The International Journal of Engineering and Information Technology
journal homepage:www.ijeit.misuratau.edu.ly

Comparative Study: Canny-LoG Edge Detection with Canny-Prewitt Edge Detection by using Huffman Encoding of Image Steganography For Grayscale Images

Areej Mokhtar Elgaier
Department of Computer Science, Elmergib University,
Alkhoms, Libya
amelghayer@elmergib.edu.ly

Ahmed Mohamed Abushaala
Department of Computer Science, Misurata University,
Misurata, Libya
a.mohamed@it.misuratau.edu.ly

***Abstract*— The security of data flow is a major concern for the public network that is the Internet. A data transmission security method called steganography conceals the message within a media container, like an image. Undoubtedly, the media cannot carry so much embedded data. Because the image edge area can withstand pixel value fluctuations better than the center, the edge area is used to accommodate additional message bits. In this study, the hybrid Canny-Log detector was compared with hybrid Canny-Prewitt detector by compress a grayscale images using the Huffman algorithm and hid them in the edges of other color images. This combined two-detector method provides a larger edge area for increased message payload while keeping hidden images unnoticeable. The hybrid Canny-LoG detector was achieved better results in embedding capacity (EC) in terms of the number of pixels available for hiding. Additionally, good results were obtained in quality metrics, such as the mean squared error (MSE) and peak signal-to-noise ratio (PSNR) for the stego image.**

Keywords—Least Significant Bit, image steganography, hybrid edge detection, dilation, Canny, LoG, Prewitt.

I. INTRODUCTION

Due to the expanding capabilities of modern communication, computer networks require extra security measures. Maintaining the confidentiality of private information and safeguarding data copyrights are crucial in the computer world. New techniques based on the idea of image processing are being created and applied to complete this goal. Steganography is a method for disguising digital messages so they are not immediately visible to humans by embedding them into other digital information, such as audio, video, photos, and others. By domain, there are two types of steganography: frequency domain and spatial domain. In spatial domains, a typical approach is the Least Significant Bit (LSB). The LSB approach is not novel. Nonetheless, there are many benefits to this approach, like an easy-to-understand algorithm and reasonably excellent stego picture imperceptibility. This means that there is still need for improvement and research on the LSB [1].

Received 22 Feb, 2025; Revised 04 Apr, 2025; Accepted 23 Apr, 2025.
Available online 31 May, 2025.
DOI: https://doi.org/10.36602/ijeit.v13i2.550

A. Least significant bits (LSB)

The Least Significant Bit (LSB) approach is one of the most straightforward and popular spatial picture steganographic techniques. This tactic is based on the idea that only the least significant portions of an image convey information and that the human eye is not sensitive enough to notice minute variations in those portions as shown in Figure (1). LSB-based spatial domain approaches, enable the secret data to be immediately integrated into the host image without compromising the original cover image's visual quality by altering the least significant bits of a select few selected pixels [2].

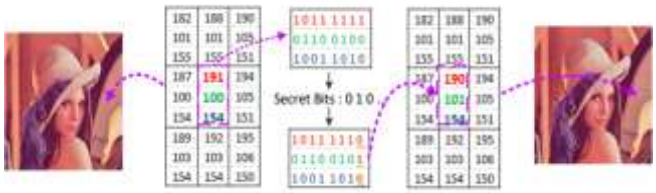

Figure (1): Basic 1-bit LSB embedding mechanism

B. Edge Detection Methods

The fundamental stage in image recognition and image analysis methods is edge detection, which is the key idea in the field of image processing for the detection of objects. Edge detection is typically a technique for separating regions of disruption in an image. Changes in an image's texture, color…etc., are what cause image disruptions. By filtering out unnecessary information and preserving just the relevant structural properties, it minimizes the amount of data that is present in an image. Edge disclosure is a key feature that helps to identify the ideal edges and precise orientation of the item in an image [3].

Traditional image edge detection techniques have been around for a while and have been proposed earlier. As a result, traditional image edge detection techniques are more developed, straightforward, and effective. The four main categories of conventional edge detection techniques are gradient change-based, gaussian difference-based, multi-scale feature-based, and structured learning-based [4].

According to traditional-based methods, an image edge is a sudden change in adjacent pixel values when there is a noticeable difference between two values. Thus, image edge detection involves applying an edge detection operator along with differential technology through pixel-to-pixel grey mutation to identify and produce image edge gradients [4]:

- First-order difference-based operators, such as canny, Sobel, Prewitt, and Robert.
- Second-order difference-based operators, such as Laplace, and LoG.

1) Canny Edge Detection

One of the most used edge detection methods in image processing today is Canny Edge Detection, which continues to outperform several recently developed new algorithms [5]. The following is a summary of this approach [6]:

- The image is smoothed using a Gaussian filter with a predetermined standard deviation to eliminate noise.
- A different operator is used at each position to determine the local gradient and edge direction. The standard canny edge detection method typically chooses a 2×2 nearby region to determine the gradient's magnitude and direction since there are significant variations in gray scale at the edge of the image.
- Apply non-maximal or critical suppression to the gradient magnitude.
- Apply threshold to the non-maximal suppression image.

2) Laplacian of Gaussian Edge Detection (LoG)

The Laplacian method uses zero crossings in the image's second derivative to identify edges. A measure of an image's second-order spatial derivative, the Laplacian helps locate edges and other areas of abrupt changes. Gaussian smoothing is used before Laplacian to lessen its impact because Laplacian is very sensitive to noise in edge detection. This two-step process is known as the LoG operation [7]. The 2-D LoG function centered on zero with Gaussian standard deviation σ is formulated in Equation (1) [8]:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (1)$$

Where:

- (x, y) represents the dimensions of the image.
- (σ) The Gaussian blur coefficient, which determines the spread of the blur. The larger the value of σ, the wider the blur and the smoother the image.
- (x^2,y^2) The squared Euclidean distance from the point (x, y) to the center of the blur.

3) Prewitt Edge Detection

This operator is similar to Sobel but with different mask coefficients. The following qualities ought to be present in the derivatives [9]:

- The mask should contain opposite signs.
- The sum of the mask has to be equal to zero.
- More edge detection due to more weight.

The masks are defined as following Figure (2).

1	1	1
0	0	0
-1	-1	-1

G_x

-1	0	1
-1	0	1
-1	0	1

G_y

Figure (2): [3×3] Prewitt edge detection masks

C. Edge Dilation

When employing picture steganography, which depends on concealing the secret data in the edges, one often aims to expand the image edge area. Applying the dilation approach to the edge region will increase the number of pixels that can be used to conceal the secret data. Edge dilation is used to increase the edge area in binary images. The dilation operator utilized in the technique takes two inputs: the dilated picture and the structuring element. The structuring element is a matrix of coordinate points with a default size of 3×3. The dilation caused by the structural element SE on the image f is represented by f ⊕ SE and may be computed using Equation (2) [10]:

$$f = f \oplus SE = \{z|(\hat{S})z \cap I \neq \emptyset\} \qquad (2)$$

Where:

- (f) Any grayscale image.
- (SE) A structuring element.
- (z) A point or location in the image.
- (Ŝ) A reflection of the structure element on pixel loop z.
- (I) region on which the operation is applied.

D. Compression

In order to record or transmit image data more effectively, image compression is employed to remove unnecessary and redundant data. As a result, image compression speeds up and reduces the duration of network transmission. There are two types of compression techniques: lossless compression and Lossy compression. The original image ought to be an exact replica of the first approach's compressed image. The two halves of an image compression system are the compressor and the decompressor. Whereas the compressor consists of two stages: pre-processing and encoding, the decompressor consists of a decoding step followed by a post-processing stage [11]. In our study, the entropy-encoding algorithm known as Huffman coding is used for lossless image compression.

- ***Huffman Encoding***

Huffman coding is a useful technique for image compression. Variable length coding, like the Huffman code, is widely used to increase coding efficiency. The encoder uses the Huffman source-coding algorithm to create a precisely decipherable Huffman code with a minimum projected code word length when it is aware of the probability distribution of a data source. It will be able to estimate the number of bits needed to store the information for a given picture. Huffman coding, which uses a specific

process to choose the representation for specific images, generates a prefix code [12].

II. RELATED WORKS

There are many previous research and studies based on combining masking and compression techniques, including:

In 2018 [13], the paper suggested a novel steganography algorithm based on the local reference edge detection method and the exclusive disjunction (XOR) property. The secret message bits are inserted in the sharp sections using local reference pixels that are identified by the Canny edge approach and optimized by the dilation morphological operator. The embedding procedure was made more secure and capable by using a bit plane-dependent XOR coding technique that alters edge pixel LSB bits as little as feasible. While currently in use, edge-based steganography methods provide enhanced imperceptibility, their embedding capacity is still limited. The proposed method successfully boosts embedding capacity while preserving the appropriate degree of imperceptibility and resilience.

The researchers suggested in 2019 [14] that the embedding in the three most significant bits (MSB) pixels of cover images using dilated hybrid edge detection with the goal of enlarging the edge region to improve the ability to insert data in image steganography. With this method, the extraction can be done without using the original cover image for edge detection. Since messages are embedded at x LSB and y LSB values, where x is the number of LSB bits replaced in the edge area and y is the number of bits replaced in the non-edge area. Also x and y cannot reach 3-bits of MSB, so the edge detection of the cover image and the stego image will be the same. According to the findings of the tests, the proposed steganography methodology improved imperceptibility to the point where the PSNR value increased by roughly 1 to 2 dB in comparison to some ways that have been suggested in the past. Similarly, to this, a broader edge region can boost the message embedding capacity.

J. C. T. Arroyo et. al [15] in 2020 developed an improved data handling technique that utilizes both steganography and cryptography to secure sensitive text data with many layers of protection and compression is used. They converted plaintext data into ciphertext, an unintelligible format, using the Polybius cipher. Then Least Significant Bit (LSB) approach is used to embed the result into an image file after the ciphertext has been compressed using the Huffman coding algorithm. According to simulation results, the proposed methodology produced stego images that performed better than those created using the lone LSB steganography technique in terms of file size, Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), and error metrics. Results showed that using the suggested method on the carrier leaves no evidence of data change, making it impossible to identify the embedding of a hidden message. Additionally, the compression technique resulted in smaller files than the LSB alone for image steganography.

The research in 2021 [16] is a steganography method that employed the edge detection mechanism for larger payloads to conceal data. Well-known edge detectors were utilized to generate as many edge pixels as possible like Sobel, Roberts, Prewitt, LOG, Canny, Fuzzy Logic, hybrid and Dilate (5×5) and hybrid dilate edge detector (10×10). The number of bits used to embed was between 2-4 bits in smooth and sharp areas, which has improved the least significant bit (LSB) method. In their proposal, steganography was used in both grayscale image and color images to compare outcomes and show the highest level of quality and payload possible. The proposed approach kept the original image's nature while achieving a high payload to incorporate into the cover image. The results of the experiments demonstrated that the PSNR was raised from a minimum of 5% to a maximum of 8%, indicating that the suggested steganography technique approach was the best in terms of payload, confidentiality, and quality.

In 2021 [10], it was proposed a novel approach based on the edge area for enhancing the suggested embedding capacity in images. The novel method dilated the edge region after combining canny and prewitt edge detection algorithms utilizing OR binary operation. The Least Significant Bit (LSB) technique was used to hide the secret text message on the cover image. The experimental findings demonstrated that employing the Canny-Prewitt approach in conjunction boosts embedding capacity more than using the two procedures independently. Experiments have also shown that dilatation on the edge area boosts embedding capacity.

In 2022 [17], it was proposed a new method that combined steganography, compression, and encryption. The study contributed to solving the problem of the security and capacity of information when sent over the Internet. The advanced encryption standard (AES) technique was used to encrypt the compressed data after the discrete wavelet transform (DWT) algorithm compressed the secret image. The least significant bit (LSB) approach has been used to conceal the encrypted data. The sender used the DWT technique to compress the secret image and the AES algorithm to encrypt the compressed image to create ciphered bits, which were then embedded into the cover image using the LSB algorithm. The cipher image will be taken from the stego-image during the extraction step in order to recover the secret image. Then, using the same key as during encryption, the ciphered data was decrypted using the AES algorithm. The DWT technique was then used to decompress the secret image. The results were good with an average PSNR of 47.8 dB and an average SSIM of 0.92. The study findings indicated that the stego-image quality is still good.

In 2022 [18], it was proposed a new approach to LSB steganography based on hybrid edge detection between Canny edge detection and Laplacian of Gaussian edge detection. Used grayscale images with different sizes to hide secret text cipher messages with different lengths. The affine hill cipher method was proposed to encrypt the secret message and then 1-bit LSB to embed this encrypted message in the edge cover image. The results indicated that hybrid edge detection (Canny- LoG) with LSB for hiding data could provide higher embedding capacity than hybrid edge detection (Canny- Sobel) with LSB. This algorithm proved that hiding in the image edge area could preserve the imperceptibility of the stego-image with good values of PSNR and MSE compared to the traditional LSB. Histogram distribution for the original image and the stego-image was almost the same, which indicates that there was no significant change in the image after the embedding

process. The method also proved that the secret message was extracted successfully without any distortion or loss of information.

III. THE DATASET

Standard images were employed as a storage medium, including three color images (Lake, Baboon, and Pepper) and two grayscale secret images (Man and Clock). These images are available for download on the website [19]. Figure (3) illustrates the images used in the study. The cover image size was standardized to [512 × 512], while the dimensions of the hidden image in Figure (3) were set to [128 × 128] with an 8-bit grayscale depth. The size [128 × 128] was chosen because it allows for complete integration into a cover image of [512 × 512].

IV. THE PROPOSED METHOD

It was identified through prior research that there is a challenge in providing sufficient space to conceal images without compromising the quality of the stego image [20]. To address this issue, a technique is proposed in this study for concealing grayscale images within the borders of RGB color images. This is achieved by integrating a hybrid edge detection method, which combines Gaussian Laplacian and Canny edge detection through an OR operation to accurately identify edges. Subsequently, dilation is applied to enhance the edge pixels, ensuring robust boundary detection. Additionally, this hybrid approach is compared with another hybrid combination of Canny and Prewitt edge detection. Following edge detection, Huffman coding is utilized to compress the hidden grayscale image, reducing its size for efficient embedding. Finally, LSB (Least Significant Bit) steganography is employed to embed the compressed image into the boundaries of the RGB cover image. This technique ensures minimal distortion to the stego image while maximizing the capacity for hidden data. The proposed method aims to provide a balanced solution that maintains image quality while optimizing the space available for concealing grayscale images.

Lossless compression was employed in the suggested approach to preserve quality while reducing the size of the secret image. There are two main processes: embedding the secret image and extracting the secret image process.

Before the embedding process, some image preprocessing was used, like resizing the secret image and then compressing the secret image by applying a lossless Huffman coding algorithm. The cover image should be used as an RGB color image because the edges in the color image are three channels (Red, Green, and Blue), which are three times the size of the grayscale image. The cover image was split into three channels, and then Canny, Prewitt and LoG edge detection was applied, and then using OR operation to get the hybrid image. These steps were applied on every channel in the cover image and after that, all edges in the three channels were ORed to get clear edges from the cover image. After that, the edges were dilated using a structuring element of size [5 × 5] to get an increasing number of pixels on the edges, and finally, were saved the results as a hybrid image to be used in the embedding process.

(a) Lake (b) Baboon (c) Peppers

(a) Man (b) Clock

Figure (3): Dataset used in testing our proposed method.

A. *Embedding Process*

- **Step 1:** Read the cover image.
- **Step 2:** Read the coordinates of the edge pixels to hold the secret image.
- **Step 3:** Check the compressed secret image's length to see whether it can fit inside the edge area pixels. If not, a notification with an error will be shown by the system.
- **Step 4:** Embed the secret compressed image in the edges of the color image, where each bit of the compressed secret image is embedded in the least significant bits (LSB) of each in every channel of RGB pixel. This process is repeated until all parts of the secret image have been included.
- **Step 5:** Finally, save the produced stego image.

Figure (4) illustrates the diagram of the embedding process.

B. *Extraction Process*

- **Step 1:** Read the stego image.
- **Step 2:** For the extraction, use the hybrid edge area pixels coordinates as a key.
- **Step 3:** Get the least significant bit of each edge pixel area sequentially.
- **Step 4:** Step 3 is repeated until the final compressed secret image is obtained.
- **Step 5:** Apply the Huffman decoding algorithm to get the secret image.
- **Step 6:** Finally, the receiver reads the secret image.

Figure (5) illustrates the diagram of the extraction process.

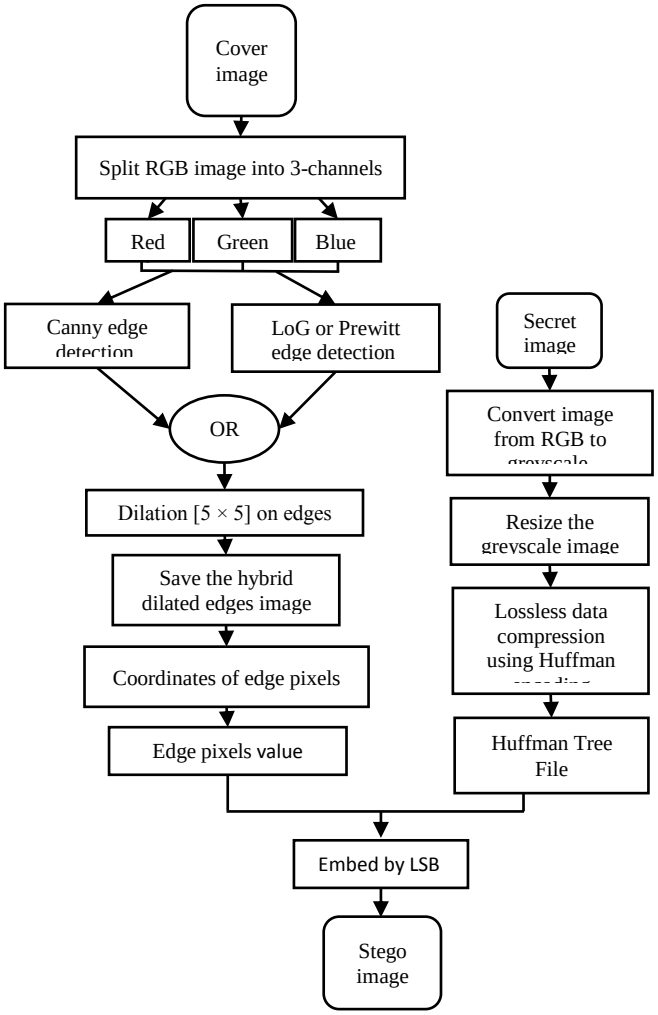

Figure (4): Diagram of the Embedding Process

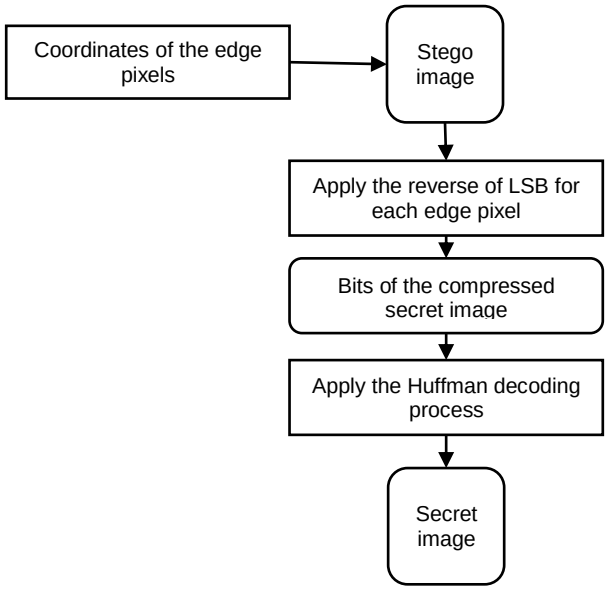

Figure (5): Diagram of the Extracting Process

V. METRICS OF RESULT

The method will be evaluated using Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), Embedding Capacity (EC), and Compression Ratio (CR) metric. The PSNR and MSE are the most commonly used metrics for measuring the quality of Stego images [21]. It is expected that the results of this method will be good in increasing the capacity of embedding confidential data in the cover image, while ensuring the high quality of the cover image, and increasing the imperceptibility.

- **Embedding Capacity (EC):** By counting the number of possible bits that could be included in the cover image, it is calculated. The number of edge pixels in the cover image will have a significant impact on the payload capacity of messages that can be inserted in it because our suggested algorithm only stores the bits of the secret message in the edge pixels of the image [18].
- **Mean Squared Error (MSE):** The mean squared difference between the reference image and the stego image. Low MSE makes the image steganography approach more effective. MSE is determined by dividing the sum of the square differences of all pixels by the total number of pixels, which is done pixel-by-pixel [22]. MSE searches for similarities using the formula of Equation (3) below:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \qquad (3)$$

The value I (i, j) represents the original cover image. K (i, j) represents the secret image. m, n indicates the dimensions of the image.

- **Peak Signal to Noise ratio (PSNR**): There will be adjustments made to the cover image's edge pixel values as the secret data is embedded. Since the modifications directly impact how imperceptible the output Stego-image is, they need to be examined. By examining the mean squared error value between the original Cover and the Stego-image, the PSNR is one of the most often used and well-regarded metrics for evaluating the quality of the Stego-image as Equation (4) [23]:

$$PNSR = 10log_{10}\left(\frac{MAX_1^2}{MSE}\right) \qquad (4)$$

The PSNR formula is calculated based on another measurement parameter, MSE, as shown. MAX is the highest pixel value available, here MAX = 225.

- **COMPRESSION RATIO (CR):** The compression ratio is the ratio of an uncompressed image's size to that of a compressed image as Equation (5) shows. Better image quality and less storage space usage result from a greater compression ratio[24].

$$CR = \frac{UnCompressed\ \ Image\ Size}{Compressed\ Image\ Size} \qquad (5)$$

VI. RESULTS AND DISSCUCION

In this study, our suggested method is being implemented using the MATLAB programming language. It is one of the best programming languages for usage in applications involving image processing. To assess the effectiveness of the adopted system, many tests were conducted. The experiment's cover images were three generic RGB color

images of various sizes. They are (Lake, Baboon and Peppers). To hide a hidden grey image (the man and clock), these images were used as cover images. The results were then compared using some of the criteria employed in this study. The performance of our suggested approach will be assessed after comparing the hybrid Canny-LoG and hybrid Canny-Prewitt methods' embedding capacities and determining how hiding a grey picture affects the stego image's quality. The following criteria are used to evaluate measurements and results:

A. Counting the number of bits in secret image after Huffman encoding

Table (1) and Figure (6) show the results of compressing each secret image. After Huffman encoding compression, the images were reduced in size and converted into a number of simple binary bits consisting of "0s" and "1s.

Table (1): Comparative the number of bits of the secret image with/without Huffman Encoding

Size of Image	Secret Image	Number of Bits without Huffman Encoding	Number of Bits with Huffman Encoding	Compression Ratio
128 x 128	Man	131072	123850	1.1:1
	Clock	131072	110933	1.2:1

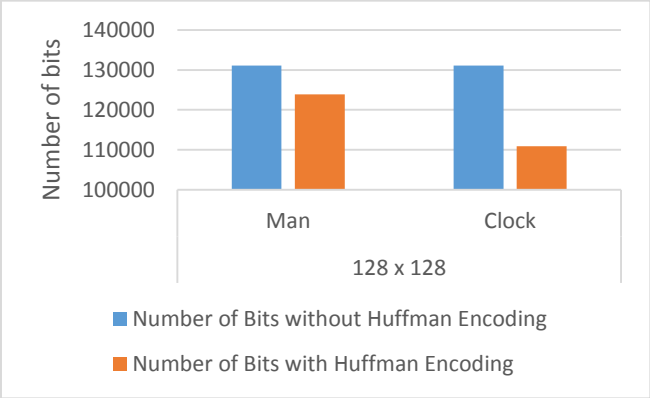

Figure (6): Comparative the number of bits of the secret image with/without Huffman Encoding

The number of bits in the secret images compressed by Huffman encryption is reduced. It should be taken into account that the compression used in this method is lossless, therefore, the reduction in the number of bits was insignificant. In addition, compression varies depending on the image. In Figure (6) and Table (1), the Clock image was compressed better than the Man image, in which the compression ratio of Clock (128 × 128) was 1.2:1, despite being the same size as the other image. The specific image content can also affect the number of bits compressed. The Huffman encoding is also useful in protecting the image because its form has been changed from a normal two-dimensional image to an encrypted vector of bits form consisting of 1's and 0's.

B. Evaluation of Payload Capacity

This work developed a steganography method based on a hybrid edge detection method, which combines (Canny-LoG) or (Canny-Prewitt) edge detectors. Before performing the process of hiding or embedding the secret message using LSB, the edge detection procedure using the proposed technique (Canny and LoG) or (Canny-Prewitt) was first applied to the RGB color image and then applied the edge dilation method of size (5 × 5). Figure (7) shows the edges of the cover image that act as a container for the messages to be stored using LSB technology. Each edge pixel will store 3 bits of the hidden message, which means one bit is stored in each (RGB) channel.

Figure (7) illustrates that the edge regions detected using different strategies are significantly different from each other. The edge area achieved with the hybrid technology is sharper. However, compared to the Laplacian of Gaussian (LoG) approach, the edge space produced by the Canny technique still contains more pixels. Also using dilation on edges resulted in a higher number of pixels of edges. Hybrid algorithms that integrate both Canny and Prewitt edge detectors for comparison were also applied on the same cover images. Figure (8) depicts the edge regions of the cover images. However, the Prewitt edge detector provided fewer details than the LoG detector, which indicated that the edge area provided by the (Canny and LoG) approach provided more room for the hidden image to be embedded.

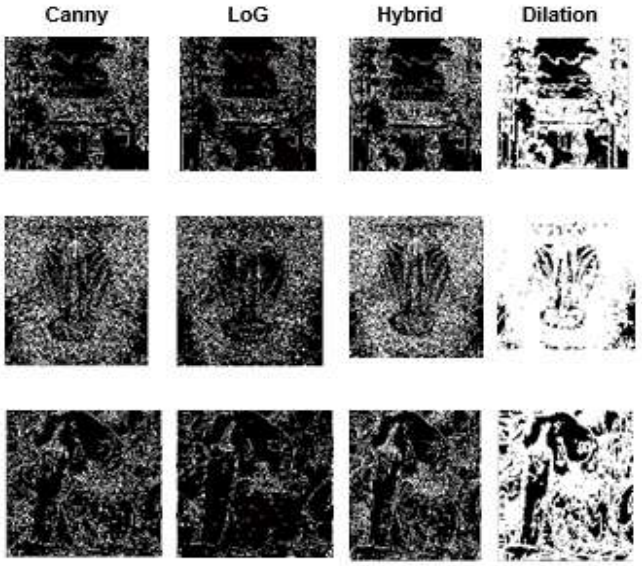

Figure (7): Edge Areas of Cover Images by (Canny-LoG)

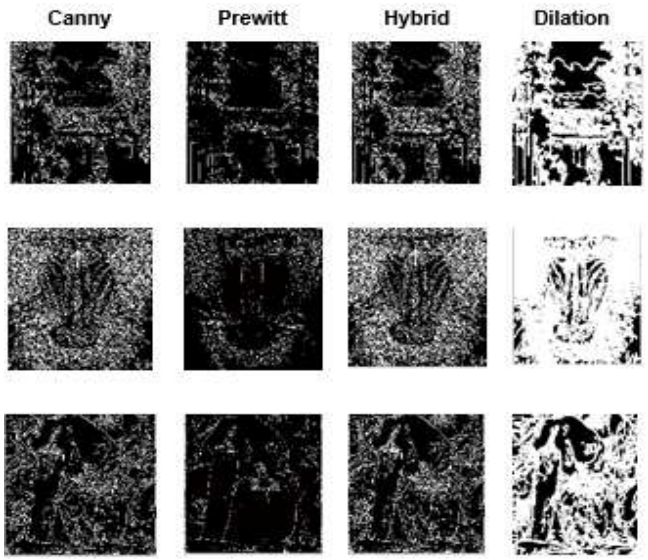

Figure (8): Edge Areas of Cover Images by (Canny-Prewitt)

Table (2) and Figure (9) show the results of the calculations, which determine the embed capacity of the cover images using the proposed (Canny-LoG) hybrid edge area technique.

Table (2): Embed Capacity Based on Canny and LoG.

Cover Image (512×512)	Embed Capacity (Bits)			
	Canny	LoG	Hybrid	Dilation
Lake	43364	33207	58367	156435
Baboon	82122	55447	105271	235796
Peppers	41650	23710	53357	168201

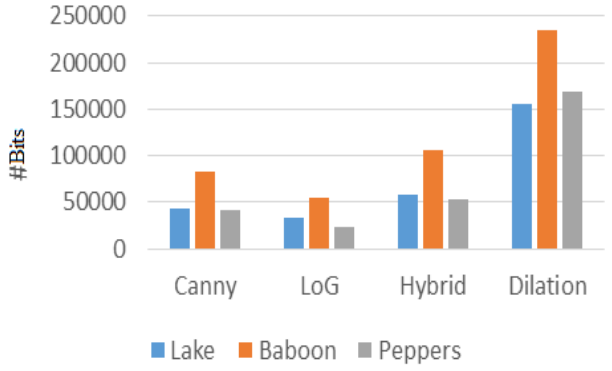

Figure (9): Embed Capacity by Using (Canny- LoG) Edge Detectors Hybrid and Dilation

The results of calculating the embed capacity using Canny, Prewitt, hybrid and dilation, are illustrated in the following Table (3) and Figure (10) .

Table (3): Embed Capacity Based on Canny and Prewitt

Cover Image (512×512)	Embed Capacity (Bits)			
	Canny	Prewitt	Hybrid	Dilation
Lake	43364	19732	51182	144846
Baboon	82122	22597	90851	226601
Peppers	41650	13902	47690	158181

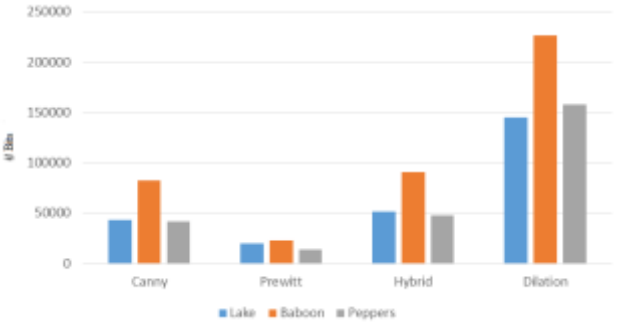

Figure (10): Embed Capacity by Using (Canny- Prewitt) Edge Detectors Hybrid and Dilation

May infer from Tables (2) and (3) that, in addition to the hybrid technique approach greatly increases the number of edge pixels as compared to employing Canny, LoG, and Prewitt independently. The dilation on edges also enhanced the number of edge pixels. The hybrid edge detection method used in the suggested algorithm, Canny-LoG, has a larger embed capacity than the hybrid method Canny-Prewitt, as demonstrated when comparing Table (2), with Table (3). Lake's cover image, for instance, has a 156435-bit embed capacity when adopting (Canny and LoG) filters, as shown in Table (2). The identical Lake cover image's embed capacity is 144846-bit when employing (Canny and Prewitt) filters as shown in Table (3).

C. Quality Evaluation of Stego Image

In summary, lower MSE values, and higher PSNR values indicate better quality stego images with less distortion and better preservation of the original cover image. By examining the MSE, and PSNR values for stego images, one can assess the effectiveness of the steganographic technique in preserving the quality and fidelity of the cover image while successfully hiding the secret image within it.

In Table (4), the results of the proposed algorithm with (Canny-LoG) edge detection are shown when the size of the cover image is [512×512] and the secret image is [128×128] with dilation of [5×5] at the edges of the cover image. The results were very close in all images, the best of which was Peppers and Clock, where obtained a high-quality PSNR of 54.890025 dB, and a low MSE of 0.210901. The results of the suggested approach for embedding in the detected edges with (Canny-LoG) of all channels are displayed in the accompanying Table (4).

Table (4): The Results of the (Canny-LoG)

Cover Image	Secret Image	MSE	PSNR (dB)
Lake	Man	0.236369	54.394902
Baboon	Man	0.235802	54.405334
Peppers	Man	0.235809	54.405194
Lake	Clock	0.211285	54.882124
Baboon	Clock	0.211833	54.870874
Peppers	Clock	0.210901	54.890025

In Table (5), the results of the proposed algorithm with (Canny-Prewitt) edge detection are shown when the size of the cover image is [512×512] and the secret image is [128×128] with dilation of [5×5] at the edges of the cover image. Here also the results were very close in all images, the best of which was Peppers and Clock, where obtained a high-quality PSNR of 54.880269 dB, and a low MSE of 0.211375.

Table (5): The Results of the (Canny-Prewitt)

Cover Image	Secret Image	MSE	PSNR (dB)
Lake	Man	0.236305	54.396070
Baboon	Man	0.236500	54.392496
Peppers	Man	0.236001	54.401659
Lake	Clock	0.211675	54.874108
Baboon	Clock	0.211758	54.872412
Peppers	Clock	0.211375	54.880269

Based on the provided results in Tables (4) and (5) it was proven that the proposed method resulted in relatively low MSE values, and high PSNR values. The results of comparing the stego-images resulting from the hybrid (Canny-LoG) with the images resulting from the hybrid (Canny-Prewitt) indicate that they are almost identical. This suggests that the processing techniques used were able to maintain a high quality and a similarity between the original and stego images.

VII. CONCLUSIONS AND FUTURE WORD

This paper focused on comparing (Canny-LoG) and (Canny-Prewitt) hybrid edge detection for improving information hiding and increasing embedding capacity by combining compression and information hiding techniques. The research involved designing an algorithm using MATLAB for image processing. This study encodes the secret grayscale image and hides it in the least significant bit of each edge pixel of color image to conceal the information effectively. The results show that the combined and expanded edge area obtained by the hybrid dilation (Canny-LoG) detector provides a higher than (Canny-Prewitt) payload capacity. Therefore, it allows for hiding a larger amount of data in the cover image, in addition, the Baboon image has the best results of embedded capacity, which was 235796-bit. The use of the Huffman encoding and LSB technique along with the hybrid and dilation Canny–LoG edge detection has been proven to increase the imperceptibility of the stego-image, the best results were achieved for Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), with values of 0.210901 and 54.890025 dB, respectively. For future work, combining with other edge detectors, such as mathematical fuzzy morphology, will be attempted to enhance the payload capacity.

REFERENCES

[1] J. Jumanto, "An enhanced LSB-image steganography using the hybrid Canny-Sobel edge detection," *Cybernetics and Information Technologies*, vol. 18, pp. 74-88, 2018.

[2] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299-326, 2019.

[3] R. K. MR and T. Mithun, "Edge Connectivity Techniques for Image Analysis–A Survey," 2020.

[4] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, W. Zhao, *et al.*, "Survey of image edge detection," *Frontiers in Signal Processing*, vol. 2, p. 826967, 2022.

[5] D. Mathur and D. P. Mathur, "Edge Detection Techniques In Image Processing With Elaborative Approach Towards Canny," *Computer Science Department, Lachoo Memorial College Of Science & Technology*, 2016.

[6] S. Singh and A. Datar, "Improved hash based approach for secure color image steganography using canny edge detection method," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 15, p. 92, 2015.

[7] S. K. Ghosal, J. K. Mandal, and R. Sarkar, "High payload image steganography based on Laplacian of Gaussian (LoG) edge detector," *Multimedia Tools and Applications*, vol. 77, pp. 30403-30418, 2018.

[8] S. Gupta, C. Gupta, and S. Chakarvarti, "Image edge detection: a review," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, pp. 2246-2251, 2013.

[9] A. S. Ahmed, "Comparative study among Sobel, Prewitt and Canny edge detection operators used in image processing," *J. Theor. Appl. Inf. Technol*, vol. 96, pp. 6517-6525, 2018.

[10] N. A. Mohsin and H. A. Alameen, "A hybrid method for payload enhancement in image steganography based on edge area detection," *Cybernetics and Information Technologies*, vol. 21, pp. 97-107, 2021.

[11] A. J. Hussain, A. Al-Fayadh, and N. Radi, "Image compression techniques: A survey in lossless and lossy algorithms," *Neurocomputing*, vol. 300, pp. 44-69, 2018.

[12] R. P. Jasmi, B. Perumal, and M. P. Rajasekaran, "Comparison of image compression techniques using huffman coding, DWT and fractal algorithm," in *2015 International Conference on Computer Communication and Informatics (ICCCI)*, 2015, pp. 1-5.

[13] K. Gaurav and U. Ghanekar, "Image steganography based on Canny edge detection, dilation operator and hybrid coding," *Journal of Information Security and Applications*, vol. 41, pp. 41-51, 2018.

[14] D. R. I. M. Setiadi, "Improved payload capacity in LSB image steganography uses dilated hybrid edge detection," 2019.

[15] J. C. T. Arroyo, C. P. Barbosa, M. V. Aborde, F. B. Yara, and A. J. P. Delima, "An improved image steganography through least significant bit embedding technique with data encryption and compression using polybius cipher and huffman coding algorithm," *International Journal*, vol. 9, 2020.

[16] M. Alanezi, I. S. M. Altaay, and S. Y. H. Mallaàloo, "Payload and quality augmentation using steganographic optimization technique based on edge detection," *Indones. J. Electr. Eng. Comput. Sci*, vol. 23, pp. 529-539, 2021.

[17] W. A. Awadh, A. S. Alasady, and A. K. Hamoud, "Hybrid information security system via combination of compression, cryptography, and image steganography," *International Journal of Electrical and Computer Engineering*, vol. 12, p. 6574, 2022.

[18] F. F. Yahia and A. M. Abushaala, "Cryptography using Affine Hill Cipher Combining with Hybrid Edge Detection (Canny-LoG) and LSB for Data Hiding," in *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic*

Control and Computer Engineering (MI-STA), 2022, pp. 379-384.

[19] *Ming Hsieh Department of Electrical Engineering USC Viterbi School of Engineering SIPI Image Database*. Available: https://sipi.usc.edu/database/database.php?volume=misc&image=39#top

[20] A. M. Elgaier and A. M. Abushaala, "An Effect of Huffman Encoding with Hybrid Edge Detection in LSB Image Steganography," in *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2024, pp. 403-409.

[21] R. Mishra, A. Mishra, and P. Bhanodiya, "An edge based image steganography with compression and encryption," in *2015 International Conference on Computer, Communication and Control (IC4)*, 2015, pp. 1-4.

[22] F. Şahin, T. Çevik, and M. Takaoğlu, "Review of the Literature on the Steganography Concept," *International Journal of Computer Applications*, vol. 975, p. 8887.

[23] H. A. W. J. Albayati and S. A. Ali, "A Comparative Study of Image Steganography Based on Edge Detection," in *Journal of Physics: Conference Series*, 2021, p. 012032.

[24] R. Patel, V. Kumar, V. Tyagi, and V. Asthana, "A fast and improved Image Compression technique using Huffman coding," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 2283-2286.