Computer Vision CS518 (Assignment 1)

Arshdeep Singh

2020CSB1074

Task 1:

Canny Edge Detection involves multiple steps

Noise reduction (using Gaussian filter), edge detection (implemented Sobel), non-maximal Suppression(using the direction matrix obtained in Sobel), double thresholding(classifying weak and strong edges), and Hysteresis (Removing the invalid weak edges and merging the valid ones with strong)

The basic idea behind detecting Threshold points,

- Initially, I observed the maximum and minimum values of nms_image (image after non_maximal_supprression)
- Then based on the above observation, chose a very loose low and high threshold
- Then tried to fine-tune them, using the observation that points above the high threshold will be classified as strong and below the low threshold as zero. So, in between points would be weak edges.

Following are a few observations:

Low Threshold	High Threshold	PSNR (%)	SSIM
0.28	0.38	71.11	12.9
0.25	0.4	70.69	13.0
0.2	0.35	66.15	12.12

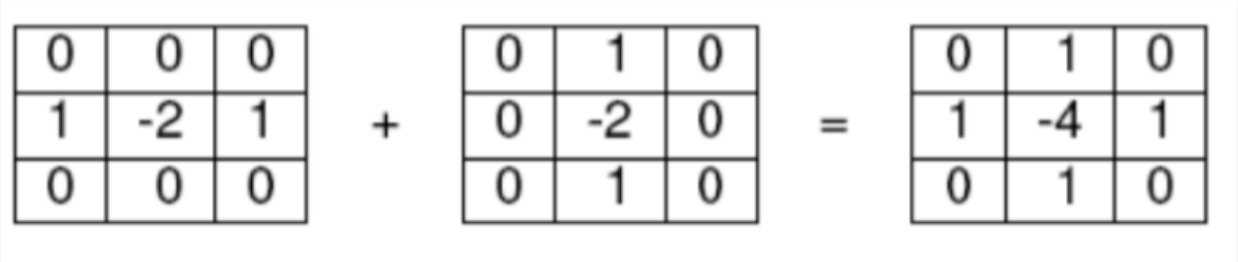
Thus, by adjusting threshold values, *myCannyEdgeDetector*() was able to product PSNR of about 65 to 75 percentage

Task 2:

Detect whether the image is Blur or Not,

Following steps were performed,

- Applied Laplacian filter



- Calculated Variance of the above-filtered image
- Then took a data set of few images and found maximum variance that ensures, sharp image
- Used linear mapping to find the corresponding Blur% of the image

BlurOrNot(), prints the % blur for given input image

Further Scope of improvement:

Since more is the variance, sharper is the image. I have used basic linear mapping. , But provided with a well-labeled dataset, we can directly use regression models for calculating more accurate results