Digital Image Processing
Assignment 4

Tejash More
M.Tech SP

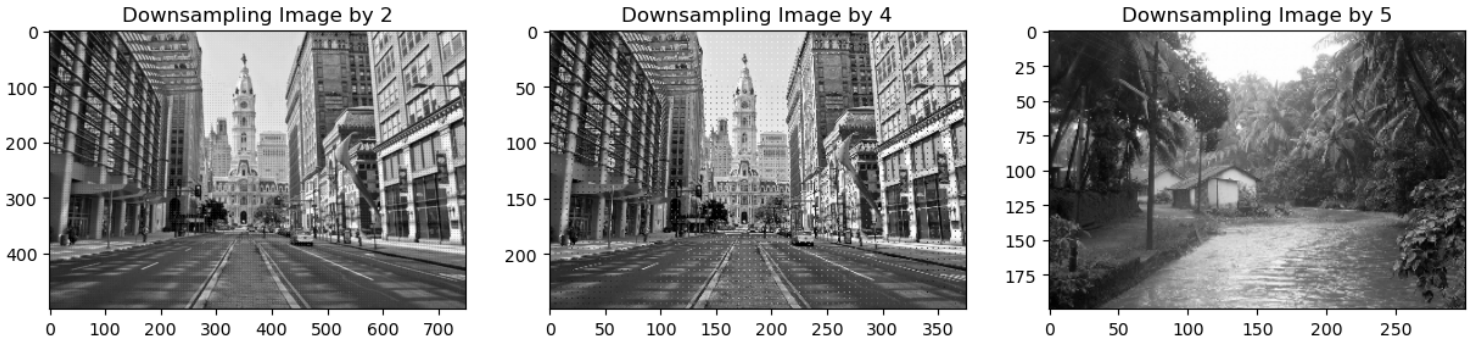
Packages used: NumPy, Pillow, Matplotlib, Scipy

1. Image Down-sampling

An image is downsampled by a factor of 2, 4, or 5. Note that in this Assignment, Interpolation has not been done to understand the effect of downsampling.

The dimension of this image is : (1000, 1500)

a. Downsample by a factor of 2, 4, and 5



The more the downsampling factor increases, the more pronounced the jarred pattern image becomes. This is due to aliasing, where the high-frequency pattern becomes aliased with the low-frequency pattern. Thus, in the image, which is downsampled by a factor of 2, the jarred line pattern is barely visible. In contrast, in the image downsampled by a factor of 4, the jarred line pattern is significantly visible.

In the Image downsampled by 5, we see another image, because the original image is a trick image, where each index multiple of 5 has a pixel value from this other image (nature).

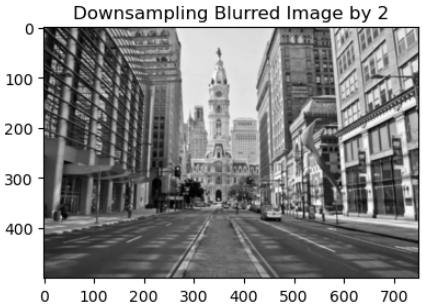
b. Anti-Aliasing Filter before downsampling

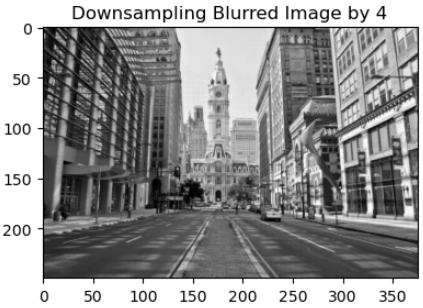
We observed that downsampling directly results in a jagged line pattern in the image. This is because of the high frequency getting aliased as a low frequency. One way to solve this is to remove the high-frequency elements before downsampling.

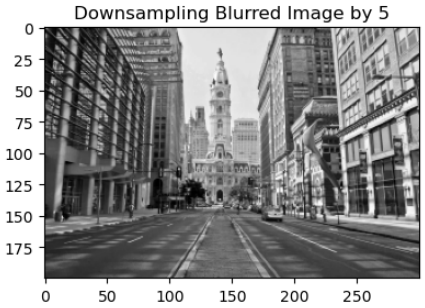
Removing high-frequency elements is done by a low-pass filter. We have used a Gaussian Low-Pass Filter with a window size of 5x5 and sigma of 2.

Digital Image Processing
Assignment 4

Tejash More
M.Tech SP



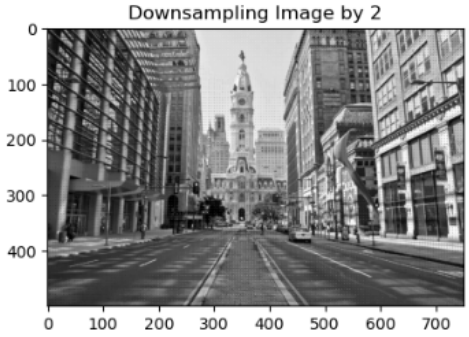




Observe that,

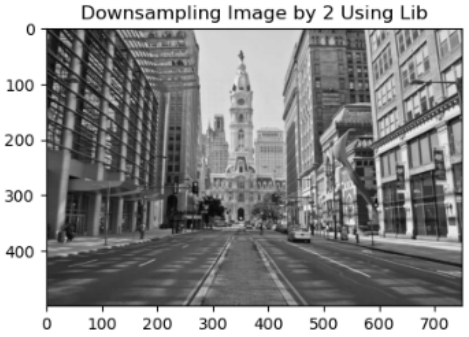
Here, downsampling by a factor of 5 does not produce the trick image, as we had seen previously. This is due to the image being passed through a low-pass filter. A low-pass filter blurs the image, removing the high-frequency components. The trick image, which has different image pixels at pixel locations that are multiples of 5, will be passed through this Low Pass Filter, resulting in an image where each pixel location has a pixel value that is the average of the pixel values of its neighbours within the window of the Gaussian Kernel.
Hence, the resultant trick image's pixel values are no longer present.

Furthermore, let's compare our downsampled image with the one obtained by the library functions.

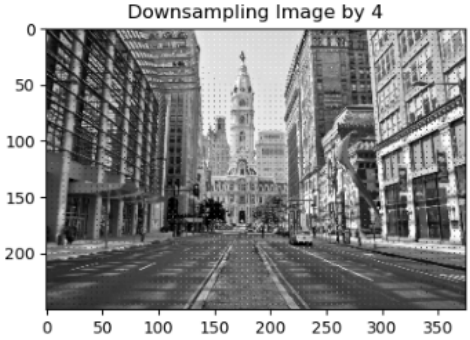


MSE: 89.45

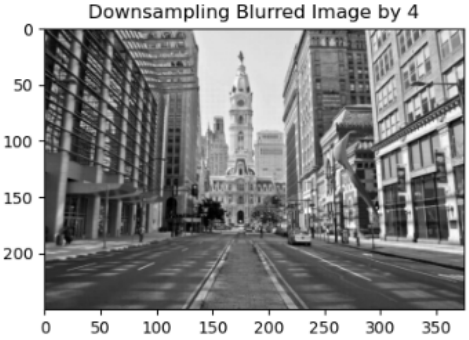
MSE: 38.79



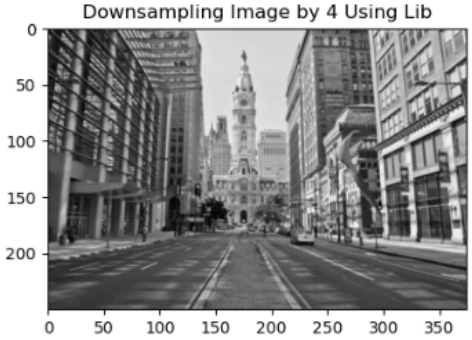
MSE: 0 (this is ref img)



MSE: 206.23



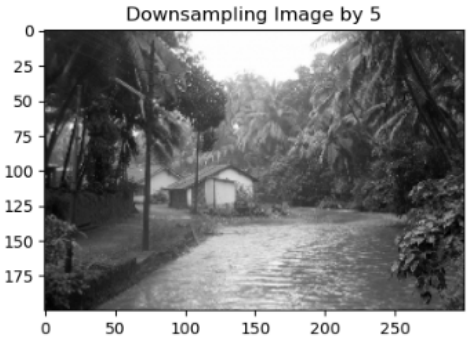
MSE: 66.98



MSE: 0 (this is ref img)

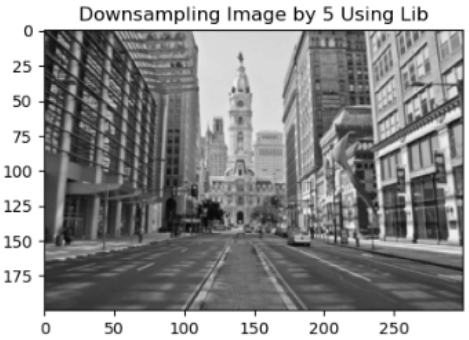
Digital Image Processing
Assignment 4

Tejash More
M.Tech SP



MSE: 1494.03

MSE: 94.06



MSE: 0 (this is ref img)

Observation:

MSE for the downsampled image after being passed through an Anti-Aliasing Filter is less in all three cases. This proves that the standard for downsampling is first to pass it through an Anti-Aliasing Filter.

c. Find the Optimal Window and Standard Deviation for the Gaussian Noise Filter for the Downsampling Factor of 5.

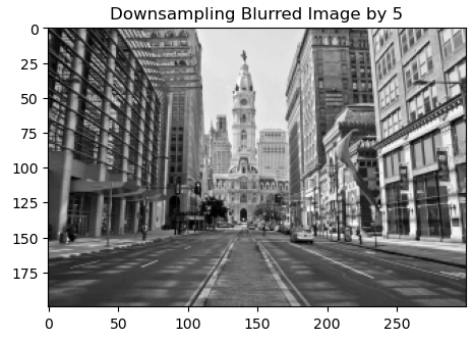
The window sizes considered are: [3, 5, 7, 9, 11, 13, 15, 17, 19].

The Standard Deviation list considered is:
[0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 , 2.2, 2.4, 2.6, 2.8, 3.0 , 3.2, 3.4, 3.6, 3.8, 4.0]

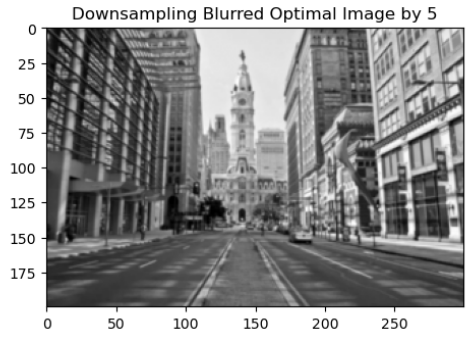
All combinations of these two are taken into consideration for the Low Pass Filter, and the MSE of the downsampled image is calculated. The reference image for the MSE is found using the library Function.

Optimal Value:

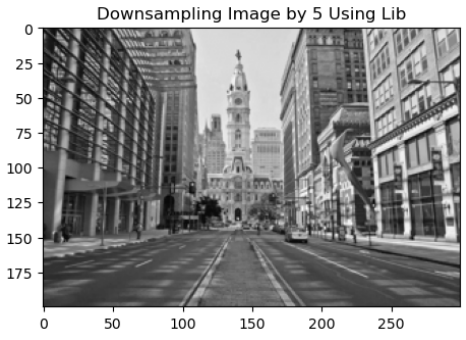
i) Optimal Window: 11
ii) Optimal Standard Deviation: 2.80
iii) MSE for this Combination: 73.01
Note that this MSE in part b, using the 5x5 window and standard deviation = 2 for the Low Pass Filter, gives the MSE = 94.06



MSE: 94.06



MSE: 73.01



MSE: 0 (this is ref img)

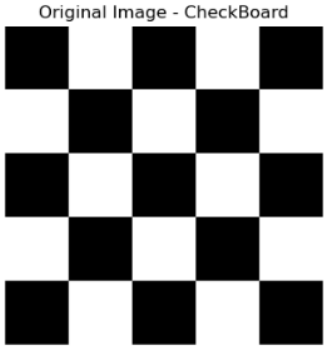
This was downsampled with the optimal low-pass filter as an anti-aliasing filter before the picture was downsampled.

Digital Image Processing
Assignment 4

Tejash More
M.Tech SP

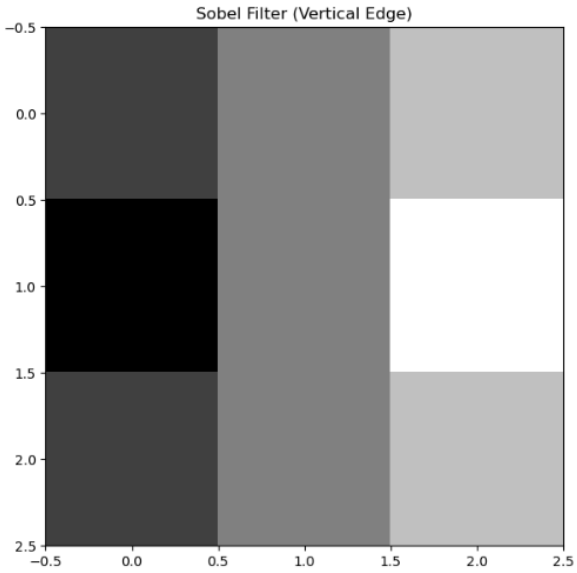
2. Edge Detection

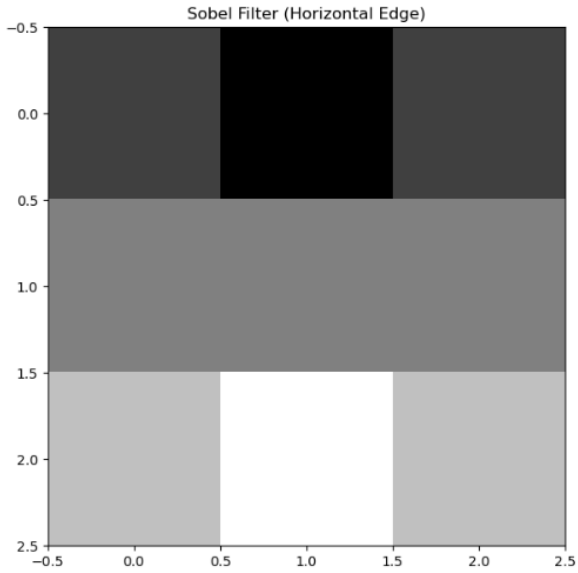
In this Question, Edge Detection has been performed on a set of 4 images.



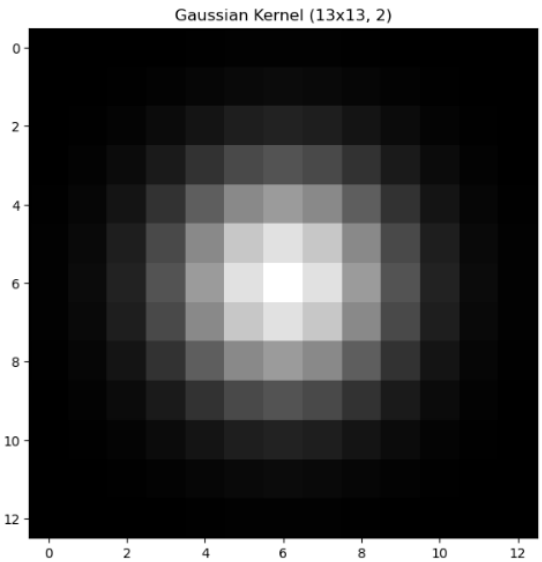
Two Approaches have been used for Edge Detection:

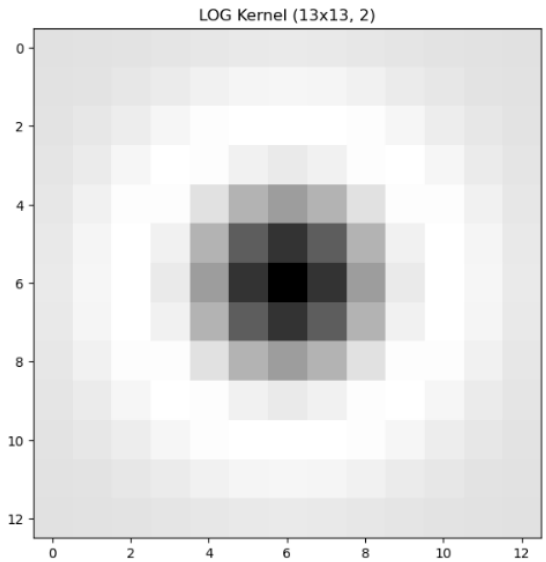
i) **Gradient-Based Approach**: Used the Sobel Filter for this





ii) LOG (Laplacian of Gaussian) Operator





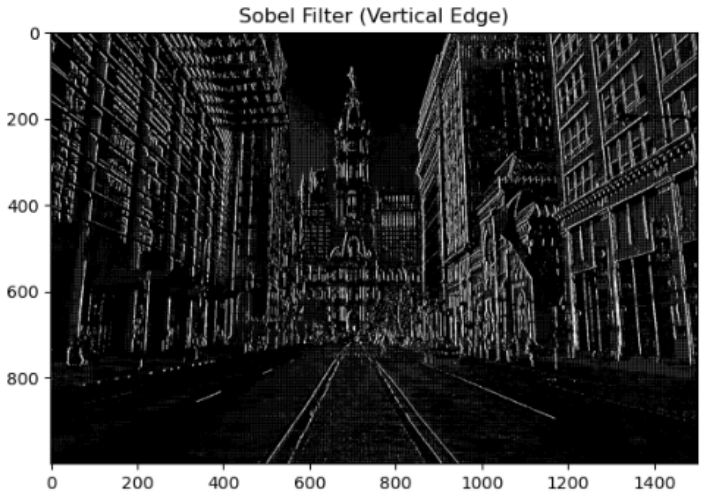
Digital Image Processing
Assignment 4

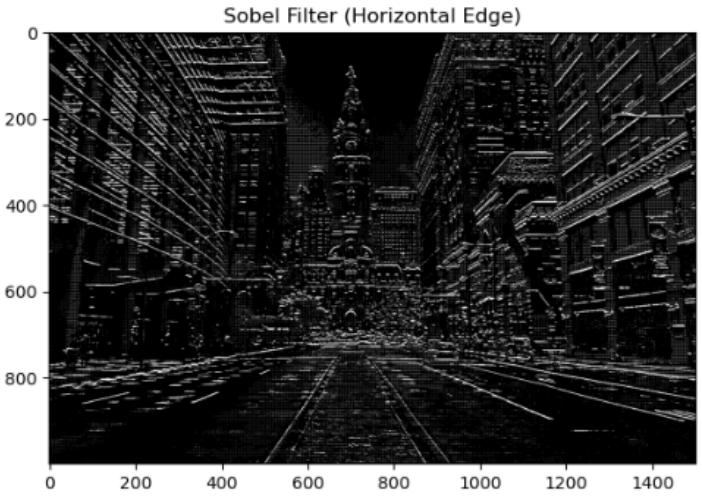
Tejash More
M.Tech SP

Theory:

In the Gradient-Based Method:

Sobel Filter, both for Horizontal Edge and Vertical Edge, is convolved with the image, resulting in two images highlighting vertical and horizontal edges (gradient)





These two gradient images are then used to calculate the Gradient Magnitude of the image, using the formula:

$$Gradient\ Magnitude = \sqrt{x^2 + y^2}$$

After calculating the Gradient Magnitude, the optimal threshold was computed using **Otsu's algorithm** to highlight the edges properly. The more the Threshold, the fewer details will be shown.

In LOG Operator:

LOG (Laplacian of Gaussian) is a filter obtained by differentiating the Gaussian Operator, and convolving the image with the said filter.

Formula used for the filter:

$$LOG = \left(\frac{i^2 + j^2 - 2\sigma^2}{\sigma^4}\right) e^{\frac{-(i^2+j^2)}{2\sigma^2}}$$

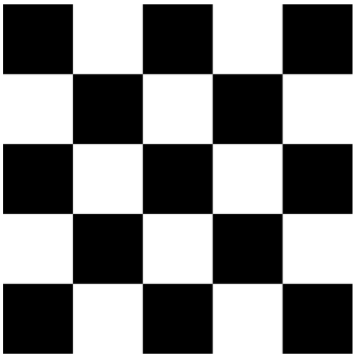
The resultant convolved image with this filter will have an edge if the product of the adjacent pixel is negative.

Result:

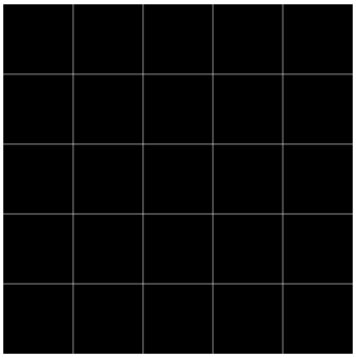
Digital Image Processing
Assignment 4

Tejash More
M.Tech SP

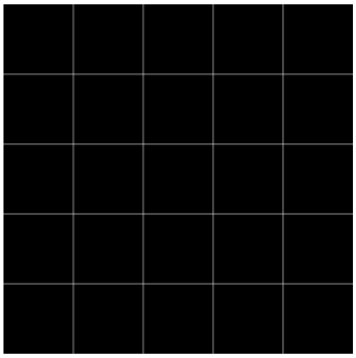
Original Image - CheckBoard



Edge Map (Sobel)

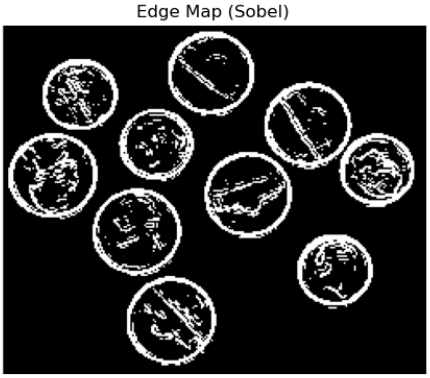


Edge Map (LOG)

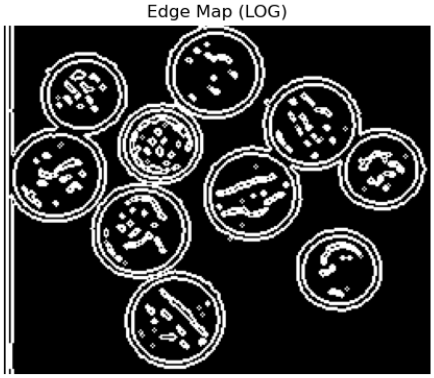


Original Image - Coins

Edge Map (Sobel)



Edge Map (LOG)



Original Image - IISc

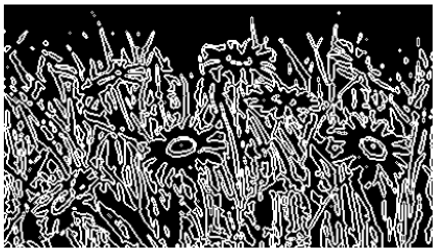
Edge Map (Sobel)

Edge Map (LOG)

Original Image - Flowers

Edge Map (Sobel)

Edge Map (LOG)



Observation

Edges processed with the Sobel Operator are thick, while those processed with the LOG operator are thin and more detailed. This is because the Sobel operator calculates the 1st derivative of the image, resulting in thick edges near the maxima, depending on the global threshold. In contrast, the LOG operator calculates the exact maxima using the 2nd derivative of the image.

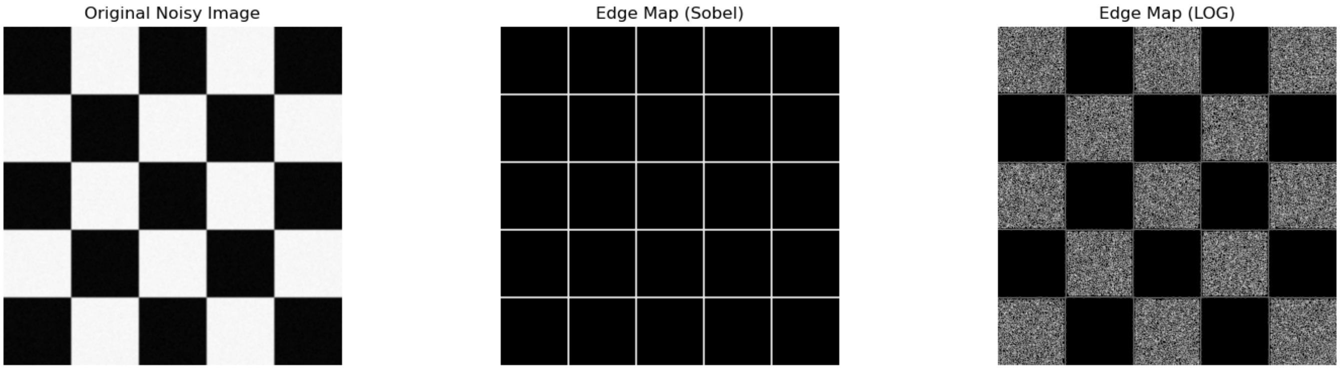
Digital Image Processing
Assignment 4

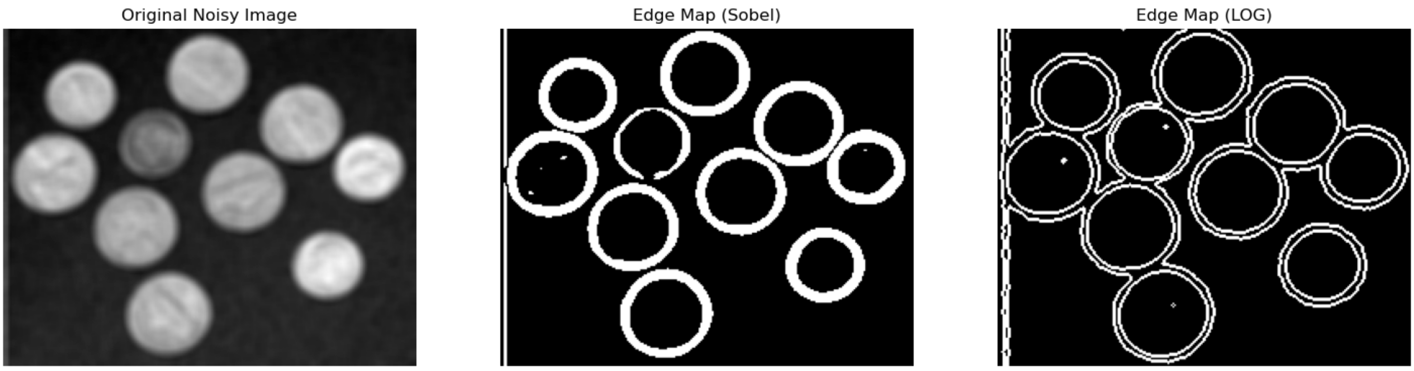
Tejash More
M.Tech SP

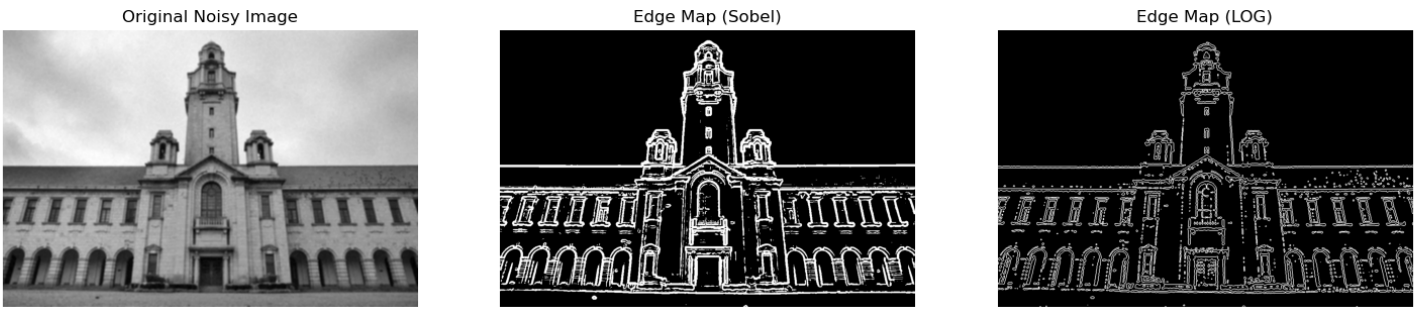
2.b) Add Gaussian Noise, then smooth the image before detecting the Edges

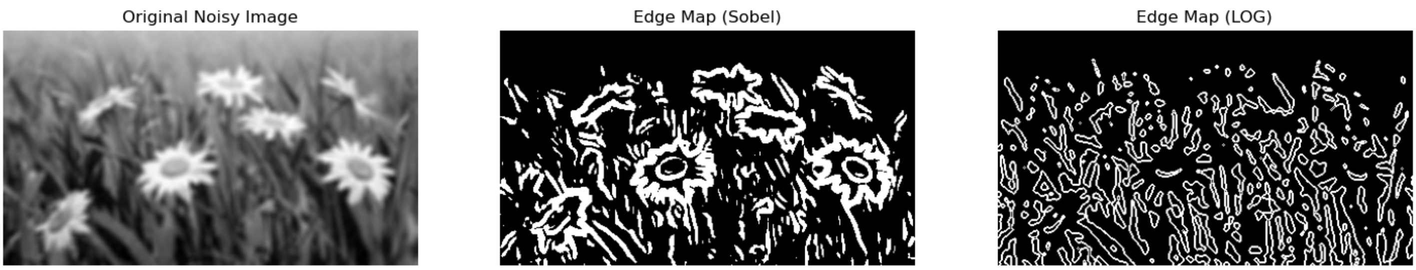
Gaussian noise with a mean of 0 and a standard deviation of 3 has been added to the image, and then it was smoothed using a Gaussian Filter of size 7x7 and a standard deviation of 3.

This Noisy Image is then used to detect its edges using the Sobel Operator and LOG Operator.









Digital Image Processing
Assignment 4

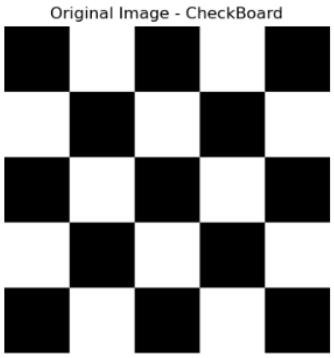
Tejash More
M.Tech SP

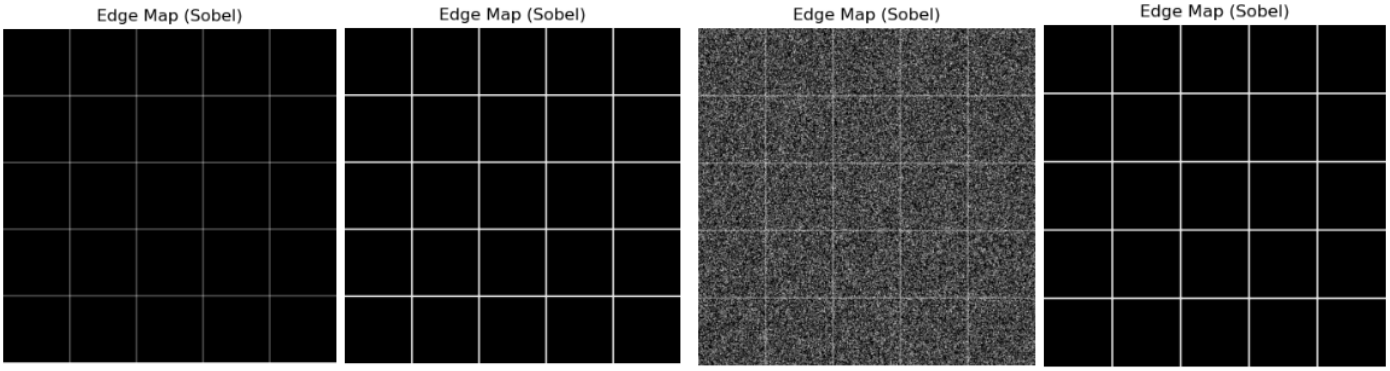
Observation:

Adding the noise introduced many high-frequency points in the image. After smoothing it, the Gaussian Filter removed some of the noise, but also softened the edges, as edges are high-frequency. Due to this, the edges detected are thicker and contain less information compared to the version without smoothening.

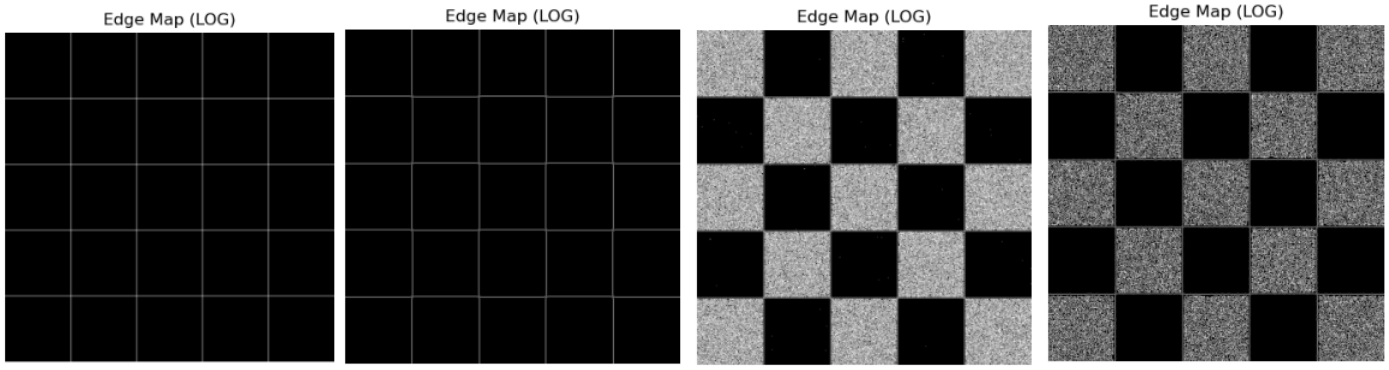
Still, the significant difference between Sobel and LOG remains unchanged: Sobel produces thick edges, while LOG produces thin edges.

Analysis:





Clean | Smooth | Noise, No Smooth | Noise and Smooth



Clean | Smooth | Noise, No Smooth | Noise and Smooth

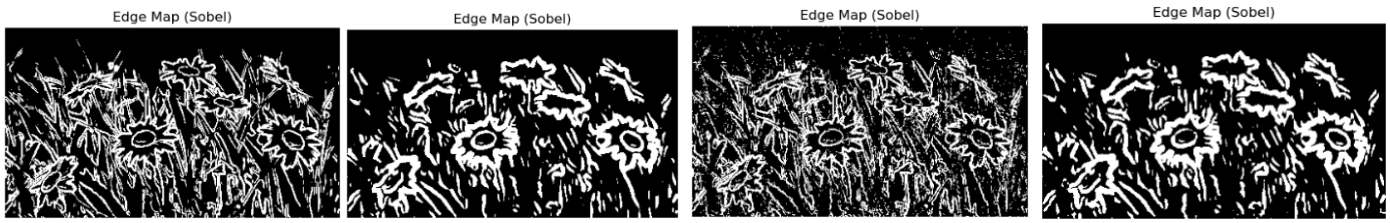
a. **Clean vs Noisy:**
   Both Sobel and LOG have captured noise in the Noisy image. Smoothing the noisy image works in the Sobel filter, as the smoothing can remove the noise to such an extent that the threshold in the Sobel operator does not classify the noisy pixel as an edge. In contrast, even after smoothing, reduced noise is still present in the LOG filter.
b. **Sobel vs LOG:**
   Sobel produces, in general, thicker edges than LOG because it considers the 1st derivative and the thresholding technique to detect the edges.

Digital Image Processing
Assignment 4

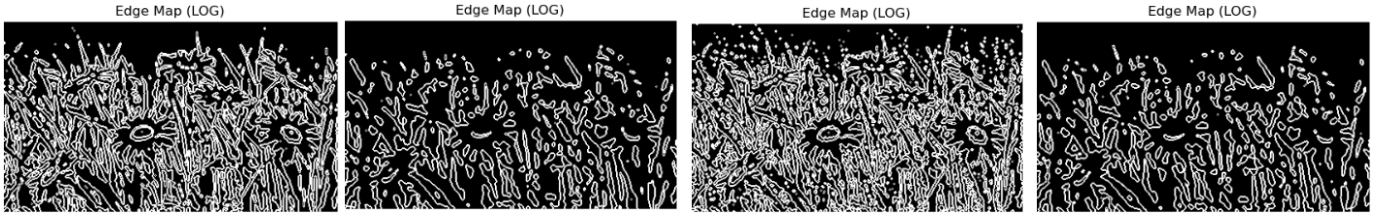
Tejash More
M.Tech SP

c. With and without Smoothing:

Smoothing the edge makes the edges a little smoother, which results in thicker detected edges compared to the clean image. In the case of LOG, the edges are not actually straight; this is because the smoothened image has thick edges, which prevent LOG from determining which pixel is the true edge.



Clean | Smooth | Noise, No Smooth | Noise and Smooth



Clean | Smooth | Noise, No Smooth | Noise and Smooth

Analysis:

a. Clean vs Noisy:

Noise has been captured with both the Sobel and the LOG operators without smoothing.

b. Sobel vs LOG:

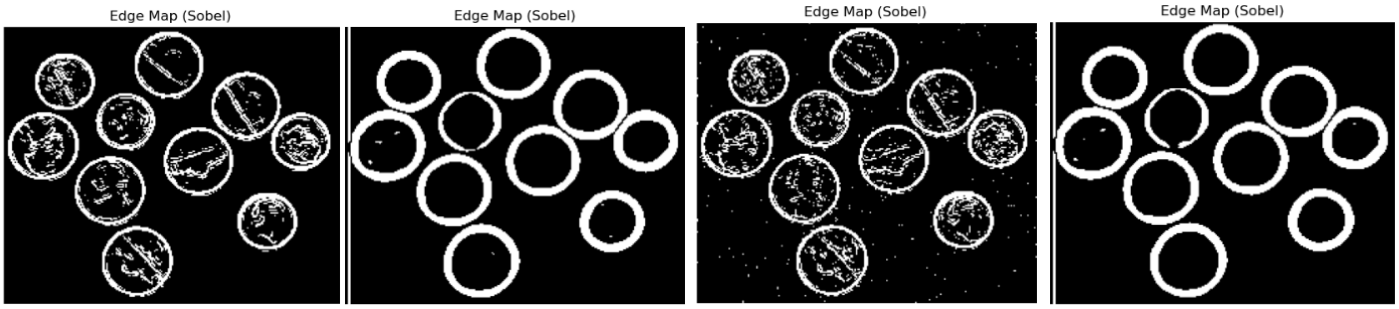
Sobel produces, in general, thicker edges than LOG because it considers the 1st derivative and the thresholding technique to detect the edges. Sobel is performing better for the smoothed image.

c. With and without Smoothing:

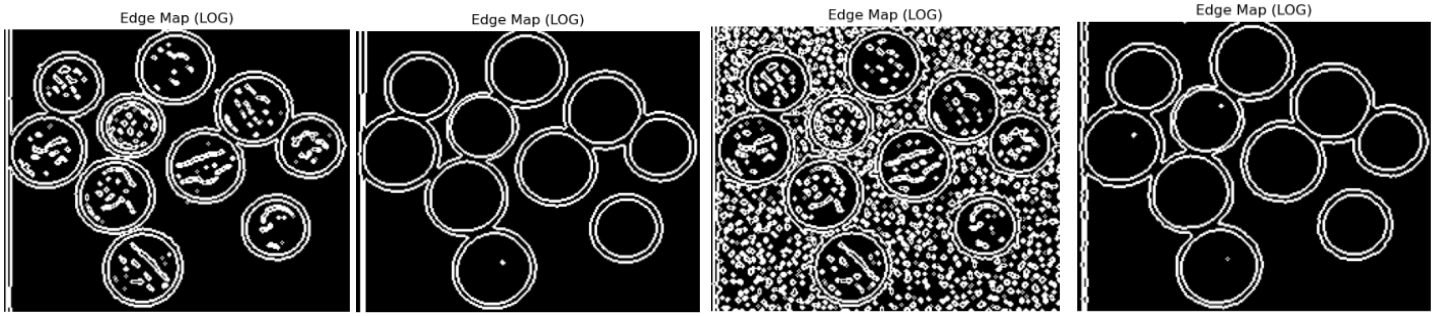
If the image contains too much detail, smoothing it may result in the loss of essential information; however, it also helps reduce noise to a great extent, so it's a trade-off.

Digital Image Processing
Assignment 4

Tejash More
M.Tech SP



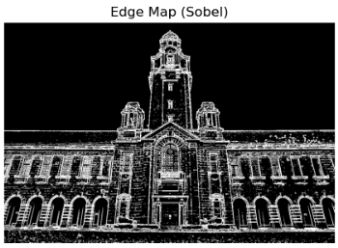
Clean | Smooth | Noise, No Smooth | Noise and Smooth



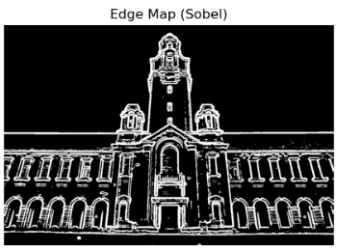
Clean | Smooth | Noise, No Smooth | Noise and Smooth

Digital Image Processing
Assignment 4

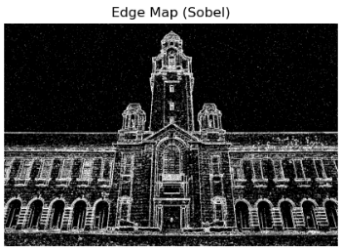
Tejash More
M.Tech SP



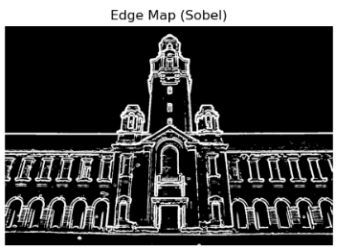
Clean



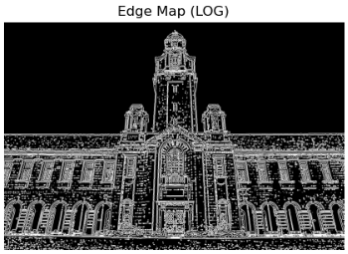
Smooth



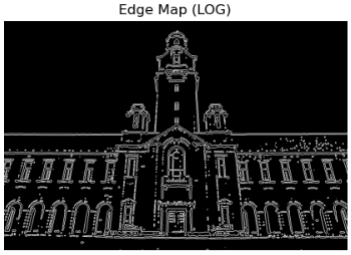
Noise, No Smooth



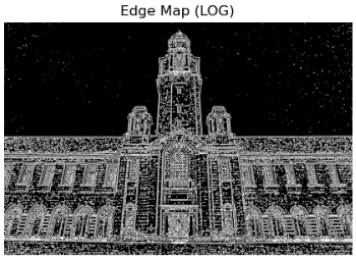
Noise and Smooth



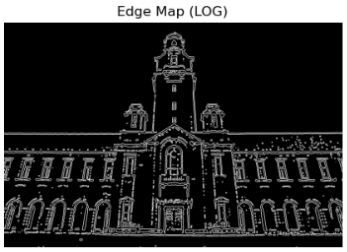
Clean



Smooth



Noise, No Smooth



Noise and Smooth

Conclusion:

Sobel and LOG are both effective edge detector filters; the difference lies in the width of the detected edges. Sobel gives thick edges, whereas LOG give thin edges. Since Noise corresponds to high frequency, both filters will detect noise as well. Smoothing helps reduce noise to some extent, but it may also remove some high-frequency, essential details, resulting in a loss of information. There is a trade-off here.

The optimal combination of these methods depends on the specific problem and dataset. With good optimized hyperparameters for Smoothness and Filters, noise can be effectively reduced without losing much information.