Edge Detection for Computer Vision

Guilherme Barros, Ivo Lavacek, Pedro Lustosa, Gustavo Raia, André Silveira

***Abstract*— This paper presents an interactive application that compares several edge detection algorithms, aiming to illustrate how these methods behave under different conditions. Developed in Python using the Streamlit framework and OpenCV library, the system allows users to apply edge detectors with and without Gaussian smoothing, enabling a clear comparison of results. The application offers an accessible and educational way to understand the impact of preprocessing and parameter choices in edge detection tasks.**

***Keywords*— Edge detection, computer vision, image processing, Gaussian smoothing, algorithm comparison**

I. Introduction

Edge detection is a fundamental technique in image processing and computer vision, helping to identify boundaries and important features within images. Many tasks, such as object recognition and segmentation, rely on robust edge detection to isolate relevant shapes.

The performance of edge detectors is often affected by image noise. Preprocessing steps like Gaussian smoothing are commonly used to reduce noise and improve the detection of meaningful edges. Understanding how these detectors behave with and without smoothing is valuable for practical applications.

This project aims to provide a simple and interactive tool to compare different edge detection methods and observe how preprocessing influences their performance. Using Streamlit, users can test multiple detectors side by side, adjust Gaussian filter parameters, and manually compare results across methods. The tool is especially useful for educational purposes, allowing a better understanding of the strengths and limitations of each algorithm.

The remainder of the paper is organized as follows: Section II describes the methodology, Section III presents the results, Section IV concludes the work, and Section V lists the references.

II. Problem Statement and System Description

Edge detection plays a crucial role in image processing and computer vision tasks, serving as a foundation for object recognition, image segmentation, and scene understanding. However, selecting an appropriate edge detection method is not always straightforward, as different algorithms behave differently under varying image conditions, particularly in the presence of noise.

In many practical scenarios, images suffer from noise that can compromise the clarity of detected edges. Preprocessing techniques like Gaussian smoothing are commonly used to reduce such noise, improving the performance of edge detectors. However, the degree of improvement depends on both the smoothing parameters and the specific detection algorithm applied.

To address this challenge, we developed an interactive application that allows users to directly compare multiple edge detection algorithms under different preprocessing conditions. By providing visual comparisons between raw and smoothed images across various detectors, the tool helps users understand how noise reduction impacts the quality and accuracy of edge detection. The system is designed to be accessible and educational, enabling experimentation with parameters and algorithms in real time through a simple web interface.

G. Barros, I. Lavacek, P. Lustosa, G. Raia, and A. Silveira are with IBMEC, Rio de Janeiro, RJ, Brazil. This project was developed as part of a Computer Vision course in the Data Science and Artificial Intelligence program.

III. Methodology

The application is implemented in Python, using OpenCV for image processing and Streamlit for the web interface. The main goal is to allow users to easily compare edge detection techniques under different conditions, particularly focusing on the impact of Gaussian smoothing as a preprocessing step.

We applied a set of detectors to both original and smoothed images, making it possible to observe how noise reduction affects the clarity and continuity of detected edges. Users can adjust the Gaussian filter parameters (kernel size and standard deviation) and instantly visualize the results. The interface also includes a manual comparison section, where users select outputs from both smoothed and unsmoothed images to compare them side by side.

A. Edge Detection Techniques

To cover a variety of approaches, the application implements eight different edge detection methods, from basic gradient operators to more advanced and composite filters. Below is a brief description of each technique:

Prewitt: A simple gradient-based method using fixed 3×3 kernels to approximate intensity changes in horizontal and vertical directions. It tends to produce thicker edges and is sensitive to noise, but offers quick and effective edge detection for prominent features.

Prewitt Compass: An extension of the basic Prewitt method, applying multiple directional filters (eight in total) to capture edges in different orientations. This improves edge coverage but can also increase sensitivity to noise.

Sobel: Similar to Prewitt but with more weight given to central pixels, adding a smoothing effect that reduces noise sensitivity. Sobel generally produces cleaner and slightly thinner edges.

Sobel Compass: A simplified compass approach using horizontal and vertical Sobel responses. By taking the maximum response between these two directions, it enhances edge detection for both axes without applying full directional filtering.

Laplacian: A second-order derivative operator that highlights regions of rapid intensity change. Unlike gradient-based methods, Laplacian captures both positive and negative transitions, making it useful for emphasizing fine details.

Roberts: One of the earliest edge detection techniques, using small 2×2 kernels to approximate gradients along diagonal directions. It is very sensitive to noise but effective for detecting fine, sharp edges.

Canny: A multi-stage algorithm combining smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding to produce thin, continuous edges. Canny is known for its accuracy and robustness, making it one of the most popular choices for edge detection.

Difference of Gaussians (DoG): A method that subtracts two blurred versions of the image, each with different levels of Gaussian smoothing. It acts as a band-pass filter, highlighting edges at specific scales while suppressing both high-frequency noise and low-frequency variations.

B. System Workflow and Interface

The system workflow is designed for ease of use and flexibility. Users begin by selecting an image from predefined options or by uploading their own. Gaussian smoothing parameters can be adjusted through interactive sliders in the sidebar, with changes applied in real time.

Once parameters are set, the application applies all edge detection methods to both the original and smoothed versions of the image. Results are displayed in side-by-side grids, allowing users to compare how each algorithm responds to noise reduction.

Additionally, the interface includes a dedicated manual comparison feature. Users can select specific detectors from the original and smoothed sets to display them side by side at the bottom of the application. This targeted comparison helps to better understand the behavior of each algorithm under different preprocessing conditions.

IV. Results

To evaluate the implemented edge detection techniques, we tested the application using several images with varying levels of detail and texture. For each image, the detectors were applied to both the original and the smoothed versions, enabling direct visual comparison of the effects of Gaussian blur as a preprocessing step.

Overall, the results confirmed that smoothing helps reduce noise and can improve the clarity of edge detection. However, the impact of smoothing varied depending on the algorithm and the image characteristics:

- **Prewitt and Sobel:** Both gradient-based detectors showed cleaner edges after applying Gaussian blur. The smoothing reduced noise in flat regions, making the main contours stand out more clearly. Sobel, in particular, produced sharper edges, especially visible in images with prominent structures.
- **Compass methods (Prewitt Compass and Sobel Compass):** These methods benefitted from smoothing as noise reduction helped to avoid multiple overlapping responses from different directional filters. Edges became more distinct, especially in complex areas of the image.
- **Laplacian and Roberts:** These detectors, being more sensitive to noise, showed noticeable improvement after smoothing. Gaussian blur helped reduce false positives caused by noise, although very fine details were sometimes less emphasized.
- **Canny:** While Canny already includes internal smoothing, applying external Gaussian blur further reduced weak and isolated edges. The result was a cleaner, more continuous edge map. This effect is illustrated in Figure 3.
- **Difference of Gaussians (DoG):** Since DoG is based on differences between blurred images, the choice of preprocessing parameters significantly influenced the outcome. External smoothing balanced edge intensity and reduced noise, as seen in Figure 2.

Specific examples of these effects are shown below.

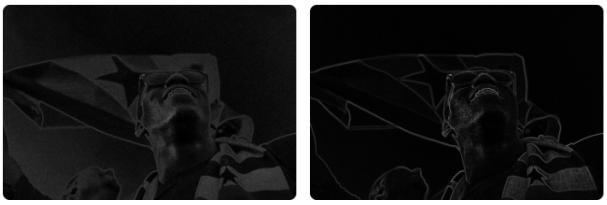

Fig. 1: Comparison of Sobel edge detection results without (left) and with (right) Gaussian smoothing. Parameters: kernel size = 9, standard deviation = 2.

Fig. 2: Comparison of Difference of Gaussians (DoG) edge detection results without (left) and with (right) Gaussian smoothing. Parameters: kernel size = 11, standard deviation = 2.

V. Discussion

The results highlight the importance of preprocessing when applying edge detection techniques, particularly in noisy images. Gaussian smoothing consistently improved the clarity of edges across all detectors, but the extent of improvement depended on the specific method and image characteristics.

For the Sobel operator (Figure 1), smoothing made the main contours of the subject more prominent, while reducing background noise. The difference was particularly noticeable

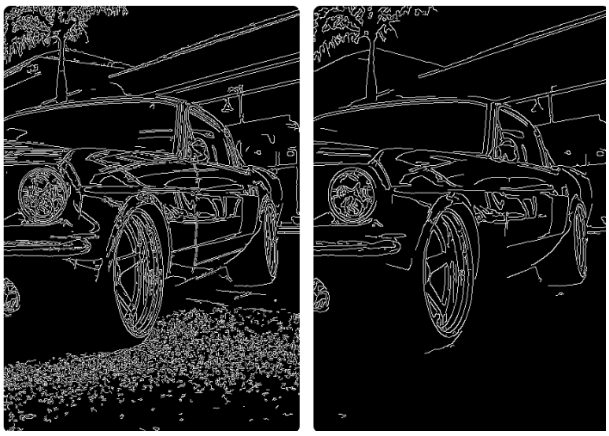

Fig. 3: Comparison of Canny edge detection results without (left) and with (right) Gaussian smoothing. Parameters: kernel size = 9, standard deviation = 1.

in regions with subtle intensity transitions, where noise previously interfered with edge definition.

In the case of the Difference of Gaussians (DoG) method (Figure 2), external smoothing complemented the internal multi-scale approach. Smoothing before applying DoG helped balance the response to mid-level frequencies and reduced spurious noise, leading to a clearer representation of object contours.

Canny edge detection (Figure 3) benefitted less dramatically from external smoothing, as the method already includes an internal Gaussian filter. However, additional smoothing improved the continuity of edges and reduced minor artifacts, especially in highly textured areas.

Across all methods, Gaussian blur helped suppress random noise and emphasized significant structural edges. However, excessive smoothing risks blurring fine details and reducing edge sharpness. Therefore, selecting appropriate filter parameters is essential for achieving a balance between noise reduction and detail preservation.

These observations reinforce the value of our application as an educational tool. By adjusting the smoothing parameters and directly comparing outputs, users can better understand the trade-offs involved in edge detection and preprocessing techniques.

VI. Conclusion

This project successfully developed an interactive application that allows users to compare various edge detection techniques in real time, with a focus on understanding the impact of Gaussian smoothing as a preprocessing step. The tool provides an accessible way to explore how different algorithms respond to noise and how smoothing can enhance edge clarity.

Through visual comparisons, we observed that most detectors benefited from preprocessing, with clearer and more continuous edges in smoothed images. The flexibility of the application, including parameter adjustments and manual comparison features, enables users to experiment and learn about the behavior of each method.

Given the limited timeframe of one week, the project focused on creating a functional and educational prototype. Future improvements could include adding more advanced detectors, exporting results automatically, or providing quantitative metrics to complement the visual analysis.

Overall, the application serves as a useful educational resource for understanding edge detection and the role of preprocessing in computer vision tasks.

References

[1] O. Sarwar, B. Rinner, and A. Cavallaro, "A privacy-preserving filter for oblique face images based on adaptive hopping gaussian mixtures," *IEEE Access*, vol. 7, pp. 142623–142639, 2019. DOI: 10.1109/ACCESS.2019.2944861.

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. Springer, 2010. DOI: 10.1007/978-1-84882-935-0.

[3] OpenCV Library. [Online]. Available: https://opencv.org/ [Accessed: Apr. 11, 2025].

[4] Streamlit Documentation. [Online]. Available: https://docs.streamlit.io/ [Accessed: Apr. 11, 2025].