Data Hiding and Retrieval Using Edge-Guided LSB Image Steganography

Waqar Ishaq[1], Bilal Ur Rehman[2]*, Syed Waqar Shah[2], Mir Muhammad Wasiq[1]
[1]Department of Telecommunication Hazara University, Mansehra, KPK, Pakistan
[2]Department of Electrical Engineering, University of Engineering and Technology, Peshawar, KPK, Pakistan.
***Correspondence:** bur@uetpeshawar.edu.pk.

Citation | Ishaq. W, Rehman. B. U, Shah. S. W, Wasiq. M. M, "Data Hiding and Retrieval Using Edge-Guided LSB Image Steganography", IJIST, Vol. Vol. 8 Issue.9 pp 3567-3577, October 2026
DOI:
Received | Sep 03, 2026 **Revised|** Oct 02, 2026 **Accepted|** Oct 12, 2026 **Published|** Oct 16, 2026

This paper presents an edge-guided Least-Significant-Bit (LSB) method for hiding and retrieving text in Red-Green-Blue (RGB) images. A common embedding mask is generated through grayscale conversion, noise suppression, and fusion and binarization of Canny and Laplacian-of-Gaussian edge maps. The secret text is converted into a binary stream and embedded in selected low-order bit positions at edge locations across the color channels. The channel-processing stage incorporates discrete cosine transform and inverse-transform operations. The method is evaluated using 24 Kodak images, 500 BSD500 images, and an 8,104-image subset of IStego 100K. The average Peak Signal-To-Noise Ratios (PSNRs) are 70.1028, 70.0841, and 71.5475 dB, respectively, with corresponding mean squared errors of 0.0065, 0.0066, and 0.0051. Benchmark structural similarity index (SSIM) values round to 1.000 at three decimal places. Bit-plane sensitivity tests show that, for a 96-bit message, average PSNR increases from 81.3219 dB with four low-order bit positions to 86.4936 dB with two. These results indicate low image distortion under the tested conditions and demonstrate the influence of bit-plane selection on visual fidelity.

Keywords: Canny Edge Detection, Data Hiding, Image Steganography, Laplacian Of Gaussian, Least Significant Bit, MSE, PSNR, RGB Images, SSIM.

Introduction:

Digital communication routinely carries private text, credentials, and other sensitive information through channels modified by unauthorized parties. Cryptography protects message content, whereas steganography additionally attempts to conceal the existence of the message by embedding it in an ordinary-looking carrier. Recent surveys describe image steganography as a balance among imperceptibility, payload, robustness, and security rather than a single-metric optimization problem [1][2][3]. Common image-based approaches and their operational tradeoffs are reviewed in [4][5][6].

A digital image is a practical cover medium because its large number of samples provides many candidate embedding locations. Nevertheless, direct modification of smooth regions can create regular artifacts that are easier to observe or detect statistically. Introductory and survey studies therefore distinguish spatial-domain methods, which directly alter pixels, from transform-domain methods, which modify coefficients after a transformation [7][8][9]. Within the spatial domain, LSB replacement remains attractive because it is computationally simple and can offer high capacity, but predictable embedding paths and excessive bit-plane use can reduce security or visual quality.

Edges offer a useful compromise. Their naturally high local variation can mask small sample changes more effectively than uniform regions. The proposed method uses a fused Canny and Laplacian-of-Gaussian (LoG) mask to select edge pixels, separates the RGB image into three channels, converts the message to binary form, and schedules message bits across low-order positions. The principal contributions are: (1) a unified edge mask that expands the pool of eligible pixels; (2) multichannel embedding with an explicit text-to-binary and binary-to-text path; (3) evaluation on three image collections with PSNR, MSE, and SSIM; and (4) controlled tests that expose the tradeoff among payload, number of modified bit positions, and image quality.

Novelty of the Proposed Method:

The novelty of the proposed approach lies in the coordinated integration of complementary edge detection, multichannel LSB allocation, and transform-assisted channel processing rather than in any individual component. Unlike conventional edge-guided methods that rely on a single edge detector or a specific edge-selection strategy, the proposed method fuses Canny and Laplacian-of-Gaussian (LoG) responses to construct a common spatial embedding mask. This fused mask is applied consistently across the RGB channels, while the secret-message bits are distributed through scheduled low-order bit positions instead of being embedded through a single fixed channel or unrestricted pixel sequence. In addition, DCT and inverse-DCT processing are incorporated within the channel-processing stage while the actual message-placement mechanism remains edge-guided and LSB based. Consequently, the proposed framework unifies edge fusion, multichannel payload distribution, transform processing, and reversible text-to-binary/binary-to-text handling within a single end-to-end embedding and retrieval architecture. The contribution should therefore be understood as an integrated steganographic framework designed to improve embedding flexibility and visual fidelity rather than as the introduction of a new edge detector, transform, or LSB operation.

Related Work:

Spatial- and Transform-Domain Steganography:

Spatial-domain methods modify pixel values directly, while transform-domain schemes embed in coefficients produced by transforms such as the discrete cosine or wavelet transform. Classification studies emphasize that spatial methods are usually simple and capacious, whereas transform methods can offer different robustness properties at the cost of additional computation [10][11][12]. Reversible schemes add the stronger requirement that the original cover be recovered exactly after message extraction; a survey of this class is provided in [13]. The present method is primarily an edge-guided LSB scheme but includes DCT/IDCT

processing in the proposed implementation, making clear specification of the embedding domain important for reproducibility.

Security-Oriented and Adaptive LSB Methods:

Steganography is also frequently combined with cryptography so that extraction alone does not reveal plaintext. Adee and Mouratidis integrated encryption and LSB hiding into a layered cloud-security model [14]. [15] combined 3-3 LSB embedding with RC4 and a Bernoulli map, while chaos-assisted edge selection has been used to randomize or protect embedding locations [16]. Visual-sensitivity-based adaptation can further vary the allocation across RGB channels and spatial regions [17]. These designs motivate keyed selection and encrypted payloads, although the method evaluated here focuses on the embedding and retrieval mechanism rather than a complete cryptographic protocol.

Edge-Guided and Compressed-Image Methods:

Edge-guided embedding has been explored using DCT-domain edge filters [2], hybrid edge detection and chaos [18], Kirsch masks [19], and variable thresholds [20]. The common rationale is that locally complex pixels tolerate larger changes than smooth pixels. Other approaches target compressed representations: [21] proposed vector-quantization data hiding for Internet-of-Things networks, [22] used Hamming distance and pixel-value differencing in AMBTC-compressed images, and [23] adjusted AMBTC pixel values to increase capacity. Because these methods use different payloads, images, and distortion conventions, cross-paper comparisons should be interpreted as indicative rather than as controlled head-to-head experiments.

Evaluation Practice and Research Gap:

Existing studies have investigated several related directions independently, including DCT-domain edge-based embedding, chaos-assisted edge selection, Kirsch-based edge detection, variable-threshold edge selection, and compressed-image data hiding [24] [25][26][27][28][29][30]. However, these approaches differ in their edge-selection mechanisms, embedding domains, payload handling, and processing architectures. The present study addresses a more specific integration gap by combining Canny–LoG edge fusion with a common spatial mask, multichannel RGB LSB allocation, scheduled low-order bit modification, DCT/IDCT-assisted channel processing, and an explicit message-retrieval path within one framework. Accordingly, the principal contribution is not the independent use of Canny, LoG, DCT, or LSB techniques, all of which are established methods, but their coordinated use in an end-to-end architecture and the experimental assessment of its fidelity, capacity, and bit-plane sensitivity across multiple image datasets. The resulting PSNR, MSE, and SSIM measurements characterize imperceptibility under the tested conditions; they are not presented as evidence of complete steganalytic security.

Proposed Edge-Guided LSB Method:

System Overview:

The method has two complementary phases. During embedding, the cover image is preprocessed to produce an edge mask, the RGB channels are separated, and the secret text is converted to a binary stream. Eligible edge locations receive message bits in scheduled low-order positions, after which the channels are returned to the image domain and recombined. In contrast to methods in which edge detection, transform-domain embedding, or channel-wise LSB modification is applied as an isolated mechanism, the proposed architecture coordinates these operations through a single fused edge mask and a defined multichannel embedding/retrieval sequence. This integration constitutes the main methodological contribution of the proposed framework. As shown in Figure 1, pixel selection, image processing, and text preparation converge at the embedding stage.

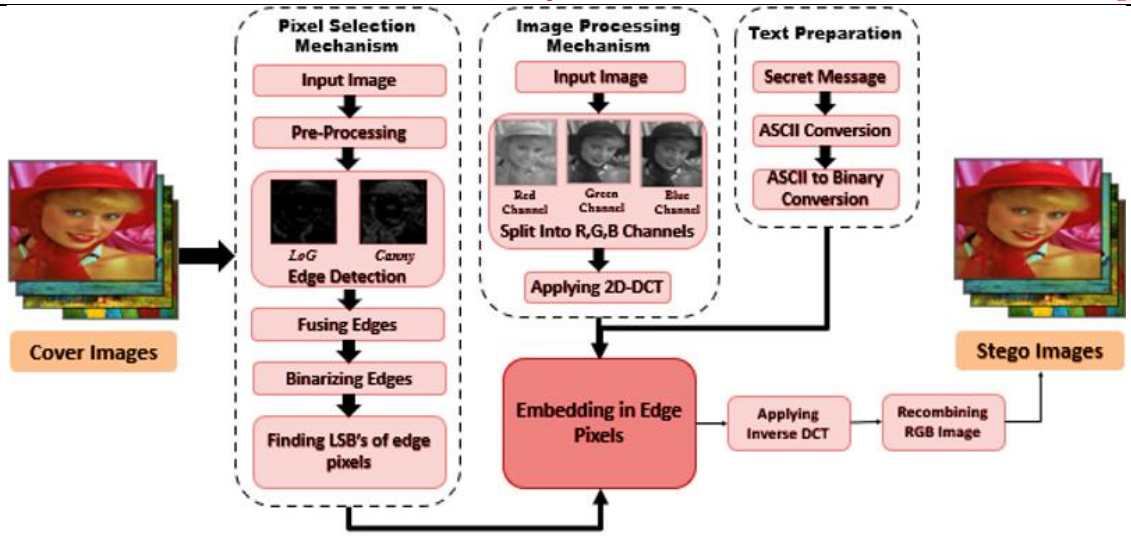

Figure 1. Proposed embedding architecture for edge-guided multichannel LSB steganography.

Preprocessing and Edge-Mask Generation:

Let the input cover image be denoted by I, with red, green, and blue components R, G, and B, respectively. To obtain a common spatial representation for edge detection across all three color channels, the RGB cover image is first converted into a grayscale image using the luminance-weighted relationship

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

The grayscale image is then smoothed using a two-dimensional Gaussian filter to suppress isolated noise responses that could otherwise produce unstable or spurious edges. For spatial offsets $x$and $y$and standard deviation σ, the Gaussian kernel is defined as

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

Here, σcontrols the degree of smoothing applied before edge detection. The smoothed grayscale image is subsequently processed using both the Canny and Laplacian-of-Gaussian (LoG) edge detectors. Canny detection provides thin and threshold-connected edge structures, whereas LoG identifies intensity transitions through second-derivative responses after smoothing. The two binary edge maps are combined using a logical OR operation:

$$E_{\text{fuse}}(x, y) = E_{\text{Canny}}(x, y) \vee E_{\text{LoG}}(x, y), \quad (3)$$

where $E_{\text{Canny}}(x, y)$and $E_{\text{LoG}}(x, y)$represent the binary outputs of the respective edge detectors. Thus, a pixel is retained in the fused edge map when it is detected by either Canny or LoG. This complementary fusion increases the pool of candidate embedding locations while maintaining the edge-guided nature of the proposed method.

The fused response is then converted into a binary embedding mask $M(x, y)$. Using an edge-positive threshold T,

$$M(x, y) = \begin{cases} 1, & E_{\text{fuse}}(x, y) \geq T, \\ 0, & E_{\text{fuse}}(x, y) < T. \end{cases}$$

Pixels for which $M(x, y) = 1$are treated as eligible locations for message embedding, whereas pixels with $M(x, y) = 0$are excluded. The detector parameters and threshold used for mask construction must remain consistent between embedding and retrieval because the receiver regenerates the edge mask during message extraction. Any disagreement in the regenerated mask can alter the sequence of eligible pixels and consequently affect correct message recovery.

Channel Processing and Message Embedding:

After generation of the common fused-edge mask, the cover image is separated into its red, green, and blue channels. The proposed workflow incorporates two-dimensional discrete cosine transform (DCT) processing within the designated channel-processing path before embedding, followed by the corresponding inverse DCT operation before final RGB recombination. The actual payload-placement mechanism remains edge-guided and based on selected low-order bit positions at the locations identified by the common embedding mask. This organization allows the edge-selection, multichannel processing, and message-embedding stages to operate within a unified framework.

The secret text is first converted into its corresponding character codes and then represented as an ordered binary sequence

$$b = \{b_1, b_2, \ldots, b_L\},$$

where Ldenotes the total number of message bits and $b_m \in \{0, 1\}$represents the m-th bit of the secret message.

For an eligible sample value p, the selected message bit $b_m$is inserted into a low-order bit position k. The replacement operation can be expressed as

$$p' = [p \text{ AND } \text{NOT}(2^k)]\text{OR}(b_m 2^k),\ k \in \{0, 1, 2, 3\}, \qquad (4)$$

where $p'$is the modified sample value, $p$is the original sample value, $b_m$is the binary message bit being embedded, and $k$identifies the selected low-order bit position.

Rather than modifying all four low-order positions simultaneously in every eligible sample, the proposed implementation schedules the available low-order positions and distributes message bits across the edge pixels and available RGB channels. Embedding continues according to this predefined order until the complete payload has been inserted. Restricting modification to edge-selected locations exploits regions of naturally higher local intensity variation, where small changes are generally less visually noticeable than in smooth areas.

The same channel order, eligible-pixel sequence, and bit-position schedule used during embedding must also be available during retrieval. In an operational implementation, a message-length header or another explicit termination mechanism is therefore required so that the receiver can determine when the complete payload has been recovered. In the present work, such framing is treated as implementation metadata and is not included in the reported image-quality calculations, consistent with the existing methodology. Figure 2 illustrates the red-channel path from RGB separation and fused-edge mapping to the replacement of selected binary positions.

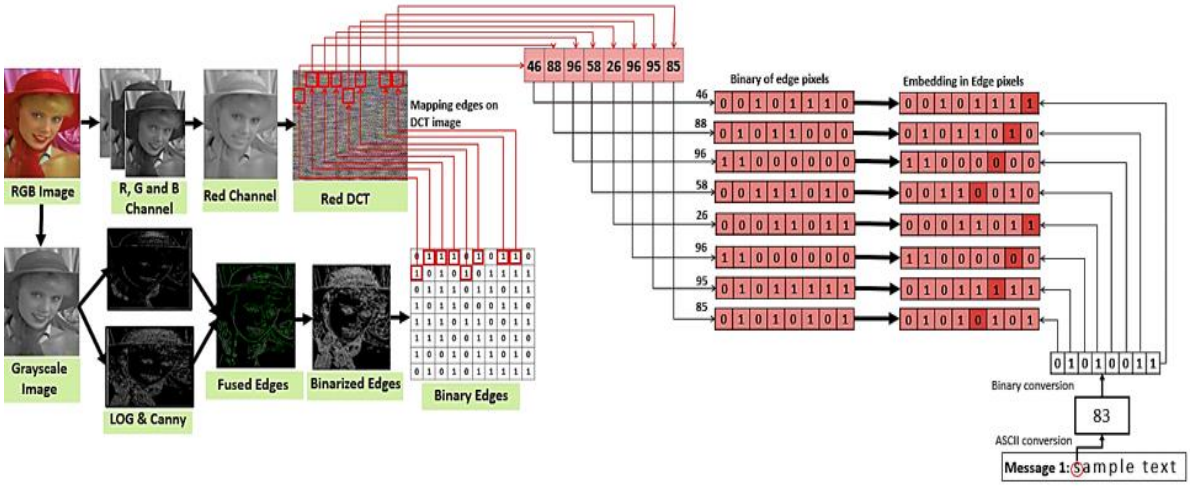

Figure 2. Detailed edge mapping and LSB replacement for one RGB channel.

Stego-Image Formation and Retrieval:

After embedding, inverse DCT processing is applied where required by the channel path, and the three processed channels are concatenated to form the stego image. Retrieval reverses the procedure. The stego image is converted to grayscale for mask regeneration, Canny and LoG edges are fused and binarized, and the RGB channels are separated. The scheduled low-order positions at eligible locations are read in the same order, grouped into character codes, and converted back to text. The receiver-side sequence is summarized in Figure 3.

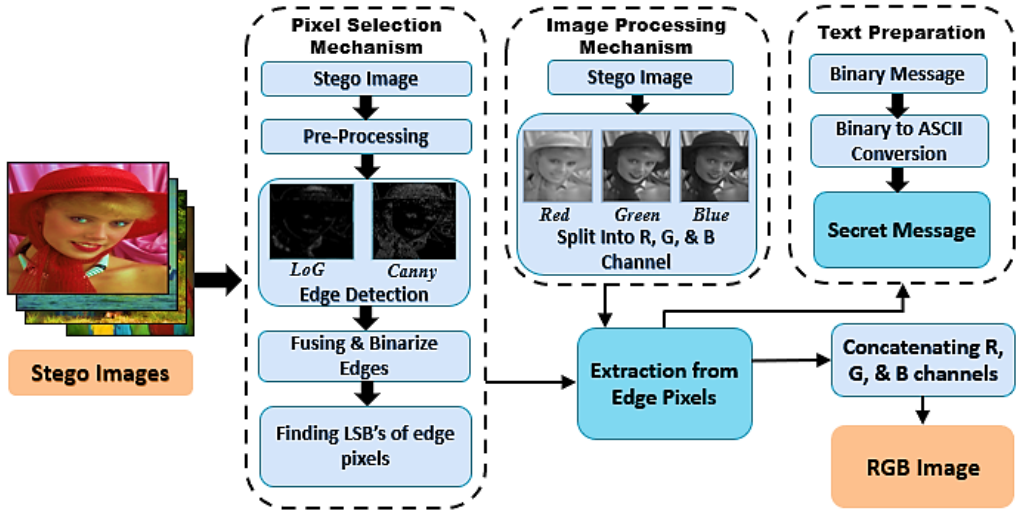

Figure 3. Proposed extraction and message-retrieval architecture.

Correct recovery assumes that the embedding order, channel schedule, bit-plane schedule, transform parameters, and edge mask can be reproduced at the receiver. Any disagreement changes the readout positions and can corrupt the recovered message.

Experimental Design:

Image Datasets:

The evaluation uses three RGB image collections with different resolutions and content diversity. Table 1 summarizes the datasets used to evaluate the proposed method. The Kodak collection contains 24 images in landscape and portrait orientations, with dimension of 768 × 512 or 512 × 786 pixels. BSD 500 contains 500 images at 481 × 321 or 321 × 481. For IStego 100K, the experiment conducted in this study uses 8,104 images of 1024 × 1024 pixels, which is described here as the evaluated subset rather than the full nominal collection.

Table 1. Datasets used in simulation.

Dataset	Color space	Image dimensions (pixels)	Format	Images used
Kodak	RGB	768 × 512 or 512 × 768	JPEG	24
BSD 500	RGB	481 × 321 or 321 × 481	JPEG	500
IStego 100K subset	RGB	1024 × 1024	JPEG	8,104

Fidelity Metrics:

Let X and Y denote the cover and stego images, respectively, each containing M × N samples for the evaluated channel representation. MSE measures the average squared sample difference:

$$MSE = (1/MN)\ \Sigma_{i=1}^{M}\ \Sigma_{j=1}^{N}\ [X(i,j) - Y(i,j)]^2 \quad (5)$$

For 8-bit samples, PSNR in decibels is computed from the same error term as

$$PSNR = 10\ \log_{10}(255^2/MSE \quad (6)$$

SSIM compares local luminance, contrast, and structure. With means μ_x and μ_γ, standard deviations σ_x and σ_γ, covariance $\sigma_{x\gamma}$, and stabilizing constants C_1 and C_2,

$$SSIM(X,Y) = [(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)] / [(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)] \quad (7)$$

Higher PSNR and SSIM and lower MSE indicate smaller image distortion. These measures quantify fidelity but do not, by themselves, establish secrecy or resistance to a trained steganalyzer. Payload is also reported alongside the fidelity measure where applicable.

Results and Discussion:

Aggregate Dataset Results:

Table 2 reports the aggregate PSNR and MSE values for the three evaluated collections. IStego 100K produces the highest average PSNR, 71.5475 dB, and the lowest MSE, 0.0051. Kodak and BSD 500 give closely matched results near 70.1 dB. The consistency across different resolutions suggests that the visual fidelity is not confined to one image size. Aggregate SSIM was not separately reported for all three datasets, so the table does not infer missing values.

Table 2. Aggregate Image-Quality Results by Dataset.

Dataset	Images	Average PSNR (dB)	Average MSE	Interpretation
Kodak	24	70.1028	0.0065	High fidelity
BSD 500	500	70.0841	0.0066	High fidelity
IStego 100K subset	8,104	71.5475	0.0051	Best aggregate

Comparison With Prior PSNR Results

The six-image comparison in Table 3 places the proposed method against the hybrid edge/chaos method of [18], the AMBTC method of [22], and the AMBTC pixel-adjustment method of [23]. The proposed values range from 68.2175 dB for Baboon to 72.6574 dB for Peppers, with an average of 70.8734 dB. The comparison favors the proposed method numerically, but differences in payload, images, preprocessing, and evaluation conditions prevent the table from serving as a controlled statistical ranking.

Table 3. PSNR Comparison on Six Benchmark Images.

Image	[18]	[22]	[23]	Proposed method
Lena	43.71	29.05	33.10	71.8357
Baboon	41.94	23.85	26.94	68.2175
Peppers	44.20	23.85	33.30	72.6574
Airplane	43.54	27.92	31.90	71.3229
Sailboat	43.59	—	29.81	70.6165
Barbara	43.45	28.57	—	70.7791
Average	43.97	27.74	31.01	70.8734

Capacity and Imperceptibility:

Table 4 compares embedding capacity with the vector-quantization method of [21]. The experimental results use a constant proposed capacity of 763,578 bits for each benchmark image and report an average PSNR of 70.6413 dB. The comparative method reports an average capacity of 515,602 bits and an average PSNR of 32.860 dB. These values indicate a favorable capacity-quality combination within the recorded experiment, subject again to differences in algorithmic assumptions and test configuration.

Table 4. Embedding-Capacity and PSNR Comparison.

Image	[21] capacity (bits)	[21] PSNR (dB)	Proposed capacity (bits)	Proposed PSNR (dB)
Lena	454,996	34.857	763,578	71.8150
Peppers	461,742	33.573	763,578	72.7824
Baboon	704,569	27.549	763,578	67.6832
Boat	490,188	33.208	763,578	70.3443
Goldhill	552,546	34.248	763,578	69.7951

Airplane	429,569	33.728	763,578	71.4278
Average	515,602	32.860	763,578	70.6413

Stress Testing and Payload Sensitivity:

The bit-plane stress test isolates the effect of using two, three, or four low-order positions for the same 96-bit message. Table 5 shows that reducing the number of modified positions increases average PSNR from 81.3219 dB for four positions to 86.4936 dB for two. The direction of this change is consistent for all six test images and confirms that lower modification depth improves fidelity.

Table 5. PSNR under two-, three-, and four-position LSB schedules.

Test image	Payload (bits)	Four positions (dB)	Three positions (dB)	Two positions (dB)
Test 1	96	81.4645	84.0771	86.7132
Test 2	96	80.9174	83.0865	86.7474
Test 3	96	80.4978	82.4535	85.0826
Test 4	96	82.7738	84.9655	88.0565
Test 5	96	81.2328	83.4325	86.4804
Test 6	96	81.0452	83.0050	85.8817
Average	96	81.3219	83.5033	86.4936

A separate payload-length experiment on a 512 × 512 Lena image provides the complementary trend. With two low-order positions, increasing the payload from 96 to 2,280 bits reduces PSNR from 86.7132 to 71.8721 dB and increases MSE from 0.0001 to 0.0042. Intermediate payloads of 288, 360, 552, and 1,128 bits produce PSNR values of 81.0236, 79.8201, 77.9229, and 74.8707 dB, respectively. Thus, capacity is not free: even when the stego image remains visually convincing, the measured distortion rises with payload.

Interpretation of SSIM and Visual Evidence:

For the benchmark images, SSIM is obtained as 1.000 to three decimal places, while MSE remains nonzero. These two facts are compatible only when the SSIM values are rounded; they should not be interpreted as exact pixelwise identity. Histogram inspection in the conducted experiments likewise shows no obvious visual separation between cover and stego histograms. Such evidence supports imperceptibility at the displayed scale, but histogram similarity is a weak security test because modern steganalysis can exploit higher-order spatial dependencies that are not visible in a global histogram.

Limitations and Future Work:

The evaluation has five main limitations. First, it measures image fidelity and nominal capacity but does not report detection error against statistical or learning-based steganalyzers. Second, robustness is not tested under JPEG recompression, resizing, cropping, noise, filtering, color conversion, or transmission errors. Third, extraction regenerates an edge mask from the stego image; even small edge-map changes may desynchronize the payload, yet the bit-error rate under such changes is not obtained. Fourth, the exact DCT channel assignment, coefficient quantization, payload framing, and end-of-message rule require fuller implementation-level specification. Fifth, the cross-paper tables combine values obtained under potentially different payloads and experimental conditions, limiting causal conclusions. Future work should therefore define a keyed pseudorandom traversal of eligible pixels, encrypt the message before embedding, and add payload-length framing and error-correcting codes. Detector thresholds and bit-plane allocation should be adapted to local texture and payload demand rather than fixed globally. Security evaluation should include established spatial and rich-model steganalyzers, receiver-operating-characteristic curves, and cross-dataset tests. Robustness experiments should report extraction bit-error rate after common image operations, while ablation studies should isolate the contributions of Canny detection, LoG

detection, fusion, DCT processing, and multichannel allocation. Public code, fixed seeds, and complete parameter settings would make the high PSNR values independently reproducible.

Conclusion:

This study proposed an edge-guided multichannel LSB steganography method integrating Canny-LoG edge fusion with DCT/IDCT-assisted channel processing. The method was evaluated using Kodak, BSD500, and an IStego 100K subset, achieving average PSNR values of 70.1028, 70.0841, and 71.5475 dB, respectively, with very low MSE values. The results demonstrate high visual fidelity, while the payload-sensitivity analysis confirms that image quality decreases as embedding depth and payload increase. Overall, the proposed approach provides effective imperceptible data hiding under the tested conditions. Future work will focus on steganalysis, robustness testing, extraction error analysis, and improved synchronization to further enhance the security and practical reliability of the method.

Reference:

[1] P. C. Mandal, I. Mukherjee, G. Paul, and B. N. Chatterji, "Digital image steganography: A literature survey," *Inf. Sci. (Ny).*, vol. 609, pp. 1451–1488, Sep. 2022, doi: 10.1016/j.ins.2022.07.120.

[2] N. Ayub and A. Selwal, "An improved image steganography technique using edge based data hiding in DCT domain," *J. Interdiscip. Math.*, vol. 23, no. 2, pp. 357–366, Feb. 2020, doi: 10.1080/09720502.2020.1731949.

[3] C. S. Fernandes, "Steganography and Computer Forensics - the art of hiding information: a systematic review," *ARIS2 - Adv. Res. Inf. Syst. Secur.*, vol. 2, no. 2, pp. 31–38, Dec. 2022, doi: 10.56394/aris2.v2i2.20.

[4] S. Atawneh, A. Almomani, and P. Sumari, "Steganography in digital images: Common approaches and tools," *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)*, vol. 30, no. 4, pp. 344–358, Jul. 2013, doi: 10.4103/0256-4602.116724.

[5] O. Evsutin, A. Melman, and R. Meshcheryakov, "Digital steganography and watermarking for digital images: A review of current research directions," *IEEE Access*, vol. 8, pp. 166589–166611, 2020, doi: 10.1109/ACCESS.2020.3022779.

[6] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image Steganography: A Review of the Recent Advances," *IEEE Access*, vol. 9, pp. 23409–23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[7] A. A. J. Altaay, S. Bin Sahib, and M. Zamani, "An introduction to image steganography techniques," *Proc. - 2012 Int. Conf. Adv. Comput. Sci. Appl. Technol. ACSAT 2012*, pp. 122–126, 2012, doi: 10.1109/ACSAT.2012.25.

[8] M. Hussain, M. H.-I. J. of A. S. and, and undefined 2013, "A survey of image steganography techniques," *earticle.net*, Accessed: Sep. 08, 2026. [Online]. Available: https://www.earticle.net/Article/A206914

[9] Gandharba Swain, Saroj Kumar Lenka, "Classification image steganography techniques in spatial domain: A study," *Int. J. Comput. Sci. Eng. Technol.*, 2014, [Online]. Available: https://www.researchgate.net/publication/303751769_Classification_image_steganography_techniques_in_spatial_domain_A_study

[10] D. Samidha and D. Agrawal, "Random image steganography in spatial domain," *2013 Int. Conf. Emerg. Trends VLSI, Embed. Syst. Nano Electron. Telecommun. Syst. ICEVENT 2013*, 2013, doi: 10.1109/ICEVENT.2013.6496564.

[11] M. N. Abdulwahedand, S. T. Mustafa, and M. S. M. Rahim, "Image spatial domain steganography: A study of performance evaluation parameters," *2019 IEEE 9th Int. Conf. Syst. Eng. Technol. ICSET 2019 - Proceeding*, pp. 309–314, Oct. 2019, doi: 10.1109/ICSENGT.2019.8906402.

[12] F. Q. A. Alyousuf, R. Din, and A. J. Qasim, "Analysis review on spatial and transform domain technique in digital steganography," *Bull. Electr. Eng. Informatics*, vol. 9, no. 2,

pp. 573–581, Apr. 2020, doi: 10.11591/EEI.V9I2.2068.

[13] T. C. Lu and T. N. Vo, "Reversible steganography techniques: A survey," *Digit. Media Steganography Princ. Algorithms, Adv.*, pp. 189–213, Jan. 2020, doi: 10.1016/B978-0-12-819438-6.00018-9.

[14] R. Adee and H. Mouratidis, "A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography," *Sensors 2022, Vol. 22, Page 1109*, vol. 22, no. 3, p. 1109, Feb. 2022, doi: 10.3390/S22031109.

[15] V. K. Trivedi, S. Stalin, S. Joshi, M. H. Alkinani, P. K. Shukla, and B. Gaur, "User data privacy in multimedia domain using 3–3 LSB-based color image steganography with RC4 and Bernoulli map protection," *J. Electron. Imaging*, vol. 31, no. 03, May 2022, doi: 10.1117/1.JEI.31.3.033021.

[16] O. Y. Abdulhammed, "A novel approach of steganography by using strong edge detection and chaos theory," *Multimed. Tools Appl. 2022 8113*, vol. 81, no. 13, pp. 17875–17888, Mar. 2022, doi: 10.1007/S11042-022-12643-3.

[17] A. AbdelRaouf, "A new data hiding approach for image steganography based on visual color sensitivity," *Multimed. Tools Appl.*, vol. 80, no. 15, pp. 23393–23417, Jun. 2021, doi: 10.1007/S11042-020-10224-W.

[18] A. Jan, S. A. Parah, M. Hassan, and B. A. Malik, "Realization of Efficient Steganographic Scheme Using Hybrid Edge Detection and Chaos," *Arab. J. Sci. Eng. 2022 482*, vol. 48, no. 2, pp. 1859–1872, Jul. 2022, doi: 10.1007/S13369-022-06960-W.

[19] S. K. Ghosal, A. Chatterjee, and R. Sarkar, "Image steganography based on Kirsch edge detection," *Multimed. Syst. 2020 271*, vol. 27, no. 1, pp. 73–87, Oct. 2020, doi: 10.1007/S00530-020-00703-3.

[20] S. Mukherjee and G. Sanyal, "Edge based image steganography with variable threshold," *Multimed. Tools Appl. 2018 7812*, vol. 78, no. 12, pp. 16363–16388, Dec. 2018, doi: 10.1007/S11042-018-6975-4.

[21] C. T. Huang, M. Y. Tsai, L. C. Lin, W. J. Wang, and S. J. Wang, "VQ-based data hiding in IoT networks using two-level encoding with adaptive pixel replacements," *J. Supercomput. 2016 749*, vol. 74, no. 9, pp. 4295–4314, Sep. 2016, doi: 10.1007/S11227-016-1874-9.

[22] R. Kumar, D. S. Kim, and K. H. Jung, "Enhanced AMBTC based data hiding method using hamming distance and pixel value differencing," *J. Inf. Secur. Appl.*, vol. 47, pp. 94–103, Aug. 2019, doi: 10.1016/J.JISA.2019.04.007.

[23] A. Malik, G. Sikka, and H. K. Verma, "An AMBTC compression based data hiding scheme using pixel value adjusting strategy," *Multidimens. Syst. Signal Process. 2017 294*, vol. 29, no. 4, pp. 1801–1818, Nov. 2017, doi: 10.1007/S11045-017-0530-8.

[24] M. M. Hashim, M. S. M. Rahim, F. A. Johi, M. S. Taha, and H. S. Hamad, "(PDF) Performance evaluation measurement of image steganography techniques with analysis of LSB based on variation image formats," International Journal of Engineering and Technology. Accessed: Sep. 08, 2026. [Online]. Available: https://www.researchgate.net/publication/343714046_Performance_evaluation_measurement_of_image_steganography_techniques_with_analysis_of_LSB_based_on_variation_image_formats

[25] A. Chatterjee, S. K. Ghosal, and R. Sarkar, "LSB based steganography with OCR: an intelligent amalgamation," *Multimed. Tools Appl.*, vol. 79, no. 17–18, pp. 11747–11765, May 2020, doi: 10.1007/S11042-019-08472-6.

[26] H. Al-Dmour and A. Al-Ani, "Quality optimized medical image steganography based on edge detection and hamming code," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2015-July, pp. 1486–1489, Jul. 2015, doi: 10.1109/ISBI.2015.7164158.

[27] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," *Proc. - Int. Conf. Pattern Recognit.*, pp. 2366–2369, 2010, doi: 10.1109/ICPR.2010.579.

[28] Z. Wang and A. C. Bovik, "Mean squared error: Lot it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, 2009, doi: 10.1109/MSP.2008.930649.

[29] A. A. Abdulla, H. Sellahewa, and S. A. Jassim, "Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images," *Multimed. Tools Appl. 2019 7813*, vol. 78, no. 13, pp. 17799–17823, Jan. 2019, doi: 10.1007/S11042-019-7166-7.

[30] I. Banerjee, S. Bhattacharyya, and G. Sanyal, "Hiding & Analyzing Data in Image Using Extended PMM," *Procedia Technol.*, vol. 10, pp. 157–166, Jan. 2013, doi: 10.1016/J.PROTCY.2013.12.348.



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.