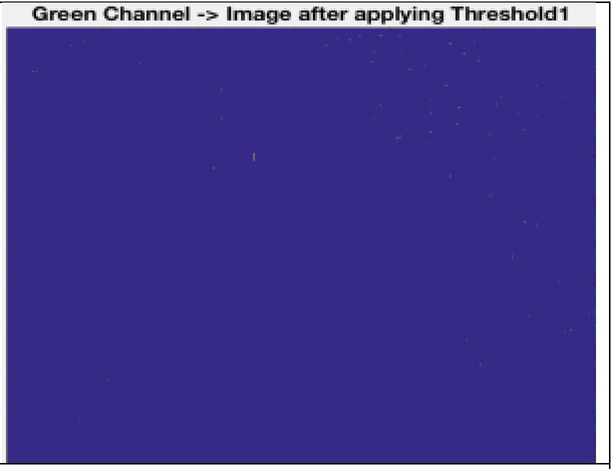
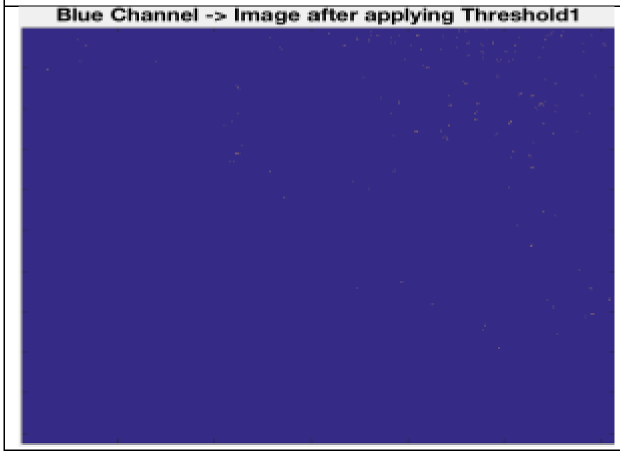




We are losing 'East Ave' written on upper board with sigma = 2.5 and threshold1 = 0.07

Below image is for standard deviation = 3, threshold1 = 0.09

We are losing 'East Ave' written on upper board with sigma = 3 and threshold1 = 0.09

- **So, we understood that we need to choose proper values of Gaussian matrix width, standard deviation and threshold1 to get the desired output. As, we increase the sigma value we will discard many edges that are not needed to achieve our goal. But if we increase it furthermore, we will start losing the important edges as well.**

Steps to apply canny: -

1. Smoothen the image using Gaussian filter.
2. Choose proper values of filter width and standard deviation(sigma).
3. Then choose red color channel.
4. Apply vertical and horizontal gradient sobel filter.
5. Compute edge magnitude.
6. Compute angle.
7. Choose threshold1.
8. For every pixel find its angle, based on the angle value choose surrounding pixel to compare.
9. Compare the edge magnitude with surrounding pixels chosen. Also compare it with threshold 1 value. Like for 0 degree compare with a pixel ahead and a pixel behind.
10. If value is greater than surrounding pixel and threshol1, then keep it else discard it by putting 0 the pixel value.
11. Repeat the step 4 to step 10 for other color channels as well.
12. Find which color channel is best to work with.

After that I have applied hysteresis which I have described in other write-up file.

New Learnings:-

Learnt how to use imwrite command to write image to a file.
Learnt how canny edge works.
How to use Hysteresis to enhance the edges.

Code to perform non-maximal suppression operation using threshold1 :-

```
temp = edge_mag;
  dims = size(temp);

  %Loops through entire image to check below conditions
  for ii = 2:dims(1)-1
    for jj = 2:dims(2)-1
        %Finding appropriate row and column based on their angle values to perform
         operation.
        if angle(ii,jj) == 0 || angle(ii,jj) == -180 || angle(ii,jj) == 180
            tempi1 = ii;
            tempi2 = ii;
            tempj1 = jj-1;
            tempj2 = jj+1;
        elseif angle(ii,jj) == 45 || angle(ii,jj) == -135
            tempi1 = ii-1;
            tempi2 = ii+1;
            tempj1 = jj+1;
            tempj2 = jj-1;
        elseif angle(ii,jj) == 90 || angle(ii,jj) == -90
            tempi1 = ii+1;
            tempi2 = ii-1;
            tempj1 = jj;
            tempj2 = jj;
        elseif angle(ii,jj) == 135 || angle(ii,jj) == -45
            tempi1 = ii-1;
            tempi2 = ii+1;
            tempj1 = jj-1;
            tempj2 = jj+1;
        end
        %chekcing if the selected 2 pixels have smaller values
        %then the current pixel and if current pixel is atleast
        %threshold1 else reject that current pixel by setting its
        %value to 0
        if ((edge_mag(ii,jj) >= edge_mag(tempi1,tempj1)) && (edge_mag(ii,jj) >=
            angle(tempi2,tempj2)) && (edge_mag(ii,jj) >= threshold1))
            continue;
        else
            temp(ii,jj) = 0;
        end
    end
  end
  %saving final outpur to returning variable
  intermediate_out = temp;
```

Final output: -

I am able to achieve the goal of reading the text on the sign boards by using values as follows: -

Parameter	Value
Color channel	Red
Standard Deviation	20
Size of Gaussian filter matrix	14
Threshold1	0.03