Computer Vision Assignment-1

Edge Detection and Smoothing Techniques

Saptarshi Halder, Roll No. 22283

Date of Submission: 12-09-2024

Abstract

This report discusses the implementation of filters, edge detection techniques, and noise handling strategies in image processing. The questions in this assignment have been solved sequentially by applying filters such as Sobel and Laplacian of Gaussian (LoG), thresholding, and Gaussian smoothing. Each section details the process and the impact of various kernel sizes and noise on the image edges.

1 Introduction

The objective of this assignment is to implement basic image processing techniques, particularly filters, edge detection, and noise reduction, from scratch using Python. Specialized libraries like OpenCV or PIL are avoided in favor of coding the algorithms from scratch, except for basic operations like image loading. The assignment consists of several tasks, each requiring analysis and visualization of results. The report follows the structure of the assignment, with each section focusing on the implementation of specific filters and their impact on the provided image.

2 Question 1: Edge Detection using 3x3 and 5x5 Filters

This question explores edge detection using filters with different kernel sizes, specifically 3x3 and 5x5. The same filters are used to analyze their impact on the edges of the provided image.

2.1 3x3 and 5x5 Filter Application

In this part, a single image is generated that combines the results from applying both the 3x3 and 5x5 filters. The filters are:

$$\text{3x3 Filter:} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{5x5 Filter:} \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -3 & 0 & 3 & 2 \\ -3 & -5 & 0 & 5 & 3 \\ -2 & -3 & 0 & 3 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

The results of both filters are visualized in Figure 1.

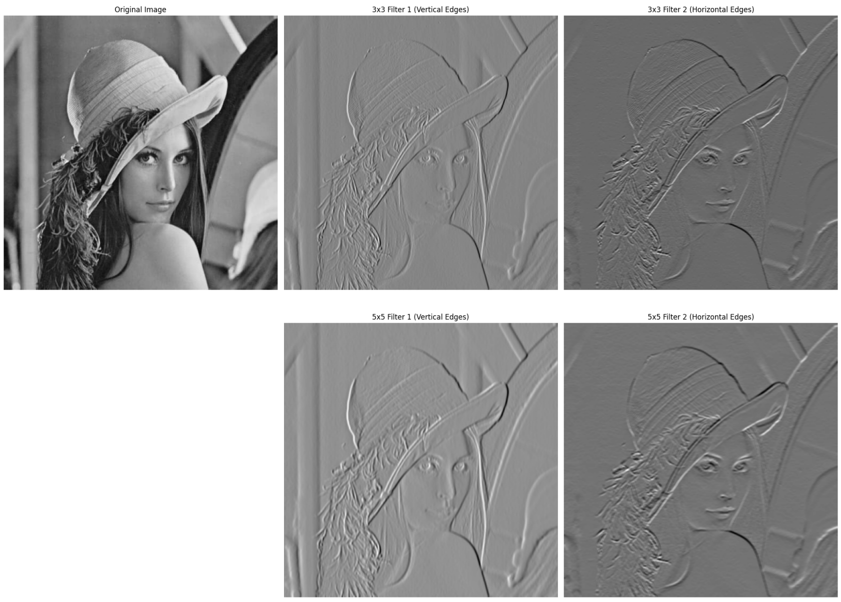

Figure 1: Combined output of edge detection using 3x3 and 5x5 filters.

2.2 Observation

The results indicate that the 5x5 filter produces thicker and more pronounced edges compared to the 3x3 filter. This is because the larger kernel captures a wider range of neighboring pixels, smoothing out fine details and emphasizing larger edges. Conversely, the 3x3 filter detects finer details with sharper but less prominent edges. By combining both results in a single image, we observe how kernel size affects edge detection, with the 5x5 filter being more suitable for images requiring more generalized edge outlines, and the 3x3 filter performing better on images where finer details are crucial.

3 Question 2: Sobel Edge Detection

3.1 Q2a: Sobel Filter Implementation and Edge Detection

In this section, we implement the Sobel edge detection filter to compute the gradient magnitude in both the x and y directions. The Sobel filters used are:

$$\text{Sobel X: } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{Sobel Y: } \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The results are combined to form the gradient magnitude image as shown in Figure 2.

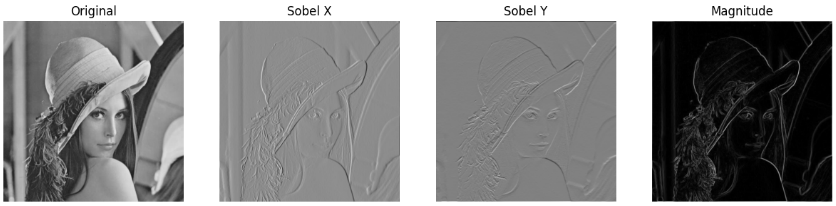

Figure 2: Sobel edge detection: Gradient magnitude image (combined x and y).

3.2 Q2b: Sobel Edge Detection with Varying Kernel Size

In this part, the kernel size of the Sobel filter is varied, and the results are analyzed.The figure below shows the impact of varying kernel size on edge detection.

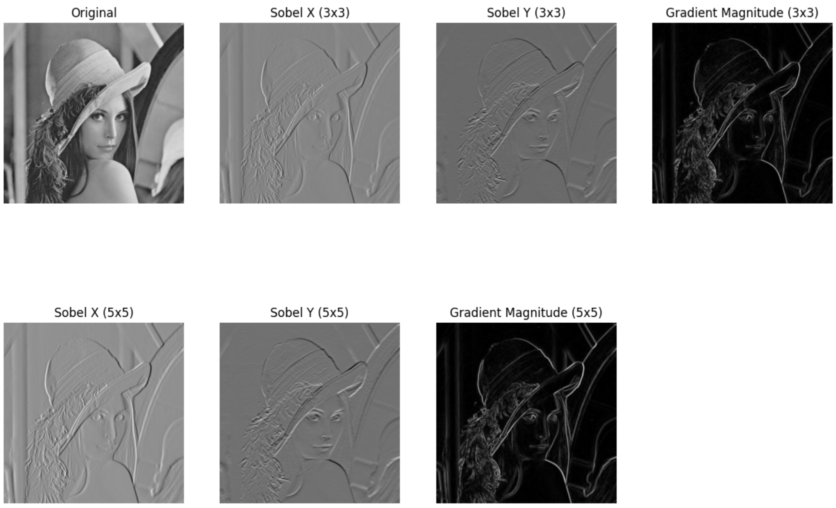

Figure 3: Sobel edge detection with varying kernel size.

We Note That The 5x5 Sobel Filters Yield Much Better Results And More Prominent Edges Are Detected Than 3x3 Filters

3.3 Q2c: Thresholding the Gradient Magnitude

To further enhance the detected edges, thresholding is applied. The gradient magnitude image is converted into a binary edge image by manually experimenting with different threshold values. The output is shown in Figure 4.

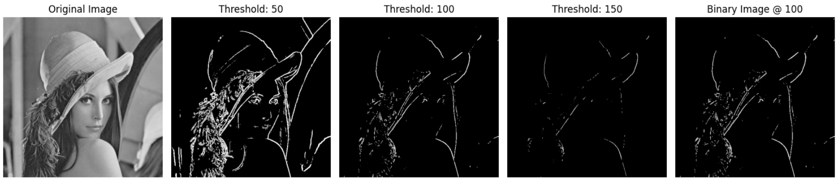

Figure 4: Binary edge image after thresholding the gradient magnitude.

3.4 Q2d: Impact of Noise on Sobel Edge Detection

In this question, synthetic noise is added to the image, and the impact of noise on edge detection is analyzed. Two outputs are generated: one showing the change in gradient magnitude due to noise and the other showing the visual appearance changes caused by noise.

Output 1: Change in Gradient Magnitude due to Noise

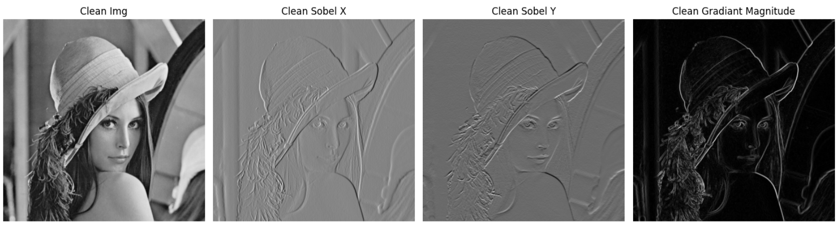

Figure 5: Clean Image Sobel And Gradient Magnitude

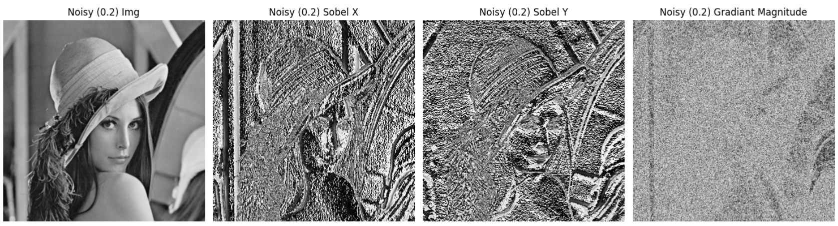

Figure 6: 0.2 noise and Image variance Image Sobel And Gradient Magnitude

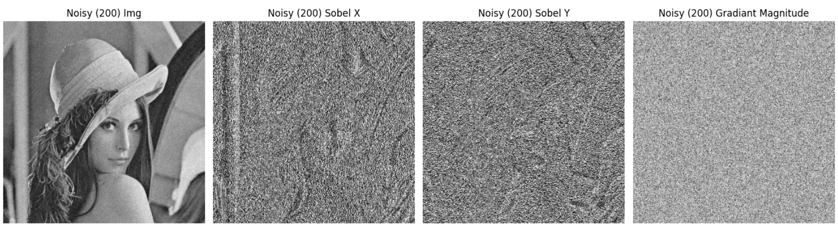

Figure 7: 200 noise and Image variance Image Sobel And Gradient Magnitude

We notice that even a small amount of noise is significant enough to reduce the edge detection of a image via sobel filtration, and the quality of gradiants gets reduced as well. Noises can affect images visually too, however a huge amount of noise is needed for humans to detect it with their eyes, such isnt the case for computors. In the next example we would be noticing , how noises can hamper audio-visual quality.

Output 2: Change in Visual Appearance due to Noise

Figure 8: Change in visual appearance due to noise.

We notice that with increase in Noise, the quality of image also starts distorting

4 Question 3: Laplacian of Gaussian (LoG) Edge Detection

4.1 Q3a: Gaussian Smoothing

In this section, Gaussian smoothing is applied to reduce noise before edge detection. The Gaussian filter used has a kernel size of 5x5 and sigma value of 1.0. The result is shown in Figure 9.

Figure 9: Smoothed image using Gaussian filter (5x5 kernel, sigma = 1.0).

4.2 Q3b: Laplacian Filter for Edge Detection

The Laplacian filter is then applied to the smoothed image to detect edges. Zero-crossings in the Laplacian result are used to detect the edges. The result is shown in Figure 10.

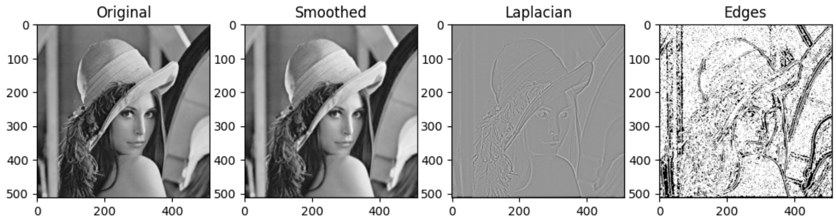

Figure 10: Edges detected using the Laplacian of Gaussian (LoG).

4.3 Q3c: Comparison of Smoothed and Non-smoothed Image Edges

In this final section, we apply the Laplacian to detect edges without smoothing the image, and the results are as follows:11.

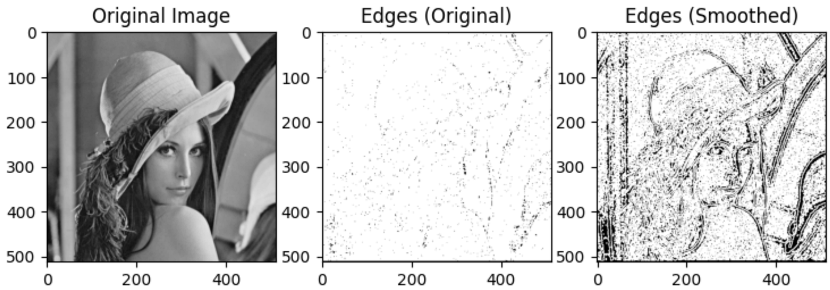

Figure 11: Comparison of edges: Smoothed image vs. non-smoothed image.

We find that using a Gaussian filter before applying the Laplacian results in more detailed edge detection, as the noise is significantly reduced. In contrast, if we attempt edge detection without the Gaussian filter, we lose more detail and finer edges because of the noise. For this process, we used a 5x5 Laplacian kernel where all the kernel values are set to 1, and the center value is -24.

5 Conclusion

In this assignment, various filtering techniques were implemented to detect edges and handle noise in images. The combination of different kernel sizes (3x3 and 5x5) demonstrates how the choice of filter affects edge detection results, with larger kernels producing more generalized edges and smaller kernels capturing finer details. The Sobel filter and its thresholding mechanism allow for flexible edge detection based on gradient magnitude, while the Laplacian of Gaussian method effectively reduces noise and detects edges with high accuracy in noisy environments.