E9 241 Digital Image Processing
Assignment 04

Name: Bitupan Arandhara Sr. No.: 25910

Q1. Image Downsampling:

A. Vanilla Downsampling:

Results:

- The image was downsampled for the given factors 2, 4, and 5
- It is observed that on the building on the left, the horizontal lines of the windows look jagged and broken. And on the road,we can see new, wavy patterns that weren't in the original image.
- It was also observed that the image that was obtained for the downsampling factor 5 was a completely different image than the original one. It is observed that every fifth pixel in the original image has a different data that is not related to that image, and when we select only those pixels we get a new image.
- The resulting images are shown below:

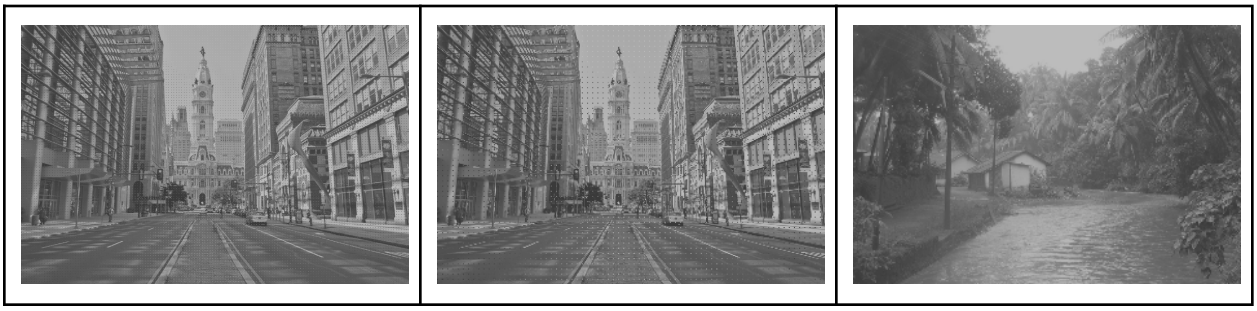


Figure 1: Downsampled images (factors: 2, 4, 5)

Inferences:

- When we are doing downsampling we are throwing away pixels, but those pixels contained information about fine details and sharp edges (high frequency information)

- The high-frequency information doesn't just vanish, it gets misinterpreted and shows up as new, false patterns in the smaller image. This is called aliasing

B. Low Pass Filtering then Downsampling:

Results:

- The original image was first filtered using a 5x5 spatial domain Gaussian Low Pass filter with $\sigma = 2$
- It is observed that aliasing artifacts seen in the vanilla downsampled images are now eliminated in the pre-filtered versions.
- It is also observed that applying the gaussian blur had removed the pre-planted pixels in the original image that lead to a new image in the previous step.

Figure 2: Downsampled images obtained after applying gaussian low pass filter (factors: 2, 4, 5)

Figure 3: Downsampled images obtained using library function (factors: 2, 4, 5)

Inferences:

- By blurring the image, the filter removes the high-frequency components (like the fine lines on the buildings and road) before we downsample.

- When we select every n-th pixel, there is no longer any high-frequency information to fold down into the lower frequency range. This prevents the creation of false patterns.
- A comparison with a library function confirms this observation. This function performs anti-aliasing internally before subsampling, producing a clean, non-aliased image very similar to our pre-filtered results.

C. Finding Optimal Window Size:

Results:

- An iterative search was performed to find the parameters that produced the lowest MSE.
- The optimal values were found to be:
  - Optimal Window Size: 13×13
  - Optimal Sigma (σ): 3.0

Q2. Edge Detection:

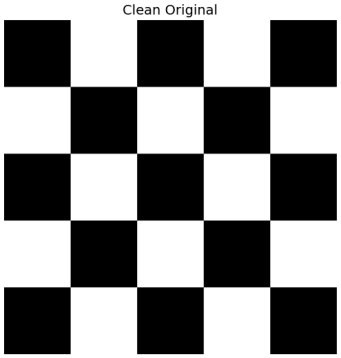
A. Edge Detection on clean image:

Results:

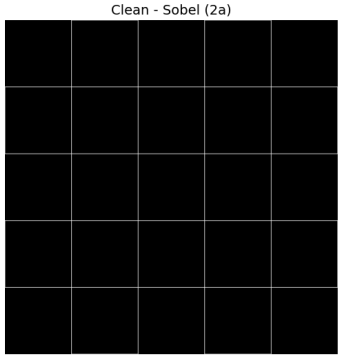
- Edge detection was performed on the four clean grayscale images. Two methods were used: a gradient-based approach (Sobel operator) and a Laplacian of Gaussian (LoG) operator.
- The Sobel operator produces thicker, more pronounced edges, while the LoG operator produces much thinner edges.
- The LoG operator is more sensitive to fine-grained, high-frequency detail. In the flowers image, it picks up almost every individual blade of grass, whereas Sobel only finds the main flower outlines. In the MainBuilding image, it highlights the fine brick texture that Sobel ignores.
- The LoG produces "ringing" artifacts for the Checkerboard image.
- The comparison is shown below:

Part 2a: Clean Image Edge Detection - Checkerboard

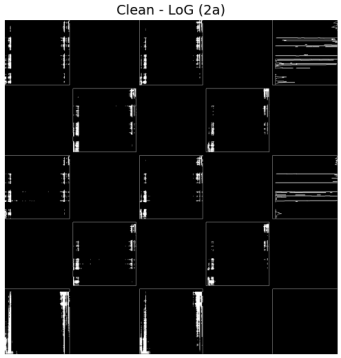
Clean Original



Clean - Sobel (2a)



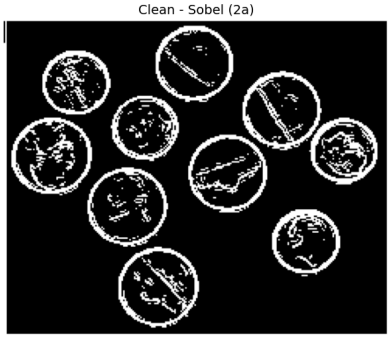
Clean - LoG (2a)



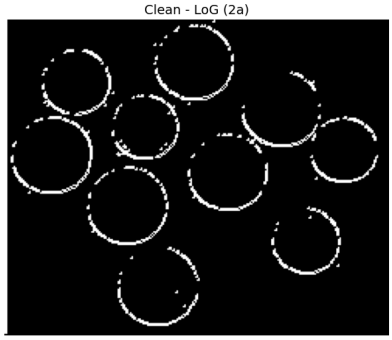
Part 2a: Clean Image Edge Detection - Coins

Clean Original

Clean - Sobel (2a)



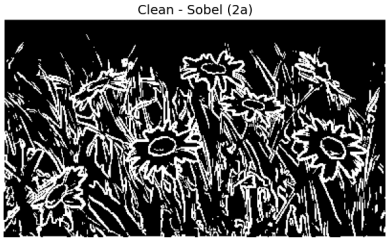
Clean - LoG (2a)



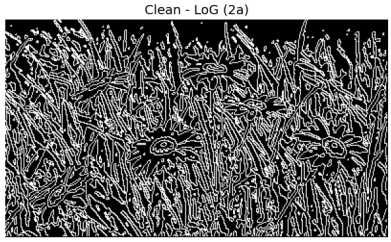
Part 2a: Clean Image Edge Detection - Flowers

Clean Original

Clean - Sobel (2a)



Clean - LoG (2a)



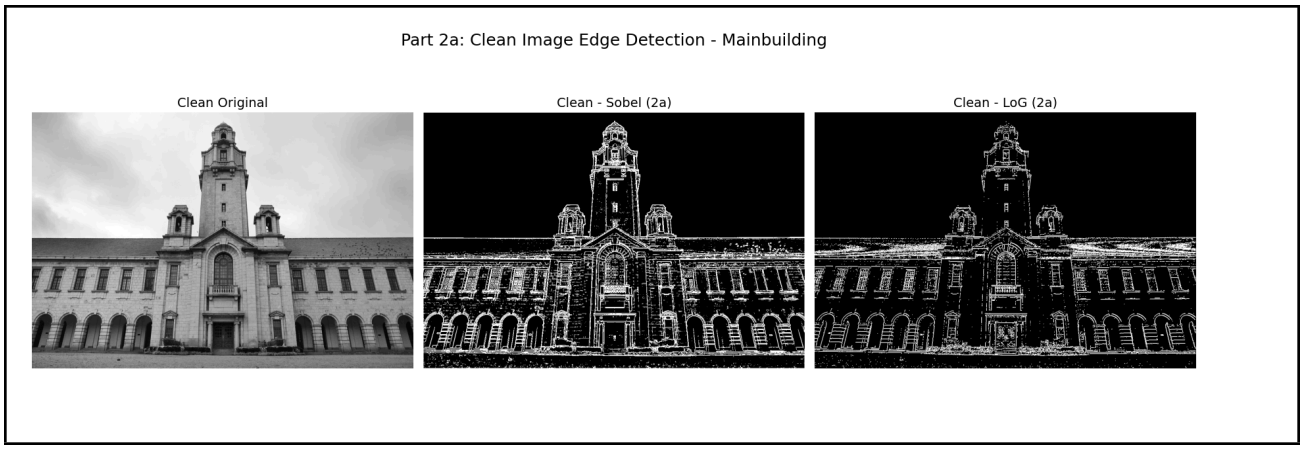


Figure 4: Edge Detection using Sobel operator and Laplacian of Gaussian operator.

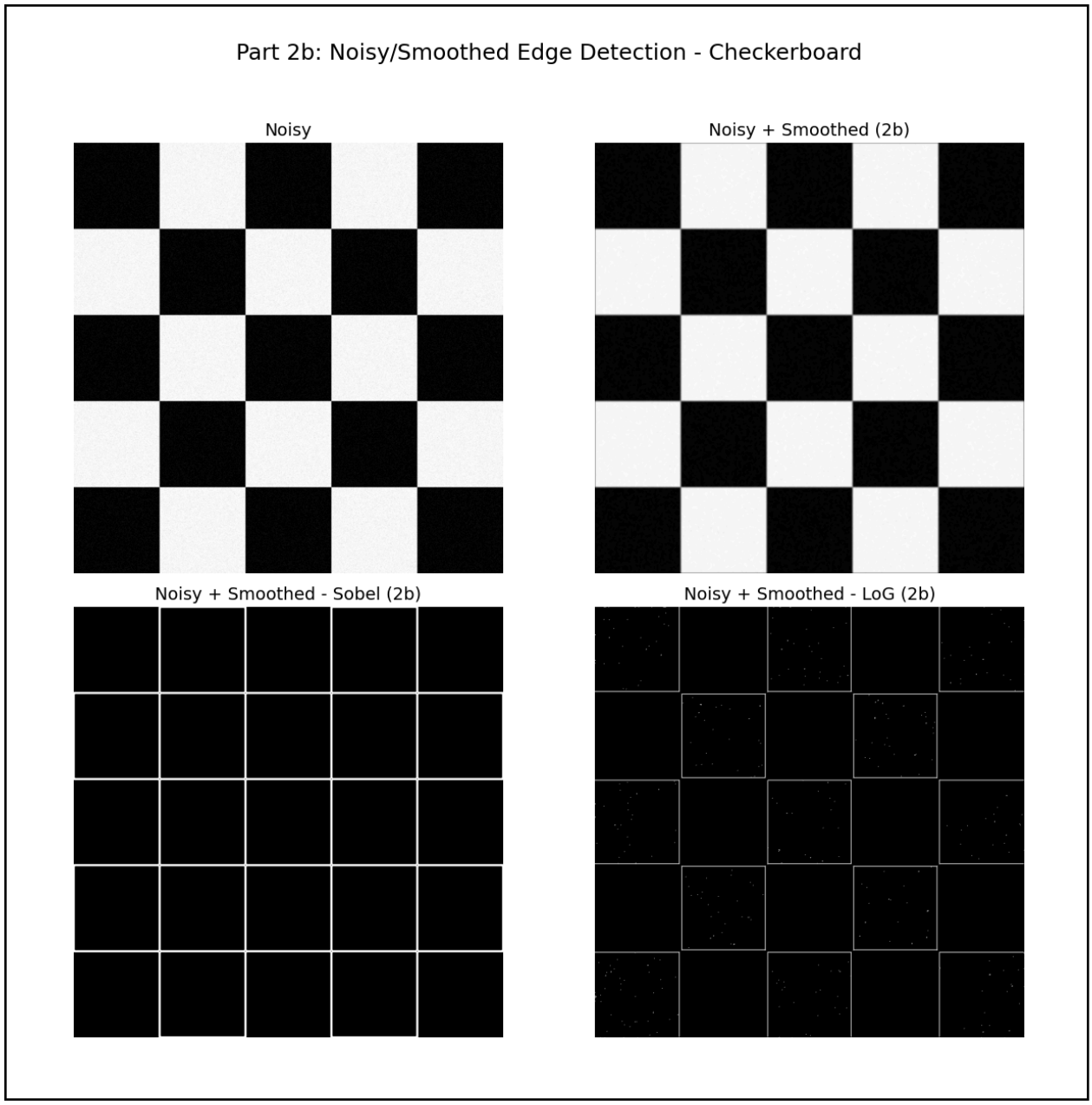
Inferences:

- The Sobel operator finds the gradient magnitude. It finds pixels where the intensity is changing rapidly. Since this change can occur over several pixels, the resulting edge map is thicker.
- The LoG operator finds the zero-crossing of the second derivative. This pinpoints the exact center of an edge, leading to its superior localization, and hence thinner edges. But, the second derivative is extremely sensitive to noise and fine texture.

B. Edge Detection on Noisy and then Gaussian Blurred Image:

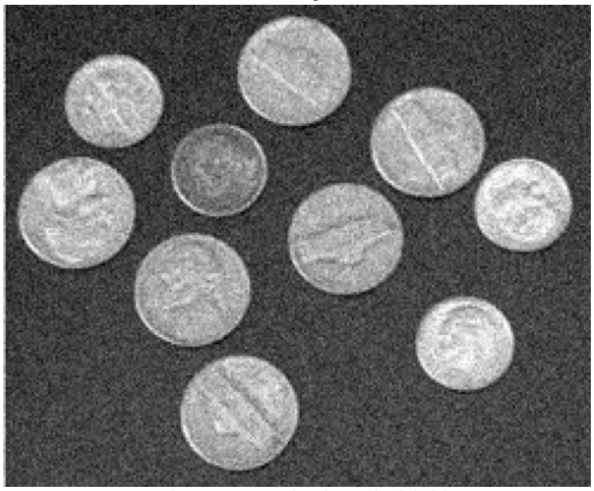
Results:

- **Gaussian Noise** was added to each of the four original images.
- A 7×7 **Gaussian smoothing filter** with **σ=3** was applied to each noisy image.
- Finally, the Sobel and LoG operators were applied to these smoothed images. The results for all four images are shown below:

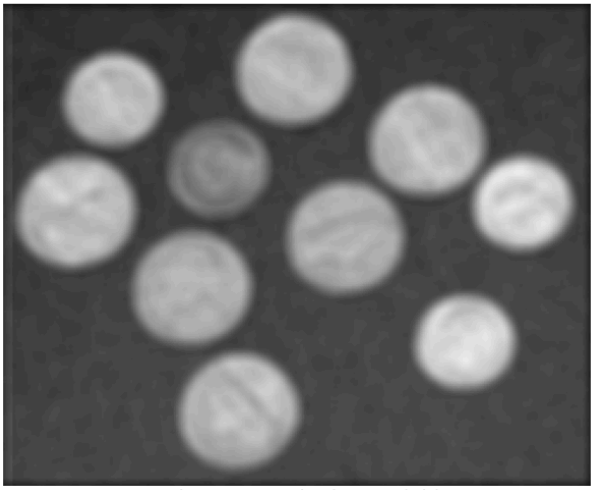

Part 2b: Noisy/Smoothed Edge Detection - Checkerboard
Noisy
Noisy + Smoothed (2b)
Noisy + Smoothed - Sobel (2b)
Noisy + Smoothed - LoG (2b)

Part 2b: Noisy/Smoothed Edge Detection - Coins

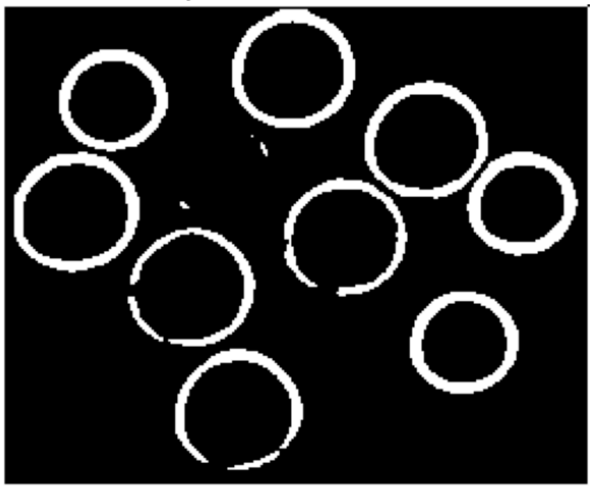
Noisy



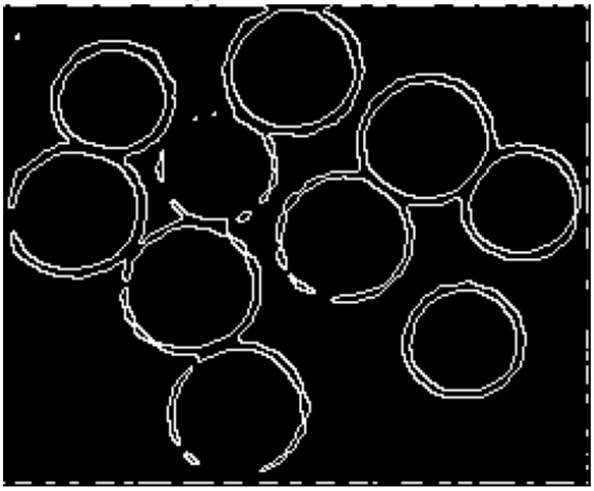
Noisy + Smoothed (2b)



Noisy + Smoothed - Sobel (2b)



Noisy + Smoothed - LoG (2b)



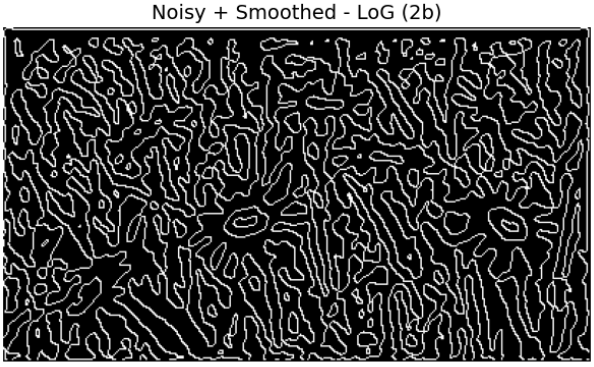
Part 2b: Noisy/Smoothed Edge Detection - Flowers

Noisy

Noisy + Smoothed (2b)

Noisy + Smoothed - Sobel (2b)

Noisy + Smoothed - LoG (2b)



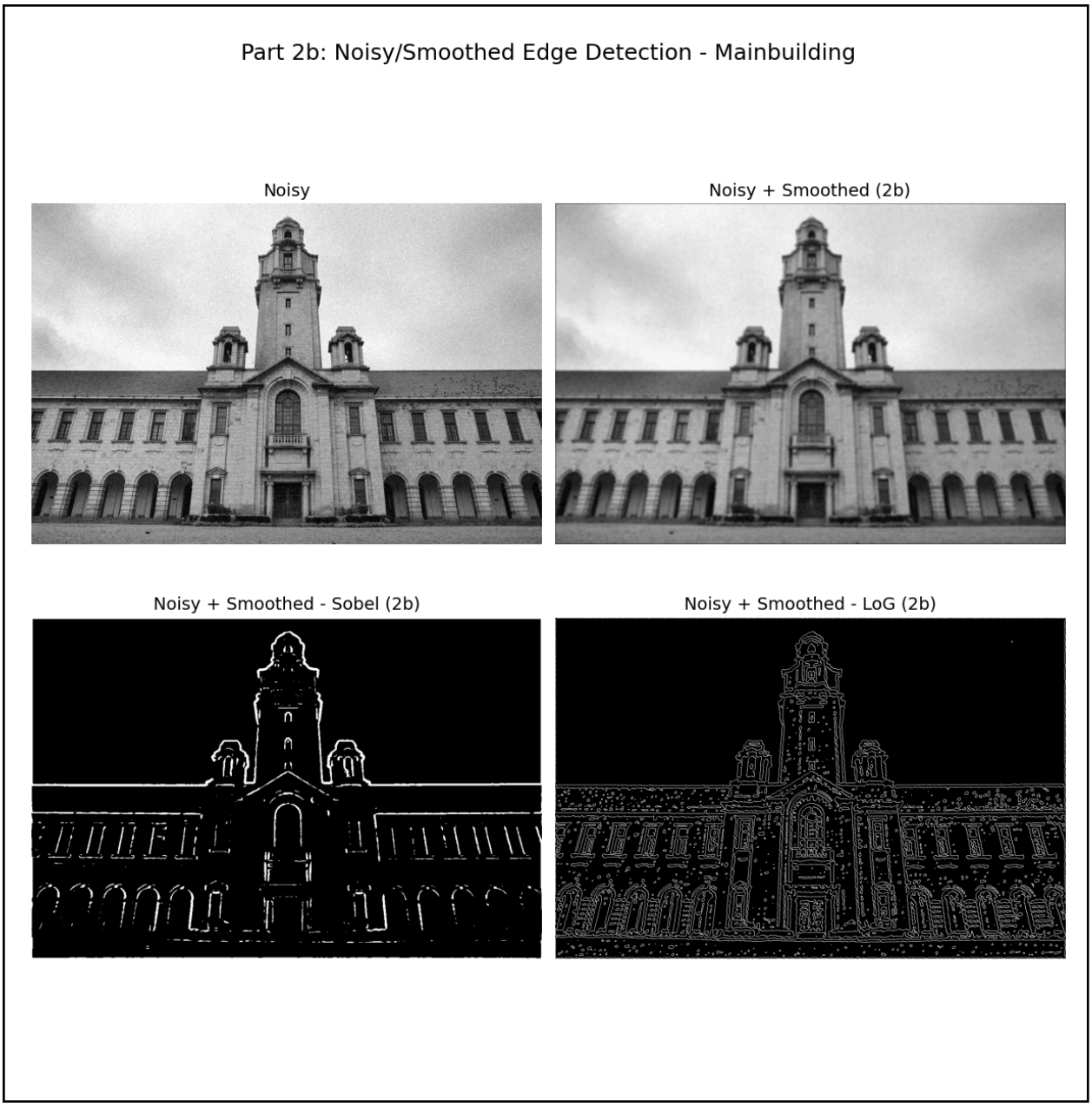


Figure 5: Noisy, Smoothed, Sobel on the Smoothed, and LoG on the Smoothed images

Inferences:

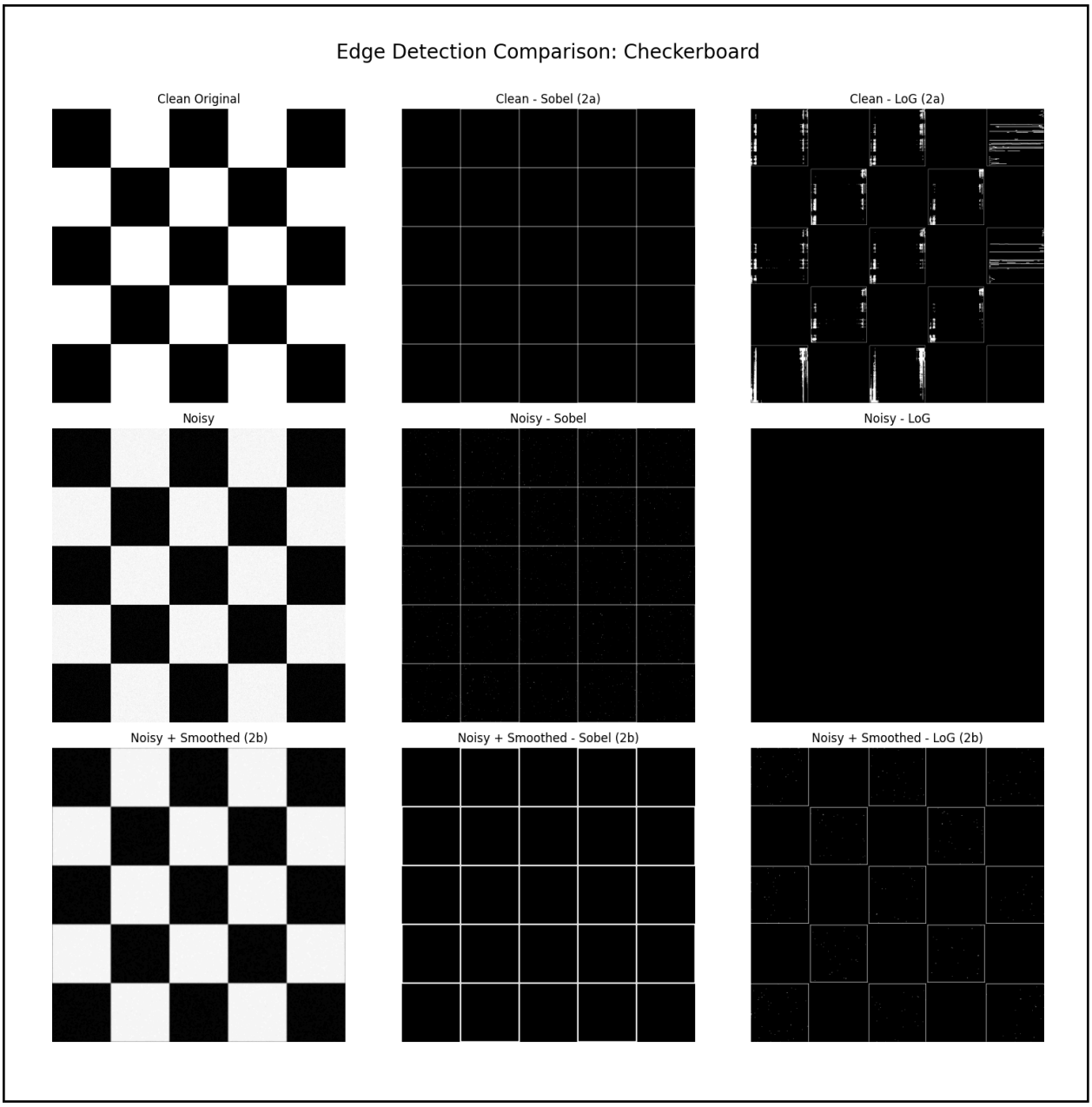
- When applied to the smoothed images, the Sobel operator successfully extracts the most prominent structural edges. Most of the fine-grained texture that was detected in 2a has been suppressed. This is because the Gaussian smoothing blurred these high-frequency details, along with the noise.
- The Log operator continues to produce thinner, more localized edges than Sobel. The LoG still detects more intricate patterns than Sobel, but these patterns now

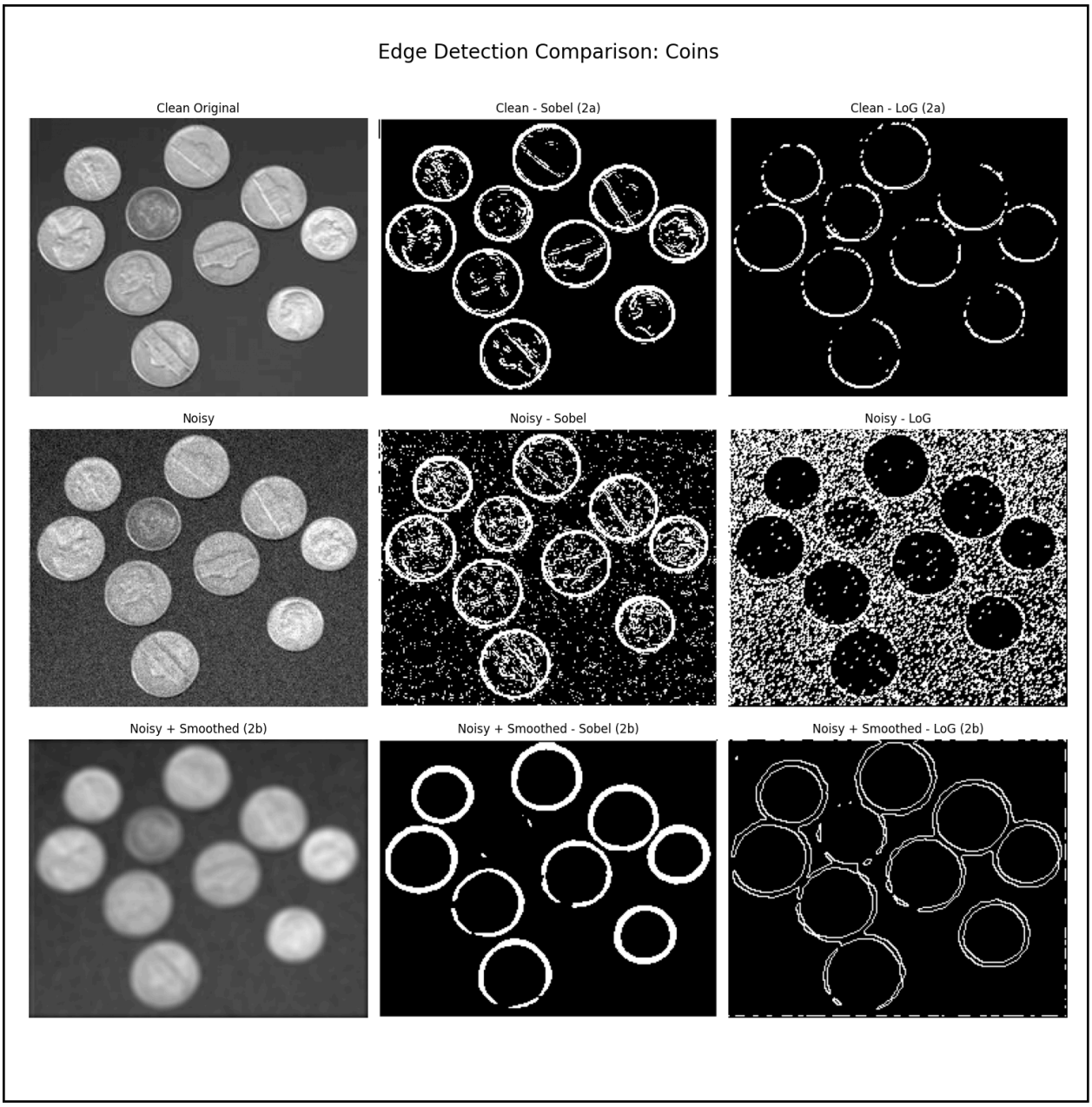
correspond to the larger, blurred shapes rather than the original, fine-grained texture.

C. Comparison:

Results:

- The 3x3 comparison plots for all four images provide a clear, qualitative basis for comparing the edge detection methods under different conditions.


Edge Detection Comparison: Checkerboard
Clean Original
Clean - Sobel (2a)
Clean - LoG (2a)
Noisy
Noisy - Sobel
Noisy - LoG
Noisy + Smoothed (2b)
Noisy + Smoothed - Sobel (2b)
Noisy + Smoothed - LoG (2b)


Edge Detection Comparison: Coins
Clean Original
Clean - Sobel (2a)
Clean - LoG (2a)
Noisy
Noisy - Sobel
Noisy - LoG
Noisy + Smoothed (2b)
Noisy + Smoothed - Sobel (2b)
Noisy + Smoothed - LoG (2b)

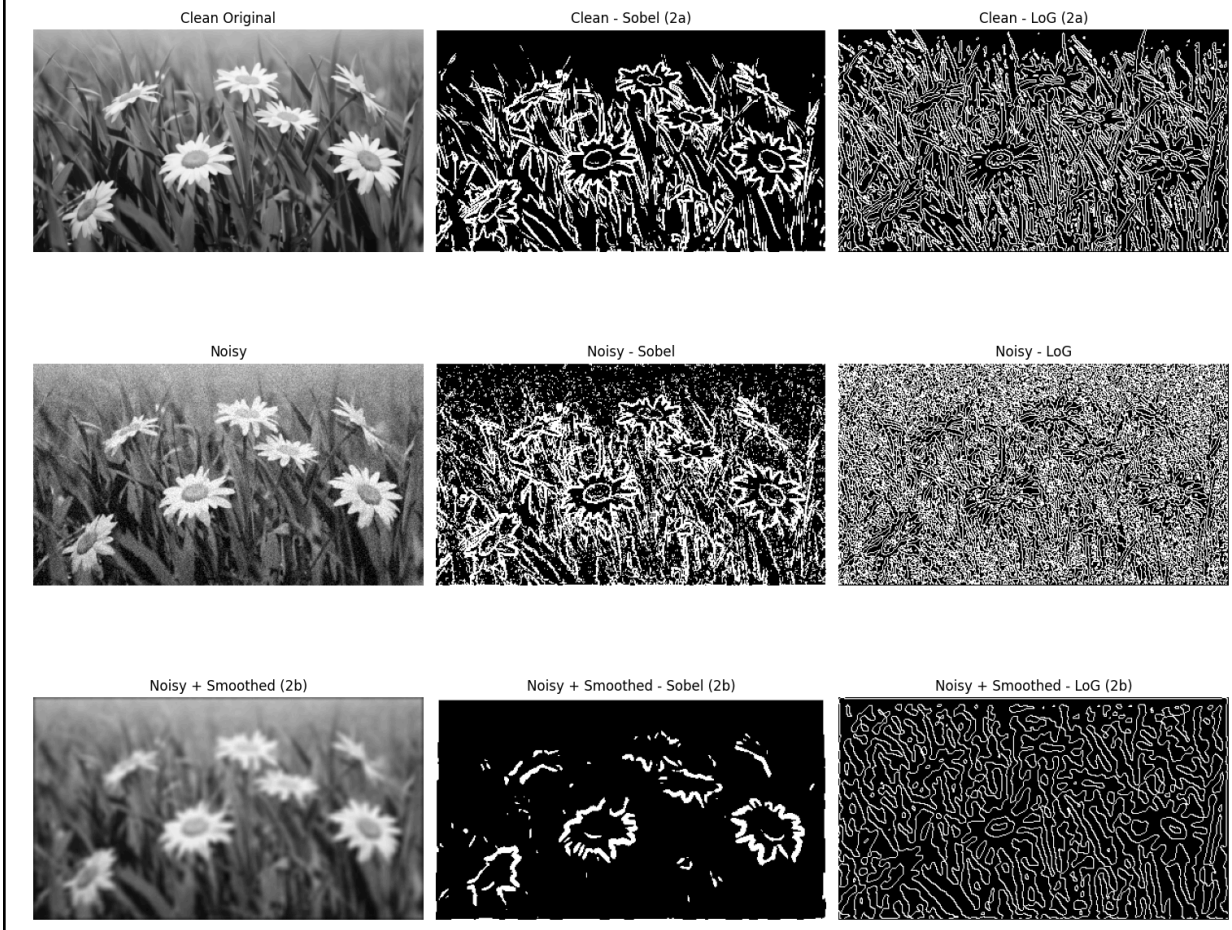

Edge Detection Comparison: Flowers
Clean Original
Clean - Sobel (2a)
Clean - LoG (2a)
Noisy
Noisy - Sobel
Noisy - LoG
Noisy + Smoothed (2b)
Noisy + Smoothed - Sobel (2b)
Noisy + Smoothed - LoG (2b)

Figure 6: The optimum Gaussian frequency response fit and its inverse.

Clean vs. Noisy Images:

- Comparing **Row 1 (Clean)** with **Row 2 (Noisy)** across all plots demonstrates the impact of noise on edge detectors.
- The plots show that the edge maps also find the "salt-and-pepper" as edges.
- The detectors are not able to distinguish between the signal (the true edges of the objects) and the noise. The second-derivative LoG operator is more sensitive to

high-frequency changes, which is why it fails even more dramatically than the Sobel operator on noisy images.

Gradient-based vs. LoG Edge Detectors:

- Comparing **Column 2 (Sobel)** with **Column 3 (LoG)** across all rows reveals the trade-offs between the two methods.
- In all clean and smoothed examples (Row 1 and Row 3), the LoG operator produces consistently thinner, more localized edges. The Sobel operator produces thicker, stronger edges.
- The LoG is more sensitive to fine-grained detail and texture than the Sobel.

With and without Gaussian Smoothing:

- Comparing **Row 2 (Noisy)** with **Row 3 (Noisy + Smoothed)** provides insight about the two approaches.
- The edge maps in Row 3 are clean, coherent, and successfully capture the main structures of the images whereas the edge maps in Row 2 are almost unusable.
- By applying the low-pass filter first, we remove the high-frequency noise. This leaves only the lower-frequency, structural edges in the blurred image. The edge detectors can now be applied to this cleaned image and will only find a better representation of the edges, as all the noise has been filtered out.