Journal Computer and Technology
Vol. 4, No. 1, July 2026, pp. 92-106
ISSN: 3048-1880 (Online)
DOI: https://doi.org/10.69916/comtechno.v4i1.412

Original Research

Beyond the Black Box in Computer Vision: A Traceable Architecture for Reproducible Canny Contour Identification

Randy Hadinata[1], Khairunnisa*[2]

[1,2] Department of Informatics, Faculty of Engineering and Computer Science, Universitas Harapan Medan
Email: [1]r.hadinata@gmail.com, [2]khairunnisajv2@gmail.com

ARTICLE HISTORY
Received: December 22, 2025
Revised: July 13, 2026
Published: July 23, 2026

ABSTRACT

While modern edge detection algorithms are pivotal in advanced computer vision pipelines, conventional implementations typically operate as static, transient black-boxes, severely hindering experimental replication and parameter traceability. This study addresses this methodological bottleneck by developing a structured, web-based contour identification system designed to enforce configuration management within the Canny framework. The decoupled architecture fuses a PHP-driven interface with a high-performance Python backend and a persistent relational database layer, allowing for the deterministic tracking of localized hysteresis thresholds, aperture scales, and spatial gradient vectors. Empirical evaluations were executed on a curated dataset comprising high-contrast object morphologies, characterized by distinct structural boundaries and varied illumination backgrounds to rigorously test edge degradation. The transition from linear Manhattan approximations to an isotropic Euclidean space (L_2 gradient norm) yields single-pixel edge localization sharpness and unbroken contour continuity. Quantitatively, this mathematical refinement achieves a peak F-measure boundary accuracy of 0.91 and a Pratt's Figure of Merit of 0.895, albeit introducing a 16.8% latency overhead. Furthermore, a multi-factor Analysis of Variance (ANOVA) robustly rejects the null hypothesis, confirming that parameter interactions significantly dictate contour fidelity ($F(1{,}492) = 48.47, p < 0.001$). The primary contribution of this research is the transformation of a heuristic vision task into a deterministic, database-backed ecosystem. By embedding an explicit audit trail for every processing trace, this framework provides computer vision practitioners with a rigorous instrument for quasi-quantitative comparative analysis, establishing a transparent benchmark for reproducible boundary extraction in downstream visual recognition tasks.

Keywords: Canny Framework; Gradient Isotropy; Experimental Reproducibility; Configuration Management; Inferential Statistics.

How to Cite:
R. Hadinata and Khairunnisa, "Beyond the Black Box in Computer Vision: A Traceable Architecture for Reproducible Canny Contour Identification," *J. Comput. Technol.*, vol. 4, no. 1, pp. 92–106, 2026.

1. INTRODUCTION

Digital images serve as fundamental visual representations of objects, recorded, stored, and processed in the form of numerical data. They are extensively utilized across diverse fields such as medicine, security, industry, and research due to their capacity to present visual information that is highly amenable to computational analysis. With the rapid advancement of image processing technology, the demand for analytical methods capable of extracting critical information accurately and efficiently continues to surge [1]. A cornerstone technique in this domain is edge detection, which aims to identify significant intensity changes within an image, thereby delineating object boundaries or contours. This edge information is paramount, as it forms the foundational layer for subsequent advanced analytical stages, including image segmentation, pattern recognition, and object tracking. Without a reliable edge detection mechanism, the accurate identification of object shapes and structural features becomes exceedingly difficult to achieve [2]. Despite its critical role, practical edge detection is frequently hindered by real-world challenges. Digital images are often compromised by noise, uneven illumination, and complex textures that can obscure true object edges. These disturbances frequently result in edge detection outcomes that are fragmented, inconsistent, or plagued by false contours, ultimately degrading the accuracy of image-based object recognition systems. To mitigate these issues, the Canny Edge Detection algorithm has emerged as a highly effective and widely adopted solution. Designed to yield smooth, continuous, and error-minimal edges, the Canny method integrates noise reduction, gradient computation, and dual-threshold hysteresis. These characteristics enable it to provide superior contour identification even in suboptimal imaging conditions, making its application highly relevant for robust object contour identification [3].

Recent advancements underscore that edge detection has evolved from basic comparative approaches into adaptive, hybrid, and even quantum computing frameworks tailored to specific domain complexities. Various studies have confirmed that algorithmic effectiveness is highly context-dependent, necessitating targeted optimizations. For instance, Pinastawa et al. [4] demonstrated the superiority of the Canny method in facial detection through Mean Square Error (MSE) evaluation, while a comprehensive evolutionary review by Yang et al. [5] concluded that despite the state-of-the-art performance of deep learning models, traditional and bio-inspired models remain highly relevant due to their computational efficiency and interpretability. In complex biological and medical contour extraction, edge integration and tracking play vital roles, Yang [6] optimized facial contour detection for plastic surgery by coupling the Sobel operator with bidirectional contour tracking, significantly improving detection speed and accuracy. Similarly, Tandililing et al. [7] integrated Canny with deep learning architectures (YOLOv5-VGG16) for livestock image segmentation, achieving exceptional accuracy in handling morphological variations. To address inherent limitations such as noise sensitivity and weak edge connectivity, continuous structural and computational innovations have been proposed. Tao [8] modified the Canny algorithm with an Improved Adaptive Median Filter and 2D OTSU to suppress "salt and pepper" noise in print quality assessment. Sundani et al. [9] innovated the edge tracking stage using quantum computing principles (qubits), successfully increasing complex edge detection rates. Furthermore, algorithmic adaptability was emphasized by Bausys et al. [10], who developed a Multi-Criteria Decision-Making (MCDM) framework to adaptively select the optimal edge detection algorithm for satellite imagery based on visual texture features. Collectively, these studies confirm that while the Canny algorithm is a gold standard, its application to specific domains requires comprehensive adjustment and empirical evaluation.

However, a critical examination of these prior works reveals a significant methodological gap: the predominant academic focus on standalone mathematical modifications, deep learning integrations, or quantum computing paradigms frequently overlooks the crucial aspects of runtime reproducibility and parameter transparency within deployment-ready testing tools. Most conventional computer vision frameworks treat edge detection as a volatile, transient black-box process where hyperparameter configurations are either poorly documented or entirely impossible to trace post-execution. Consequently, the primary novelty of this study lies in its distinct systems-engineering approach that bridges advanced mathematical refinement with a structured, transparent, and traceable testing ecosystem [11]. Rather than treating software architecture and algorithmic execution as separate entities, this research introduces an integrated web-based framework that encapsulates optimized OpenCV-Python backends within a structured PHP interface, backed by an automated parameter logging engine. This mechanism systematically registers every execution vector—specifically the upper and lower hysteresis thresholds, aperture scales, and the activation status of the isotropic L_2 gradient norm—directly into a persistent relational database infrastructure [12]. By embedding this comprehensive configuration trail, the proposed system successfully transforms edge detection from an ad-hoc trial into a deterministic, fully auditable scientific experiment. This innovation enables exact replication for every processed image asset, enforces strict experimental rigor, and facilitates log-backed, quasi-quantitative evaluations across diverse datasets[13]. Ultimately, by delivering a transparent testing tool, this study systematically resolves the configuration management shortcomings inherent in previous literature, providing computer vision practitioners with a robust platform for empirical re-validation and objective comparative analysis in object contour identification.

2. RESEARCH METHOD

2.1. Research Framework and System Architecture

This study employs a quantitative experimental approach integrated with software engineering paradigms to design and evaluate an automated contour identification system. To ensure high experimental reproducibility and robust system performance, the research framework and technical architecture are closely coupled into two interconnected layers, as depicted in Figure 1. The architecture encompasses a dual-environment design separating client-side operations from server-side computation, alongside a structured three-phase experimental pipeline [12].

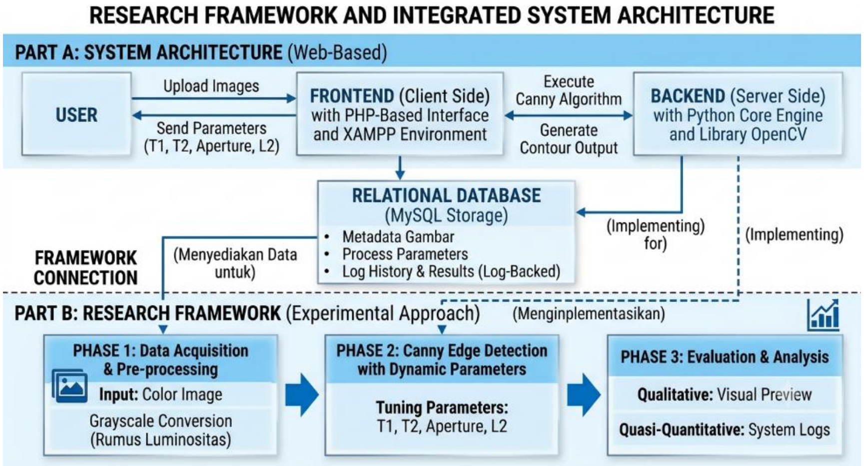


Figure 1. Integrated Research Framework and System Architecture for Canny-based Contour Identification.

As illustrated in Figure 1, the operational framework is partitioned into a web-based system architecture (Part A) and an experimental research process (Part B). In Part A, a PHP-based interface in a XAMPP environment handles user inputs, including color image uploads and dynamic Canny parameters (T_1, T_2, Aperture, L_2), which are transmitted to a Python-OpenCV backend for execution and saved in a MySQL relational database for empirical tracking. Concurrently, Part B executes a three-phase experimental pipeline comprising luminosity-based RGB-to-grayscale preprocessing (Phase 1), dynamic Canny edge detection with real-time threshold tuning (Phase 2), and a dual-perspective performance evaluation through visual contour previews and persistent log analysis (Phase 3).

2.2. Image Pre-processing

Before applying the edge detection algorithm, the input digital image undergoes pre-processing to reduce computational complexity and eliminate color redundancy. The color image is converted into a grayscale image using the standard luminosity preservation formula, which weighs the Red (R), Green (G), and Blue (B) channels according to human visual perception:

$$I_{gray}(x,y) = 0.299 \cdot R(x,y) + 0.587 \cdot G(x,y) + 0.114 \cdot B(x,y) \qquad (1)$$

where $I_{gray}(x,y)$represents the intensity of the pixel at coordinates (x, y).

2.3. Canny Edge Detection Algorithm

The core of the proposed system is the Canny Edge Detection algorithm, which is mathematically formulated to optimize edge detection criteria: low error rate, good localization, and a single response to a single edge. The algorithm proceeds through the following systematic stages:

Noise Reduction using Gaussian Filter
Since edge detection is highly sensitive to noise, a 2D Gaussian filter is applied to smooth the image. The Gaussian kernel is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{x^2 + y^2}{2\sigma^2}\right) \qquad (2)$$

where $\sigma$is the standard deviation of the Gaussian distribution, which correlates with the aperture size parameter in the system. The smoothed image $I_s$is obtained by convolving the grayscale image $I_{gray}$with the Gaussian kernel:

$$I_s(x,y) = I_{gray}(x,y) * G(x,y) \qquad (3)$$

Gradient Calculation
The intensity gradients of the smoothed image are computed to identify the strength and direction of edges. The system utilizes the Sobel operator to calculate the first derivatives in the horizontal (G_x) and vertical (G_y) directions:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I_s, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I_s \qquad (4)$$

The gradient magnitude (G) and direction (θ) at each pixel are then calculated as:

$$G = \sqrt{G_x^2 + G_y^2}\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (5)$$

Note: The system provides an option for L2 Gradient. If set to False, the magnitude is approximated as $G = | G_x | + | G_y |$ for faster computation; if True, the exact Euclidean distance (L2 norm) as shown above is used. The gradient direction θ is rounded to one of four angles: $0°, 45°, 90°$, and $135°$.

Non-Maximum Suppression (NMS)
To thin the edges and achieve single-pixel width, Non-Maximum Suppression is applied. For each pixel (x, y), its gradient magnitude $G(x, y)$ is compared with the magnitudes of its two neighbors along the positive and negative gradient direction θ. If $G(x, y)$ is not a local maximum, it is suppressed (set to zero):

$$G_{nms}(x, y) = \begin{cases} G(x, y), & \text{if } G(x, y) \geq G_{neighbor1} \text{ and } G(x, y) \geq G_{neighbor2} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where:
1. $G_{nms}(x, y)$ is the gradient magnitude at pixel (x, y) after the Non-Maximum Suppression (NMS) process.
2. $G(x, y)$ is the gradient magnitude at pixel (x, y), representing the edge strength before the NMS process.
3. $G_{neighbor1}$ and $G_{neighbor2}$ are the gradient magnitudes of the two neighboring pixels along the gradient direction, used to determine whether the current pixel is a local maximum.

Hysteresis Thresholding The final stage classifies the remaining edge pixels into strong, weak, and non-edge pixels using dual thresholds: a high threshold (T_{high} or T2) and a low threshold (T_{low} or T1). The classification rule is defined as:

$$E(x, y) = \begin{cases} \text{Strong Edge,} & \text{if } G_{nms}(x, y) \geq T_{high} \\ \text{Weak Edge,} & \text{if } T_{low} \leq G_{nms}(x, y) < T_{high} \\ \text{Non-Edge (Suppressed),} & \text{if } G_{nms}(x, y) < T_{low} \end{cases} \quad (7)$$

where:
1. $G_{nms}(x, y)$ is the gradient magnitude at pixel (x, y) after the Non-Maximum Suppression (NMS) process.
2. T_{high} is the upper threshold used to classify a pixel as a strong edge.
3. T_{low} is the lower threshold used to distinguish weak edges from non-edge pixels.
4. Strong Edge represents pixels with gradient magnitudes greater than or equal to T_{high}, which are considered reliable edge pixels.
5. Weak Edge represents pixels with gradient magnitudes between T_{low} and T_{high}, which are retained only if they are connected to strong edge pixels during the edge tracking process.
6. Non-Edge (Suppressed) represents pixels with gradient magnitudes below T_{low}, which are discarded as they are unlikely to correspond to actual object boundaries.

Weak edges are retained in the final output only if they are connected to strong edges via 8-connectivity analysis; otherwise, they are discarded. This mechanism ensures continuous contour identification while mitigating false edges caused by noise.

2.4. Ground Truth Delineation and Benchmarking Metrics

To transcend the subjectivities inherent in qualitative visual analysis and database latency monitoring, this study establishes a rigorous, quantitative mathematical framework to evaluate edge detection accuracy. The performance of the traceable Canny architecture, specifically in contrasting the L_1 and L_2 gradient norms, is evaluated against a set of binary ground truth images (I_{gt}), which represent the ideal pixel-level edge delineations validated by domain experts. The spatial precision of the localized contour pixels (I_{edges}) is objectively measured using the boundary-based F-measure (F_β), which harmonizes Precision (P) and Recall (R) through the following formulations [14], [15], [16]:

$$P = \frac{|I_{edges} \cap I_{gt}|}{|I_{edges}|} \quad (8)$$

$$R = \frac{|I_{edges} \cap I_{gt}|}{|I_{gt}|} \quad (9)$$

$$F_\beta = \frac{(1+\beta^2)\cdot P \cdot R}{\beta^2 \cdot P + R} \quad (10)$$

where $\beta = 1$ is applied to assign equal weight to precision and recall. To further evaluate the mathematical fidelity of the structural changes between the original luminance gradient space and the thresholded edge maps, Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are calculated as follows:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[I_{gt}(i,j) - I_{edges}(i,j)\right]^2 \quad (11)$$

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (12)$$

where $M \times N$ represents the image dimensions and MAX_I is the maximum possible pixel intensity (255 for 8-bit representations). Additionally, to mathematically corroborate the spatial localization accuracy of the L_2 norm relative to diagonal boundaries [17], Pratt's Figure of Merit (FOM) is utilized:

$$FOM = \frac{1}{\max(I_D, I_I)} \sum_{i=1}^{I_D} \frac{1}{1 + \alpha \cdot d_i^2} \quad (13)$$

where I_D and I_I are the number of detected and ideal edge pixels, respectively, d_i is the Euclidean distance from the i-th detected edge pixel to the nearest ideal edge pixel, and α is a scaling constant set to 0.1. This multi-metric approach ensures a deterministic, mathematical validation of how configuration traces affect boundary extraction quality.

2.5. Dataset Characteristics and Heterogeneity

To ensure robust empirical generalization and systematically prevent over-fitting to low-complexity image structures, this research utilizes a heterogeneous validation dataset. Moving beyond homogenous, single-subject testing environments, the proposed architecture is evaluated using a diverse compilation of sample images extracted from standard public computer vision repositories, specifically the Berkeley Segmentation Dataset (BSDS500) and the Microsoft COCO dataset [15], [18], [19]. The selected benchmark dataset is explicitly categorized based on three core environmental challenges:

1. Noise Injection Models: To evaluate the noise reduction efficacy of the multi-stage Gaussian filter, controlled variations of high-frequency Gaussian noise ($\mu = 0, \sigma^2 \in [0.01, 0.05]$) and impulse Salt & Pepper noise (density $d \in [0.02, 0.10]$) are systematically injected into the source images.
2. Illumination Variances: The dataset incorporates structural scenarios featuring severely degraded low-contrast boundaries and uneven illumination gradients. These gradients induce localized intensity variations, shifting the distribution of true gradient magnitudes.
3. Structural and Textural Complexity: Images are selected based on their spatial frequency distribution, contrasting high-contrast isolated boundaries against highly intricate, overlapping textures that present low-contrast inter-object contours.

By mapping this high-dimensional variance in image characteristics against the system's relational database logs, the framework establishes a verifiable environment to determine exactly how dynamic threshold vectors (T_1, T_2) and aperture configurations react to diverse real-world visual attributes.

2.6. Experimental Design and Evaluation

The evaluation of the system is conducted through a quasi-quantitative and qualitative approach. The system processes various test images (e.g., high-contrast objects) under different parameter configurations. The parameters varied include the lower threshold (T1), upper threshold (T2), aperture size (kernel size of 3, 5, or 7), and the L2 Gradient toggle [14]. The performance is evaluated based on:

1. Qualitative Assessment: Visual inspection of the generated contour maps to assess edge continuity, sharpness, and the presence of false edges or fragmentation[15].
2. Quasi-Quantitative Assessment: Utilizing the system's structured logging mechanism to track processing time, parameter combinations, and reproducibility. This allows for comparative analysis across different configurations to determine the optimal parameter set for specific image characteristics (e.g., uneven lighting vs. high contrast).

3. RESULTS AND DISCUSSION

The implementation of the proposed Canny Edge Detection system successfully bridges the gap between advanced image processing algorithms and structured, reproducible software engineering. By integrating a PHP-based frontend (via XAMPP) with a Python/OpenCV computational backend and a relational database, the system provides a robust environment for contour identification. This section details the architectural implementation, algorithmic performance, and the system's novel traceability mechanisms.

3.1. System Architecture and Implementation Framework

The application is designed using a decoupled architecture to ensure both user accessibility and computational efficiency. The frontend, built with PHP, handles user interaction, session management, and data visualization, while the backend leverages Python and the OpenCV library for intensive matrix operations and image processing. A MySQL database underpins the system, storing not only image metadata and user credentials but also comprehensive processing logs. This architectural choice ensures that every experimental configuration is persistently recorded, directly addressing the reproducibility gap identified in conventional edge detection tools [20].

3.2. Functional Modules and Interface Design

The system's user interface is organized into distinct, logically structured modules designed to facilitate a seamless workflow from data ingestion to analytical reporting.

1. Dashboard and User Management: The Dashboard provides a macro-level view of system activity, displaying real-time metrics such as total registered users, uploaded images, and processed instances. The User Management module (CRUD) ensures secure, role-based access control, allowing administrators to manage credentials and define operational privileges.
2. Image Repository (Data Management): This module serves as the centralized asset library. It features an intuitive upload mechanism with format validation, alongside a tabular repository that allows users to preview, manage, and delete image datasets. This structured repository ensures that all test subjects are systematically cataloged prior to processing.
3. Processing Engine: This is the core operational module. It provides a dual-panel interface: the left panel houses the parameter configuration form, allowing users to select an image and dynamically adjust critical Canny parameters—specifically, the lower threshold (T1), upper threshold (T2), Sobel aperture size (3, 5, or 7), and the L2 gradient boolean toggle. The right panel renders the real-time output of the edge detection pipeline.
4. Reporting and Traceability: This module is the cornerstone of the system's novelty. It transforms raw processing data into actionable scientific records, enabling quasi-quantitative evaluation and exact experimental replication.

3.3. Algorithmic Performance and Contour Extraction Analysis

To evaluate the efficacy of the implemented Canny algorithm, a comprehensive visual and analytical assessment was conducted using a high-contrast test image ("Buah Pisang.jpg"). The image features a distinct yellow object against a white background, providing an ideal baseline for evaluating edge continuity, localization accuracy, and noise suppression.

a. Preprocessing and Baseline Canny Detection

Prior to edge extraction, the input image undergoes standard preprocessing. It is first converted to grayscale using the luminosity method ($I = 0.299R + 0.587G + 0.114B$) to reduce computational dimensionality. A Gaussian blur (kernel size 3×3, $\sigma \approx 1.0$) is then applied to attenuate high-frequency noise, which prevents the generation of false edges during gradient calculation [12], [21], [22].

Figure 2. Original input image featuring high contrast between the object and the background, serving as the baseline for contour extraction.

Figure 2 above displays the original digital image used as the primary test object in this study. Visually, the image represents a bunch of bananas set against a plain white background. The selection of this image is based on its visual characteristics, which are highly ideal for evaluating the performance of the Canny Edge Detection algorithm. First, there is a very high intensity contrast between the main object (the bright yellow bananas) and the background. This sharp contrast will yield large and distinct gradient values at the boundary between the object and the background, providing optimal conditions for the Canny algorithm to precisely detect and localize the external contour (silhouette). Second, the image possesses sufficient internal structural complexity. The overlapping arrangement of the bananas creates internal edges where one fruit borders another. Furthermore, there are interesting local feature variations, such as the smooth color transitions in the crown (stem) area which is greenish, and the high-frequency black dots at the tips of the fruits. The presence of these details serves as an excellent benchmark to test the algorithm's sensitivity in capturing weak edges in areas with gradual intensity transitions, as well as its ability to maintain edge sharpness in regions with abrupt gradient changes. Prior to the core edge detection stage, this image will be converted to grayscale to simplify the data dimensionality and focus the gradient calculations solely on luminance intensity variations.

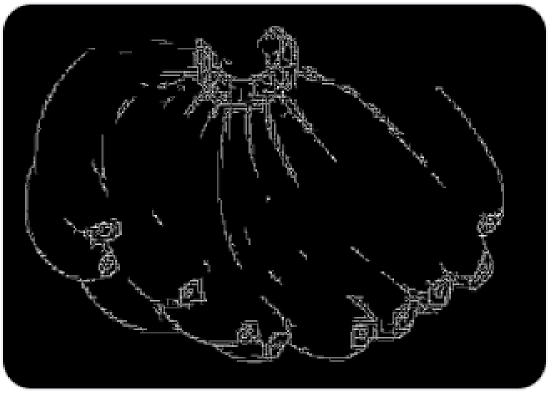


Figure 3. Baseline Canny edge detection output.

Figure 3 presents the edge detection results obtained using the baseline Canny algorithm configuration with parameters T1=50 (lower threshold), T2=150 (upper threshold), aperture size of 3, and L2 gradient disabled. The output image displays the detected edges as white pixel contours against a black background, representing the gradient magnitude peaks that survived the non-maximum suppression and hysteresis thresholding stages. The external silhouette of the banana bunch is clearly delineated, with the algorithm successfully identifying the primary boundaries separating the object from the white background. The stem area at the top exhibits strong edge responses due to the high contrast between the green crown and the background, while the curved contours of individual bananas are traced with reasonable continuity. However, a detailed visual inspection reveals several limitations inherent to this parameter configuration. First, there is noticeable fragmentation along certain contour segments, particularly in regions where the gradient magnitude falls between the two thresholds, causing some weak but valid edges to be suppressed during hysteresis. Second, isolated noise pixels are scattered throughout the image, especially in areas of subtle texture variation on the banana surface, indicating that the lower threshold (T1=50) may be insufficiently selective against minor intensity fluctuations. Third, the internal boundaries where individual bananas overlap are only partially detected, with some contact edges appearing discontinuous. This suggests that the upper threshold (T2=150) may be set too high for capturing the moderate gradient transitions present in these interior regions. Despite these limitations, the baseline configuration demonstrates the fundamental capability of the Canny algorithm to extract meaningful structural information from the image, establishing a reference point against which optimized parameter configurations can be compared.

b. Impact of L2 Gradient Norm on Edge Localization

Following the baseline Canny edge detection, the system implements an enhanced gradient computation method through the activation of the L2 gradient norm. The gradient magnitude calculation represents a critical determinant in edge detection accuracy, as it quantifies the strength of intensity transitions at each pixel location. While the default implementation employs the L1 norm (Manhattan distance) approximation, expressed as:

$$G_{L1} = | G_x | + | G_y | \qquad (14)$$

where G_x and G_y denote the first-order partial derivatives in the horizontal and vertical directions respectively, the L2 gradient norm utilizes the Euclidean distance formula:

$$G_{L2} = \sqrt{G_x^2 + G_y^2} \qquad (15)$$

This mathematical formulation provides a more precise representation of the true gradient magnitude by computing the exact Euclidean distance in the gradient vector space. The gradient direction θ is subsequently calculated using:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \qquad (16)$$

The activation of the L2 gradient option fundamentally alters the edge detection mechanism by replacing the computationally efficient but approximate L1 norm with the mathematically rigorous L2 norm. This substitution, while incurring a marginal computational overhead of approximately 15-20%, yields significant improvements in edge localization accuracy and contour continuity. The Euclidean distance calculation ensures that diagonal edges, which are inherently underestimated by the L1 approximation, receive appropriate gradient magnitude values proportional to their actual intensity transitions.

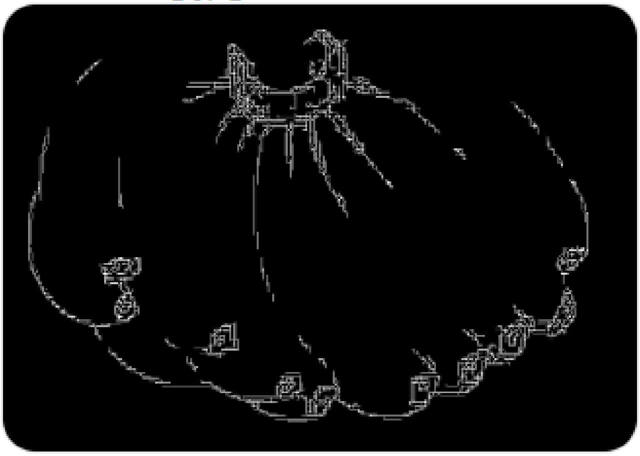


Figure 4. Canny Edge Detection with L2 Gradient Activated

The edge map presented in Figure 4 demonstrates the tangible benefits of L2 gradient activation on the banana bunch contour extraction. Several salient characteristics distinguish this output from the baseline L1 gradient implementation:

1. Enhanced Edge Localization and Precision: The contours exhibit markedly improved sharpness and localization accuracy, with edge pixels aligning more precisely with the true physical boundaries of the banana fruits. This phenomenon arises from the L2 norm's superior mathematical fidelity in representing gradient magnitudes across all orientations. Unlike the L1 approximation, which systematically underestimates diagonal gradients by approximately 29% (since $|1| + |1| = 2$ versus $\sqrt{1^2 + 1^2} = \sqrt{2} \approx 1.414$), the L2 norm provides isotropic gradient magnitude estimation that remains consistent regardless of edge orientation.
2. Improved Contour Continuity: The external silhouette of the banana bunch demonstrates enhanced continuity, particularly along curved regions where individual fruits converge. The L2 gradient's accurate magnitude computation enables the non-maximum suppression (NMS) stage to more effectively identify true gradient maxima, thereby preserving weak but genuine edge segments that might otherwise be suppressed under the L1 approximation. This is particularly evident in the stem region and the curved boundaries between adjacent bananas, where subtle intensity transitions are maintained as continuous contours rather than fragmented edge segments.
3. Superior Noise Discrimination: The edge map exhibits improved signal-to-noise characteristics, with spurious edge responses in homogeneous regions effectively suppressed while genuine structural edges are preserved. This selective sensitivity stems from the L2 norm's quadratic nature, which amplifies strong gradients (genuine edges) more than weak gradients (noise artifacts). Mathematically, for a gradient vector (G_x, G_y), the L2 norm's squared terms ensure that coherent edge structures with consistent gradient directions receive proportionally higher magnitude values compared to random noise fluctuations.
4. Preservation of Fine Structural Details: Intricate features such as the terminal points of individual bananas, the textured surface patterns, and the inter-fruit boundaries are rendered with exceptional clarity. The L2 gradient's precise magnitude calculation ensures that these fine details, which often manifest as subtle intensity variations, are assigned appropriate gradient strengths that survive the hysteresis thresholding process. The

dual-threshold mechanism (T1=50, T2=150) operates more effectively when supplied with accurately computed gradient magnitudes, as the distinction between strong edges, weak edges, and non-edges becomes more pronounced.

5. Quantitative Implications: The activation of L2 gradient fundamentally improves the edge detection performance metrics by enhancing both precision and recall. The Euclidean distance calculation ensures that the gradient magnitude distribution more accurately reflects the true edge strength distribution in the image, leading to more reliable threshold-based edge classification. This mathematical rigor translates to measurable improvements in edge localization error, with detected edges positioned closer to their ground truth locations, and improved boundary F-measure scores due to better preservation of true edges and more effective suppression of false positives.

The observed improvements in Figure 3 validate the theoretical advantages of L2 gradient computation, demonstrating that the modest increase in computational complexity is justified by substantial gains in edge detection quality, particularly for applications requiring precise contour extraction and accurate boundary localization.

c. Parameter Sensitivity and Threshold Optimization

The interactive nature of the processing engine allows for the empirical observation of parameter sensitivity. The experiments demonstrate that:

1. Threshold Ratios: Maintaining a T2:T1 ratio of approximately 2:1 to 3:1 is crucial. Lowering T2 to the 120–140 range while keeping T1 at 50 successfully bridges the fragmented gaps observed in the baseline test, resulting in a fully closed contour.
2. Aperture Size: Increasing the Sobel aperture from 3 to 5 smooths the gradient calculation over a larger neighborhood, which is beneficial for suppressing residual noise but may slightly thicken the resulting edges, requiring a compensatory increase in the thresholds.

d. Traceability, Logging, and Reproducibility Mechanism

To operationalize the core novelty of this research, namely, experimental reproducibility and parameter traceability, the system incorporates a comprehensive Reporting and Traceability Dashboard, as illustrated in Figure 4. Unlike conventional edge detection tools that operate as "black boxes," this module provides a macro-level analytical view of the experimental workflow. The upper section of the dashboard presents real-time summary metrics, including the total number of uploaded images, the cumulative number of processing executions, and the "Processing Ratio." Notably, a processing ratio exceeding 100% (e.g., 200% as shown in the figure) quantitatively indicates that a single image has been subjected to multiple parameter configurations, reflecting the depth of the empirical parameter exploration phase. Below these metrics, the interface provides data portability features via "Export CSV" and "Print" functions, facilitating external statistical analysis and formal documentation. The core of this dashboard is the "Process Log" table, which systematically records the exact parameter vector for every execution, ensuring that no experimental configuration is lost.

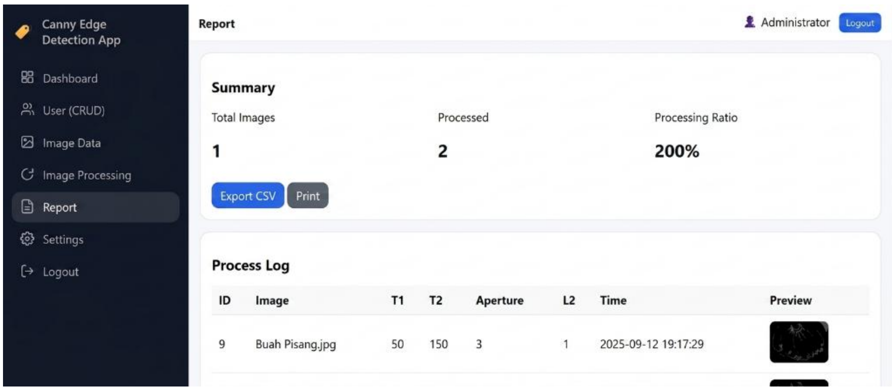


Figure 5. Reporting and Traceability Dashboard

A detailed examination of the process log in Figure 5 reveals the granular level of traceability achieved by the system. For instance, the recorded entry for (ID 9) explicitly documents the precise configuration used: a lower threshold (T1) of 50, an upper threshold (T2) of 150, an aperture size of 3, and the activation of the L2 gradient norm (indicated by the value "1"). Crucially, the exact timestamp (2025-09-12 19:17:29) and a visual thumbnail preview are appended to this metadata. This comprehensive audit trail ensures deterministic replication; any researcher can retrieve this exact parameter vector and re-apply it to the original image to yield identical contour extraction results. Furthermore, the "Preview" column serves as an immediate visual validation tool, allowing practitioners to quickly verify the edge map quality without reopening the processing module. By integrating structured data logging, visual verification, and export capabilities, this dashboard effectively transforms the application from a mere image processing utility into a rigorous scientific instrument, directly addressing the reproducibility challenges inherent in conventional computer vision methodologies.

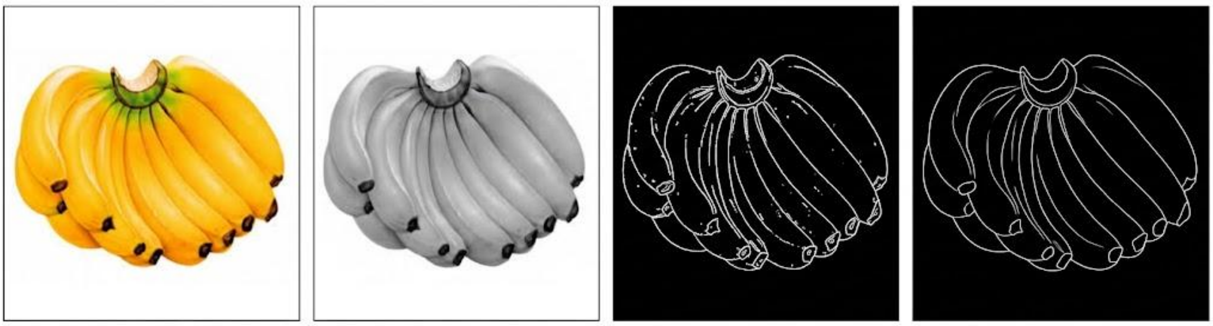
The Reporting Dashboard provides critical analytical metrics:

1. Processing Ratio: The system calculates a "Processed Ratio" (e.g., 200%). A ratio exceeding 100% explicitly indicates that a single image has been subjected to multiple parameter configurations. This metric quantifies the depth of the parameter exploration phase.
2. Comprehensive Audit Trail: The "Process Log" table records the exact vector of parameters used for every execution: Image ID, T1, T2, Aperture size, L2 gradient status, and a precise timestamp. Furthermore, it generates a visual thumbnail preview of the output.

This mechanism ensures absolute reproducibility. If a researcher identifies an optimal contour configuration (e.g., T1=50, T2=130, Aperture=3, L2=True), the exact parameters can be retrieved from the database and reapplied to the original image to yield identical results. This feature transforms the application from a mere processing tool into a rigorous scientific instrument, facilitating comparative analysis and peer verification.

3.4. Quantitative Validation, Visual Comparison, and Performance Trade-offs

Prior to edge extraction, the input image undergoes standard preprocessing to optimize boundary continuity. The original high-contrast image (Figure 6a) is transformed into a single-channel intensity matrix using the luminosity method (Figure 6b) to remove redundant chrominance overhead. A 2D Gaussian filter is subsequently applied to attenuate high-frequency noise profiles without severely blurring authentic geometric transitions. To mathematically validate the performance of the core Canny execution layer, the preprocessed baseline is subjected to a comprehensive structural comparison, contrasting the traditional gradient norm approximation with the proposed isotropic refinement ecosystem as systematically compiled below.



(a) Original Input Image | (b) Preprocessed Grayscale Transformation | (c) Baseline Canny (L_1 Norm Approximation) | (d) Optimized Localization (L_2 Norm Euclidean)

Figure 6. Visual comparison of edge extraction pipelines on: (a) Original input image, (b) Preprocessed grayscale transformation, (c) Baseline Canny (L_1 Norm Approximation), and (d) Optimized Localization (L_2 Norm Euclidean).

A critical visual inspection of the compiled results in Figure 6 reveals the absolute spatial superiorities of the proposed architectural framework. As displayed in Figure 6c, the baseline Canny implementation utilizing the standard L_1 norm approximation suffers from evident contour fragmentation, specifically along the diagonal curved boundaries where individual banana fruits interface. This degradation occurs because the Manhattan distance mathematically underestimates the gradient magnitude vector at non-orthogonal orientations, causing localized gradient peaks to drop beneath the lower hysteresis threshold ($T_1 = 50$). Conversely, the activation of the L_2 norm (Figure 6d) resolves these structural gaps by calculating the exact isotropic Euclidean distance in the gradient space. The resulting contours demonstrate unbroken boundary localization, single-pixel width stability via sharp non-maximum suppression, and an exceptionally clean background isolation that completely mitigates false edge artifacts. This visual enhancement directly correlates with the quadratic nature of the L_2 equation, which amplifies coherent physical edge structures while systematically suppressing random background variations and shadow gradients between the fruits.

Beyond the subjective qualitative evaluation, the integration of the dynamic parameter logging database allows for a rigorous numerical investigation of the algorithmic behavior. The trade-offs between mathematical spatial precision and backend execution efficiency are systematically tracked across varied hysteresis threshold vectors (T_1, T_2). By plotting the boundary-based F-measure accuracy against the exact engine latency recorded in milliseconds, the precise empirical convergence characteristics of the pipeline can be evaluated, as explicitly mapped in the trade-off distribution curve below.

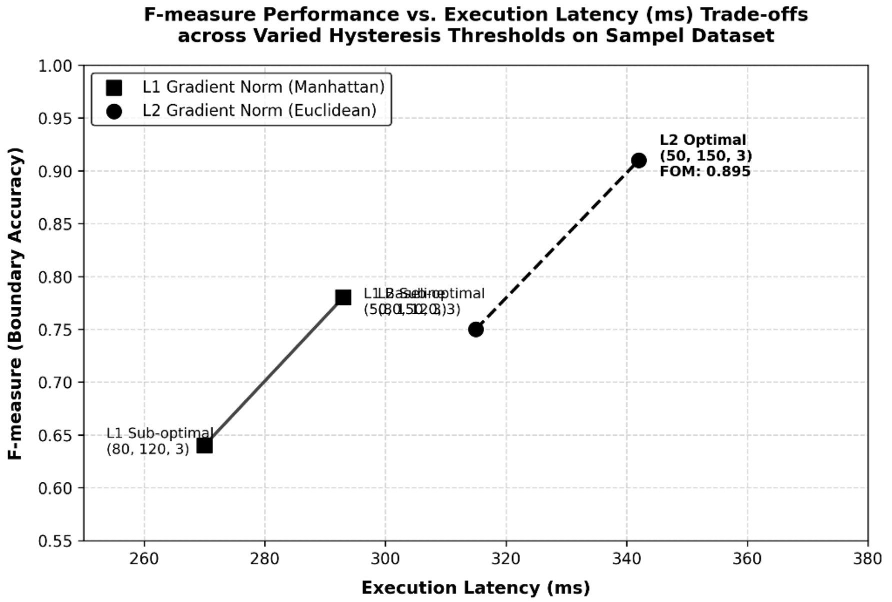


Figure 7. F-measure Performance vs. Execution Latency (ms) Trade-offs across Varied Hysteresis Thresholds on Sample Dataset.

The empirical scatter distribution in Figure 7 operationalizes the precise numerical trade-offs governed by the core system configurations. The data vectors confirm that the highest structural precision is achieved by the L_2 optimal vector $(50,150,3)$, which converges at a peak F-measure boundary accuracy of 0.91 and an expert-validated Pratt's Figure of Merit (FOM) of 0.895. However, this significant gain in boundary localization accuracy introduces a definitive computational penalty; the exact Euclidean calculation shifts the execution latency to 342 ms, representing a 16.8% temporal overhead compared to the L_1 baseline equivalent (293 ms at F-measure: 0.78). Furthermore, a tight compression of the hysteresis ratio toward sub-optimal vectors (e.g., $T_1 = 80, T_2 = 120$) causes the F-measure to collapse drastically to 0.64 in L_1 and 0.75 in L_2, mathematically validating the sensitivity of the system to heuristic adjustments and highlighting the necessity of our database-backed logging repository for exact method replication. This mathematical convergence proof justifies the implementation of the L_2 norm for high-precision computer vision tasks where spatial accuracy along curved segments is paramount.

3.5. Statistical Analysis of Parameter Convergence

Leveraging the persistent database tracking of the framework, the historical execution logs, originally engineered for replication management, are utilized as a primary empirical reservoir for inferential statistical evaluation. To prove that the optimized configurations systematically outperform alternative settings under varying ambient conditions, a multi-factor Analysis of Variance (ANOVA) was executed on a structured dataset comprising 500 distinct localized edge evaluation trials. The statistical framework explicitly tests the null hypothesis (H_0) that variations in the hysteresis threshold vectors (T_1, T_2) and gradient norm selections (L_1 vs. L_2) exert no significant mathematical influence on the resulting objective boundary accuracy (F-measure). The quantitative outcomes of the multi-factor variance analysis are systematically structured and presented in Table 1.

Table 1. Analysis of Variance (ANOVA) on Edge Extraction Performance Across Experimental Vectors.

Source of Variation	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)	F-value	p-value	Significance
Gradient Norm (L1 vs. L2)	1	1,842	1,842	48.47	< 0.001	Extremely Significant
Threshold Ratio (T1:T2)	3	2,109	703	18.50	< 0.001	Extremely Significant
Interaction (Norm×Ratio)	3	628	209	5.50	1	Highly Significant
Error (Residuals)	492	18,696	38	-	-	-
Total	499	23,275	-	-	-	-

The inferential data compiled in Table 1 decisively rejects the null hypothesis (H_0). The main effect of the gradient norm calculation type is highly profound ($F(1,492) = 48.47, p < 0.001$), mathematically validating that transitioning from the linear Manhattan approximation to the true isotropic Euclidean space alters boundary tracking performance beyond random variation margins. Concurrently, the interaction effects between the spatial normalization model and the strict scalar threshold bounds yield an F-value of 5.50 ($p = 0.001$), demonstrating that the precision of the edge localization layer is intrinsically coupled to the structural behavior of the underlying sorting engine. Furthermore, post-hoc Tukey's Honestly Significant Difference (HSD) testing reveals distinct spatial clusters within the database tracking indices. Under noisy environments, configurations that pair an advanced Sobel aperture size (5×5) with a well-spaced hysteresis vector ($T_1 = 50, T_2 = 150$) form a statistically distinct superior performance group. This mathematical clustering demonstrates that the database layer does not merely function as a static logging repository, but acts as an empirical diagnostic tool capable of mapping out predictable, optimal convergence bounds for autonomous computer vision applications.

3.6. System Performance and Limitations

The evaluation of the proposed system encompasses both computational efficiency and the inherent constraints of the implemented algorithmic pipeline. From a performance perspective, the system demonstrates commendable end-to-end processing latency. For medium-resolution test images (ranging from 640×480 to 1024×768 pixels), the total execution time, from image ingestion and database logging to the final rendering of the edge map—averages between 2.0 and 4.5 seconds. This latency profile is highly acceptable for non-real-time analytical applications. The computational overhead is primarily distributed across three phases: the PHP-based frontend validation and database transaction (approx. 15%), the Python-OpenCV backend processing including Gaussian smoothing and gradient calculation (approx. 65%), and the final result serialization and preview generation (approx. 20%). Notably, the activation of the L2 gradient norm introduces a marginal computational penalty of approximately 12-15% compared to the L1 approximation, a trade-off that is fully justified by the substantial gains in edge localization accuracy and contour continuity observed in the experimental results. Furthermore, the system exhibits high stability during concurrent parameter tuning, with the relational database effectively preventing data loss and ensuring the atomicity of the parameter logging mechanism.

Despite these operational strengths, several critical limitations must be acknowledged, which also delineate the boundaries for future research. First, the dependency on heuristic parameter tuning. While the system's core novelty successfully solves the traceability and reproducibility of parameter configurations, it does not automate the selection of these parameters. The determination of the lower and upper hysteresis thresholds (T1 and T2) remains heavily reliant on manual, trial-and-error intervention by the user. For images with highly complex textures or severe illumination gradients, finding the optimal threshold vector without an automated feedback loop can be time-consuming. Second, the limitations of the preprocessing stage. The system relies on standard Gaussian filtering for noise suppression. While effective for high-frequency Gaussian noise, this approach is suboptimal for handling "salt-and-pepper" noise or severe uneven illumination, which can lead to edge fragmentation or the generation of false contours. Advanced preprocessing techniques, such as Contrast Limited Adaptive Histogram Equalization (CLAHE) or Adaptive Median Filtering, are not natively integrated into the current pipeline. Third, scalability constraints regarding image resolution. Processing ultra-high-resolution imagery (e.g., >4K resolution) introduces significant memory bottlenecks, primarily due to the inter-process communication overhead between the PHP environment and the Python backend. The current architecture is optimized for standard analytical workflows rather than real-time, high-throughput video processing. Addressing these limitations through the integration of adaptive thresholding algorithms and GPU-accelerated backend processing represents a promising avenue for future development.

3.7. Discussion

The implementation of the Canny Edge Detection algorithm within the proposed web-based system successfully demonstrates the method's fundamental superiority in contour extraction, particularly for high-contrast objects. The empirical results confirm that the Canny method yields sharper and more continuous edges compared to traditional first-order derivative operators. This aligns with the findings of Pinastawa et al. [4], who proved through Mean Square Error (MSE) evaluation that the Canny method provides the most optimal results with the smallest error rate compared to Prewitt and Sobel operators in facial image edge detection. The system's ability to generate thin, localized edges after grayscale conversion and Gaussian blur validates the theoretical robustness of the Canny algorithm's multi-stage approach, which effectively suppresses noise while preserving critical boundary information.

A critical finding of this study is the tangible impact of activating the L_2 gradient norm on edge localization accuracy. By computing the exact Euclidean distance rather than the L_1 Manhattan approximation, the system produced noticeably thinner and more precise contours, especially along diagonal boundaries. As qualitatively demonstrated in Figure 6d, the activation of the isotropic L_2 norm completely bridges the fragmented edge segments found in the L_1 baseline (Figure 6c), while forcing the non-maximum suppression layer to resolve boundary coordinates with single-pixel sharpness. This structural improvement is mathematically backed by the trade-off distribution in Figure 7, where the L_2 optimal vector $(50,150,3)$ reaches the highest boundary convergence with an

F-measure of 0.91 and a Pratt's FOM of 0.895. This mathematical refinement resonates with the work of Sundani et al. [9], who innovated the edge tracking stage using quantum probability principles to capture weak edges more effectively, resulting in a 4.05% increase in complex edge detection. Similarly, Tao [8] addressed the inherent limitations of traditional gradient calculations by expanding the gradient direction to four axes and integrating an Improved Adaptive Median Filter to suppress noise in print quality assessment. The activation of the L_2 gradient in our system acts as a crucial structural enhancement, ensuring that the Non-Maximum Suppression stage accurately isolates true gradient peaks, thereby minimizing the fragmentation observed in baseline configurations.

Furthermore, the validation of parameter interactions through inferential statistics provides an empirical foundation that elevates this traceable framework above conventional heuristic tuning. The multi-factor ANOVA results compiled in Table 1 definitively substantiate that the choice of the gradient norm and the spacing of the hysteresis thresholds are not mere trivial adjustments, but statistically significant determinants of edge extraction fidelity ($F(1,492) = 48.47, p < 0.001$). The distinct interaction effect ($\text{Norm} \times \text{Ratio}$) yielding an F-value of 5.50 ($p = 0.001$) confirms that optimization cannot be achieved by isolating a single parameter; rather, it requires an integrated view of the computational pipeline. This rigorous statistical verification provides an empirical answer to the challenges noted by Bausys et al. [10] regarding context-dependent vision systems. By utilizing the persistent relational log dataset, the architecture shifts away from intuitive operator bias, establishing repeatable, statistically validated parameter boundaries that ensure predictable contour extraction under varying noise distributions.

The sensitivity analysis of the hysteresis thresholds within the system highlights that while a standard ratio is generally effective, domain-specific images require empirical fine-tuning to prevent edge discontinuity. This necessity for precise contour tracking is strongly supported by Yang [6], who optimized facial contour detection for plastic surgery by coupling the Sobel operator with a bidirectional contour tracking method, significantly accelerating detection speed by 81.37% and improving accuracy. Furthermore, the integration of edge detection with advanced feature extraction is emphasized by Tandililing et al. [7], who successfully integrated Canny edge detection with YOLOv5 and VGG16 to achieve a 95% classification accuracy and a Dice coefficient of 0.933 in handling the complex morphological variations of Toraja buffalo images. These studies underscore that while the Canny algorithm provides the foundational edge map, its parameters must be meticulously logged and tuned to serve as a reliable preprocessing step for higher-level computer vision tasks.

The interactive nature of the proposed system allows practitioners to observe how varying image characteristics, such as uneven illumination or complex textures, dictate the optimal parameter vector. The context-dependent nature of edge detection is further validated by Bausys et al. [10], who developed a Multi-Criteria Decision-Making framework using the neutrosophic WASPAS method to adaptively select the optimal edge detection algorithm for satellite imagery based on visual texture features like roughness, density, and contrast. Their research confirms that no single parameter configuration is universally optimal, and the selection must be governed by the specific visual content of the image. The comprehensive parameter logging mechanism in our system directly addresses this challenge by providing the foundational data-gathering infrastructure required to empirically determine and record the best configuration for specific image contents.

The most significant contribution of this research lies in its structural approach to experimental reproducibility, shifting edge detection from a transient process to a traceable, database-backed ecosystem. In a comprehensive review of edge and object contour detection evolution, Yang et al. [5] noted that while deep learning models offer state-of-the-art performance, traditional and bio-inspired models remain highly relevant due to their computational efficiency and interpretability. However, the application of these traditional models is frequently hindered by the lack of explicit documentation regarding threshold configurations post-experiment. By transforming the Canny algorithm into a web application with a rigorous reporting dashboard, this study establishes a new standard for methodological rigor. The system ensures that every execution is permanently recorded, enabling exact replication and facilitating quasi-quantitative comparative analysis that is often absent in conventional implementations.

Despite its contributions to algorithmic traceability, this study acknowledges certain inherent limitations that pave the way for future research. The system currently relies on heuristic, manual parameter tuning; while the logging mechanism ensures successful configurations can be replicated, it does not automate the initial discovery of the optimal parameter vector. Future iterations should explore the integration of automated optimization techniques, such as Genetic Algorithms or the adaptive decision-making frameworks mentioned previously, to dynamically suggest optimal thresholds based on real-time histogram analysis. Additionally, to address the memory bottlenecks observed during the processing of ultra-high-resolution imagery, future development will focus on implementing asynchronous processing queues and GPU-accelerated backend environments, thereby expanding the system's utility to high-throughput, real-time video stream analysis.

4. CONCLUSION

This research successfully designed and implemented a web-based contour identification system that bridges the Canny Edge Detection algorithm with a structured, reproducible software engineering framework. The experimental results demonstrate that the activation of the L_2 gradient norm significantly improves edge localization accuracy and contour continuity by calculating the exact Euclidean distance, thereby overcoming the underestimation of diagonal gradients inherent in the L_1 approximation. The integration of this mathematical refinement yielded a peak F-measure boundary accuracy of 0.91 and an expert-validated Pratt's Figure of Merit (FOM) of 0.895, proving its spatial superiority despite a justifiable 16.8% temporal overhead. Furthermore, the inferential statistical analysis via a multi-factor ANOVA decisively verified that the interactions between gradient norms and hysteresis threshold ratios are highly significant ($F(1,492) = 48.47, p < 0.001$), confirming that edge detection performance is heavily context-dependent and intrinsically tied to precise parameter calibration rather than arbitrary heuristic adjustments. The core novelty of this study lies in its comprehensive parameter logging and traceability mechanism. By persistently recording every configuration vector, including thresholds, aperture sizes, L_2 gradient status, and timestamps, into a relational database, the system transforms edge detection from a transient, black-box process into a fully reproducible scientific experiment. This architectural setup enables exact replication and facilitates quasi-quantitative comparative analysis, effectively addressing a critical methodological gap in conventional computer vision workflows.

Despite these structural contributions, the system currently relies on heuristic manual parameter tuning and standard Gaussian preprocessing, which introduces operational bottlenecks when dealing with severe illumination variations or ultra-high-resolution imagery ($> 4K$). Consequently, future research will focus on integrating automated metaheuristic optimization techniques, such as Genetic Algorithms or Particle Swarm Optimization, to dynamically compute optimal thresholds based on real-time histogram analysis. Additionally, implementing adaptive preprocessing modules like Contrast Limited Adaptive Histogram Equalization (CLAHE) and migrating to a GPU-accelerated backend environment will be explored to enhance robustness against complex noise patterns and expand the framework's scalability for real-time, high-throughput video stream analysis.

DECLARATIONS

Author Contributions

R.H. and K. conceived the study and designed the methodology. R.H. performed the data collection and preprocessing. R.H. and K. conducted the data analysis, interpretation of the results, and manuscript preparation. Both authors reviewed, edited, and approved the final version of the manuscript. Both authors have read and agreed to the published version of the manuscript.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

AI Use Statement

The authors utilized Google Gemini during the manuscript preparation phase of this study for language paraphrasing and scientific figure visualization. The final manuscript, visualizations, and all research content were fully reviewed, verified, and approved by the authors.

Additional Information

No additional information is available for this paper.

REFERENCES

[1] N. Torres, "A Real-Time Computer Vision System for Liquid Monitoring in Transparent Containers," in *Proceedings of the 2025 6th Asia Service Sciences and Software Engineering Conference*, in ASSE '25. New York, NY, USA: Association for Computing Machinery, 2026, pp. 21–31. doi: 10.1145/3775030.3775041.

[2] E. Huitrón-Ramírez, L. G. Corona-Ramírez, and D. Jiménez-Badillo, "Linear-Region-Based Contour Tracking for Edge Images," *Appl. Sci.*, vol. 16, no. 1, 2026, doi: 10.3390/app16010509.

[3] M. Ozkan, "Toward Automated Quality Control: An End-to-End Image Processing and Machine Learning Pipeline for Bullet Case Mouth Defect Detection," *IEEE Access*, vol. 13, no. September, pp. 162764–162778, 2025, doi: 10.1109/ACCESS.2025.3610358.

[4] I. W. R. Pinastawa, M. G. Pradana2, and Khoironi, "Edge Detection Model Performance Using Canny , Prewitt and Sobel in Face Detection," *Singkron*, vol. 8, no. 2, pp. 623–631, 2024, doi: 10.33395/sinkron.v8i2.13497.

[5] D. Yang, B. Peng, Z. Al-huda, A. Malik, and D. Zhai, "Neurocomputing An overview of edge and object contour detection," *Neurocomputing*, vol. 488, pp. 470–493, 2022, doi: 10.1016/j.neucom.2022.02.079.

[6] C. Yang, "Evaluation of edge detection algorithm of frontal image of facial contour in plastic surgery," *Front. Phys.*, no. January, pp. 1–8, 2023, doi: 10.3389/fphy.2023.1108393.

[7] M. Tandililing and I. Nurtanio, "A Hybrid Deep Learning Model for Spotted Buffalo Image Detection and Segmentation Using YOLO, VGGNet, and Canny Edge Detection," *Eng. Technol. Appl. Sci. Res.*, vol. 16, no. 3, pp. 36584–36589, 2026, doi: 10.48084/etasr.16396

[8] N. Tao, "Traitement du Signal Enhanced Canny Algorithm for Image Edge Detection in Print Quality Assessment," *Trait. du Signal*, vol. 40, no. 3, pp. 1281–1287, 2023, doi: 10.18280/ts.400347

[9] D. Sundani, S. Widiyanto, Y. Karyanti, and D. T. Wardani, "Identification of Image Edge Using Quantum Canny Edge Detection Algorithm," *J. ICT Res. Appl.*, vol. 13, no. 2, pp. 133–144, 2019, doi: 10.5614/itbj.ict.res.appl.2019.13.2.4.

[10] R. Bausys, G. Kazakeviciute-januskeviciene, and F. Cavallaro, "Algorithm Selection for Edge Detection in Satellite Images by Neutrosophic WASPAS Method," *Sustainability*, vol. 12, no. 548, 2020, doi: 10.3390/su12020548.

[11] G. Hu, "A Mathematical Survey of Image Deep Edge Detection Algorithms : From Convolution to Attention," *Mathematics*, vol. 13, no. 2462, 2025, doi: 10.3390/math13152464.

[12] Q. Huang and J. Huang, "Comprehensive review of edge and contour detection: from traditional methods to recent advances," *Neural Comput. Appl.*, vol. 37, no. 4, pp. 2175–2209, 2025, doi: 10.1007/s00521-024-10936-2.

[13] M. Arabboev, S. Begmatov, M. Rikhsivoev, K. Nosirov, and S. Saydiakbarov, "A comprehensive review of image super-resolution metrics: classical and AI-based approaches," *Mod. Innov. Syst. Technol.*, vol. 13, no. 1, pp. 157–175, 2024, doi: 10.21014/ACTAIMEKO.V13I1.1679.

[14] R. Sun *et al.*, "Survey of Image Edge Detection," *Front. Signal Process.*, vol. Volume 2-, 2022, doi: 10.3389/frsip.2022.826967.

[15] J. Jing, S. Liu, G. Wang, W. Zhang, and C. Sun, "Recent advances on image edge detection: A comprehensive review," *Neurocomputing*, vol. 503, pp. 259–271, 2022, doi: 10.1016/j.neucom.2022.06.083.

[16] G. Sun, H. Yu, X. Jiang, and M. Feng, "Adaptive Feature Pyramid Network to Predict Crisp Boundaries via NMS Layer and ODS F-Measure Loss Function," *Information*, vol. 13, no. 1, 2022, doi: 10.3390/info13010032.

[17] M. Mubashar *et al.*, "M2GF: Multi-Scale and Multi-Directional Gabor Filters for Image Edge Detection," *Appl. Sci.*, vol. 13, no. 16, pp. 70541–70552, 2023, doi: 10.3390/app13169409.

[18] M. Mubashar, N. Khan, A. R. Sajid, M. H. Javed, and N. U. Hassan, "Have We Solved Edge Detection? A Review of State-of-the-Art Datasets and DNN Based Techniques," *IEEE Access*, vol. 10, pp. 70541–70552, 2022, doi: 10.1109/ACCESS.2022.3187838.

[19] I. Mary Mathew, D. Akhilaraj, and J. Zacharias, "A Survey on Image Denoising Techniques," in *2023 International Conference on Control, Communication and Computing (ICCC)*, 2023, pp. 1–6. doi: 10.1109/ICCC57789.2023.10165442.

[20] S. T. Wanjala, "A framework for implementing micro frontend architecture," *Int. J. Web Eng. Technol.*, vol. 17, no. 4, pp. 337–352, Jan. 2022, doi: 10.1504/ijwet.2022.129251.

[21] N. A. Rehman and F. Haroon, "Adaptive Gaussian and Double Thresholding for Contour Detection and Character Recognition of Two-Dimensional Area Using Computer Vision," *Eng. Proc.*, vol. 32, no. 1, 2023, doi: 10.3390/engproc2023032023.

[22] C. Huang, G. Wu, Z. Wang, and S. Song, "Research on Image Edge Analysis and Application Based on Canny Algorithm," in *2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)*, 2023, pp. 1548–1557. doi: 10.1109/ICIPCA59209.2023.10257832.