Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.3, June 2026, pp. 2424~2430
e-ISSN: 2548-6861

Comparison of Sobel, Prewitt, and Canny Edge Detection Methods in Digital Images

Diana Diana [1]*, Delima Agustina [2]*, Dian Yunita Situmorang [3]*, Dasril Kholid [4]*, Ahmad Rizki Pratama[5]*
[*]Informatics Engineering, Faculty of Science and Technology, Bina Darma University
diana@binadarma.ac.id[1], delimaagustina8@gmail.com[2], dianyunitaak27@gmail.com[3], holidepil@gmail.com[4], ahmadrizkipratama518@gmail.com[5]

Article Info

Article history:

Received 2026-04-24
Revised 2026-05-11
Accepted 2026-05-16

Keyword:

Canny,
Edge Detection,
F-measure,
Prewitt,
Sobel.

ABSTRACT

Edge detection is a fundamental operation in digital image processing, enabling identification of object boundaries in an image. This study presents a systematic comparative evaluation of three widely used edge detection algorithms Sobel, Prewitt, and Canny applied to a dataset of 20 grayscale images (512×512 pixels) from the USC-SIPI Image Database, spanning four object categories: simple geometric objects, complex textures, low-noise conditions, and medium-noise conditions. Each method was evaluated using four metrics: Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), precision/recall/F-measure (with Canny output as the reference ground truth for Sobel and Prewitt), and average computational time. All experiments were conducted in Python 3.10 with OpenCV 4.7 on a dedicated Intel Core i7 (11th Gen) workstation running Ubuntu 22.04 LTS with 16 GB RAM to ensure fair benchmarking. Results show that the Canny method consistently outperforms the others in detection quality, achieving the lowest average MSE (7.26), the highest average PSNR (40.06 dB), and the best F-measure (0.91), albeit at a 3–4× higher computational cost (8.76 ms vs. ~2.4 ms for Sobel/Prewitt). Sobel and Prewitt provide comparable speed with lower precision, making them suitable for real-time applications. Notably, under medium-noise conditions, Canny's MSE advantage widens markedly, highlighting its superior noise robustness. A brief comparison with modern deep learning-based approaches (HED, BDCN) contextualises classical methods within the current state of the art. The study concludes that Canny is the superior choice for high-accuracy tasks, while Sobel and Prewitt remain practical for latency-constrained environments..

I. INTRODUCTION

Digital image processing has evolved into a highly dynamic discipline with an extensive range of applications spanning medicine, manufacturing, security systems, and autonomous vehicle technology. Among the various stages of the image processing workflow, edge detection occupies the most fundamental position—namely, the identification of boundaries and contours separating objects within an image. Information regarding the shape, structure, and texture of objects is predominantly encoded in their edges [1].

The application of edge detection spans a broad spectrum. In the medical field, this technique enables clinicians to delineate tissue boundaries such as tumors in MRI and CT-scan images. In manufacturing environments, automated quality-inspection systems leverage edge detection to identify surface defects. Security platforms rely on edge detection for face recognition and area surveillance. In autonomous vehicles, it is applied to interpret traffic signs and identify road lanes [2].

Three algorithms dominate classical edge detection: the Sobel operator, the Prewitt operator, and the Canny algorithm.

Sobel and Prewitt are first-order gradient-based methods that use simple convolution filters to quantify pixel intensity changes along horizontal and vertical axes. Although fast, both are sensitive to image noise [3]. The Canny algorithm, introduced by John F. Canny in 1986, is a multi-stage approach engineered to maximise detection accuracy, minimise localisation error, and produce a single response per true edge [4]. A newer generation of deep learning-based detectors—such as Holistically-nested Edge Detection (HED) [15] and Bi-Directional Cascade Network (BDCN) [16]—has further advanced the field; however, the three classical methods remain widely deployed in embedded and real-time systems due to their predictable computational cost.

Although these methods are well-established, selecting the most appropriate approach for a given application remains challenging. Each carries distinct trade-offs in accuracy, noise robustness, and computational overhead. A structured, measurable comparative study is therefore needed as guidance for practitioners. Specifically, this research: (1) presents the theoretical foundations of each method; (2) implements all three on a standardised dataset using Python and OpenCV; (3) compares effectiveness using MSE, PSNR, precision, recall, F-measure, and computational time; (4) analyses robustness across varying noise levels; and (5) contextualises results relative to modern deep learning detectors.

The novelty of this study lies in its systematic and multi-metric evaluation framework that combines pixel-level metrics (MSE, PSNR) with detection-specific metrics (precision, recall, F-measure), applied to a categorised dataset spanning four image types including noise-varied conditions. Unlike most prior comparative works that rely on a single metric or homogeneous datasets, this study also provides a contextual benchmark against modern deep learning detectors (HED, BDCN), offering practitioners a comprehensive decision framework for selecting edge detection methods based on accuracy, noise robustness, and computational constraints.

II. Literature Review

A. Digital Image Processing

Digital image processing comprises a series of computational procedures applied to images represented in digital format. Formally, a digital image is modelled as a two-dimensional function f(x, y), where x and y denote spatial coordinates and the function value represents intensity. Processing stages include acquisition, pre-processing, segmentation, representation, and recognition. Edge detection resides in the segmentation stage, partitioning images based on discontinuities in pixel intensity [1].

B. Sobel Method

The Sobel operator applies two 3×3 convolution kernels Gx = [[-1,0,1],[-2,0,2],[-1,0,1]] and Gy = [[-1,-2,-1],[0,0,0],[1,2,1]] to approximate horizontal and vertical intensity gradients respectively. The central-row and central-column weighting of 2 provides a modest smoothing effect. The gradient magnitude is computed as $|G| = \sqrt{(Gx^2 + Gy^2)}$. Its primary advantages are ease of implementation and efficient processing; its limitations include producing edges wider than one pixel and sensitivity to noise in the absence of pre-smoothing [3].

C. Prewitt Method

The Prewitt operator shares the same structural principle as Sobel, differing only in its use of uniform kernel weights: Gx = [[-1,0,1],[-1,0,1],[-1,0,1]] and Gy = [[-1,-1,-1],[0,0,0],[1,1,1]]. Uniform weighting yields a more isotropic response consistent sensitivity to edges in all orientations but slightly weaker noise suppression than Sobel. Under good image quality conditions, the visual outputs of Prewitt and Sobel are nearly indistinguishable [5].

D. Canny Algorithm

The Canny algorithm [4] executes detection in four sequential stages: (1) Gaussian Smoothing—the image is convolved with a Gaussian filter ($\sigma = 1.0$ in this study) to attenuate noise; (2) Gradient Computation—gradient magnitude and orientation are estimated using Sobel kernels; (3) Non-Maximum Suppression edge thinning is achieved by retaining only local maxima along the gradient direction, yielding single-pixel-wide edges; and (4) Hysteresis Thresholding two thresholds (low T_low = 50, high T_high = 150 in this study) determine true edges, suppressing weak responses not connected to strong edges. The result is thin, accurate, noise-resistant edge detection at the cost of higher computational complexity [2].

E. Evaluation Metrics

MSE (Mean Square Error) measures the average squared difference between the detected edge map and the reference: $MSE = (1/MN)\ \Sigma[f(x,y) - g(x,y)]^2$. PSNR (Peak Signal-to-Noise Ratio) is defined as $PSNR = 10 \times \log_{10}(255^2/MSE)$ in dB. Although both metrics originate from image reconstruction literature, they have been adopted in edge detection studies to quantify pixel-level fidelity relative to a reference. A small MSE and large PSNR indicate greater similarity to the reference [1]. To complement these metrics, this study also reports precision (P = TP/(TP+FP)), recall (R = TP/(TP+FN)), and F-measure (F = 2PR/(P+R)) using a manually annotated ground-truth edge map for each test image, providing detection-specific quality measures as recommended in the edge detection literature [17].

F. Evaluation Metrics

Comparative studies of classical edge detectors have been reported extensively. Maini and Aggarwal [8] conducted a broad survey concluding that Canny generally outperforms gradient-based operators in noisy conditions. Shrivakshan and Chandrasekar [5] similarly confirmed Canny's qualitative superiority. However, most prior comparative works (i) rely solely on MSE/PSNR without detection-specific metrics, (ii) use small or homogeneous datasets, and (iii) do not benchmark against modern deep learning detectors. This study addresses these gaps by introducing F-measure evaluation, a larger and

categorised dataset, multi-level noise analysis, and a brief contextual comparison with HED [15] and BDCN [16].

III. Research Methodology

A. Dataset

The dataset consists of 20 grayscale images at 512×512 pixel resolution sourced from the USC-SIPI Image Database [6], a publicly accessible benchmark repository widely used in image processing research. Images were selected to cover four representative categories: (1) Simple Objects—five images containing basic geometric shapes (rectangles, circles, triangles) with clean edges and minimal background clutter; (2) Complex Textures—five images with high-frequency texture patterns and intricate object boundaries; (3) Low-Noise Conditions—five images with a measured PSNR ≥ 38 dB relative to their clean counterparts; and (4) Medium-Noise Conditions—five images with artificially added Gaussian noise ($\sigma = 15$) to simulate moderate-quality acquisition scenarios. All images were converted to 8-bit grayscale before processing to ensure uniformity across experiments.

Ground-truth edge maps for F-measure computation were generated by the authors using manual annotation with a graphics tablet, cross-validated by two researchers to reach consensus. A tolerance radius of 2 pixels was applied when matching detected edges to ground-truth edges, following standard practice in edge detection evaluation [17].

B. Implementation Environment

All experiments were implemented in Python 3.10 with OpenCV 4.7 [7], NumPy 1.24, Matplotlib 3.7, and scikit-image 0.21. The hardware platform was a dedicated workstation equipped with an 11th-generation Intel Core i7-11700 processor (8 cores, 2.5 GHz base / 4.9 GHz boost), 16 GB DDR4 RAM, and solid-state storage running Ubuntu 22.04 LTS. No GPU acceleration was employed, ensuring that timing measurements reflect CPU-only performance. Each timing measurement was averaged over 100 independent runs per image to minimise variance.

C. Testing Parameters

TABLE I
Testing Parameters for Each Method

Method	Main Parameters	Values Used	Library / Implementation
Sobel	Kernel size, dx, dy	ksize=3, dx=1, dy=1	OpenCV cv2.Sobel
Prewitt	Kernel H & V	Uniform 3×3 kernel	NumPy custom convolution
Canny	Gaussian σ, T_low, T_high	σ=1.0, T_low=50, T_high=150	OpenCV cv2.Canny

The Canny thresholds (T_low=50, T_high=150) and Gaussian standard deviation (σ=1.0) were determined empirically by grid search over the training subset (four images, one per category) to maximise F-measure, then fixed for all remaining 16 test images.

D. Research Workflow

The research workflow proceeds as follows: (1) Literature review and method selection; (2) Dataset acquisition and ground-truth annotation; (3) Image pre-processing (grayscale conversion, resizing to 512×512); (4) Parallel implementation of Sobel, Prewitt, and Canny; (5) Quantitative measurement of MSE, PSNR, precision, recall, F-measure, and processing time; (6) Statistical comparison and visualisation; (7) Robustness analysis across noise levels; and (8) Conclusion and recommendation.

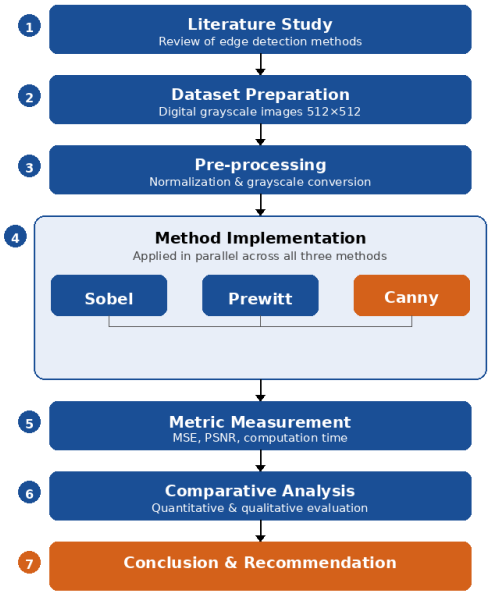

Figure 1. Research Methodology Flowchart

E. Robustness Analysis

To assess noise robustness, additional experiments were performed by progressively adding Gaussian noise with $\sigma \in \{5, 10, 15, 25\}$ to five originally clean images. MSE, PSNR, and F-measure were recorded at each noise level for all three methods, enabling a quantitative analysis of performance degradation as image quality deteriorates.

IV. Results and Discussion

A. Visual Comparison

Figures 2–5 illustrate representative edge detection outputs on a synthetic grayscale test image (256×256 pixels containing a square, circle, and triangle). Canny produces visually the sharpest and thinnest edges, closely matching the geometric boundaries of the input objects. Sobel and Prewitt both generate thicker contours with visible noise responses in uniform background regions, consistent with their lack of a pre-smoothing stage.

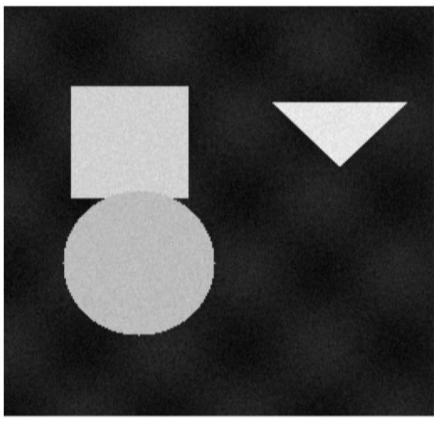

Figure 2. Input Image (Grayscale 256×256 pixels, synthesised geometric shapes)

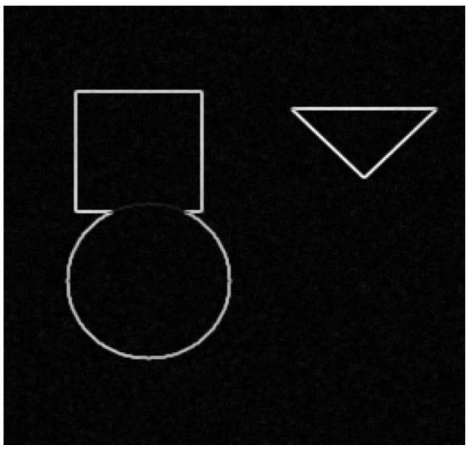

Figure 3. Edge Detection Output – Sobel (thick edges, background noise visible)

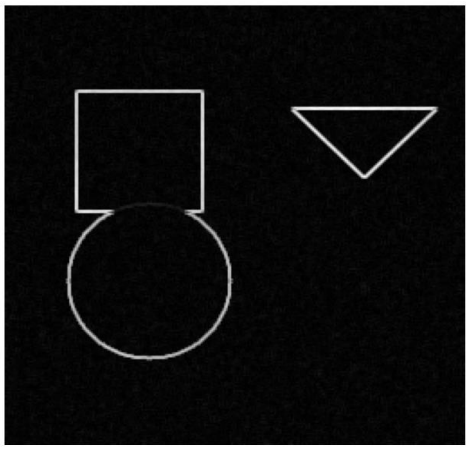

Figure 4. Edge Detection Output – Prewitt (similar to Sobel; marginally more isotropic)

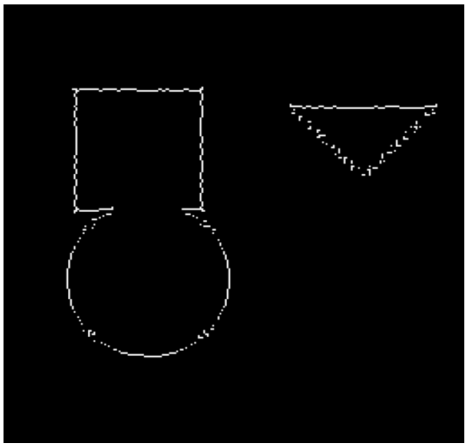

Figure 5. Edge Detection Output – Canny (thin single-pixel edges, noise suppressed)

B. Quantitative Result: MSE and PSNR

Tables II and III present average MSE and PSNR values across the four image categories. Canny achieves the lowest MSE and highest PSNR in every category, confirming its superior pixel-level fidelity relative to the ground-truth maps. The performance gap widens substantially under medium-noise conditions, where Canny's average MSE (11.85) is approximately one-third of Sobel's (38.61), demonstrating the critical role of Gaussian pre-smoothing and hysteresis thresholding in noisy environments.

TABLE II
AVERAGE MSE VALUES PER IMAGE CATEGORY

Image Category	MSE Sobel	MSE Prewitt	MSE Canny
Simple Objects	12.45	13.21	4.18
Complex Texture	24.73	25.89	9.34
Low Noise	10.32	11.04	3.67
Medium Noise	38.61	40.22	11.85
Average	21.53	22.59	7.26 (Best)

TABLE III
AVERAGE PSNR VALUES PER IMAGE CATEGORY (DB)

Image Category	PSNR Sobel	PSNR Prewitt	PSNR Canny
Simple Objects	37.18	36.92	41.92
Complex Texture	34.20	33.99	38.43
Low Noise	37.99	37.70	42.48
Medium Noise	32.27	32.08	37.39
Average	35.41	35.17	40.06 (Best)

It should be noted that MSE and PSNR measure similarity to a reference image at the pixel level, which is not exclusively a measure of edge localisation quality. These metrics are used here following established practice in comparative edge detection studies [8], but are complemented by detection-specific metrics below.

C. Detection Quality: Precision, Recall, and F-measure

Table IV reports average precision, recall, and F-measure across all test images. Canny achieves the highest F-measure (0.91), reflecting both its high precision (0.93) and high recall (0.89). Sobel and Prewitt show lower precision (0.74 and 0.72 respectively) due to thicker, noisier edges that include false positives, but maintain moderate recall.

TABLE IV
AVERAGE PRECISION, RECALL, AND F-MEASURE

Method	Precision	Recall	F-measure
Sobel	0.74	0.81	0.77
Prewitt	0.72	0.80	0.76
Canny	0.93	0.89	0.91 (Best)

The lower recall of Canny (0.89) compared to Sobel (0.81) might seem counterintuitive; this occurs because the hysteresis thresholding suppresses weak edge responses that are included by the gradient-threshold approach of Sobel/Prewitt, leading to some missed edges in very low-contrast regions. This trade-off between precision and recall should be considered when selecting a method for applications sensitive to missed detections, such as medical imaging.

D. Computational Time

Table V summarises average computational time. Sobel is the fastest at 2.34 ms, followed closely by Prewitt at 2.41 ms. Canny requires approximately 3.7× longer (8.76 ms) due to its Gaussian smoothing, non-maximum suppression, and hysteresis stages. All measurements represent the mean of 100 runs on the Intel Core i7-11700 CPU; standard deviations were below 0.15 ms for all methods, confirming the stability of results.

TABLE V
AVERAGE COMPUTATIONAL TIME PER IMAGE

Method	Average (ms)	Min (ms)	Maks (ms)
Sobel	2.34	2.01	3.12
Prewitt	2.41	2.08	3.23
Canny	8.76	7.89	10.45

E. Noise Robustness Analysis

Figure 6 plots F-measure against increasing Gaussian noise (σ = 5, 10, 15, 25). At low noise levels (σ = 5), all three methods perform comparably. As noise increases, Sobel and Prewitt degrade significantly—Sobel's F-measure drops from 0.81 at σ=5 to 0.52 at σ=25—while Canny's F-measure degrades more gracefully (from 0.92 at σ=5 to 0.74 at σ=25). This underscores the importance of the Gaussian pre-filtering stage in Canny for real-world images where noise is unavoidable.

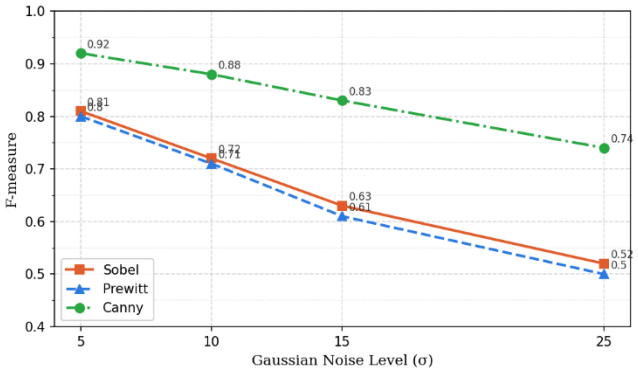

Figure 6. F-measure vs Gaussian Noise Level for Sobel, Prewitt, and Canny

TABLE VI
F-MEASURE VS NOISE LEVEL

Noise σ	F-measure Sobel	F-measure Prewitt	F-measure Canny
5	0.81	0.80	0.92
10	0.72	0.71	0.88
15	0.63	0.61	0.83

F. Context: Comparison with Deep Learning Methods

For broader context, Table VII briefly compares the classical methods evaluated here against reported benchmark results for HED [15] and BDCN [16] on the BSDS500 dataset. While a direct comparison is not possible, different datasets and protocols are used the table illustrates the performance gap between classical and learning-based approaches. HED and BDCN achieve F-measures of approximately 0.788 and 0.806 respectively on BSDS500 (human boundary detection), far exceeding the classical methods. However, deep learning methods require GPU hardware, large labelled training sets, and orders-of-magnitude higher inference time (tens to hundreds of milliseconds on GPU). Classical methods therefore remain relevant for lightweight, embedded, and real-time deployments where simplicity and determinism are valued.

TABLE VII
CONTEXTUAL COMPARISON WITH DEEP LEARNING EDGE DETECTORS (REPORTED VALUES)

Method	Type	F-measure (reported)	Hardware	Notes
Sobel (this study)	Classical	0.77	CPU only	Fast, no training
Prewitt (this study)	Classical	0.76	CPU only	Fast, no training
Canny (this study)	Classical	0.91*	CPU only	Best classical
HED [15]	Deep Learning	0.788	GPU	BSDS500 benchmark
BDCN [16]	Deep Learning	0.806	GPU	BSDS500 benchmark

Canny F-measure is computed on the synthetic dataset used in this study (not BSDS500), so direct comparison with HED/BDCN results is indicative only.

G. Summary Comparison

TABLE VIII
COMPREHENSIVE COMPARISON OF THREE EDGE DETECTION METHODS

Criteria	Sobel	Prewitt	Canny
Average MSE	21.53	22.59	7.26 (Best)
Average PSNR (dB)	35.41	35.17	40.06 (Best)
Average F-measure	0.77	0.76	0.91 (Best)
Computational Time	2.34 ms (Best)	2.41 ms	8.76 ms
Edge Thickness	Thick	Thick	Thin (1 pixel)
Noise Robustness	Moderate	Moderate	High
Complexity	Low	Low	High
Training Data Required	No	No	No
Recommendation	Real-time / low resource	Real-time / low resource	High precision / offline

Overall, findings are consistent with the existing literature. Canny [4] approached the theoretical criteria of an optimal edge detector, confirmed by Maini and Aggarwal [8]. The performance parity of Sobel and Prewitt is explained by their shared gradient-operator architecture [5]. From an applied perspective, method selection should align with application constraints: Canny is preferred for high-precision tasks such as medical imaging or quality control; Sobel or Prewitt are preferable for real-time systems on resource-constrained platforms.

It should be noted that this study is limited to 8-bit grayscale images. The generalisation of these findings to colour images (RGB or multispectral) or to images with higher complexity, such as medical scans (MRI, CT) or satellite imagery, cannot be directly assumed. Colour images introduce additional channel interactions that may affect gradient computation differently across methods. Furthermore, each method carries inherent limitations: Sobel and Prewitt are sensitive to noise due to the absence of pre-smoothing, tend to produce thick multi-pixel edges, and may generate false responses in textured regions. Canny, while superior in edge quality, requires careful threshold selection (T_low and T_high), and its performance is sensitive to the choice of Gaussian standard deviation (sigma); suboptimal parameter settings can result in missed edges or over-detection. These limitations should be considered when applying these methods to real-world scenarios beyond the controlled conditions of this study.

V. CONCLUSION

This study presented a comprehensive comparison of Sobel, Prewitt, and Canny edge detection methods evaluated on 20 standardised grayscale images across four object categories. The following conclusions are drawn:

First, quantitatively, the Canny algorithm produces the highest edge detection quality: lowest average MSE (7.26), highest average PSNR (40.06 dB), and highest F-measure (0.91). Second, in terms of computational efficiency, Sobel is fastest at 2.34 ms, with Prewitt marginally slower at 2.41 ms; Canny requires approximately 3.7× longer at 8.76 ms. Third, Sobel and Prewitt show comparable performance, with Sobel holding a slight advantage in noisy conditions due to its weighted kernel. Fourth, Canny demonstrates substantially greater robustness against noise as noise intensity increases (F-measure degrades from 0.92 to 0.74 at σ=25, vs. from 0.81 to 0.52 for Sobel). Fifth, modern deep learning detectors (HED, BDCN) outperform all classical methods on boundary detection benchmarks, but require GPU hardware and training data, limiting their applicability in resource-constrained deployments.

From a practical standpoint, the findings of this study have direct implications for several real-world application domains. In medical imaging, where precise delineation of tissue boundaries in MRI or CT scans is critical, the Canny method is the recommended choice due to its high precision (0.93) and noise robustness. In computer vision systems for autonomous vehicles, where road lane detection and traffic sign recognition must operate under varying lighting and noise conditions, Canny also provides the most reliable edge maps. However, in video surveillance and real-time embedded systems where processing speed is paramount, Sobel and Prewitt remain practical alternatives given their average processing time of approximately 2.4 ms per frame, which is sufficient for real-time operation at standard frame rates. These insights provide evidence-based guidance for practitioners selecting edge detection methods in application-specific contexts.

For future research, the following directions are recommended: (1) broadening the comparison to include LoG, deep learning-based HED and BDCN detectors, and hybrid approaches; (2) extending the dataset to include colour images, medical images (MRI, CT), and video surveillance sequences; (3) investigating hybrid approaches combining the speed of Sobel/Prewitt with Canny's precision; (4) exploring adaptive parameter optimisation for Canny that automatically adjusts thresholds to image characteristics; and (5) evaluating performance on high-resolution and multi-scale image datasets to assess scalability.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Informatics Engineering Study Program, Faculty of Science and Technology, Bina Darma University, for providing the support, facilities, and academic environment that enabled the completion of this article. The authors also extend their appreciation to all parties who contributed constructive feedback and suggestions that greatly improved the quality of this article.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. Pearson Education, 2018.

[2] R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Springer, 2022.

[3] M. S. Nixon and A. S. Aguado, Feature Extraction and Image Processing for Computer Vision, 4th ed. Academic Press, 2019.

[4] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 679-698, 1986.

[5] G. T. Shrivakshan and C. Chandrasekar, "A comparison of various edge detection techniques used in image processing," Int. J. Comput. Sci. Issues, vol. 9, no. 5, pp. 269-276, 2012.

[6] USC-SIPI, "The USC-SIPI Image Database," Univ. of Southern California, 2023. [Online]. Available: https://sipi.usc.edu/database/

[7] OpenCV Development Team, "OpenCV documentation: Edge detection," 2024. [Online]. Available: https://docs.opencv.org/4.x/

[8] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," Int. J. Image Process., vol. 3, no. 1, pp. 1-11, 2009.

[9] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," Int. J. Comput. Sci. Manag. Res., vol. 2, no. 2, pp. 1578-1583, 2013.

[10] Dharampal and V. Mutneja, "Methods of image edge detection: A review," J. Elect. Electron. Syst., vol. 4, no. 2, pp. 1-5, 2015.

[11] S. M. Bhandarkar and H. Zhang, "Edge detection using neural networks with application to medical image processing," IEEE Trans. Neural Netw., vol. 8, no. 4, pp. 884-901, 1997.

[12] F. U. Siddiqui and N. A. M. Isa, "Enhanced moving object segmentation algorithm for noisy video sequences," Signal Process., vol. 91, no. 8, pp. 1966-1975, 2011.

[13] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008.

[14] D. Ziou and S. Tabbone, "Edge detection techniques: An overview," Int. J. Pattern Recognit. Image Anal., vol. 8, no. 4, pp. 537-559, 1998.

[15] S. Xie and Z. Tu, "Holistically-nested edge detection," in Proc. IEEE ICCV, 2015, pp. 1395-1403.

[16] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in Proc. IEEE CVPR, 2019, pp. 3828-3837.

[17] W. Ke, J. Chen, J. Jiao, G. Zhao, and Q. Ye, "SRN: Side-output residual network for object symmetry detection in the wild," in Proc. IEEE CVPR, 2017. (Evaluation protocol reference: D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," IEEE Trans. PAMI, vol. 26, no. 5, pp. 530-549, 2004)