Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

An Efficient Image Steganography Method Using Block-Based Indexing and LSB Substitution

Anilkumar Patel[1], Daxa Vekariya[2]

[1]Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.
[2]Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.

E-mail: [1]anilpatel11@gmail.com, [2]Vasoyadaxa@gmail.com

Abstract

A secure and efficient image steganography algorithm is proposed in this paper that uses block-wise indexing and LSB embedding. First, the cover image is divided into blocks of size 4 × 4. Blocks are classified as embedding blocks (E-Blocks) and index blocks (I-Blocks). Here, the 3-bit LSB substitution scheme is applied. The secret bits are matched with the least significant bits of the row pixels of the image, and the index values of those pixels are saved separately. In case of a mismatch, the secret bits are embedded in default pixels. The effectiveness of the proposed scheme is increased because the approach involves fewer changes to the cover image and does not use any deterministic extraction technique using indexing. The experimental results carried out by implementing the proposed scheme on various images of sizes ranging from 64 × 64 pixels to 512 × 512 pixels revealed that it produces high-quality images and efficient extraction. In the case of gray-scale images, the average PSNR, SSIM, and UIQI measures were found to be 55.64583 dB, 0.99869, and 0.99800, respectively, whereas for color images, their values turned out to be 49.19303 dB, 0.99209, and 0.99671. Overall, the presented technique provides an effective compromise between embedding capacity, invisibility, and robustness, thus being a good choice for secure image transmission applications.

Keywords: Embedding, E-Blocks, Imperceptibility, I-Blocks, Security.

1. Introduction

The internet communications have grown tremendously, making possible the transmission of different types of information, including text messages, voice communications, and media files. With the increased usage of internet services for transmitting classified information, there arises the need for a safe and reliable channel of communication. There are basically two fundamental approaches to securing information, which include cryptography and steganography [20].

In cryptography, data is converted from plain readable data into coded data that is readable only to those who are supposed to read it. The use of coding, in most cases, means that there is some secret in the message that makes it susceptible to unauthorized decoding. In steganography, however, the secret is hidden in another medium to ensure the security of the

DOI: https://doi.org/10.36548/jiip.2026.2.006
Received: 19.02.2026, received in revised form: 22.03.2026, accepted: 09.04.2026, published: 25.04.2026

message. This does not generate any form of suspicion since no change is made to the message medium [21].

Steganography involving images is a common field, embedding data in the pixel values of an image. Image steganography encompasses the following techniques:

- Text steganography: Hides data within formatted text.
- Image steganography: Embeds secret data into the pixel values of digital images [19].

Among the many techniques, spatial domain techniques manipulate the pixel values directly with a mathematical model of 2D or 3D matrices in the case of grayscale and RGB images, respectively [28]. Substitution of LSBs forms a major part of this category [28]. On the other hand, frequency domain techniques perform the operation by first transforming the image with the help of algorithms such as DCT and DWT, and then embedding the data within their coefficients; they are highly resilient to compression [15].

Though many attempts have been made toward developing LSB-based steganographic techniques, the majority of them fail due to their lack of flexibility, low payload size, or poor robustness to even small perturbations. In particular, any algorithm based on the principle of substitution fails to provide the necessary control over embedding positions, and hence is incapable of providing any trade-off between invisibility and robustness. These issues necessitate the requirement of an embedding scheme that not only controls the amount of distortion but also guarantees deterministic extraction.

2. Related Work

Watermarking, which is almost similar to steganography, attempts to claim ownership and secure the copyright of digital data, whereas steganography is mainly concerned with communication in secrecy [1]. To overcome the issues of capacity and distortion in steganography, Por et al. [2] presented a sequential color cycle LSB embedding technique. On the other hand, Khandelwal and Bisht [3] adopted a data hiding technique through the energy cost function model.

There have been some advanced studies that considered the use of models for assessing the performance of steganographic techniques, focusing on three parameters: imperceptibility, robustness, and capacity. For example, Zhang et al. [4] used ISGAN for embedding gray-scale information into color images. Similarly, Ogras [5] proposed LSB embedding using a chaos-based approach. Moreover, Mahana and Aggarwal [6] performed performance evaluation using traditional measures. Likewise, Siddiqui et al. [7] developed an IRD-based steganography method for medical images using intensity-based segmentation. Finally, Eyssa et al. [8] highlighted secure data embedding in OFDM channels using DWT and chaotic mapping.

In an effort to improve the scalability and robustness of techniques, Liu et al. [9] developed a Dense Net-based approach in coverless image steganography, whereas Li et al. [10] employed masking techniques to ensure the GAN-generated images look like processed images. Other techniques that proved useful include the application of transforms such as Arnold and FrRnWT [11], and enhancements made by applying flipping operations [12]. Some techniques involving hybrid methods that combine DWT and LSB substitution [13] are still

successful and further randomness can be improved in LFSR-based pseudo-random embedding [14]. To protect confidential medical records, Ahmad et al. [15] incorporated wavelet transform along with DES and arithmetic coding into their technique. In another attempt, the NAS-Stego framework [16] was developed by utilizing neural architecture search for encoder-decoder generation. Patel and Vekariya [17] presented a modified LSB embedding technique by employing segmentation of the image into 4×4 blocks referred to as E-Blocks and I-Blocks to embed the secret message. Supporting this technique, Ye et al. [18] presented another scheme of encryption-based image steganography using interpolation and histogram shifting.

Shmuel et al. [24] used the concept of maximum energy seam embedding to achieve invisibility by hiding the secret data on perceptually redundant paths. The authors Mohamed et al. [25] proposed the use of stochastic computing to provide the concept of probabilistic logic-based large-capacity hiding. On the other hand, StyleGAN steganography [26] combined the style transfer and embedding concepts. Furthermore, Josephus permutation combined with LSB embedding [27] increased randomness and confusion.

The concept of metadata-guided enhanced LSB techniques for character sequence optimization was provided by Jayapandiyan et al. [28], whereas the authors Chuang et al. [31] proposed the technique of large payload embedding using adjacent mean prediction in RGB images. Recently, Zheng et al. [22] and Yang et al. [23] proposed an advanced adversarial training framework for JPEG steganography. Moreover, hybrid models using RSA and compression [29] and high-capacity LSB substitutions [30] were discussed. Furthermore, Mahana et al. [6] and Siddiqui et al. [7] highlighted the importance of invisibility in healthcare applications. In addition, LSB was used with cryptography (AES, Blowfish) [20][21].

Kanimozhi et al. [32] have provided their ideas for secure and reliable image watermarks, while [33] has indicated that there will be almost no loss in the recovery of the secret along with higher PSNR value. [34] has presented Discrete Cosine Transformation along with Deep Learning techniques and GAN, providing high levels of security and integrity. GAN was used by Ambika et al. [35] where it changes the appearance of the images in certain areas without compromising the originality of those areas in the image.

Hence, although traditional LSB-based data embedding is commonly practiced owing to its simplicity, it does not demonstrate robustness or selectivity. The simple representation of data embedding using the 3-LSB algorithm within one column of a 4×4 pixel grayscale image is illustrated in Fig. 1. Although the method incorporates direct data embedding through bit replacement, it does not take into consideration distortion reduction or robust extraction. Generally, the available literature suggests that the above three factors are inherently incompatible with each other, posing a common dilemma in the design process. Other than the application of machine learning methods, another direction in research involves using classification algorithms within image steganography. Vasoya et al. developed an image steganography system with the inclusion of classification algorithms for guiding data hiding processes, showing improved concealment results in certain cases [38]. On a similar note, Siddhapura and Vekariya conducted studies on privacy-protected data security employing a range of perturbation and cryptographic methods [39].

However, these restrictions can be mitigated with the introduction of the structured embedding technique that we propose through block partitioning using a 4×4 matrix and index-based embedding. The blocks are designated as either an E-Block or I-Block, allowing for selective and distortionless data hiding. In addition to this, our contribution to the field involves presenting a more advanced 3-LSB embedding based on secret-aware indexing.

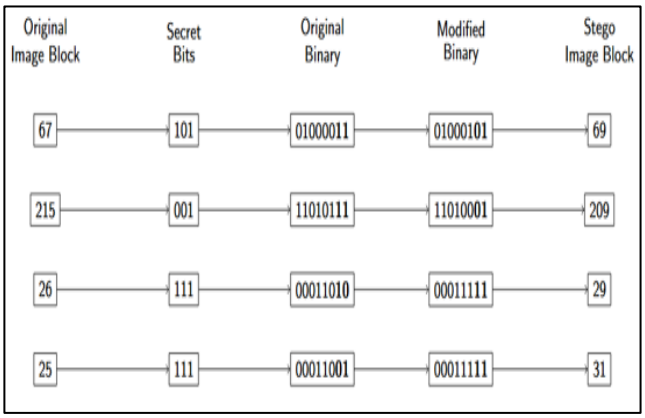

Figure 1. Illustration of 3-LSB Substitution

2.1 Major Contributions

This paper makes the following key contributions.

- Proposed an innovative steganography technique to embed hidden information by partitioning the cover image into smaller segments (4×4 blocks). Some blocks can be used to embed actual hidden data (E-Blocks) while others can be used to embed the index/position of the hidden data (I-Blocks).
- An intelligent technique to embed 3-bits using the LSB technique by storing matched pixel positions to minimize image distortion while maintaining retrieval feasibility.
- The proposed scheme can support any type of secret data, whether it is in text or image format. The scheme can support a maximum of 999,999 characters or complete images either in grayscale or color.
- The proposed technique was experimented with using multiple images of various sizes (64×64 to 512×512). Excellent visual quality was achieved based on the experimental results (PSNR = 48.86 dB, SSIM = 0.99834, and UIQI = 0.99838).

This technique can be employed for practical purposes considering robustness against minor alterations or saving the image in various file formats.

3. Methodology

This paper implements the concept of block-based image steganography where the image is divided into disjoint 4x4 blocks. The use of small-sized blocks ensures that the embedding process occurs locally, thus reducing any possible visual degradation of the image. In addition, the use of the 4x4 block format allows for the representation of the position of the pixel being embedded using only two bits, hence ensuring that the index-based storage system is supported. Larger blocks will require increased storage space for indexing purposes and additional computations, whereas smaller blocks will limit the capacity of information being embedded in the image. This paper uses secret information such as text, grayscale images, and color images in a color image.

The next step after partitioning is categorizing the blocks into two categories based on their logical content. These include index blocks (I-Blocks) and embedding blocks (E-Blocks).

The purpose of the I-Blocks is to store information about the specific pixel positions from which the data will be embedded. On the other hand, the responsibility of the E-Blocks is to carry the data payload. Specifically, selected pixel least significant bits are modified in order to embed the data within the 4×4 E-Blocks. Meanwhile, the indices are stored in separate I-Blocks.

Embedding the payload is done in an organized way. For instance, depending on the need, the data could be embedded using only one-color channel or be spread across different color channels in order to avoid inter-channel correlations. Nevertheless, because changes are made at the level of least significant bits of pixels only, the color channel statistics are not disturbed.

3.1 Methodology Overview

The proposed method employs a number of steps that help in embedding and retrieving data. First, the cover image is split into separate 4x4 blocks. Some of the blocks are selected as I-Blocks to store positional information, while others are designated as E-Blocks to embed the secret message. A Three-bit LSB match algorithm is used to embed the data into the image. During decoding, the stored index helps in extracting embedded data bits from their correct pixel positions, thereby leading to the successful retrieval of hidden information.

Embedding data can be achieved by employing two approaches. One involves chossing one color component at a time to embed the data while the other involves distributing the process equally among all the components. This approach is adopted based on the notion that changes are made to the least significant bits. Figure 2 and 3 shows the embedding and extracting flow chart.

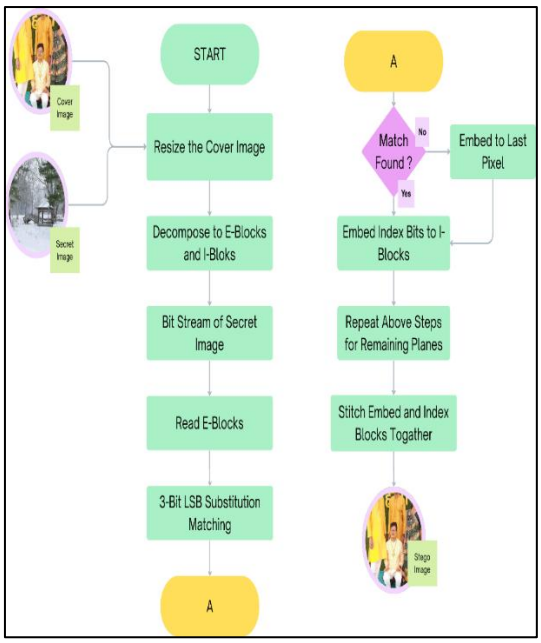

Figure 2. Embedding Flow Chart

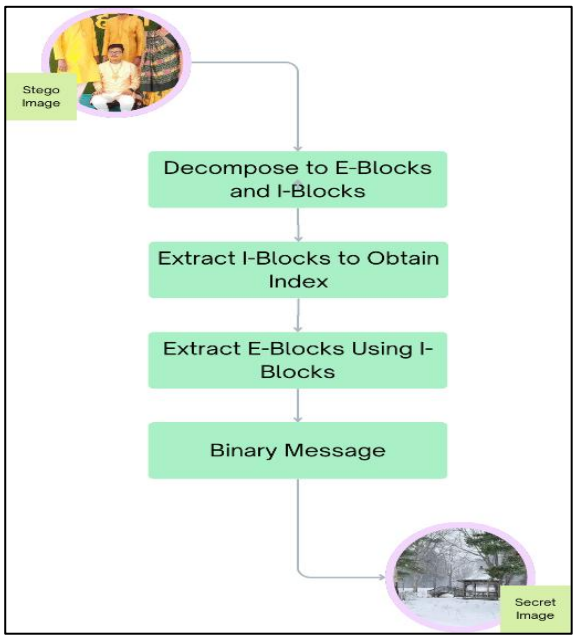

Figure 3. Extracting Flow Chart

3.2 Embedding Algorithm

The embedding process is done on the rows of the 4×4 E-blocks. In each row, three bits of information from the message are embedded independently of the others. The resulting

pixels will be visually undetectable and can be extracted deterministically because of the retained index information. Out of the different possible pixels with regard to the LSB condition, the first one from the left will be selected.

The choice of using three least significant bits as the embedding size is a compromise between embedding capacity and invisibility. If we were to use fewer bits, the embedding capacity would be reduced, whereas if we were to use more bits, there would be a higher chance of visibility. In the proposed method, all changes are made only in the least significant bit position in the pixel value, thus greatly minimizing the possibility of any visible block artifact. Since there is a minimum intensity change in pixels without any alteration of high-frequency content, any block edge discontinuity becomes practically insignificant.

Algorithm: Embedding Algorithm Using I-Blocks and E-Blocks

Input: Color cover image C of size M × N; Secret message S (text/image)
Output: Stego image C'

1. Partition C into non-overlapping 4 × 4 blocks
2. Classify first 25% of the blocks as I-Blocks, rest as E-Blocks
3. Convert S to binary → S_b
4. Let n ← length(S_b)
5. Append n at beginning of S_b: S_b ← append (n, S_b)
6. Pad S_b to make length divisible by 3
7. k ← 1
8. for each E-Block B: do
9. t ← S_b [k: k+2] (Next 3 bits)
10. k ← k + 3
11. r ← first row of B
12. match ← false
13. for i = 1 to 4: do
14. t1 ← 3 LSBs of r[i]
15. if t == t1: then
16. Save index i for B in the corresponding I-Block
17. match ← true
18. break
19. end if
20. end for
21. if not match: then
22. Overwrite 3 LSBs of r[4] with t
23. Save index 4 for B in corresponding I-Block
24. end if
25. if k > length(S_b): then
26. break
27. end if
28. end for
29. Embed saved indices into I-Blocks using 2 LSBs per pixel.

An index value represents the column number (1-4) of the selected pixel in each row, making it safe to represent the index value in terms of the two least significant bits only. The

index value is recorded sequentially without any collisions in the I-Block. This method of assignment makes it possible to locate each hidden data block uniquely at extraction time.

3.3 Illustrative Example of Embedding

Given: A 4 × 4 E-Block and a 12-bit secret message.

Original E-Block:

67	96	149	55
215	103	82	204
26	27	235	218
25	231	156	118

Secret Bits to Embed: 101 001 111 111

Selected pixels with matching 3 LSBs: (0,2), (1,3), (2,3), (3,1) → column indices = (2,3,3,1)

Modified E-Block (Stego):

67	96	149	55
215	103	82	201
26	27	235	223
25	231	156	118

I-Block (Original):

151	87	194	140
242	83	69	67
39	99	117	15
85	90	174	243

Modified I-Block:

150	87	195	141
242	83	69	67
39	99	117	15
85	90	174	243

3.4 Extracting Algorithm

Input: Stego image C' of size M × N
Output: Reconstructed secret message S

1. Partition C' into non-overlapping 4 × 4 blocks
2. Classify first 25% of the blocks as I-Blocks, rest as E-Blocks
3. Extract 2 LSBs from I-Blocks to retrieve index values
4. Initialize binary stream $S_b \leftarrow \emptyset$
5. for each E-Block B: do

6. Retrieve stored index i from corresponding I-Block
7. Extract 3 LSBs from i^{th} pixel in first row of B
8. Append extracted bits to S_b
9. end for
10. Extract message length n from the beginning of S_b
11. Extract next n bits and decode to get original message S

As the specific pixel location employed for embedding purposes is directly recorded, there will be no ambiguities during extraction, even in scenarios where more than one pixel carries an identical pattern of Least Significant Bits (LSBs).

3.5 Evaluation Metrics

To measure the degree of similarity and invisibility between the stego image and the cover image, two measures are often used, known as Peak Signal-to-Noise Ratio (PSNR) and structural Similarity Index Measure (SSIM).

3.5.1 Peak Signal-to-Noise Ratio (PSNR)

PSNR is an extensively employed measure that calculates the distortion caused by comparing the values of pixels of the original and steganographic images. The PSNR formula can be calculated in terms of MSE as follows:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[C(i,j) - C'(i,j)]^2 \quad (1)$$

Where

- C(i,j) is the pixel value at position (i,j) in the original image.
- C'(i,j) is the pixel value at position (i,j) in the stego image.
- M x N is the image dimension.

Using MSE, the PSNR is calculated as:

$$PSNR = 10 * log_{10}(\frac{MAX_I^2}{MSE}) \quad (2)$$

Where

- MAX_I is the maximum pixel value (255 for 8-bit images).
- Higher PSNR values (typically above 40 dB) indicate that the stego image is visually indistinguishable from the original.

3.5.2 Structural Similarity Index (SSIM)

SSIM measures the perceptual similarity between two images based on luminance, contrast, and structural information. It is computed as:

$$SSIM(C, C') = \frac{(2\mu_C\mu_{C'}+C_1)(2\sigma_{CC'}+C_2)}{(\mu_C^2+\mu_{C'}^2+C_1)(\sigma_C^2+\sigma_{C'}^2+C_2)} \quad (3)$$

Where

- μc and μc′ are the average intensities of images C and C′.
- $\sigma^2 c$, $\sigma^2 c'$ are the variances.
- σcc′ is the covariance between C and C′.
- C_1 and C_2 are constants to avoid division by zero (usually $C_1 = (0.01 \cdot L)^2$, $C_2 = (0.03 \cdot L)^2$ where L = 255).

SSIM values range from 0 to 1, where values closer to 1 denote higher structural similarity.

3.5.3 Universal Image Quality Index (UIQI)

In an image, pixel values available at different positions show different effects on the Human Visual System (HVS). If some distortion or changes are introduced in the image, such distortion is calculated as a combination of three factors; loss of correlation, contrast distortion, and luminance distortion.

$$UIQI(A,B) = L(A,B) * C(A,B) * S(A,B) \quad (4)$$

Where

- A is cover image, μA and σA are mean and standard deviation, respectively of A.
- B is stego image, μB and σB is mean and standard deviation, respectively of B.
- σAB is covariance between A and B.
- Luminance distortion, L (A, B) = $2\ \mu A\mu B/\mu A^2+\mu B^2$
- Contrast distortion, C (A, B) = $2\sigma A\sigma B/\sigma A^2 +\sigma B^2$
- Loss of correlation, S (A, B) = $2\sigma AB/\sigma A+\sigma B$

SSIM and UIQI are related perceptual quality metrics that assess structural similarity and visual distortion; however, they are distinct measures derived from similar perceptual principles rather than one being an improved version of the other.

3.5.4 Robustness Metrics

In this work, practical robustness refers to the ability of the steganographic scheme to reliably extract the embedded secret data after the stego image undergoes common and moderate image processing operations, such as noise addition or brightness and contrast variations. It does not imply resistance to aggressive distortions or strong lossy compression, which are known limitations of spatial-domain LSB-based methods.

In addition to visual quality metrics, robustness is evaluated to measure the reliability of secret extraction under common image processing operations. For secret text payloads, robustness is quantified using the Bit Error Rate (BER), defined as the ratio of incorrectly extracted bits to total embedded bits. For secret image payloads, Normalized Correlation (NC)

between the original and extracted secret images is used. Robustness experiments include Gaussian noise, salt-and-pepper noise, speckle noise, Poisson noise, and brightness and contrast variations.

3.5.5 Scalability

Since the cover image is partitioned into non-overlapping 4×4 blocks, the number of embedding and index blocks increases linearly with image resolution. Consequently, payload capacity scales proportionally with image size, while computational complexity remains linear with respect to the total number of pixels. Experimental results have been extended to include images ranging from 64×64 to 512×512, demonstrating consistent trends in PSNR, SSIM, and payload capacity across resolutions. This confirms that the proposed method is scalable and suitable for both low- and high-resolution images.

3.5.6 Normalized Correlation

$$NC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} C(i,j)C'(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} C(i,j)^2} \quad (5)$$

Where

- C (i, j) = pixel value of the cover image
- C’ (i, j) = pixel value of the stego image
- M×N = image size
- NC=3 / (NCR+NCG+NCB) R, G and B for Color Image Channels

3.5.7 Bit Error Rate

It is the ratio of erroneous extracted bits to the total number of embedded bits

$$BER = \frac{1}{N}\sum_{i=1}^{N} | b_i - b_i' | \quad (6)$$

Where

- bi = original secret bit
- bi′ = extracted secret bit

3.6 Security and Capacity Analysis

3.6.1 Security Discussion

The proposed method enhances security through indirection and controlled randomness. Instead of directly embedding secret bits in a sequential or fixed location, it:

- Embeds secret data only where the LSBs match the message bits,

- Separates index information into I-Blocks, which act as a key to locate embedded data,
- Embeds only 3 LSBs per selected pixel to avoid significant alteration,
- Avoids altering all blocks—only selected E-Blocks are used, increasing statistical resistance to steganalysis.

This strategy makes it significantly harder for attackers to use statistical or visual attacks to detect or extract the hidden message. Furthermore, additional layers of security such as encrypting the index stream or randomizing block assignment (based on a secret key) can be incorporated to further enhance protection.

Additionally, the embedding order of E-Blocks and the mapping between E-Blocks and I-Blocks can be randomized using a secret key-driven pseudo-random sequence. This randomization disrupts predictable embedding patterns and significantly increases resistance to steganalysis and unauthorized extraction.

3.6.2 Embedding Capacity

The payload capacity is an important performance metric that defines the total number of secret bits that can be embedded into a cover image without noticeable distortion. It is one of the three fundamental performance dimensions, along with imperceptibility and robustness. Higher payload capacity enables the transmission of larger amounts of secret data but must be carefully balanced to avoid perceptual degradation and loss of robustness. The proposed scheme achieves high payload capacity by leveraging LSB matching and index-based embedding, which significantly reduces pixel modifications. This balance between capacity, visual quality, and robustness has been clearly articulated to emphasize the practical relevance of payload capacity in real-world applications.

Each embedding block requires a small number of bits to store the position index of the matching pixel. Allocating one-quarter of the blocks for index storage provides sufficient capacity to store all required indices without overflow, while allowing the majority of blocks to be used for payload embedding. This allocation strikes a practical balance between index storage requirements and embedding capacity, ensuring reliable extraction while maintaining high payload efficiency.

Index values correspond to pixel positions within a row and are therefore limited to a small, fixed range. This ensures that index values can be represented safely using two bits without overflow. Additionally, each index is stored sequentially and mapped deterministically to index blocks, preventing collisions

Let the cover image be of size M × N pixels. Each 4 × 4 block contains 16 pixels. Assume:

- 25% of total blocks are I-Blocks (used for index storage),
- 75% are E-Blocks (used for embedding),
- 4 pixels per E-Block are used for embedding (1 per row),
- Each selected pixel embeds 3 bits using LSB substitution.

Then, the total number of E-Blocks is:

- E-Blocks = (3/4) × (MN / 16)

Each E-Block contributes 4 × 3 = 12 bits. Therefore, the total embedding capacity is:

- Capacitybits = (3/4) × (MN / 16) × 12 = (9MN / 16)

For example, for a 512 × 512 color image:

- Capacity = (9 × 512 × 512) / 16 = 147,456 bits = 18,432 bytes ≈ 18 KB.

This is sufficient for hiding text messages, grayscale images, or small-sized color images with minimal visual distortion.

4. Results and Discussion

Table 1. Sample Images Used as Cover-Secret Images

 1.jpg	 2.jpg	 3.jpg
 4.jpg	 5.jpg	 6.jpg
 7.jpg	 8.jpg	 9.jpg

Table 2. Sample Stego Images

Cover Image	Secret Message	Stego Image
 512x512.jpeg	 128x128.jpeg	 512x512.jpeg128x128.jpeg
 256x258.jpg	 64x64.tif	 256x256.jpg64x64.tif.png

Table 3. Nomenclatures for Table 3, Table 4 and Table 5

CIS	Cover Image Size in Bytes	SMS	Secret Message Size in Bytes
PSNR	Peak Signal to Noise Ratio in dB	SIS	Secret Image Size in Bytes
SSIM	Structure Similarity Index	SMPL	Secret Message Payload in Bytes
UIQI	Universal Image Quality Index	MCPL	Maximum Cover Pay Load in Bytes

Table 1 shows sample images used as cover-secret images, whereas Table 2 shows the sample stego images. Table 3 shows the nomenclatures used in Tables 3-5.

Table 4. Performance Metrics for Secret Message in Image Steganography

SN	Stego Image	CIS		PSNR	SSIM	UIQI	SMS	MCPL
1	64x64glena.png	64	64	56.13200	0.99982	0.99886	80	282
2	64x64gr1.jpg	64	64	60.50700	0.99990	1.00000	80	282
3	64x64j11.jpg	64	64	56.08300	0.99995	0.99999	262	282
4	64x64J27.jpg	64	64	60.72000	0.99985	1.00000	80	282
5	64x64gr3.jpg	64	64	60.93700	0.99995	1.00000	80	282
6	64x64b1.bmp	64	64	50.82200	0.99983	0.99959	262	282
7	64x64p13.png	64	64	58.84700	0.99946	1.00000	262	282
8	64x64gray.jpg	64	64	55.45400	0.99990	0.99999	262	282
9	64x64t1.tif	64	64	55.65100	0.99933	0.99999	80	282
10	64x64w1.jpg	64	64	59.88700	0.99998	1.00000	80	282
11	128x128gr2.jpg	128	128	61.26600	0.99995	0.99980	262	1146
12	128x128gr2.jpg	128	128	60.39100	0.99993	0.99973	320	1146
13	128x128j9.jpg	128	128	60.53100	0.99949	1.00000	320	1146
14	128x128j14.jpg	128	128	61.62800	0.99956	1.00000	262	1146
15	128x128p5.png	128	128	60.60900	0.99968	0.99999	320	1146
16	128x128glena.png	128	128	55.85600	0.99918	0.99932	320	1146
17	128x128p1.png	128	128	55.92900	0.99962	0.99997	320	1146
18	128x128t2.tiff	128	128	60.84300	0.99965	1.00000	320	1146
19	128x128b1.bmp	128	128	55.85800	0.99975	0.99988	320	1146
20	128x128t1.tif	128	128	55.80800	0.99819	0.99996	320	1146
21	256x256b1.bmp	256	256	53.04800	0.99897	0.99962	2515	4602
22	256x256p1.png	256	256	53.03200	0.99882	0.99992	2515	4602
23	256x256t1.tif	256	256	53.02800	0.99677	0.99999	2515	4602
24	256x256glena.png	256	256	53.18900	0.99861	0.99210	2515	4602
25	256x256p14.png	256	256	50.70200	0.98614	0.95852	2515	4602
26	256x256g5.gif	256	256	52.98700	0.99975	0.99862	2515	4602
27	256x256j14.jpg	256	256	57.83400	0.99904	0.99999	2515	4602
28	256x256p11.png	256	256	58.59100	0.99963	0.99917	2515	4602
29	256x256w7.jpg	256	256	57.78600	0.99984	0.99997	2515	4602
30	256x256t5.tif	256	256	54.41400	0.99831	1.00000	2515	4602
31	512x512b1.bmp	512	512	52.92500	0.99968	0.99966	10414	18426
32	512x512glena.png	512	512	52.97100	0.99938	0.99621	10414	18426
33	512x512p1.png	512	512	53.00000	0.99967	0.99992	10414	18426
34	512x512t1.tif	512	512	52.93300	0.99870	0.99997	10414	18426
35	512x512j19.jpg	512	512	57.67900	0.99960	1.00000	10414	18426
36	512x512p9.png	512	512	54.22600	0.99715	0.98461	10414	18426
37	512x512w8.jpg	512	512	57.56600	0.99969	0.99996	10414	18426
38	512x512j9.jpg	512	512	58.01200	0.99964	1.00000	10414	18426
39	512x512p15.png	512	512	57.17500	0.99991	0.99749	10414	18426
40	512x512w19.jpg	512	512	57.78900	0.99992	0.99999	10414	18426
41	128x1281.lena.tif	128	128	50.92200	0.99741	0.99996	1024	1146
42	256x2561.lena.tif	256	256	54.06200	0.99794	0.99998	2050	4602
43	256x2561.lena.tif	256	256	51.00600	0.99610	0.99997	4102	4602
44	512x5121.lena.tif	512	512	53.95500	0.99943	0.99998	8206	18426
45	512x5121.lena.tif	512	512	50.95500	0.99899	0.99996	16414	18426

46	256x2565.camera	256	256	69.59300	0.99995	1.00000	50	4602
47	256x2568.Kids.jpg	256	256	69.12700	0.99991	1.00000	50	4602
48	256x2567.Chips.jp	256	256	69.33200	0.99999	1.00000	50	4602
	Average			56.90829	0.99898	0.99838		

Table 5. Performance Metrics for Secret Gray Image in Image Steganography.

SN	Stego Image	CIS		SIS		PSNR	SSIM	UIQI	SMPL	MCPL
1	64x64j8.jpg	64	64	16	16	55.48800	0.99959	0.99999	768	846
2	64x64j11.jpg	64	64	16	16	56.06300	0.99994	0.99999	768	846
3	64x64J23.jpg	64	64	16	16	55.85100	0.99975	0.99997	768	846
4	64x64j22.jpg	64	64	16	16	55.81400	0.99989	0.99995	768	846
5	64x64j21.jpg	64	64	16	16	55.25300	0.99935	0.99522	768	846
6	64x64j20.jpg	64	64	16	16	55.95500	0.99844	0.99999	768	846
7	64x64p5.png	64	64	16	16	55.84100	0.99959	0.99997	768	846
8	64x64p11.png	64	64	16	16	56.14800	0.99980	0.99959	768	846
9	64x64w6.jpg	64	64	16	16	55.52500	0.99894	0.99997	768	846
10	64x64w2.jpg	64	64	16	16	57.18700	0.99898	1.00000	768	846
	Average					55.91250	0.99943	0.99946		
11	128x128j12.jpg	128	128	32	32	55.80700	0.99869	0.99999	3072	3438
12	128x128j15.jpg	128	128	32	32	55.87600	0.99886	0.99999	3072	3438
13	128x128j16.jpg	128	128	32	32	55.52500	0.99943	0.99993	3072	3438
14	128x128J28.jpg	128	128	32	32	56.13400	0.99871	0.99999	3072	3438
15	128x128p13.png	128	128	32	32	58.32300	0.99912	1.00000	3072	3438
16	128x128w25.jpg	128	128	32	32	55.52200	0.99981	0.99998	3072	3438
17	128x128w18.jpg	128	128	32	32	55.78600	0.99944	0.99999	3072	3438
18	128x128p15.png	128	128	32	32	55.60500	0.99941	0.99903	3072	3438
19	128x128p12.png	128	128	32	32	56.11100	0.99965	0.99999	3072	3438
20	128x128J35.jpg	128	128	32	32	55.74300	0.99965	0.99978	3072	3438
	Average					56.04320	0.99928	0.99987		
21	256x256j2.jpg	256	256	64	64	55.84100	0.99985	0.99997	12288	13806
22	256x256j13.jpg	256	256	64	64	55.72600	0.99859	0.99999	12288	13806
23	256x256w17.jpg	256	256	64	64	55.79100	0.99931	0.99999	12288	13806
24	256x256w3.jpg	256	256	64	64	55.73500	0.99870	0.99999	12288	13806
25	256x256p10.png	256	256	64	64	56.35800	0.99930	0.99999	12288	13806
26	256x256p3.png	256	256	64	64	55.91200	0.99966	0.99998	12288	13806
27	256x256w24.jpg	256	256	64	64	55.74000	0.99942	0.99999	12288	13806
28	256x256t2.tiff	256	256	64	64	55.73600	0.99871	0.99999	12288	13806
29	256x256w10.jpeg	256	256	64	64	55.92300	0.99964	0.99997	12288	13806
30	256x256j11.jpg	256	256	64	64	55.77500	0.99972	0.99998	12288	13806
	Average					55.85370	0.99929	0.99998		
31	512x512w11.jpg	512	512	128	128	55.74500	0.99941	0.99997	49152	55278
32	512x512w21.jpg	512	512	128	128	55.72100	0.99959	0.99934	49152	55278
33	512x512p14.png	512	512	128	128	48.42000	0.97050	0.92595	49152	55278
34	512x512w6.jpg	512	512	128	128	55.23400	0.99928	0.99997	49152	55278
35	512x512t2.tiff	512	512	128	128	55.79700	0.99967	0.99999	49152	55278
36	512x512p13.png	512	512	128	128	58.96900	0.99977	1.00000	49152	55278
37	512x512w6.jpg	512	512	128	128	55.24700	0.99929	0.99997	49152	55278
38	512x512J30.jpg	512	512	128	128	55.74800	0.99972	0.99998	49152	55278
39	512x512J35.jpg	512	512	128	128	55.85100	0.99975	0.99972	49152	55278
40	512x512w3.jpg	512	512	128	128	55.70000	0.99967	0.99998	49152	55278
	Average					55.24320	0.99667	0.99249		
			Average			55.64583	0.99869	0.99800		

Table 6. Performance Metrics for Secret Color Image in Image Steganography.

SN	Stego Image	CIS		SCIS		PSNR	SSIM	UIQI	SMPL	MCPL
1	64x64w2.jpg	64	64	16	16	46.57592	0.98442	0.99935	768	846
2	64x64j3.jpg	64	64	16	16	50.45410	0.99459	0.99916	768	846
3	64x64j7.jpg	64	64	16	16	50.16748	0.99413	0.99825	768	846
4	64x64j4.jpg	64	64	16	16	47.22206	0.99793	0.99829	768	846
5	64x64j13.jpg	64	64	16	16	50.27211	0.99621	0.99972	768	846
6	64x64j1.jpg	64	64	16	16	49.51807	0.99317	0.99402	768	846
7	64x64j19.jpg	64	64	16	16	49.80407	0.98772	0.99993	768	846
8	64x64j11.jpg	64	64	16	16	49.75653	0.99931	0.99969	768	846
9	64x64p5.png	64	64	16	16	49.26716	0.99498	0.99476	768	846
10	64x64t2.tiff1	64	64	16	16	50.46679	0.99669	0.99967	768	846
	Average					49.35043	0.99391	0.99828		
11	128x128j15.jpg	128	128	32	32	49.27141	0.99354	0.99918	3072	3438
12	128x128j15.jpg	128	128	32	32	49.75166	0.99350	0.99921	3072	3438
13	128x128j1.jpg	128	128	32	32	49.87294	0.99130	0.99456	3072	3438
14	128x128j14.jpg	128	128	32	32	48.85205	0.99158	0.99980	3072	3438
15	128x128J23.jpg	128	128	32	32	49.24575	0.99331	0.99930	3072	3438
16	128x128w6.jpg	128	128	32	32	47.93549	0.97880	0.99814	3072	3438
17	128x128J27.jpg	128	128	32	32	49.01661	0.99200	0.99945	3072	3438
18	128x128w21.jpg	128	128	32	32	50.90752	0.99475	0.99494	3072	3438
19	128x128p10.png	128	128	32	32	48.37402	0.99230	0.99980	3072	3438
20	128x128w19.jpg	128	128	32	32	51.22172	0.99697	0.99928	3072	3438
	Average					49.44492	0.99181	0.99837		
21	256x256w12.jpg	256	256	64	64	49.71431	0.99128	0.99928	12288	13806
22	256x256J35.jpg	256	256	64	64	51.39218	0.98838	0.99727	12288	13806
23	256x256p13.png	256	256	64	64	43.16056	0.97382	0.99359	12288	13806
24	256x256p15.png	256	256	64	64	52.50944	0.97411	0.94164	12288	13806
25	256x256J33.jpg	256	256	64	64	48.66874	0.99546	0.99860	12288	13806
26	256x256J27.jpg	256	256	64	64	49.41206	0.98978	0.99942	12288	13806
27	256x256p11.png	256	256	64	64	46.02495	0.98348	0.99710	12288	13806
28	256x256t2.tiff	256	256	64	64	49.48751	0.98908	0.99957	12288	13806
29	256x256w9.jpeg	256	256	64	64	49.32401	0.99256	0.99958	12288	13806
30	256x256w22.jpeg	256	256	64	64	50.25864	0.99349	0.99937	12288	13806
	Average					48.99524	0.98714	0.99254		
31	512x512j4.jpg	512	512	128	128	47.27869	0.99704	0.99582	49152	55278
32	512x512w6.jpg	512	512	128	128	48.47767	0.99140	0.99723	49152	55278
33	512x512p7.png	512	512	128	128	50.84619	0.99449	0.99882	49152	55278
34	512x512j19.jpg	512	512	128	128	49.02035	0.99304	0.99993	49152	55278
35	512x512t2.tiff	512	512	128	128	49.70767	0.99737	0.99954	49152	55278
36	512x512w9.jpeg	512	512	128	128	49.01887	0.99789	0.99940	49152	55278
37	512x512p5.png	512	512	128	128	49.37108	0.99391	0.99314	49152	55278
38	512x512J35.jpg	512	512	128	128	49.84027	0.99673	0.99707	49152	55278
39	512x512w7.jpg	512	512	128	128	50.29107	0.99832	0.99745	49152	55278
40	512x512p12.png	512	512	128	128	45.96359	0.99482	0.99802	49152	55278
	Average					48.98154	0.99550	0.99764		
			Average			49.19303	0.99209	0.99671		

Table 4, Table 5 and Table 6 show the Performance Metrics for Secret Message, Secret Gray Image and Secret Color Image on various Cover Images of different sizes and types in image steganography.

As can be seen from the results in Table 4, we achieve an average PSNR of 56.90829, SSIM of 0.99898 and UIQI of 0.99838 for variable sizes of the secret message as per Secret Message Payload (SMPL) given in bytes. The results indicate that higher PSNR values are

typically obtained for smaller payload sizes, while PSNR decreases gradually as the payload size increases for the same cover image. This behaviour is expected, as larger payloads require a greater number of embedded bits, resulting in increased pixel modifications.

Table 5 gives an average PSNR of 55.64583, SSIM of 0.99869 and UIQI of 0.99800. This table shows the results of embedding a secret gray image over gray or color cover images. Table 6 gives an average PSNR of 49.19303, SSIM of 0.99209, and UIQI of 0.99671. This table shows the result of embedding a secret color image over color cover image.

Table 7. Comparison Between the Proposed and Other Methods: Gray Covers

	[3]				**[12]**			
Q = 3, Grayscale Cover Images	**MSE**	**PSNR**	**SSIM (Q=4)**	**BPP**	**MSE**	**PSNR**	**SSIM**	**BPP**
128 x 128	2.277	44.6754	-	-	-	42.15	-	-
256 x 256	2.258	44.1502	0.9939	-	-		-	-
512 x 512	1.96	45.2079	-	-	-		-	-
	[13]				**Proposed**			
Q = 3, Grayscale Cover Images	**MSE**	**PSNR**	**SSIM**	**BPP**	**MSE**	**PSNR**	**SSIM**	**BPP**
128 x 128	-	37.1980			-	56.0432	0.99924	0.5
256 x 256	-				-	55.8537	0.99928	0.5
512 x 512	-				-	55.2432	0.99667	0.5

Table 7 shows a comparison of [3], [12] and [13] with the proposed method with respect to PSNR. It can be seen that the PSNR obtained as per [3] is around 45.0, 42.15 as per [12] and 37.1980 as per [13]. As can be seen from the table, we obtain a PSNR of around 55.85, which shows a significant improvement over the existing methods mentioned in the table.

Table 8. Comparison Between the Proposed and Other Methods: Color Covers

	[17], [22], [23], [29], [30]				**Proposed**			
Cover Image Size (Secret Image Size)	**MSE**	**PSNR**	**SSIM (Q=4)**	**BPP**	**MSE**	**PSNR**	**SSIM (Q=3)**	**BPP**
128 x 128 Color (32 x 32 Secret) [17]	-	44.534	0.9848	1.5	-	49.44492	0.99181	1.5
128 x 128 Color [22]	-	30.730	0.9845	-				
256 x 256 Color (64 x 64 Secret) [17]	-	44.747	0.9870	1.5	-	48.99524	0.98714	1.5
256 x 256 Color [23]	-	40.390	0.9908	0.333				
512 x 512 Color (128 x 128 Secret) [17]	-	44.874	0.9965	1.5	-	48.98154	0.99550	1.5
512 x 512 Color (128 x 128 Secret) [29]	2.717	40.310	0.9451	1.318				
512 x 512 Color (128 x 128 Secret) [30]	-	31.200	0.8220	-				

Table 8 shows a comparison of [17], [22], [23], [29] and [30] for different sizes of cover and secret images, along with the respective PSNR and SSIM Values. The results clearly indicate an improvement in the PSNR and SSIM.

Table 9. Comparison of the Proposed and Other Methods: Secret Message Length and LSB

	[6]			Proposed Method	
Cover Image / Size	**Secret Message Length**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Cameraman.jpg	50 Bytes	61.42	1.000	69.593	1.000
Kids.jpg	50 Bytes	64.68	1.000	69.127	1.000
coloredchips.jpg	50 Bytes	66.32	1.000	69.332	1.000
	[7]			**Proposed Method**	
128 x 128	1KB	45.610	0.974	50.922	0.997
256 x 256	2KB	47.080	0.985	54.062	0.998
256 x 256	4KB	44.710	0.097	51.006	0.996
512 x 512	8KB	49.330	0.980	53.955	0.996
512 x 512	16KB	45.960	0.097	50.955	0.999
LSB Method	**[20]**			**Proposed Method**	
1 LSB - Color	-	42.750		NA	-
1 LSB - Gray	-	42.760		NA	-
4 LSB - Color	-	29.280		-	-
4 LSB - Gray	-	29.280		-	-
3 LSB - Color	-	NA		49.193	-
3 LSB - Gray	-	NA		55.645	-

[6], [7] and [20] are compared in Table 9 in terms of size and LSB methods as stated. Here, also obtained improved results for PSNR and SSIM. [6] shows an impactful PSNR of 69.127 compared to 61.42 for the same size of Cover and Secret message. Likewise, one can note that very good results are obtained compared to [7] using the proposed method. [20], shows that using 1 LSB for embedding purposes yields a PSNR of 42.750, and 29.280 PSNR using 4 LSB. We can see that as embedding capacity is increased, PSNR s significantly decreases. However, in the proposed approach, even with increased embedding capacity using 3 LSB, the PSNR obtained is notably 49.193 for color and 55.645 for gray cover images.

All cover images are 8-bit color/gray images stored in lossless formats such as PNG, JPG, TIFF or BMP unless otherwise stated. Embedding success depends on local image content: textured regions generally offer a higher matching probability, whereas smooth regions may require more pixel modification.

The PSNR primarily depends on the number of pixels modified during embedding. When fewer bits are embedded, fewer pixel values are altered, resulting in higher PSNR. Although the 3-LSB strategy increases embedding capacity, imperceptibility is preserved because pixel values are modified only when no matching LSB pattern exists. Since index-based matching avoids unnecessary changes, the proposed method maintains consistently high visual quality across different payload sizes and image formats. Robustness experiments indicate that the proposed scheme maintains reliable extraction under common noise conditions and brightness/contrast adjustments. However, like most spatial-domain LSB-based approaches, the method is sensitive to aggressive lossy compression such as JPEG, which may disturb the least significant bits. This limitation is acknowledged and motivates future extensions toward hybrid spatial-frequency domain designs.

Smooth regions may exhibit repetitive LSB patterns, which can influence matching probability. However, this behavior does not compromise extraction correctness because the exact index of the selected pixel is stored.

Table 10. Entropy, Histogram Characteristics and Noise Level of Various Datasets

Dataset	Entropy Present	Histogram Diversity	Noise Level
DIV2K	High	Wide	Low–Moderate
LFW Funneled	Low–Medium	Narrow (skin tones)	Low
USC-SIPI	Low → High (image-dependent)	Controlled & diverse	Very Low
ImageNet-mini	Low → High	Very Wide	Moderate

Table 11. Bit Error Rate and Normalized Correlation Between Cover and Stego Images (Noiseless)

Data Set	Size of Cover	Cover	Secret	BER (Cover and Stego)	BER(Secret and Extracted)	NC(Cover and Stego)	NC(Secret and Extracted)
DIV2K	64x64	0001	16x160002.png	0.043569	0.000000	0.999635	1.000000
	64x64	0002	16x160003.png	0.043538	0.000000	0.999931	1.000000
	64x64	0003	16x160004.png	0.042826	0.000000	0.999999	1.000000
	64x64	0004	16x160005.png	0.043915	0.000000	0.999995	1.000000
	128x128	0005	32x320006.png	0.041751	0.000000	1.000000	1.000000
	128x128	0006	32x320007.png	0.041967	0.000000	0.999777	1.000000
	128x128	0007	32x320008.png	0.042130	0.000000	0.999936	1.000000
	128x128	0008	32x320009.png	0.042699	0.000000	0.999759	1.000000
	128x128	0005	32x320013.png	0.041934	0.000000	0.999999	1.000000
LFW-Funneled	256x256	Eckhart	64X64Guiel	0.044919	0.000000	0.997134	1.000000
	256x256	Guiel	64X64Patterson	0.046226	0.000000	0.998312	1.000000
	256x256	Patterson	64X64Peirsol	0.042735	0.000000	0.999969	1.000000
	256x256	Peirsol	Pena	0.043781	0.000000	0.999984	1.000000
USC-SIPI-aerials	512x512	2.1.01	128x1282.1.02	0.041824	0.000000	0.999995	1.000000
	512x512	2.1.02	128x1282.1.03	0.041708	0.000000	0.999998	1.000000
	512x512	2.1.03	128x1282.1.04	0.042155	0.000000	0.999764	1.000000
	512x512	2.1.04	128x1282.1.05	0.042014	0.000000	0.999865	1.000000
Imagenet-mini	512x512	n01534433_1	128x128n01534433_2	0.039660	0.000000	0.999988	1.000000
	512x512	n01534433_2	128x128n01534433_3	0.042012	0.000000	1.000000	1.000000
	512x512	n01534433_3	128x128n01534433_4	0.042171	0.000000	0.999958	1.000000
	512x512	n01534433_4	128x128n01534433_5	0.041349	0.000000	1.000000	1.000000
USC-SIPI-misc	512x512	4.1.0.1	128x1284.1.0.2	0.043201	0.000000	0.999189	1.000000
	512x512	4.1.0.2	128x1284.1.0.3	0.044518	0.000000	0.998240	1.000000
	512x512	4.1.0.3	128x1284.1.0.4	0.045120	0.000000	0.999995	1.000000
	512x512	4.1.0.4	128x1284.1.0.5	0.043128	0.000000	0.999978	1.000000
Average				0.042834	0.000000	0.999656	1.000000

Various types and sizes of images [37] from the DIV2K, LFW FUNNELED, USC-SIPI, and IMAGENET-MINI datasets were used to measure Bit Error Rate (BER) and Normalized Correlation (NC) as shown in Table 10. The BER and NC between the cover image and the stego image were found to be ~0.042834 and ~0.999656. Whereas BER and NC between the original and extracted secret image obtained without introduction of noise is ~0.000 and 1.000 as mentioned in Table 11. To note that a complete color secret cover of one fourth the size of the color cover image was used to perform experiments to obtain BER and NC.

5. Future Work

Despite the effectiveness of the presented algorithm in regard to its imperceptibility, capacity, and general practical robustness, there is room for extending this method to other areas in the future. First of all, since the current technique does not cope well with lossy compression techniques like JPEG, which might affect the least significant bits and consequently impact data extraction, a promising solution could be adding some frequency domain-based methods, such as DWT-based or hybrid spatial-frequency embedding methods. Another option is to extend this approach to animated pictures or video steganography. The method could be made more effective by using an adaptive mechanism for choosing pixels and generating an index based on particular properties of the selected area. Simple ML models can be used to select areas with high enough entropy to make an image invisible even after data insertion. It could also be beneficial to integrate some encryption and error-correction coding to improve robustness in cases of low visibility. Another potential application might include password or biometric-controlled embedding.

Furthermore, the suggested approach is highly applicable in modern technologies like the Internet of Things (IoT) and cloud computing. Given that resources are limited in the case of IoT and that most data transmitted through IoT involves visual information, the use of a lightweight technique is ideal in ensuring that such information is safely shared in real time. In terms of cloud computing, the approach may be utilized to incorporate ownership or access control information into the images before storing them in the cloud.

6. Conclusion

This research proposes a novel steganography algorithm that is highly efficient and robust, utilizing 4×4-pixel block partitioning and indexed Least Significant Bits (LSB). This proposed algorithm uses coordinated embedding (E-Blocks) and index (I-Blocks) blocks, wherein secret data can be embedded into matching pixels without causing noticeable changes to the overall image appearance. The 3-LSB embedding scheme allows for either text or secret images to be embedded while maintaining a balance between invisibility, payload capacity, and security. Results have shown a great degree of accuracy when embedding grayscale and color images, with PSNR values reaching 55.65 and 49.19 dB, respectively. With this algorithm, users are guaranteed to extract secret data even under alterations, allowing the embedding of 999,999 characters or secret images in high resolution.

References

[1] Das, Rig, and Themrichon Tuithung. "A Novel Steganography Method for Image Based on Huffman Encoding." In 2012 3rd national conference on emerging trends and applications in computer science, IEEE, 2012, 14-18.

[2] Por, Lip Yee, Delina Beh Yin, Tan Fong Ang, and Sim Ying Ong. "An Enhanced Mechanism for Image Steganography Using Sequential Colour Cycle Algorithm." Int. Arab J. Inf. Technol. 10, no. 1 (2013): 51-60.

[3] Khandelwal, Pulkit, and Neha Bisht. "Randomly Hiding Secret Data Using Dynamic Programming for Image Steganography." In 2015 International Conference on Computing and Network Communications (CoCoNet), IEEE, 2015, 777-783.

[4] Zhang, Ru, Shiqi Dong, and Jianyi Liu. "Invisible Steganography via Generative Adversarial Networks." Multimedia tools and applications 78, no. 7 (2019): 8559-8575.

[5] Ogras, Hidayet. "An Efficient Steganography Technique for Images Using Chaotic Bitstream." International Journal of Computer Network and Information Security 12, no. 2 (2019): 21.

[6] Mahana, Sumit Kumar, and Rajesh Kumar Aggarwal. "Image Steganography: Analysis & Evaluation of Secret Communication." In proceedings of international conference on sustainable computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India. 2019.

[7] Siddiqui, Ghazanfar Farooq, Muhammad Zafar Iqbal, Khalid Saleem, Zafar Saeed, Adeel Ahmed, Ibrahim A. Hameed, and Muhammad Fahad Khan. "A Dynamic Three-Bit Image Steganography Algorithm for Medical And E-Healthcare Systems." IEEE Access 8 (2020): 181893-181903.

[8] Eyssa, Asmaa Abdelmonem, Fathi Elsaid Abdelsamie, and Abdelaziz Elsaid Abdelnaiem. "An Efficient Image Steganography Approach Over Wireless Communication System." Wireless personal communications 110, no. 1 (2020): 321-337.

[9] Liu, Qiang, Xuyu Xiang, Jiaohua Qin, Yun Tan, and Yao Qiu. "Coverless Image Steganography Based on DenseNet Feature Mapping." EURASIP Journal on Image and Video Processing 2020, no. 1 (2020): 39.

[10] Li, Mingjie, Zichi Wang, Haoxian Song, and Yong Liu. "Disguise of Steganography Behaviour: Steganography Using Image Processing with Generative Adversarial Network." Security and Communication Networks 2021, no. 1 (2021): 2356284.

[11] Sivaramakrishnan, Upasana, Namrata Panga, and G. K. Rajini. "Image Steganography based on Fractional Random Wavelet Transform and Arnold Transform with Cryptanalysis." In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2021, 618-623.

[12] Kamil, Samar, Siti Norul Huda Sheikh Abdullah, Mohammad Kamrul Hasan, and Farah Aqilah Bohani. "Enhanced Flipping Technique to Reduce Variability in Image Steganography." IEEE Access 9 (2021): 168981-168998.

[13] Sravani, S., and R. Raniith. "Image Steganography for Confidential Data Communication." In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2021, 01-05.

[14] Shunnar, Majd, Amaal Othman, and Ahmed Awad. "A Study to Improve Image Steganography Using Linear Feedback Shift Register." In 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE, 2022, 344-348.

[15] Ahmad, Mostafa A., Mourad Elloumi, Ahmed H. Samak, Ali M. Al-Sharafi, Ali Alqazzaz, Monir Abdullah Kaid, and Costas Iliopoulos. "Hiding Patients' Medical Reports Using an Enhanced Wavelet Steganography Algorithm in DICOM Images." Alexandria Engineering Journal 61, no. 12 (2022): 10577-10592.

[16] Meng, Yanger, and Jingtao Li. "Image Steganography of Convolutional Neural Network Based on Neural Architecture Search." (2022).

[17] Patel, Anilkumar, and Daxa Vekariya. "Randomly Hiding Secret Data Using I-Blocks and E-Blocks for Image Steganography." In Futuristic Trends in Networks and Computing Technologies: Select Proceedings of Fourth International Conference on FTNCT 2021, Singapore: Springer Nature Singapore, 2022, 375-390.

[18] Ye, Hanmin, Keqin Su, Xiaohui Cheng, and Shiming Huang. "Research on Reversible Image Steganography of Encrypted Image Based on Image Interpolation and Difference Histogram Shift." IET Image Processing 16, no. 7 (2022): 1959-1972.

[19] Patel, Anil Kumar, and Daxa Vekariya. "A Literature Review on Image Quality and Embedding Payload for Image Steganography." In AIP Conference Proceedings, vol. 2855, no. 1, AIP Publishing LLC, 2023, 060018.

[20] Jebur, Sabah Abdulazeez, Abbas Khalifa Nawar, Lubna Emad Kadhim, and Mothefer Majeed Jahefer. "Hiding Information in Digital Images Using LSB Steganography Technique." International Journal of Interactive Mobile Technologies 17, no. 7 (2023).

[21] Alanzy, May, Razan Alomrani, Bashayer Alqarni, and Saad Almutairi. "Image Steganography Using LSB and Hybrid Encryption Algorithms." Applied Sciences 13, no. 21 (2023): 11771.

[22] Zheng, Ziqiang, Yuanmeng Hu, Yi Bin, Xing Xu, Yang Yang, and Heng Tao Shen. "Composition-Aware Image Steganography Through Adversarial Self-Generated Supervision." IEEE Transactions on Neural Networks and Learning Systems 34, no. 11 (2022): 9451-9465.

[23] Yang, Jianhua, Fei Shang, Yi Liao, and Yifang Chen. "Toward High Capacity and Robust Jpeg Steganography Based on Adversarial Training." Security and Communication Networks 2023, no. 1 (2023): 3813977.

[24] Shmueli, Ron, Divya Mishra, Tal Shmueli, and Ofer Hadar. "A Novel Technique for Image Steganography Based on Maximum Energy Seam." Multimedia Tools and Applications 83, no. 28 (2024): 70907-70920.

[25] EL-Hady, Mohamed, Maha H. Abbas, Farooq A. Khanday, Lobna A. Said, and Ahmed G. Radwan. "DISH: Digital Image Steganography Using Stochastic-Computing with High-Capacity." Multimedia Tools and Applications 83, no. 25 (2024): 66033-66048.

[26] Li, Li, Xinpeng Zhang, Kejiang Chen, Guorui Feng, Deyang Wu, and Weiming Zhang. "Image Steganography and Style Transformation Based on Generative Adversarial Network." Mathematics 12, no. 4 (2024): 615.

[27] Yanuar, Muhammad Rafly, Suryadi MT, Catur Apriono, and Muhammad Firdaus Syawaludin. "Image-to-Image Steganography with Josephus Permutation and Least Significant Bit (LSB) 3-3-2 Embedding." Applied Sciences 14, no. 16 (2024): 7119.

[28] Jayapandiyan, Jagan Raj, C. Kavitha, and K. Sakthivel. "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization." IEEE Access 8 (2020): 136537-136545.

[29] Wahab, Osama Fouad Abdel, Ashraf AM Khalaf, Aziza I. Hussein, and Hesham FA Hamed. "Hiding Data Using Efficient Combination of RSA Cryptography, And Compression Steganography Techniques." IEEE access 9 (2021): 31805-31815.

[30] Solak, Serdar. "High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms." IEEE Access 8 (2020): 166513-166524.

[31] Chuang, Yun-Hsin, Bor-Shing Lin, Yan-Xiang Chen, and Hung-Jr Shiu. "Steganography in RGB Images Using Adjacent Mean." IEEE access 9 (2021): 164256-164274.

[32] Kanimozhi, R., and V. Padmavathi. "Robust and Secure Image Steganography with Recurrent Neural Network and Fuzzy Logic Integration." Scientific Reports 15, no. 1 (2025): 13122.

[33] Zhou, Yiqiao, Na Wang, Xiaolong Hong, Yanchun Peng, and Shuo Shao. "Deep Learning-Based Image Steganography with Latent Space Embedding and Smart Decoder Selection." Entropy 27, no. 12 (2025): 1223.

[34] Malik, Kaleem Razzaq, Muhammad Sajid, Ahmad Almogren, Tauqeer Safdar Malik, Ali Haider Khan, Ayman Altameem, Ateeq Ur Rehman, and Seada Hussen. "A Hybrid Steganography Framework Using DCT and GAN for Secure Data Communication in the Big Data Era." Scientific Reports 15, no. 1 (2025): 19630.

[35] Ambika, Virupakshappa, and Deepak S. Uplaonkar. "Deep Learning-Based Coverless Image Steganography on Medical Images Shared via Cloud." Engineering Proceedings 59, no. 1 (2024): 176.

[36] Rahman, Shahid, Jamal Uddin, Hameed Hussain, Sabir Shah, Abdu Salam, Farhan Amin, Isabel de la Torre Díez, Debora Libertad Ramirez Vargas, and Julio Cesar Martinez Espinosa. "A Novel and Efficient Digital Image Steganography Technique Using Least Significant Bit Substitution." Scientific Reports 15, no. 1 (2025): 107.

[37] https://www.kaggle.com/datasets

[38] Vasoya, Daxa L., Vipul M. Vekariya, and P. P. Kotak. "Novel Approach for Image Steganography Using Classification Algorithm." In 2018 2nd international conference on inventive systems and control (ICISC), IEEE, 2018, 1079-1082.

[39] Siddhpura, Akash, and V. Vekariya. "An Approach of Privacy Preserving Data Mining Using Perturbation & Cryptography Technique." International Journal on Future Revolution in Computer Science & Communication Engineering 4 (2018): 255-259.