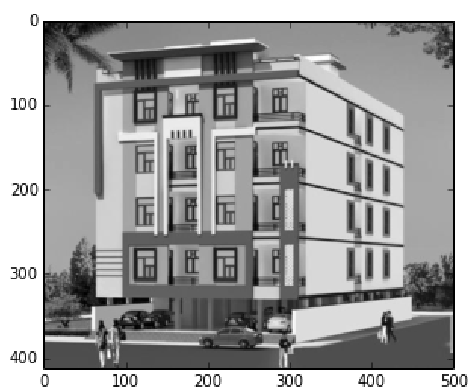
Edge and Corner Point Detection - Assignment 1

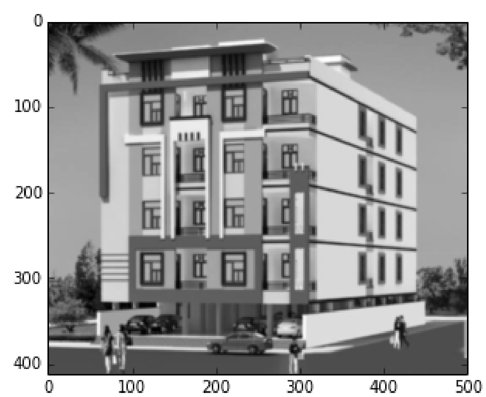
Name - Ashritha Kandiraju
Reg.No - 14CO121

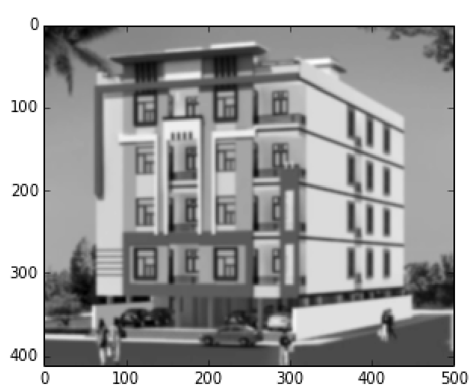
1. **GAUSSIAN FILTER:**
    a. The effect of Gaussian smoothing is to blur an image, in a similar fashion to other filters like mean filter. As the gaussian outputs a weighted average of each pixel's neighborhood, with the average weighted more towards the value of the central pixels, it has been observed that Gaussian provides gentler smoothing and preserves edges better than a similarly sized mean filter.
    b. The degree of smoothing is determined by the standard deviation of the Gaussian kernel which is calculated from the sigmax and sigmay values. In general the kernel size must increase with increasing standard deviation to maintain the gaussian nature of the filter.
    c. However, the images that have been used for experimentation were of good quality and smoothing the images for noise reduction using different gaussian kernels did not have much of a difference in output.

Kernel size = 3x3, sigmax,sigmay = 0

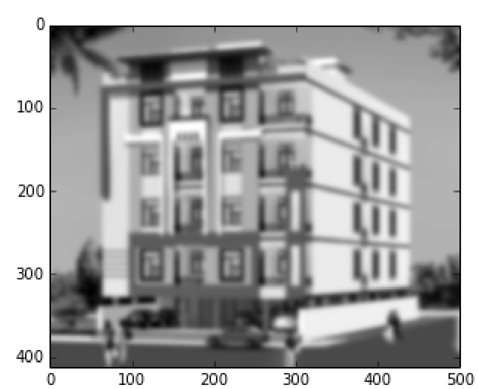


Kernel size = 5x5, sigmax,sigmay = 0





Kernel size = 9x9 , sigmax = sigmay = 3

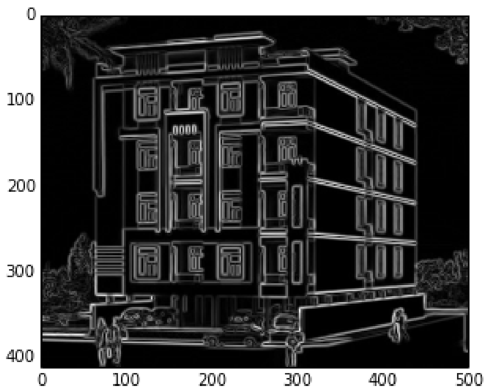


Kernel size = 25x25 , sigmax = sigmay = 3

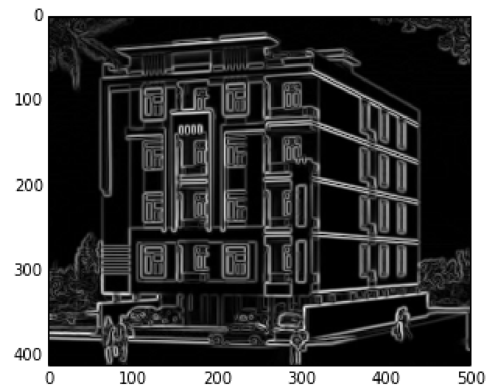
2. CALCULATION OF INTENSITY GRADIENT :

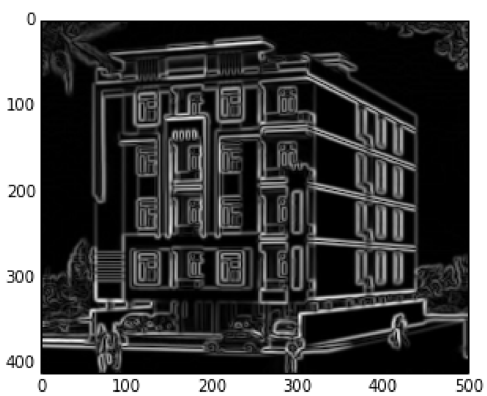
a. An edge in an image may point in a variety of directions, so the algorithm aims to filter and detect horizontal, vertical and diagonal edges in the blurred image. Any edge detector operator (Robert, Prewitt or Sobel) can be used to calculate the first derivative in horizontal and vertical direction. The magnitude of gradient and the direction of a pixel will be further calculated using these derivative values.
b. The experiments have been carried out with varying filter sizes of a **Sobel filter** and decent results have been obtained for filter_size = 3 in both x and y directions.
c. With increase in kernel size, the image tends to get more blurry as a consequence of which we might lose out on the edges. Also larger kernel size requires higher time for pixel convolution. Larger kernel size tends to make the edges unclear and bright.

Kernel size = 3x3

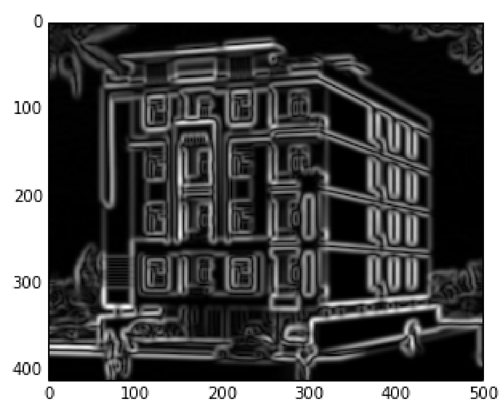


Kernel size = 5x5





Kernel size = 9x9



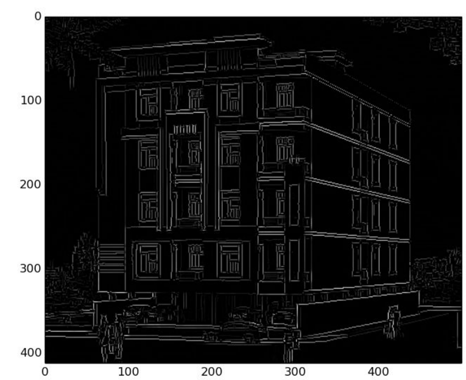
Kernel size = 25x25

3. NON-MAXIMUM SUPPRESSION

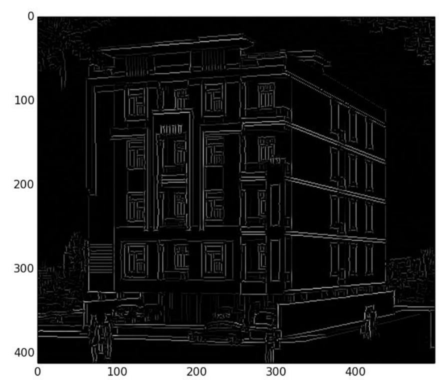
a. Non-maximum suppression is an edge-thinning technique. Non-maximum suppression can help to suppress all the gradient values (by setting them to 0) except the local maxima, which indicate locations with the sharpest change of intensity value.

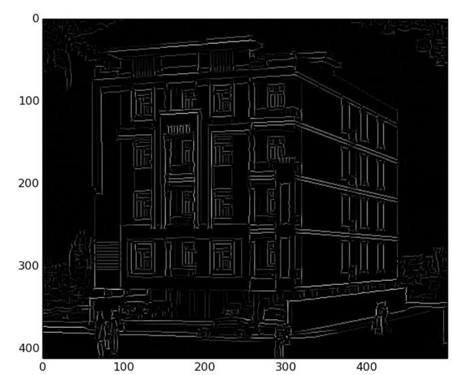
b. At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction.
c. Experiments have been carried out by varying kernel sizes of the Sobel filter. Thinner edges i.e better suppression has been observed with a kernel size of 5 as compared to kernel size of 3. Kernel sizes of 9 and 25 though have produced relatively thinner edges have resulted in unnecessary blurring. Therefore, decent results have been obtained with kernel size = 5.

Kernel size = 3x3

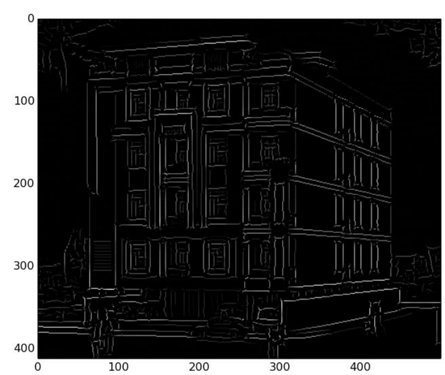


Kernel size = 5x5





Kernel size = 9x9



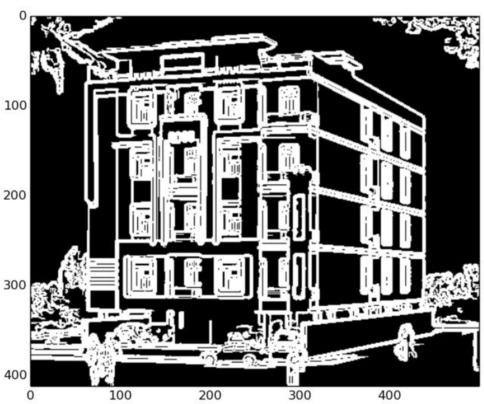
Kernel size = 25x25

4. DOUBLE THRESHOLDING

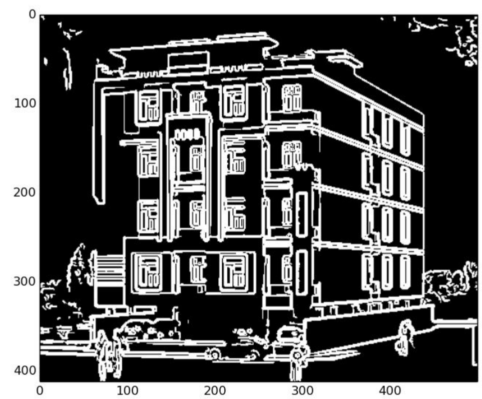
a. After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise or color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's value is smaller than the low threshold value, it will be suppressed. The two threshold values have to be empirically determined and are extremely dependent on the nature of the input image.

b. Different values have been tried out for the low and high values of thresholds. The sobel filter size did not matter much in this case and a smaller kernel size of 3 has been used for faster computation.
c. Lower thresholds result in unnecessary edges whereas higher thresholds result in the elimination of a few important edges. Decent results have been obtained by setting low = 100, high = 150.

low=50, high=100



low=100, high=250



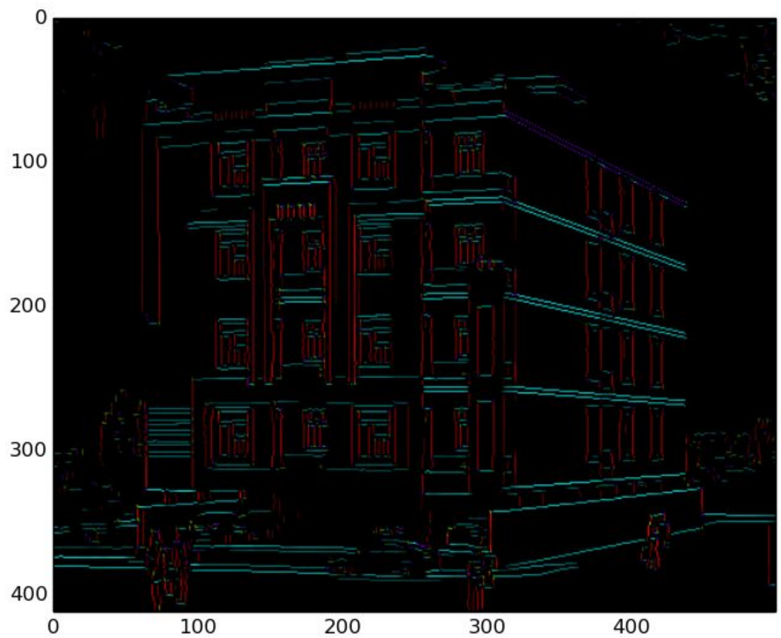
low=100, high=150

low=150, high=250

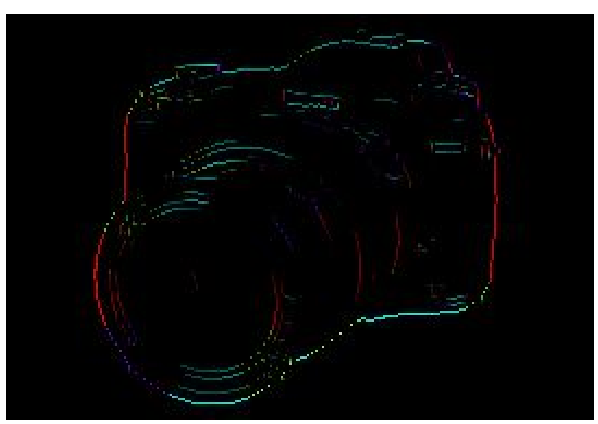
5. HYSTERESIS TRACKING

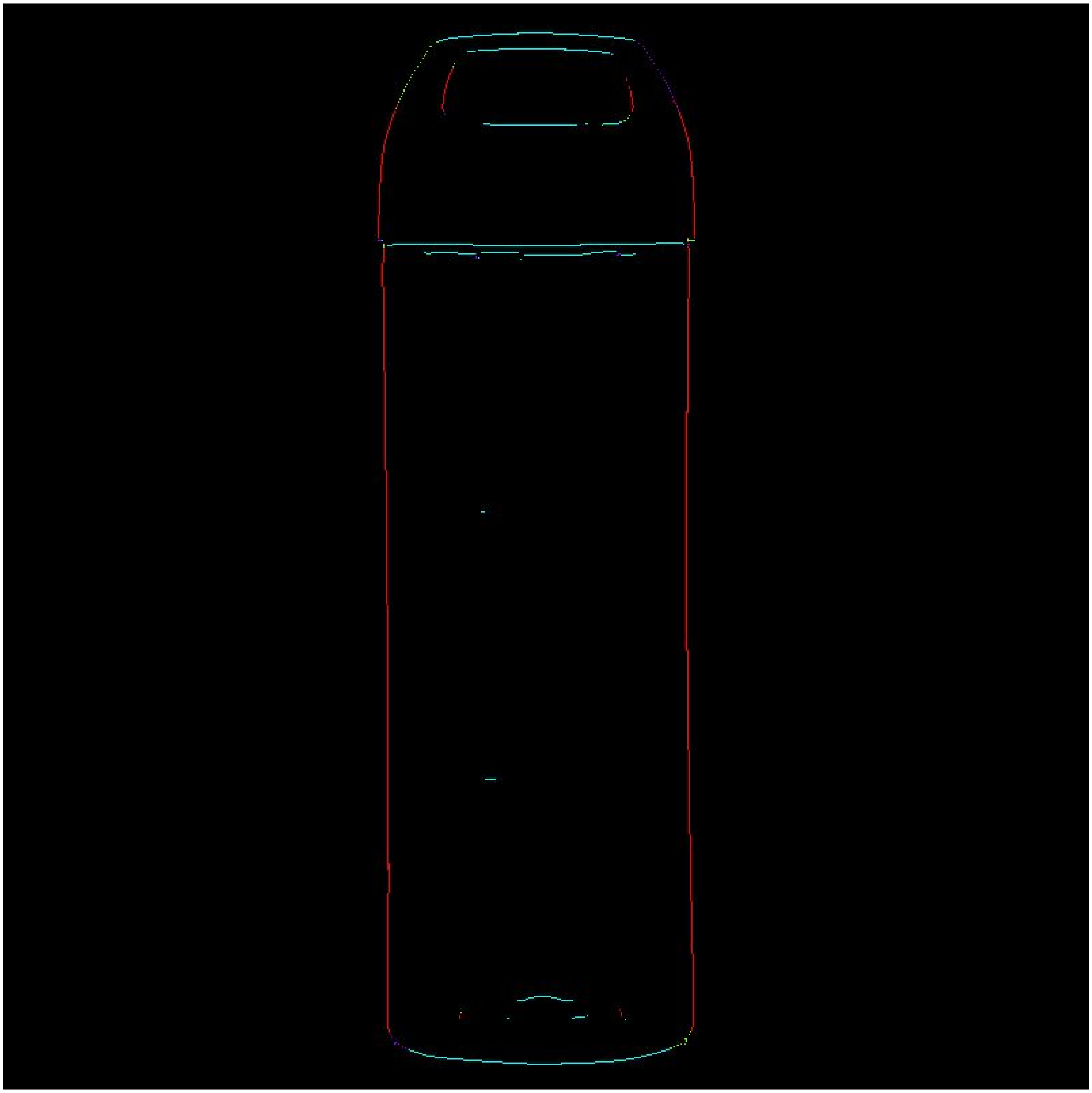
a. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved. The threshold values in this case are the same as those used in double thresholding.

b. Experiments have been carried by varying kernel sizes of gaussian and sobel filters and also the threshold values. The results of the experiments can be viewed in the **output** folder. Best results have been obtained with gaussian filter size = 5, sobel filter size = 3, low = 100 , high = 150.



c. Experiments with other images of a camera, car have been carried out and good results have been obtained. The edge detector was able to identify curved edges as well as edges along the corners without introduction of many spurious edges.





d. The edge image is colored based on HSV colorspace. The colors alloted to the edge pixels depend on the magnitude and direction of gradient at the particular pixel. The direction decides the color and magnitude decides the intensity of that color.

6. HARRIS CORNER DETECTION

a. Gaussian window has been used for corner detection. Gaussian window_size = 5 and sobel filter size = 3 have been used for the experiments. Tuning of harris constant and threshold values has been given more focus. Theoretically harris constant can assume values between 0.04 and 0.06 and the threshold values have been varied from 0.01 to 0.1.

window_size	filter_size	harris_constant	threshold	Number of corner pixels detected
5	3	0.04	0.01	21793
5	3	0.04	0.03	12569
5	3	0.04	0.05	8838
5	3	0.04	0.10	4328
5	3	0.06	0.01	20088
5	3	0.06	0.03	11564
5	3	0.06	0.05	8153
5	3	0.06	0.10	3992

b. Few observations from the above table are:
   - Increasing the harris constant value decreases the number of detected corners. This leads to a decrease in false corners as well as corners that need to be detected.
   - Increasing the threshold decreases the number of corners thereby missing out on important corner pixels which would otherwise have been detected.
   - The harris constant and threshold values have to be appropriately chosen as they depend on the image content to a large extent.

c. All the experimented outputs can be seen in the output folder.

REFERENCES :

1. https://docs.opencv.org/3.3.1/da/d22/tutorial_py_canny.html
2. https://en.wikipedia.org/wiki/Canny_edge_detector
3. http://dasl.unlv.edu/daslDrexel/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html
4. http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf

5. https://en.wikipedia.org/wiki/Corner_detection#The_Harris_.26_Stephens_.2F_Plessey_.2F_Shi.E2.80.93Tomasi_corner_detection_algorithm
6. https://stackoverflow.com/questions/29731726/how-to-calculate-a-gaussian-kernel-matrix-efficiently-in-numpy