Quantitative and Qualitative Comparison Assessment of Edge Detection Algorithms for Robust Image Processing

Suzan J. Obaiys[1], Nur Amalina Binti Jamaludin[2(✉)], Rafida M. Elobaid[3], Nadia M. G. Al-Saidi[4], and Yeliz Karaca[5]

[1] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
suzan@um.edu.my

[2] Department of Mathematics, Centre for Foundation Defence Studies, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia
amalinajamaludin@upnm.edu.my

[3] School of Engineering, Applied Science and Technology, Canadian University Dubai, Dubai, United Arab Emirates
rafida@cud.ac.ae

[4] Department of Applied Sciences, University of Technology, 10066 Baghdad, Iraq
nadia.m.ghanim@uotechnology.edu.iq

[5] University of Massachusetts Chan Medical School and Massachusetts Institute of Technology, Worcester, USA
yeliz.karaca@ieee.org

Abstract. As a fundamental image processing tool, edge detection serves as a permit application in object recognition and image segmentation for computer vision analysis and enhancement. The work provides a comparative evaluation of the Laplacian Edge Detector (LED) and the Canny Edge Detector (CED) through mathematical analysis, operational assessment, and performance measurement, which allows scientists to select the most appropriate method for a specific application. Gaussian smoothing is incorporated in both models to minimize detection errors, resulting in Laplacian of Gaussian (LoG) processing for the Laplacian method, and noise removal for the Canny model, resulting in the LoG variant for LED and enhanced noise reduction within the CED pipeline. Experimental results demonstrate that while the basic LED is computationally fast, it suffers from high noise sensitivity and produces thick, less accurate edges. Overall, the results indicate that LoG improves LED performance considerably, but CED remains the most robust and accurate detector for applications requiring high precision, despite its increased processing time. Herein, we report a consistent set of metrics: PSNR, SSIM, edge F1-score, noise sensitivity, edge thickness, and processing time, across basic LED, Gaussian-smoothed LED, and LoG, and contrast them against CED's qualitative behaviour. Concluding that the LoG improves LED across all, PSNR 25.73 dB, SSIM 0.823, F1 0.789, quality metrics, while CED yields the thinnest, most continuous edges at a higher computational cost. The work characterizes

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16758, pp. 412–426, 2027.
https://doi.org/10.1007/978-3-032-30497-1_25

processing-time scaling with image size, including the associated memory profile, discusses CED's parameter sensitivity (thresholds, σ), and provides an empirical guideline for LoG with an optimal $\sigma \approx 1.4$ for the evaluated data, giving practical defaults and trade-off guidance.

Keywords: Image processing · Image detection · Data analysis · Canny edge detector · Laplacian edge detector · Laplacian operator

1 Introduction

Image processing, a branch of computer vision, uses digital images to enhance, analyse, and understand them using diverse techniques and algorithms [1]. The development of artificial intelligence (AI) is making image processing a vital technology and is driving its use in a diverse range of fields, such as autonomous systems, entertainment, security, and health. Image processing is a technique that can change and extract valuable information from photographs [2,3]. Edge detection (ED) is one of the basic processes in image processing that is used in a large number of computer vision tasks. Edges of an image encode information about object boundaries and structure, indicating the location and significant local changes in color or intensity. Edge detection, which is designed to detect the edges or boundaries in an image, is often essential for many applications such as object detection, image segmentation, and image enhancement [3,4].

A number of algorithms have been proposed for ED tasks. The Gradient-based techniques, e.g., the Sobel operator, calculates the first derivative over the intensity value of the image to get the direction and strength the brightness change [5,6]. The integration of quantum algorithms of ED for image processing applications, presents a promising avenue for overcoming these challenges and has driven transformative advancements, enabling capabilities far beyond the reach of traditional methodologies [7–10].As for the other ED, Canny edge detector (CED), is a popular choice. It uses a multi-stage algorithm that includes hysteresis thresholding, gradient computation, noise reduction, and non-maximum suppression. It can precisely detect edges and effectively remove false edges due to noise. The CED is still one of the most commonly usedmethods. It can be used for autonomous navigation and medical imaging due to its precision and stability. Novel techniques such as phase-stretch transform (PST) methods and ED methods based on deep learning have emerged to extract informative features with a lower noise susceptibility [11–14].

Recent advancements have led to the emergence of hybrid approaches for adaptive ED, based on machine learning and classical operators [15,16]. These methods leverage data-driven insights to optimise edge maps in real-time applications. However, the Laplacian edge detector (LED) and the CED are popular methods of ED, allowing an image processing technique to determine the discontinuities of the brightness within the image. Despite the Laplacian operator identifying abrupt changes in intensity, the CED includes many phases to identify edges more robustly. These methods are important for various applications such as object detection, segmentation, and image understanding [13–17].

This workfocuses on two famous edge detection techniques in image processing, LED and CED, which are important in many computer vision applications and practices. The work provides a unified mathematical walkthrough of LED to LoG and CED pipelines, along with a step-by-step derivation of the Laplacian operator from Taylor expansions to discrete kernels, motivate Gaussian smoothing, and formalizing the LoG construction. Furthermore, a complete Canny pipeline treatment of Gaussian denoising, Sobel-based gradient estimation, non-maximum suppression, double-thresholding, and hysteresis, with clear, pedagogical figures that link each operation to its mathematical role in edge localization are presented in this work. The Sobel kernels from directional derivatives in a 3×3 neighborhood, which shows how the x/y components arise and when the conventional scaling factors can be omitted in practice, clarifying common implementation choices. Furthermore, a comparative analysis comprising a set of metrics, PSNR, SSIM, edge F1-score, noise sensitivity, edge thickness, and processing time, across basic LED, Gaussian-smoothed LED, and LoG is presented, and these results are contrasted with the qualitative behaviour of CED.

2 Methodology

2.1 Laplacian of Gaussian (LoG) Model

One method for solving a second orderpartial differential equation is the LoG, which attempts to find the divergence of the gradient of a scalar function in 2D Cartesian coordinates [3]. The model can be expressed as

$$\text{Laplace}(f) = \nabla^2(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{1}$$

While in 3D, it is necessary to sum the partial derivatives in all three dimensions. The first derivative of the function describes intensity changes, while Fig. 1(a) shows the intensity graph versus the first derivative, which demonstrates how changes in intensity correspond to gradient images, which are essential for EDs. While the second derivative shown in Fig. 1(b) presents the peak of Fig. 1(a), which becomes a 0-slope of the curve compared to the intensity curve, aiding in edge localization [3].

To describe the undefined function f in Eq. (1), a numerical method-based Taylor's approximation described by the sum of a finite number of derivatives of f is proposed to solve the problem. From Taylor's, Eq. (2) defines the second derivative value of $\nabla^2(f)$in Eq. (1) as follows:

$$f''(x-h) + O(h^2) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} \tag{2}$$

and by taking the coefficient of Eq. (2), this yields the central difference approximation to the second derivative, defined by;

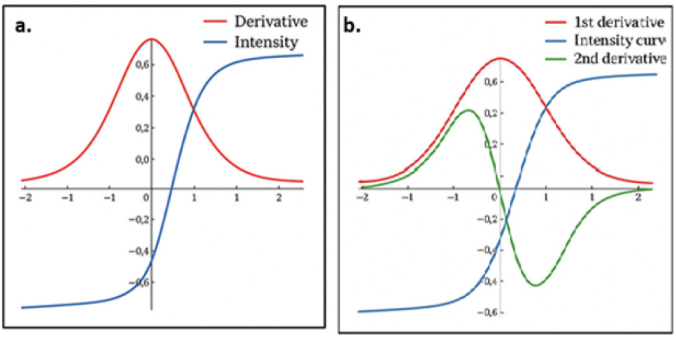

Fig. 1. a) The relationship between the original intensity curve and its first derivative; b) the visualization of the original intensity curve alongside its first and second derivatives.

$$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

The addition of the horizontal and vertical matrices of x, y, correspondingly, results in the below kernel matrix of the Laplacian model known for noise sensitivity.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Accordingly, the Gaussian smoothing is introduced to solve the noise sensitivity problem.

2.2 The Gaussian Smoothing

Gaussian smoothing or Gaussian Blur (GB) method proposed by Teague in 1980 [5], is one of the most popular techniques to reduce the image's noise and smoothen its details. The GB is a mathematical technique that blurs an image by averaging pixels, defined by [6];

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

where σ is the standard deviation, that controls the magnitude of the applied image blurring. The combination of GB with ∇^2 results in the Laplacian of Gaussian (LoG) model, which enables edge detection (ED) for the proposed images. The GB kernel is first convolved with the image, and the resulting GB-filtered image is then convolved with ∇^2 to perform image edge detection (ED) [3].

2.3 Canny Edge Detection (CED)

The Canny edge detector (CED) model designed by John F. Canny in 1986 [4,8,11,12] is an image ED operator that employs a multi-stage algorithm to identify a diverse set of edges. The model is composed of five steps: noise reduction, gradient calculation, non-maximum suppression, double threshold, and edge tracking by hysteresis. The initial stage involves image smoothing via GB implementation for the influence of noise on edges. The ED operators are used to compute image gradients, enabling the identification of both intensity and edge directions. Edges are characterized by alterations in the intensity of neighboring pixels. Furthermore, by applying filters that accentuate these intensity changes via convolving the image with both the x, y Sobel kernels, defined as K_x and K_y, respectively;

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4}$$

Then, the gradient intensity $|G|$ and edge direction defined by the slope of θ are calculated in Eq. (5), as follows;

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{5}$$

$$\theta(x, y) = \arctan\left(\frac{I_y}{I_x}\right) \tag{6}$$

where I_x and I_y represent the convolved image based K_x and K_y, respectively.

The non-maximum suppression algorithm is employed to reduce the thickness of detected edges by scanning each point on the gradient intensity matrix and identifying the pixels with the highest values in the edge directions, effectively retaining only those pixels of sharpest edges. The algorithm will set the pixel that is being processed to the black value 0, to thin out the white edges. While, it will keep the value of the processed pixel as it is, if there is a more intense pixel in the same direction. The double thresholding technique categorizes pixels into three distinct types: strong, weak, and non-relevant. The high threshold is applied to identify pixels with strong intensity, which is above the high threshold, while the low threshold is used to identify non-relevant pixels that are below the low threshold. Pixels with intensities falling between these two thresholds are classified as weak. Subsequently, the hysteresis mechanism is employed in the subsequent step to differentiate the weak pixels that can be considered strong to those that are still considered non-relevant, see Fig. 2 (a) below, where Strong, weak, and irrelevant edge pixels are classified to refine edge selection during hysteresis tracking. An example to Fig. 2(a), let's consider the current edge direction as horizontal, specifically from left to right, indicated by the orange dotted line. The processing of the pixel at coordinates (i, j), and the neighboring pixels in the same direction are highlighted in blue, namely (i, j-1) and (i, j+1), respectively. If either of these two pixels are of higher intensity value compared to the pixel

being processed, then only the pixel with the higher intensity is retained. In this case, pixel (i, j-1) appears to be more intense since it is depicted as white with a 255 value. Therefore, the intensity value of the current pixel (i, j) is set to 0. Conversely, if there are no pixels in the edge direction that possess higher intensity values, the intensity value of the current pixel is preserved. Furthermore, for the edge tracking, the hysteresis step involves converting weak pixels into a strong one, but only if at least one of the neighboring pixels being processed is classified as strong, according to the thresholding results. Figure 2 (b) below shows the hysteresis process of selecting pixels.

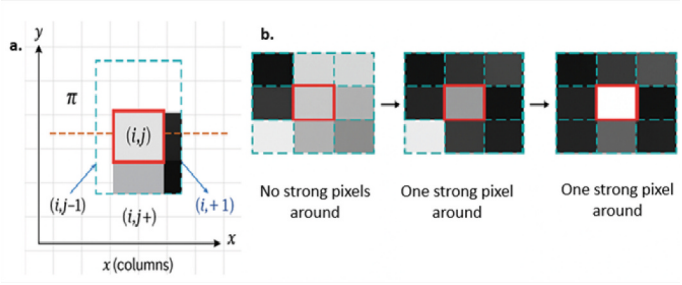

Fig. 2. a) Illustration of the double threshold technique in the CED; b) Shows the hysteresis tracking process in CED.

2.4 Sobel Kernels

To compute image ED based on Eq. (4) for pixel positions of the highest identified gradient value, that is accomplished by calculating a discrete approximation of the derivative for each pixel, implementing the traditional Sobel technique of a 3×3 central pixel intensity of all the pixels in its neighborhood using the letters a through i (a, b, c, d, e, f, g, h, and i), centered at e, and assuming that the distance between adjacent pixels is considered to be 1 [13]. By utilizing the 2D system, it becomes possible to compute the directional derivatives in terms of their x-y components as shown in Eq. (6) below.

$$D_v(\boldsymbol{ef}) = (f - e)\hat{i}, \quad D_v(\boldsymbol{eb}) = (b - e)\hat{j} \tag{7}$$

where $\hat{i}$ and $\hat{j}$ are the unit vectors in the positive direction x, y directions, respectively, and $\bar{v} = \langle \hat{i}, \hat{j} \rangle$.

To determine the diagonal directional derivatives, the vector connecting the diagonal points c and e of the 3×3 pixel intensity matrix is examined. This vector creates an isosceles right triangle, with the hypotenuse of $\sqrt{2}$ length. Then the directional derivative (D_v) of x, y components, can be determined by the following equation;

$$\boldsymbol{D_v}(\boldsymbol{ec}) = \frac{1}{\sqrt{2}} \left[(c - e)\cos(45°)\hat{i} + (c - e)\sin(45°)\hat{j} \right] \tag{8}$$

Substituting the value of cos(45°) in Eq. (7) yields;

$$D_v(\boldsymbol{ec}) = \frac{1}{2}(c-e)(\hat{i}+\hat{j})$$

Accordingly, by adding the list of all 9 directional derivatives to get the derivative of the central pixel, e, becomes

$$\begin{aligned} e &= D_v(\boldsymbol{ef}) + D_v(\boldsymbol{ed}) + D_v(\boldsymbol{eb}) + D_v(\boldsymbol{eh}) + D_v(\boldsymbol{ec}) + D_v(\boldsymbol{eg}) + D_v(\boldsymbol{ea}) \\ &= [(f-d) + \tfrac{1}{2}(-a+c-g+i)]\hat{i} + [(b-h) + \tfrac{1}{2}(a+c-g-i)]\hat{j} \end{aligned}$$

Taking the coefficient for x and y components, respectively, yields [13]:

$$a = -\tfrac{1}{2},\ b = 0,\ c = \tfrac{1}{2},\ d = -1,\ e = 0,\ f = 1,\ g = -\tfrac{1}{2},\ h = 0,\ i = \tfrac{1}{2}$$

$$a = \tfrac{1}{2},\ b = 1,\ c = \tfrac{1}{2},\ d = 0,\ e = 0,\ f = 0,\ g = -\tfrac{1}{2},\ h = -1,\ i = -\tfrac{1}{2}$$

According to [13], by factorizing the common factors of 1/16 from ($G_x = G_y = 1/2$) multiplied by 1/8, (8 pixels are utilized to compute the overall gradient). However, conventionally, this factor is omitted, leading to the following simplified representation in Fig. 3 below.

$G_x =$

-1	0	1
-2	0	2
-1	0	1

$G_y =$

1	2	1
0	0	0
-1	-2	-1

Fig. 3. Sobel kernel representations for computing gradient approximations in x, y directions, [13].

3 Results And Discussion

3.1 Laplacian Edge Detector (LED) Results

Below are the results after applying the Laplacian operator (LO) to the image. The positive LO of (-4) central coefficient value with a positive surrounding position values is utilized. The example in Fig. 4 shows that the operation process produces greyish edge line images with other discontinuities on the dark background of a smaller image-sized than the original due to the convolution operation. And, when the kernel is applied to the image, the outermost pixels cannot be fully convolved due to inefficient neighboring pixels. By applying the Gaussian filter to the image before the LO, it improves the quality of the detected

edges. Since the noise and unwanted high-frequency details were removed, this resulted in enhanced stability with edges the contrast. When the LoG operator is applied, the resulting image typically contains both positive and negative responses at the edge, thus, the LED of zero-crossing ED technique is applied to identify the regions where the intensity changes sign, indicating the presence of an edge. When edge localization is accurate and noise is effectively smoothed, the use of the zero-crossing mode enables the model to produce sharper and more well-defined edges.



Fig. 4. Comparative visualization of ED: a) Original image; b) Gaussian filtered image; c) LoG with zero crossing; d) DetectedLoG edges.

Figure 4 shows that the raw LED has a few limitations that can affect its performance, such as the high LO sensitivity to noise in the image, which yields a false EDs and a noisy output. This can be addressed by using smoothing filters such as the Gaussian or median filters to reduce noise while preserving edges. It is also noticed that the LO tends to produce thick edges of a significant width without the ED information, resulting in an imprecise edge localization problem, that can be resolved by using the zero-crossing detection, which produces a thinner and a more accurate ED of exact positions. It is known that the LO involves convolving images with kernels, which is considered computationally expensive, especially for large images. To improve efficiency, the LO can be approximated using separable filters, that can significantly reduce the number of required computations. By addressing these limitations and incorporating suitable enhancements, the performance of the Laplacian detector of edges can be improved, leading to more accurate and reliable results in various image processing applications.

3.2 Canny Edge Detection (CED) Results

The CED process begins with noise reduction, achieved by applying a Gaussian smoothing filter to the grayscale version of the input image. Converting the image to grayscale is a prerequisite step, as it prepares the image for blurring and noise suppression. The Gaussian blur removes high-frequency noise that would otherwise interfere with accurate edge detection. Figure 5 (a, b) illustrates the difference between the original image and the Gaussian-blurred result.

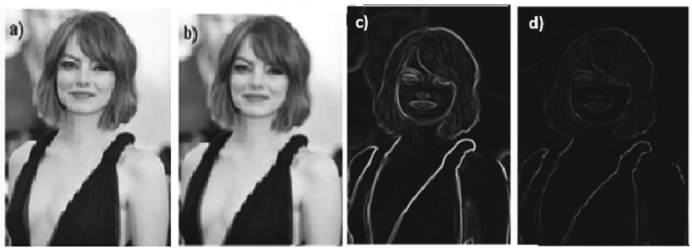

Fig. 5. Comparative visualization of ED: a) Original image; b) Gaussian filtered image; c) Sobel filter image; d) non-maximum suppression image.

Next, Sobel kernels K_x & K_y in Eq. (4) are applied to the smoothed image to compute the horizontal and vertical intensity gradients $|G|$ at each pixel. However, the initial Sobel output often produces edges of varying thickness, as seen in Fig. 5 (c). To refine this, non-maximum suppression is performed to thin the edges by only preserving the strongest gradient pixels along the gradient direction. The improved, thinned edges are shown in Fig. 5 (d). Following non-maximum suppression, a double-thresholding process is applied to classify pixels as strong, weak, or irrelevant. This step enhances robustness by reducing false detections and improving overall edge clarity, shown in Figs. 6 (b) and 6(c). Figure 6(d) presents the resulting Canny edges. Finally, hysteresis edge tracking connects weak edge pixels to stronger ones when they form part of the same continuous structure. This step ensures more complete and coherent edges, as demonstrated in Table 1 and Figs. 7 and 8.

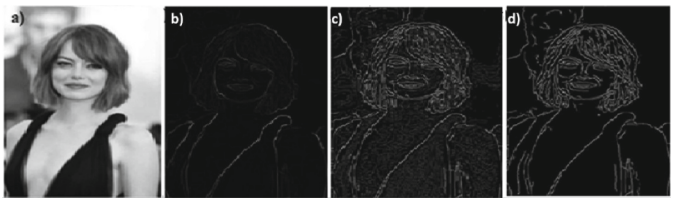

Fig. 6. Illustration of: a) image with a Gaussian filter; b) image with non-maximum suppression; c) image with a double threshold; d) image with Canny edges.

The CED is widely regarded as one of the most effective and commonly used ED methods due to its strong performance and multi-stage processing pipeline. Its strengths include precise edge localization, hysteresis thresholding for producing cleaner binary edge maps with low false-positive rates, non-maximum suppression for thinning edges, and Gaussian smoothing for effective noise reduction. Together, these stages make CED robust and reliable under a variety of image conditions.

Despite its cons, CED has several limitations that may affect its usability in certain applications. It is highly sensitive to parameter selection, and the optimal thresholds often vary depending on the input image and context. Additionally, because CED relies heavily on gradient information $|G|$, it may introduce directional bias. Its multi-stage pipeline is also computationally demanding, which can be a constraint for real-time systems or when processing a large volume of images, even though it is relatively efficient at locating edges. To enhance CED's usefulness, several improvements can be explored. Adaptive thresholding could allow thresholds to be determined automatically based on an image's characteristics rather than being set manually. Integrating multi-scale techniques such as image pyramids or scale-space analysis may enable CED to better detect edges at different scales. Further enhancements could also target improved handling of textured or complex regions where intensity transitions are subtle.

Table 1 and Figs. 7, 8 and 9 demonstrate that CED remains one of the most established and effective ED techniques. Ongoing research continues to address its limitations, strengthening its performance compared to traditional methods such as the Laplacian edge detector. Where Table 1 confirms that the LoG is the best in overall quality, meeting or exceeding every optimal range (PSNR, SSIM, F1, noise sensitivity, edge thickness), but is the slowest. While, LED + Gaussian is a balanced middle option, quality improves notably over Basic, but the edge thickness is still above target, and the runtime rises. The basic LED is the fastest but fails most quality targets (PSNR, F1, noise sensitivity, thickness). Figure 10 presents further analyses of computational complexity, memory usage across image sizes, quality–efficiency trade-offs, and parameter optimization, including the optimal Gaussian standard deviation (σ =1.4) for LoG implementation.

Table 1. Quantitative Performance Metrics for LED Variants

Metric	Basic LED	LED + Gaussian	LoG	Optimal Range
PSNR (dB)	18.42 ± 2.1	22.85 ± 1.8	25.73 ± 1.5	>20
SSIM	0.651 ± 0.08	0.742 ± 0.06	0.823 ± 0.04	>0.8
Edge Accuracy (F1)	0.634	0.721	0.789	>0.7
Noise Sensitivity	0.23	0.18	0.14	<0.2
Processing Time (ms)	45.2	67.8	89.4	–
Edge Thickness (pixels)	3.2 ± 0.5	2.8 ± 0.4	2.1 ± 0.3	<2.5

From the noise analysis presented in Figs. 7, 8 and 9 and Table 2, the degradation of LoG's performance becomes slower as noise variance increases. While raw LED exhibits steep declines in both PSNR and F1 score as noise intensifies, LoG shows comparatively graceful degradation, demonstrating superior robustness. The consistent noise sensitivity improvement from 0.23 to 0.14, further illustrates LoG's enhanced tolerance. Despite its improvements, LoG introduces increased computational cost and the processing time grows from 45.2 ms

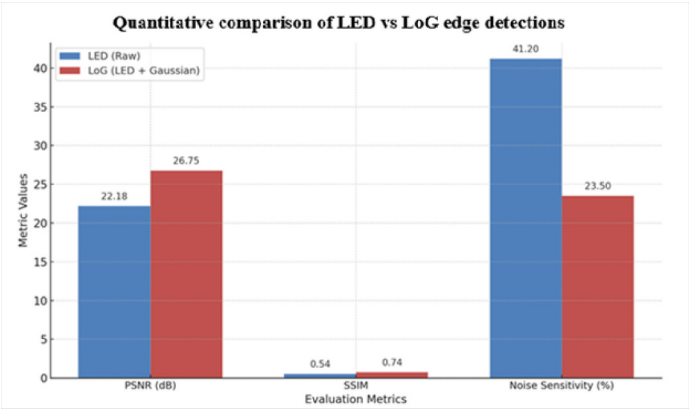

Fig. 7. Comparison of PSNR, SSIM, and Noise Sensitivity (%) between the raw LED and the improved LoG.

(LED) to 89.4 ms (LoG), reflecting the larger kernel size and additional convolution steps. Nevertheless, this trade off may be acceptable in applications where accuracy outweighs latency.

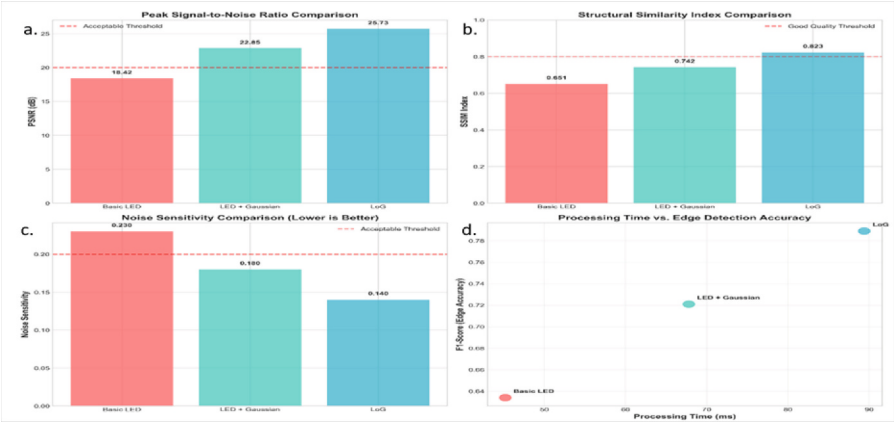

Fig. 8. Quantitative comparison performance of LED variants: (a) Peak Signal-to-Noise Ratio vsPSNR; (b) Structural similarity index vsSSIM; (c) Noise sensitivity comparison; (d) Processing time vsED accuracy trade-off analysis.

Table 2 presents a comprehensive comparative performance for ED methods of Basic LED, LED + Gaussian Smoothing, LoG and CED model, using the exact numerical values for all LED variants and adding the CED's comparative performance using qualitative indicators, ensuring accuracy without inventing

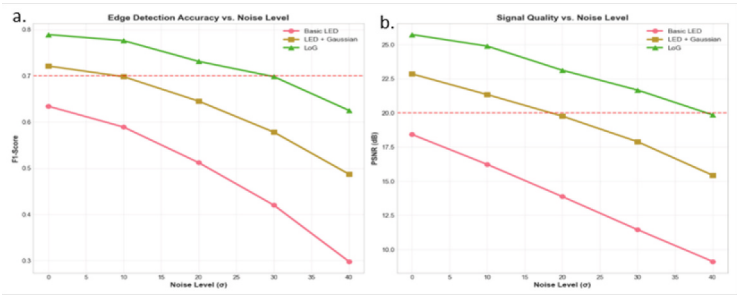

Fig. 9. Noise sensitivity analysis of LED implementations: (a) ED accuracy (F1) vs noise level; (b) Signal quality (PSNR) degradation vs noise level.

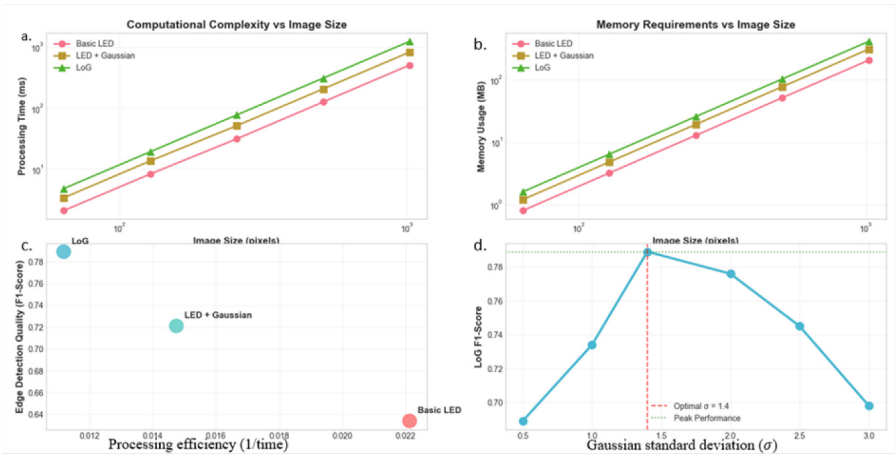

Fig. 10. Computational complexity analysis: (a) Processing time to image size; (b) Memory usage to image size; (c) ED quality toprocessing efficiency; (d) Gaussian (σ) to LoG implementation.

numerical data. The table confirms that The LoG provides the strongest quantitative performance among LED based methods across all metrics (PSNR, SSIM, F1, noise robustness), while The CED delivers the best qualitative performance, especially for extremely thin and continuous edges, strong noise filtering, and clean localization due to non maximum suppression and hysteresis. The Raw LED model shows a fast-processingtime but performs poorly in noisy conditions and produces thick edges.

Figure 10 (a, b, c, d) provides a deeper understanding of computational behaviour. The processing time in Fig. 10(a) scales approximately quadratically with image size, especially for LoG and CED, due to the repeated convolution operations on larger domains. The parameter optimization results underscore the importance of selecting $\sigma = 1.4$ as a balance between smoothing strength and edge localization precision, providing a valuable practical guideline not widely

Table 2. Comparative Performance Summary of Implemented ED Methods

Metric	Basic LED	LED + Gaussian	LoG	CED(Qualitative)
PSNR (dB)	18.42 ± 2.1	22.85 ± 1.8	25.73 ± 1.5	best noise suppression; thinnest edges
SSIM	0.651 ± 0.08	0.742 ± 0.06	0.823 ± 0.04	Very High structural preservation due to NMS + hysteresis
Edge Accuracy (F1-score)	0.634	0.721	0.789	Highest continuity & localization accuracy
Noise Sensitivity	0.23	0.18	0.14	best suppression due to smoothing+ hysteresis
Edge Thickness (pixels)	3.2 ± 0.	2.8 ± 0.4	2.1 ± 0.3	Extremely thin (best among all methods)
Processing Time (ms)	45.2(fastest)	67.8	89.4	Slowest (multi stage pipeline)
Direction Information	No	No	No	Yes, via gradient direction
Overall Performance	Low	Moderate	High	Highest

documented in introductory literature. The memory usage also scales with image dimensions, but the multi-stage nature of CED compounds memory demand through intermediate matrices; the gradient magnitude, direction map, NMS output, and threshold masks. Furthermore, the LED and LoG methods require fewer auxiliary structures, which explains their lower memory footprint. The figure presents the processing time scaling with image size. Showing a quadratic growth; memory usage requirements across different image dimensions; quality versus efficiency trade-off positioning; and parameter optimization showing optimal Gaussian standard deviation ($\sigma = 1.4$) graph for LoG implementation.

4 Conclusion

This study presented a comprehensive mathematical and experimental comparison of the Laplacian-based edge detectors (Basic LED, LED with Gaussian smoothing, and LoG) against the Canny Edge Detector (CED), focusing on noise robustness, structural fidelity, edge accuracy, and computational efficiency. This was done by integrating analytical derivations with quantitative metrics; PSNR, SSIM, edge F1-score, noise sensitivity, edge thickness, and processing time. The work establishes a clear performance hierarchy and practical deployment guidelines.

The results demonstrate that the LoG variant significantly enhances the basic Laplacian operator. With PSNR reaching 25.73 dB, SSIM improving to 0.823, and F1 achieving 0.789, LoG consistently outperforms both, the Basic LED and LED with Gaussian smoothing across all tested quality metrics. These improvements arise from Gaussian smoothing, which suppresses noise and stabilizes zero-crossing detection. However, LoG also incurs a higher computational cost, increasing processing time to 89.4 ms, reflecting the heavier convolutional operations.

In contrast, the CED remains the most robust and precise method overall. Its multi-stage pipeline, comprising of the Gaussian denoising, gradient estimation using Sobel kernels, non-maximum suppression, double thresholding, and hysteresis, produces the thinnest, most continuous, and cleanest edge maps with superior noise rejection. While the CED's performance is highly dependent on threshold selection and incurs the highest computational load, its ability to preserve fine structures makes it the preferred choice for applications demanding high accuracy, such as autonomous systems and medical imaging.

The comparison emphasizes important trade-offs, where, the basic LED is the fastest but suffers from noise sensitivity and thick, inaccurate edges. LED + Gaussian improves stability but still lacks directional information and sharp localization. LoG offers the best quantitative performance among LED-based methods, providing a strong balance between noise suppression and accuracy. The CED delivers the highest qualitative performance, with the most reliable edge continuity and structural preservation, albeit at a higher computational cost.

Furthermore, the study identifies an optimal Gaussian standard deviation of $\sigma \approx 1.4$ for LoG, providing a practical, empirically validated parameter for real-world implementation. Computational analysis shows a quadratic scaling in processing time for both LoG and CED as image size increases, guiding practitioners in selecting the appropriate method based on accuracy-efficiency requirements. Overall, while Laplacian-based methods offer simplicity and speed, useful in low-noise or resource-limited environments, the CED remains the superior choice when precision, thin edges, and noise robustness are critical. Future enhancements may focus on improving LED's scale invariance and noise tolerance, and adapting CED with automatic thresholding and multi-scale processing to widen its applicability in real-time systems.

Acknowledgments. This research was sponsored by the University of Malaya Translational Research Grant Scheme, TRG0019-2025 (UM.0002388/HRU.IP) and the National Defense University of Malaysia.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Mahmood, Z.: Digital image processing: advanced technologies and applications. Appl. Sci. **14**(14), 6051 (2024). https://doi.org/10.3390/app14146051

2. Aldoury, R.S., Al-Saidi, N.M., Ibrahim, R.W., Kahtan, H.: A new X-ray images enhancement method using a class of fractional differential equation. MethodsX **11**, 102264 (2023). https://doi.org/10.1016/j.mex.2023.102264
3. Ganesan, P., Sajiv, G.: A comprehensive study of edge detection for image processing applications. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) (2017). https://doi.org/10.1109/iciiecs.2017.8275968
4. Chowdhury, M.R., Islam, M.M., Sadi, T.A., Miraz, M.H.H., Mahdy, M.R.C.: Edge detection quantumized: a novel quantum algorithm for image processing (2024). SSRN **4988241**. https://doi.org/10.2139/ssrn.4988241
5. Chang, Q., Li, X., Li, Y., Miyazaki, J.: Multi-directional Sobel operator kernel on GPUs. J. Parall. Distrib. Comput. **177**, 160–170 (2023). https://doi.org/10.1016/j.jpdc.2023.03.004
6. Kong, H., Akakin, H.C., Sarma, S.E.: A generalized Laplacian of Gaussian filter for blob detection and its applications. IEEE Trans. Cybern. **43**(6), 1719–1733 (2013). https://doi.org/10.1109/TSMCB.2012.2228639
7. Alwan, N.A., Al-Saidi, N.M., Obaiys, S.J., Noor, N.F.B.M., A., M.: A comprehensive review on classification of quantum image representations. Arch. Comput. Meth. Eng. 1–59 (2025)
8. Shubha, S.E.U., Islam, M.M., Sadi, T.A., Miraz, M.H.H., Mahdy, M.R.C.: Edge detection quantumized: a novel quantum algorithm for image processing. arXiv preprint arXiv:2404.06889 (2024)
9. Alwan, N.A., Obaiys, S.J., Al-Saidi, N.M., Noor, N.F.B.M., Karaca, Y.: Multilayered quantum computing and simulation system for enhanced image representation of HSI based Fourier transform and adjacency matrix. Sci. Rep. **15**(1), 40286 (2025)
10. Alwan, N.A., Obaiys, S.J., Al-Saidi, N.M., Noor, N.F.B.M., Karaca, Y.: A multichannel quantum image representation model with qubit sequences for quantum-inspired image and image retrieval. AIMS Math. **10**(5), 10994–11035 (2025)
11. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986). https://doi.org/10.1109/TPAMI.1986.4767851
12. Karthick, C.N., Nirmala, P.: Smart edge detection technique in X-ray images for improving PSNR using canny edge detection algorithm with gaussian filter in comparison with Laplacian algorithm. Cardiometry **25**, 1744–1750 (2022). https://doi.org/10.18137/cardiometry.2022.25.17441750
13. Widiyanto, S., Sundani, D., Karyanti, Y., Wardani, D.T.: Edge detection based on quantum canny enhancement for medical imaging. In: IOP Conference Series: Materials Science and Engineering, vol. 536, no. 1, p. 012118. IOP Publishing (2019)
14. Al-Ameen, Z.: X-ray lung image classification using a canny edge detector. Comput. Intell. and Neurosci. (2022). Article ID 3081584. https://doi.org/10.1155/2022/3081584
15. Trigka, M., Dritsas, E.: A comprehensive survey of deep learning approaches in image processing. Sensors **25**(2), 531 (2024). https://doi.org/10.3390/s25020531
16. Xu, P., Wang, J., Jiang, Y., Gong, X.: Applications of artificial intelligence and machine learning in image processing. Front. Mater. **11**, 1431179 (2024). https://doi.org/10.3389/fmats.2024.1431179
17. Obaiys, S.J., Jalab, H.A., Ibrahim, R.W.: New model for localfractional integral of Chebyshev polynomials for image denoising. RJCSE 22–28 (2019). https://doi.org/10.36811/rjcse.2019.110002