IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 5, October, 2024, pp. 1731-1741
Journal homepage: http://iieta.org/journals/isi

Steganography in Spatial Domain Images: Using Image Edge to Hide the Secret Data with a Quality Stego Image

I Putu Bagus Gede Prasetyo Raharja[1], Deka Julian Arrizki[1], Riki Mi'roj Achmad[1], Ntivuguruzwa Jean De La Croix[1,2], Tohari Ahmad[1*]

[1] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2] African Center of Excellence in Internet of Things, College of Science and Technology, University of Rwanda, Kigali 3900, Rwanda

Corresponding Author Email: tohari@if.it.ac.id

https://doi.org/10.18280/isi.290507

Received: 10 December 2023
Revised: 16 July 2024
Accepted: 29 July 2024
Available online: 24 October 2024

Keywords:
information security, network infrastructure, national security, steganography, spatial domain images, image edge detection

ABSTRACT

Securing communication in our highly digitalized world has become a pressing issue due to the escalating threats of unauthorized data access and violations of network policies. Cryptographic techniques are employed to encrypt data for protection to address these challenges. However, a potential vulnerability arises during data transmission. Sophistic intruders may discern the encrypted information, leading to suspicions and unauthorized access. In response, steganography emerged as an alternative method for communication security. Steganography involves concealing confidential information within the codes of digital files, providing a unique approach that focuses on disguising the presence of communication to enhance data security. In this context, this paper introduces an enhanced information-hiding method implemented by utilizing image edges and modulus functions. This study provides a comparative analysis of various steganographic methods, highlighting the trade-offs between image quality, as evaluated by the Peak Signal-to-Noise Ratio (PSNR), and the payload size. The experimental results indicate that the proposed method has efficient data-hiding capabilities with minimum degradation in the quality of the resulting stego image.

1. INTRODUCTION

In the rapidly evolving landscape of information security, the field of information hiding, which includes cryptography and steganography, stands as a critical discipline encompassing diverse research areas. While cryptography and steganography share the overarching goal of safeguarding sensitive information, they exhibit distinct conceptual frameworks. Cryptography, a well-established practice, involves data encryption to ensure secure communication. However, it does not inherently conceal the existence of the communication itself, allowing encrypted data to be observed by third parties during transmission [1]. This potential susceptibility to interception raises concerns about confidentiality. In stark contrast, steganography operates with the explicit aim of preventing the detection of communication by embedding information within the digital fabric of files, such as audio [2], text [3], photos [4], and videos [5]. This deliberate concealment ensures that only the intended participants in communication are privy to the exchange, offering an additional layer of security. The main component of a steganographically-based communication system involves key elements: the cover, used for carrying the secret bits of the confidential information for transmission; the hidden bits, considered as the bitstreams making the confidential information; and the stego, resulting from combining the cover and the secret information.

Considering the substantial redundancy inherent in digital images, numerous steganographic techniques for concealing data have been discussed in existing literature. The study [6] proposed a method where pixels for hiding secret data are chosen randomly; post-processing of the stego media is executed using a hybrid fuzzy difference expansion through an adaptive approach embedding data in regions of interest within the cover image. Based on the sophistication of the presence of the hidden data, several other research works [7-11] have been proposed to improve the visibility quality of the stego image after concealing a substantial amount of data. In the research [10], a steganographic method has been proposed that enhances the visual quality of the stego image by arranging two pixels in ascending order and applying an optimal pixel adjustment procedure. Their approach involved sorting the pixels of the cover image in ascending order before concealing the secret data, resulting in improved visibility quality with enhanced imperceptibility of the distortion in the stego image.

Further research by Abdollahi et al. [12] emphasizes the selection of embedding positions in smooth and edge areas before concealing data. To ensure data security, secret sharing, and steganography were applied to embed data into images. Nevertheless, the existing steganographic approaches present several drawbacks, mainly based on the non-optimal use of the cover image's pixels, which results in the distortion of the

visibility quality of the stego image. This makes the steganographic algorithms vulnerable to steganalysis-based attacks of different forms [13-18]. To address this while addressing the image's distortion issues, the steganographic algorithms always seek to minimize the trade-off between the payload size concealment and the quality of the stego image.

Considerable efforts are underway to fortify the security of the steganography process by incorporating a steganography key. In the study by Al-Jarah and Arjona [19], a steganographic method was introduced that uses a novel approach to enhance the security aspect of steganography. The technique proposed in that paper involves the utilization of a confidential steganography key. Simultaneously, a concerted effort is to leverage edge detection techniques to augment steganography's embedding capacity and overall security [6]. However, the reliance on an external key transmitted to the receiver via a communication link introduces vulnerability to interception and compromise.

To address the challenges in steganography, particularly the optimal use of the pixels within cover images, this research presents a new approach to bolster security by combining picture edges and the modulus function. The primary goal of this method is to enhance both the embedding capacity and the security layer without transmitting the key through external channels. Employing a fuzzy logic-based approach, the study generates image edges to extract the original secret. During the extraction phase, the modulus function is applied, using edge images as a key to retrieve concealed sensitive data. The overarching objective is to augment embedding capacity and security while maintaining an acceptable level of quality in the resulting stego picture. This proposed system is implemented within the spatial domain, and it offers significant improvements over previous techniques by eliminating the need for external key transmission and optimizing pixel usage to minimize distortion. The contribution of this paper is summarized in these three points:

(1) Introducing a new approach combining picture edges and the modulus function to improve security and embedding capacity.

(2) Utilizing fuzzy logic-based edge generation and modulus function for data retrieval eliminates the need for external key transmission.

(3) Implementing the system in the spatial domain to optimize pixel usage, minimize distortion, and maintain the quality of the stego picture.

The remaining sections of this paper delve into the literature study in Section 2, followed by an elucidation of the proposed method in Section 3. Section 4 presents the results and discussion, and the paper concludes in Section 5.

2. RELATED WORKS

In recent years, significant strides have been made in digital image steganography, driven by the integration of various technologies. Notably, there has been a commendable advancement in performance, particularly with the emergence of intelligent algorithms designed to secure secret information. These algorithms, categorized based on their application domain, can be broadly delineated into spatial domain and transform (frequency) domain techniques. The concealment of confidential data within the spatial domain involves a direct manipulation of the pixel values of the cover image to achieve the desired enhancement [8]. This approach signifies a noteworthy achievement in steganographic techniques, showcasing the effectiveness of spatial domain methodologies in ensuring the security of secret information.

Several extant methodologies exist for steganographic techniques, particularly in the spatial domain. The Least Significant Bit (LSB) approach, renowned for its simplicity within practitioners' circles, involves substituting the LSB of each pixel during the embedding process [19]. Despite its advantageous payload capacity, this method is susceptible to various picture-processing processes, including compression and cropping [20]. AlKhodaidi and Gutub [21] introduced a technique to refine secret data distribution to enhance steganographic security through secret sharing. Adaptive image refining (AIR) is proficient at securing confidential data distribution, making it suitable for information security. However, certain AIR techniques encounter boundary issues, potentially affecting data extraction and image quality.

Moreover, several other research works have been proposed to enhance payload capacity while preserving image quality. Introducing a novel approach, Gaurav and Ghanekar [22] incorporate dilated hybrid edge detection on the three most significant bits (MSB) of cover images, amplifying data embedding capability within steganography's domain. In this work, two innovative approaches to Reversible Data Hiding (RDH) in image steganography address low embedding capacity challenges. The first approach enhances dual image-based Least Significant Bit (LSB) matching with reversibility, maintaining stego-image quality while enabling complete data recovery. The second approach combines n-rightmost bit replacement (n-RBR) with modified pixel value differencing (MPVD) using four identical cover images, presenting significant improvements in RDH [22].

Furthermore, the research work [23] introduced an improved method for hiding data by combining difference expansion and the modulus function. This research puts forward a new approach using difference expansion and the modulus function for concealing data in the spatial domain. The primary aim was to enhance the amount of data that can be hidden while maintaining a reasonable quality in the stego image. This method hides data in negative and positive variations between adjacent pixels. Two ranges were defined: negative values from negative two to zero and positive values from zero to positive two, outlining acceptable variations for data embedding. A modulus two operation was applied to the stego image pixels to retrieve data using this method. However, a limitation of this approach is its use of a narrow range of differences, impacting the data embedding capacity, as many pixels are left unused for data hiding. Hence, this method may not be suitable for concealing large data. The research proposed [11] developed a different steganographic algorithm, utilizing a difference expansion paradigm to balance the trade-off between payload size and stego image distortion. Their approach suggests concealing a secret message through two steps. Firstly, differences between the neighboring pixels within the cover image are computed, forming the foundation for data hiding. Secondly, a technique called difference expansion embeds data within the obtained differences. Difference expansion conceals secret data bits within the calculated differences between neighboring pixels in the same pair of the cover image.

Recent advancements in image steganography on increasing security spatial domain have been researched. Liu et al. [24] introduced a novel distortion cost function utilizing quaternion representation to improve spatial image steganography. Their

approach defines image complexity through quaternion magnitude and phase, resulting in a distortion cost function that efficiently allocates embedding modifications in complex image regions. This method demonstrates superior security performance compared to state-of-the-art schemes like S-UNIWARD, HiLL, and MiPOD. Additionally, their generalized QMP (GQMP) model further enhances security by balancing the effects of quaternion magnitude and phase using an exponential model.

Drawing inspiration from existing works, our newly proposed algorithm in this study is rooted in edge computation, employing the grayscale images from the commonly used dataset known as the SIPI database [25] to validate our method experimentally.

3. METHODOLOGY

The proposed method suggests a new steganographic scheme consisting of computing for the image's edge to be used as a secret key. This approach utilizes the unique features of edge images to embed hidden information within the pixels for the system's robustness.

3.1 Fuzzy edge detection

In the context of this research, a preliminary phase precedes the embedding process, wherein both the cover image and the secret message undergo rigorous mathematical processing. Specifically, applying fuzzy logic to cover images is employed to generate a key matrix. Subsequently, the edge detection method is executed systematically, encompassing four distinct steps, as elucidated below:

Step 1: Compute Gradients

Gradients play a foundational role in the detection of edges within images. In image analysis, an edge denotes a substantial alteration in intensity or color, corresponding to elevated gradient values. Through the computation of gradients, the analytical function identifies regions with pronounced changes indicative of potential edges. The gradients are distinctly calculated in both the horizontal (D_y) and vertical (D_x) directions. Sobel operators are employed for the precise computation of these gradients. The Sobel operator matrix is depicted in Eqs. (1)-(2), where (G_x) and (G_y) represent the horizontal and vertical gradients, respectively.

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (1)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

Step 2: Fuzzified Gradients

Fuzzified gradients are computed in this step by applying Gaussian Membership Functions (MF) to the gradients, resulting in the Sobel Operator. We calculate three levels of gradient intensity (low, middle, and high) for both. D_x and D_y Using predefined means and a standard deviation. The Gaussian MF transforms each gradient value into a fuzzy value, indicating how strongly it belongs to each intensity level. Calculating different levels of gradient intensity allows a more sophisticated and accurate interpretation of the edges of the image. The formula to calculate Gaussian MF is shown in Eq. (3), where $\mu(x)$ is the membership value for the input x, and e represents the base of a natural algorithm. We use different mean and standard deviation values on the membership function calculation for each gradient level. To calculate three levels of gradient intensity, we will use the relations for the vertical gradient along the y-axis Eqs. (4)-(6), and the relations Eqs. (7)-(9) are used to compute the gradients along the x-axis. The sample of the fuzzified gradients is illustrated in Figure 1.

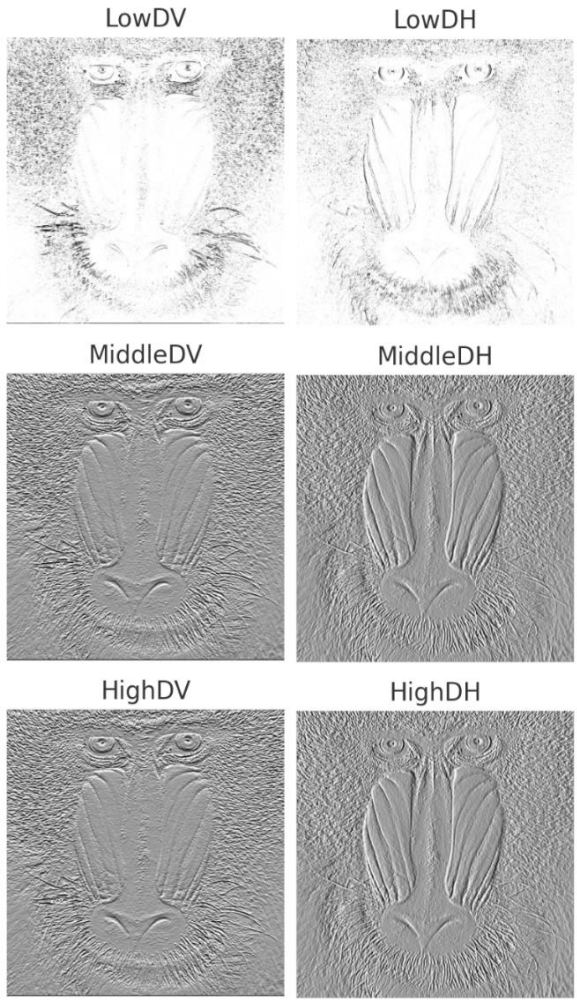

Figure 1. Results of the fuzzified gradients

$$\mu(x) = e^{-\frac{1}{2}\left(\frac{x-mean}{std\_dev}\right)} \quad (3)$$

$$\mu_{LowDH}(Dx) = e^{-\frac{1}{2}\left(\frac{Dx-0}{255}\right)} \quad (4)$$

$$\mu_{MiddleDH}(Dx) = e^{-\frac{1}{2}\left(\frac{Dx-255}{255}\right)} \quad (5)$$

$$\mu_{HighDH}(Dx) = e^{-\frac{1}{2}\left(\frac{Dx-255}{255}\right)} \quad (6)$$

$$\mu_{LowDV}(Dy) = e^{-\frac{1}{2}\left(\frac{Dy-0}{255}\right)} \quad (7)$$

$$\mu_{MiddleDV}(Dy) = e^{-\frac{1}{2}\left(\frac{Dy-255}{255}\right)} \quad (8)$$

$$\mu_{HighDV}(Dy) = e^{-\frac{1}{2}\left(\frac{Dy-255}{255}\right)} \quad (9)$$

Step 3: Fuzzy Rules

The fuzzified gradient information needs to be interpreted, and it needs to be determined whether each pixel in the image should be classified as an edge or background. We use Eqs. (10)-(12) as the set rules to determine whether a pixel is a background or an edge.

$$Rule_1 = \max\left(\mu_{HighDH}, \mu_{HighDV}\right) \quad (10)$$

$$Rule_2 = \max\left(\mu_{MiddleDH}, \mu_{MiddleDV}\right) \quad (11)$$

$$Rule_3 = \max\left(\mu_{MiddleDH}, \mu_{MiddleDV}\right) \quad (12)$$

The edge output is obtained by combining the maximum of Rules 1 and 2, and the background output is determined by Rule 3. These rules effectively differentiate between edge and non-edge regions in an image by considering the intensity of gradients in both horizontal and vertical directions. High and middle-intensity gradients result in edge, while low intensity in both directions results in background.

Step 4: Defuzzification

Defuzzification operation involves translating the fuzzy logic results into a binary edge-detected image. This is done by comparing the edge output with the background output for each pixel. Given the edge output (E) and background output (B) the defuzzified value for each pixel can be determined by Eq. (13).

$$Pixel\ (i,j) = \begin{cases} 255, & E(i,j) > B\ (i,j) \\ 0, & otherwise \end{cases} \quad (13)$$

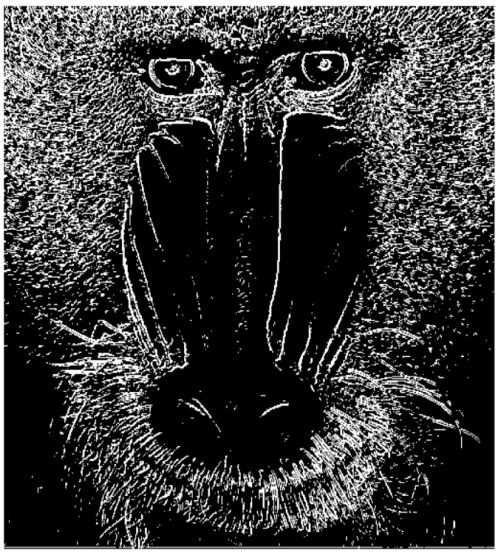

Figure 2. Identified edges

Defuzzification is critical in processing fuzzy logic systems, translating the fuzzy quantities into precise actions or outputs. In the context of edge detection in image processing, defuzzification helps finalize the decision for each pixel, determining whether it is part of an edge. The example result of the defuzzification process can be seen in Figure 2. Algorithm 1 summarizes the steps of Fuzzy Edge Detection

Algorithm 1. Fuzzy rules for edge detection

```
1: function FuzzyEdgeDetection(Cover)
2:     for each Pixel in the Cover do
3:         Gx ← ComputeGradientX(Pixel)
4:         Gy ← ComputeGradientY(Pixel)
5:         LowX ← GaussianMF(Gx, "low")
6:         MidX ← GaussianMF(Gx, "middle")
7:         HighX ← GaussianMF(Gx, "high")
8:         LowY ← GaussianMF(Gy, "low")
9:         MidY ← GaussianMF(Gy, "middle")
10:        HighY ← GaussianMF(Gy, "high")
11:        EdgeRule1 ← max(HighX, HighY)
12:        EdgeRule2 ← max(MidX, MidY)
13:        BackgroundRule ← min(LowX, LowY)
14:        EdgeOutput ← max(EdgeRule1, EdgeRule2)
15:        if EdgeOutput > BackgroundRule, then
16:            Edges[Pixel] ← 255
17:        else
18:            Edges[Pixel] ← 0
19:        end if
20:    end for
21:    return Edges
22: end function
```

3.2 Data embedding

As illustrated in Figure 3, the data embedding process is made of steps starting from edge identification until the secret data is concealed in the cover image. At the first embedding stage, the cover image is processed to get all the edges of the image using the Sobel operator. To generate the key, we convert all the pixels that have 255 values to 1. The resulting key will be a $n \times n$ matrix with binary values, with n being the size of the image. The binary key matrix is critical in enhancing security in embedding and extracting concealed information within the cover image.

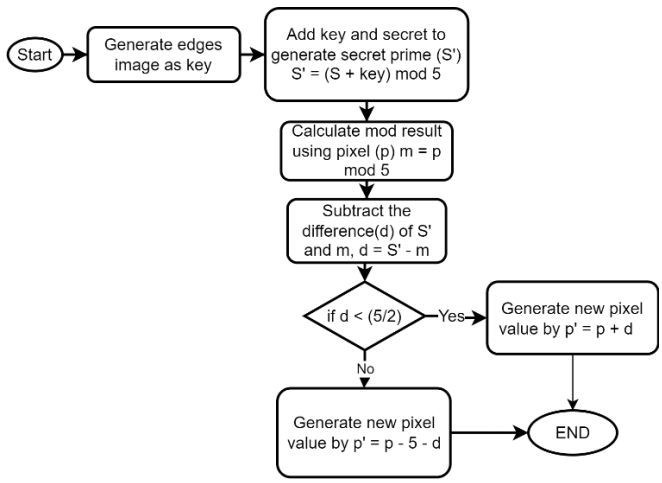

Figure 3. The flow of embedding process

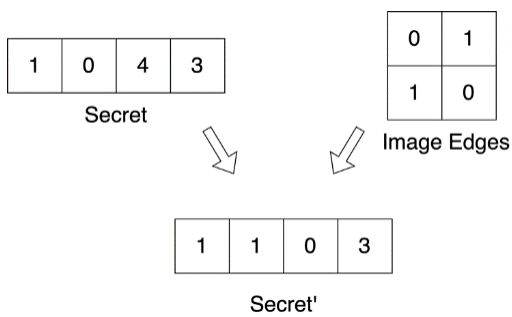

Figure 4. Generating secret prime

The key metric will generate a transformed secret key (see Figure 4). Let S represent the vector of the secret base 5 bits, and E represent the reshaped edge image matrix flattened into a vector. The operation to generate the transformed secret key (S') is mathematically expressed in Eq. (14). This secret key transformation adds a layer of security to the process.

$$S' = (S + E)\ mod\ 5 \tag{14}$$

After generating the transformed secret, we continue to embed the secret prime into the image. The embedding process is as follows:

Step 1: Calculate the modulo-5 value of the pixel using Eq. (15).

$$m = p\ mod\ 5 \tag{15}$$

Step 2: Compute the difference between the secret prime and the modulo-5 value of the pixel using Eq. (16).

$$d = S' - m \tag{16}$$

Step 3: Let p' the new pixel value. If d is less than half of the base of the secret prime (in this case $\frac{5}{2}$) the pixel value is increased by d. Otherwise, the pixel's value is decreased by 5-d to wrap around the base five systems (see Eq. (17)).

$$p' = \begin{cases} p + d, & d < \frac{5}{2} \\ p - (b - d), & d \geq \frac{5}{2} \end{cases} \tag{17}$$

Algorithm 2 summarizes the steps of Data Embedding for embedding secrets into stego image.

Algorithm 2. Data embedding algorithm

```
function StegoEmbedding (CoverImage, SecretData)
    Cover ← ReadCoverImage (CoverImage)
    Secret ← ReadSecretData (SecretData)
    Edges ← FuzzyEdgeDetection (Cover)
    BinaryKey ← ConvertEdgestoBinary (Edges)
    TransformedSecret ← TransformSecret
    for each Pixel in the Cover, do
      ModValue ← Pixel mod 5
      Difference ← TransformedSecret - ModValue
      if Difference < 2.5 then
        NewPixel ← Pixel + Difference
      else
        NewPixel ← Pixel - (5 - Difference)
      end if
      UpdateCoverImage(Pixel, NewPixel)
    end for
    StegoImage ← WriteStegoImage (Cover)
    return StegoImage
end function
```

3.3 Data extraction

As illustrated in Figure 5, the proposed method considers three key elements in the data recovery process: extracting the transformed secret data, generating the key matrix, and recovering the original secret data.

(1) Transformed secret data extraction: For each pixel (p), calculate the remainder of the pixel value when divided by the base 5 using Eq. (18) to generate a secret prime (S').

$$S' = p\ mod\ 5 \tag{18}$$

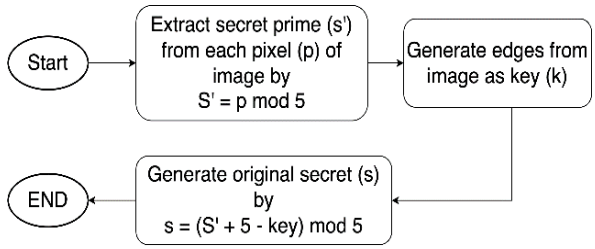

Figure 5. The flow of data recovery

(2) To obtain the key matrix, we generate edges from the stego images using the Sobel operator we used previously in an embedding process to ensure the key generated is consistent. The resulting key is the same size and value as the key in the embedding process, an $n\ x\ n$ matrix with binary values, with n being the size of the image.

(3) To recover the original secret data, we proceed as follows: Let S' be the vector representing the transformed secret data extracted from the image and let E be the vector representing the edge image that is already flattened. Both vectors are of the same length n, and their elements correspond to the individual values of the secret prime and the edge image, respectively. The operation to retrieve the original secret s from the S' and the edge image E is defined for each element i (where i ranges from 1 to n) based on Eq. (18).

$$S' = p\ mod\ 5s = (S'_i + 5 - E_i)\ mod\ 5 \tag{19}$$

Algorithm 3 summarizes the steps of Data extraction for extracting secrets from stego images.

Algorithm 3. Data extraction algorithm

```
function DataExtraction (StegoImage)
    Stego ← ReadStegoImage (StegoImage)
    Edges ← FuzzyEdgeDetection (Stego)
    BinaryKey ← ConvertEdgestoBinary (Edges)
    TransformedSecret ← []
    for each Pixel in Stego do
      ModValue ← Pixel mod 5
      TransformedSecret.append (ModValue)
    end for
    SecretData ← []
    for each i in TransformedSecret, do
      OriginalSecret ← (TransformedSecret[i] + 5 -
BinaryKey[i]) mod 5
      SecretData.append(OriginalSecret)
    end for
    return SecretData11:    return SecretData
end function
```

4. RESULTS AND DISCUSSION

To implement the proposed algorithm experimentally, we use images from the SIPI image database [25] with sample images illustrated in Figure 6. These 512×512 pixel grayscale images, each with eight bits per pixel, are known as Zelda, Plane, Baboon, Lake, Boat, and Goldhill. The secret data used

in this research consists of bitstreams in base five ranging from 10 to 100KB. The proposed method's performance is analyzed using the six selected cover images.

We use the Peak Signal-to-Noise Ratio (PSNR) evaluated in decibels (dB) to assess the steganographic image's quality. It is worth noting that the threshold value for the PSNR to be admissible for a stego image is 30 dB. The PSNR is computed using the Mean Square Error (MSE) obtained from Eq. (20) and calculated using Eq. (21) with $cover_{im}(i,j)$ the pixel value at the position(i,j) in the cover image and $sto_{im}(i,j)$ the pixel value at the position(i,j) in the stego image.

$$MSE = \frac{1}{k \times l}\sum_{i=1}^{k}\sum_{j=1}^{l}[cover_{im}(i,j) - sto_{im}(i,j)]^2 \quad (20)$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \quad (21)$$

Table 1 compares three steganographic methods with the same cover images and payload sizes. The results of the proposed method are compared to those reported in several studies [26, 27]. The evaluation is centered around varying payload sizes in kilobits (KB) and the resulting PSNR values for each method across the considered cover images, namely "Plane," "Baboon," "Lake," "Boat," "Goldhill," and "Zelda."

Based on the obtained results, it is demonstrated that the proposed method showcases the highest PSNR values, which indicate the superior quality of the stego images obtained with this newly proposed approach, emphasizing the effectiveness of the proposed method in minimizing information loss during data embedding. The IPPVO results reported by Ding et al. [26] are also among the good results recently achieved. However, they are consistently inferior to those reported by Chang et al. [27]. In this method, we propose an improvement. The outperformance of the proposed method identifies a contextual understanding of how the proposed steganographic technique fares concerning an existing approach.

The analysis of the average PSNR values in Figure 7 shows that the PSNR values generally decrease as the payload size increases from 10KB to 50KB. This trend is consistent across all three methods. The proposed method consistently outperforms the IPPVO and HPPVO methods across all payload sizes in this range, indicating better image quality retention after steganographic processing. Higher PSNR in smaller payloads suggests that the Proposed Method is particularly effective in maintaining image quality in less complex or lower-resolution images, making the steganographic modifications less detectable. In the 60-100KB range, the PSNR values tend to decrease as the Payload size increases, as shown in Figure 8. This trend holds for all methods. However, the proposed method shows a notably higher PSNR than the IPPVO and HPPVO methods, suggesting a substantial improvement in maintaining image quality in this file size range. The overall trend across all payload sizes indicates that as the amount of data to be encoded increases, the quality of the resultant image, as measured by PSNR, decreases. This is expected due to the increased compression required to maintain the payload within the specified size. The IPPVO and HPPVO methods, while effective to a degree, seem less capable of maintaining high image quality during steganographic processing than the proposed method.

Table 2 shows how the Structural Similarity Index (SSIM) values vary for different cover images (Plane, Baboon, Lake, Goldhill, and Zelda) across payload capacities ranging from 10KB to 100KB. As expected, the SSIM values decrease with increasing payloads, indicating a decline in image quality as more data is embedded. For instance, the Plane image starts with an SSIM of 0.94 at a 10KB payload, dropping to 0.77 at 100 KB, demonstrating a gradual quality degradation. Similarly, Baboon shows high initial quality with an SSIM of 0.93 at 10KB, reducing to 0.75 at 100KB. The Lake image begins at 0.92 SSIM at 10KB and falls to 0.64 at 100KB, suggesting it is more prone to quality loss compared to Plane and Baboon. Goldhill's SSIM values start at 0.91 for 10KB and decrease to 0.64 for 100KB, while Zelda shows a similar trend with SSIM values from 0.87 at 10KB to 0.61 at 100KB. These results highlight the trade-off between payload capacity and image quality, with higher payloads leading to greater structural dissimilarities in the cover images.

Figure 6. Sample cover images

Table 1. Comparison of the obtained and the existing results

Payload (in KB)	Cover Images	PSNR Result (in dB)		
		Proposed Method	IPPVO [26]	HPPVO [27]
10	Plane	60.92	64.09	64.18
	Baboon	61.10	54.75	55.45
	Lake	60.96	60.53	60.42
	Goldhill	60.95	60.79	60.91
	Zelda	60.64	59.96	60.51
20	Plane	57.83	60.42	60.40
	Baboon	58.15	56.54	56.84
	Lake	57.88	55.36	55.52
	Goldhill	57.87	55.60	55.97
	Zelda	57.67	56.12	56.81
30	Plane	56.09	58.31	58.52
	Baboon	56.33	53.12	53.70
	Lake	56.18	52.38	53.11
	Goldhill	56.12	53.53	53.91
	Zelda	55.96	54.81	54.95
40	Plane	54.87	56.51	56.72
	Baboon	55.04	52.14	52.42
	Lake	54.94	52.33	52.51
	Goldhill	54.87	52.73	53.07
	Zelda	54.72	51.41	59.43
50	Plane	53.88	54.75	54.96
	Baboon	54.07	51.36	51.64
	Lake	53.87	51.56	51.73
	Goldhill	53.87	50.97	51.31
	Zelda	53.77	51.17	51.41
60	Plane	53.12	54.17	54.39
	Baboon	53.24	50.25	50.53
	Lake	53.13	50.50	50.67
	Goldhill	53.13	50.39	50.82
	Zelda	52.96	50.24	50.59
70	Plane	52.46	52.91	53.13
	Baboon	52.60	48.73	49.01
	Lake	52.45	48.99	49.15
	Goldhill	52.37	49.13	49.56
	Zelda	52.28	49.23	49.48
80	Plane	51.84	51.33	51.56
	Baboon	51.98	47.15	47.15
	Lake	51.85	47.39	47.39
	Goldhill	51.86	47.55	47.98
	Zelda	51.71	47.65	47.90
90	Plane	51.38	49.89	50.02
	Baboon	51.49	45.71	45.99
	Lake	51.31	45.95	46.11
	Goldhill	51.32	46.11	46.54
	Zelda	51.25	46.21	46.46
100	Plane	50.87	48.44	48.59
	Baboon	51.07	44.27	44.56
	Lake	50.89	44.51	44.68
	Goldhill	50.88	44.66	45.09
	Zelda	50.77	44.76	45.21

The Structural Similarity Index (SSIM) is a metric used to evaluate the similarity between two images. It considers changes in structural information, luminance, and contrast to provide a comprehensive assessment of image quality. SSIM values range from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect dissimilarity. Unlike traditional metrics such as Mean Squared Error (MSE), which only consider pixel differences, SSIM models the human visual perception of image quality, making it more sensitive to structural distortions. It is widely used in image processing tasks, including image compression, denoising, and quality assessment, to ensure that the processed images retain high visual fidelity. An SSIM value above 0.9 is generally considered good, indicating that the image retains most of its original quality, while values between 0.7 and 0.9 indicate moderate quality, and values below 0.7 suggest noticeable degradation in image quality.

To illustrate the practical application of the proposed algorithm in information security, this study incorporates a steganalysis attack using adaptive steganalysis models, as cited by several researchers [13, 16]. The stego images are generated using the proposed steganographic algorithm with payload capacities of 50KB and 100KB. To ensure robust attack results, the stego images undergo one or more common attack operations (such as cropping and compression) before being used for training. The data presented in Table 3 indicate the proposed algorithm's strong resistance, as the detection accuracy in all scenarios remains below 50%. The highest

detection accuracy, achieved with preprocessed images during a strong steganalysis attack, is only 42.83%. This explains the robustness of the proposed method against such attacks.

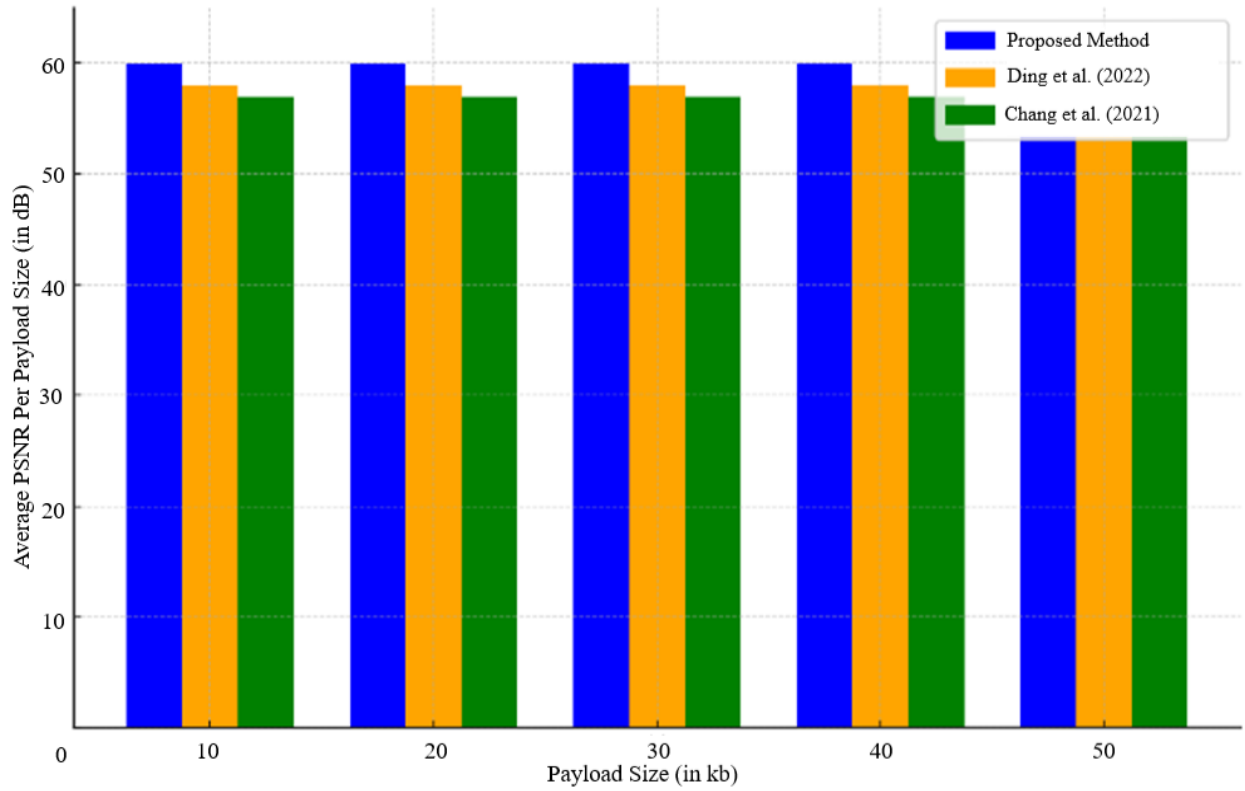

Figure 7. Average PSNR generated using secret data sizes ranging from 10KB to 50KB

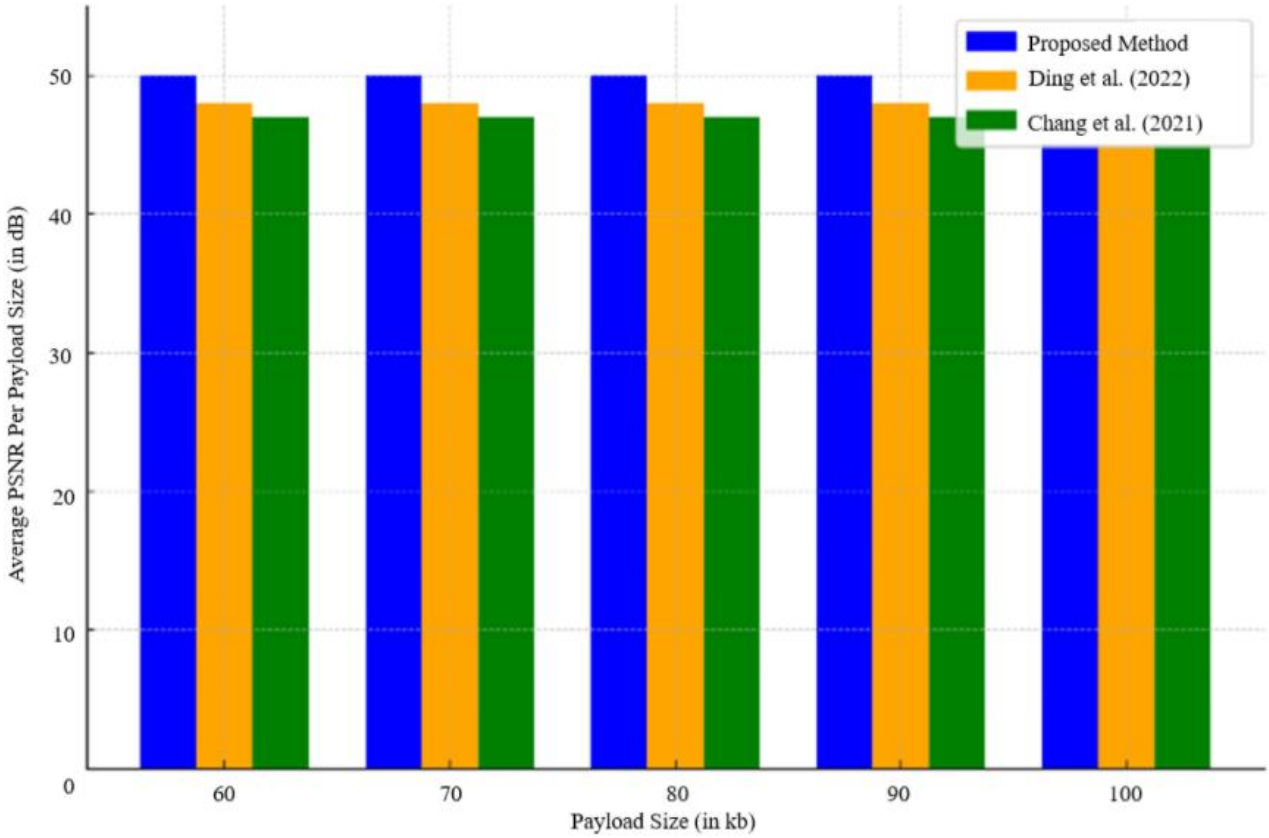

Figure 8. Average PSNR generated using secret data sizes ranging from 60KB to 100KB

Table 2. SSIM value for each cover image

Payload (in KB)	SSIM Value	
	Cover Images	Proposed Method
10	Plane	0.94
	Baboon	0.93
	Lake	0.91
	Goldhill	0.87
	Zelda	0.81
20	Plane	0.71
	Baboon	0.67
	Lake	0.63
	Goldhill	0.62
	Zelda	0.95
30	Plane	0.89
	Baboon	0.88
	Lake	0.79
	Goldhill	0.74
	Zelda	0.72
40	Plane	0.70
	Baboon	0.69
	Lake	0.65
	Goldhill	0.63
	Zelda	0.89
50	Plane	0.87
	Baboon	0.87
	Lake	0.86
	Goldhill	0.85
	Zelda	0.73
60	Plane	0.67
	Baboon	0.64
	Lake	0.63
	Goldhill	0.60
	Zelda	0.91
70	Plane	0.90
	Baboon	0.86
	Lake	0.82
	Goldhill	0.82
	Zelda	0.77
80	Plane	0.72
	Baboon	0.71
	Lake	0.71
	Goldhill	0.62
	Zelda	0.87
90	Plane	0.87
	Baboon	0.85
	Lake	0.80
	Goldhill	0.78
	Zelda	0.77
100	Plane	0.75
	Baboon	0.64
	Lake	0.64
	Goldhill	0.61
	Zelda	0.60

Table 3. Detection accuracy in percentage (%) of the proposed method by steganalysis attacks

Staganalysis Method (Attaching Algorithm)	Payload Capacity of 50KB	Payload Capacity of 100KB
Algorithm in [13]	25.21	36.44
Algorithm in [16]	32.51	42.83

5. CONCLUSION

In conclusion, this study has significantly improved the efficacy of existing data-hiding methodologies, enhancing the security of concealed payloads by incorporating edge images as an additional security layer. Including this extra security layer is vital to safeguarding the embedded secret payload, rendering it less susceptible to detection or extraction by unauthorized entities. Moreover, integrating edge images introduces a heightened complexity to the embedding process, thereby increasing potential attackers' difficulty deciphering the concealed information. Beyond offering an efficient means of concealing information within images with minimal impact on quality for smaller data loads, the proposed method

introduces an innovative security feature. This feature significantly enhances the resilience of steganographic content against unauthorized access and detection [1]. Such a strategic advantage positions the proposed method as a highly effective technique for secure steganographic practices, particularly in scenarios where data integrity and security are paramount considerations. The findings of this study contribute not only to the advancement of steganographic methodologies but also to the broader discourse on secure information transmission within digital images.

The practical applications of this method are extensive, ranging from secure communication in military and diplomatic contexts to protecting intellectual property and personal privacy in the digital age. One of the main challenges that could arise is the computational complexity introduced by the edge detection and embedding processes, which might require optimization for real-time applications. Additionally, there may be limitations in the method's robustness against highly sophisticated steganalysis techniques that continue to evolve. Addressing these challenges in future research will be crucial for enhancing the method's practical applicability and overall effectiveness.

The comparative analysis of various steganographic methods has yielded valuable insights into the trade-offs between image quality, as measured by PSNR, and payload size. Notably, the proposed method, leveraging edge images as a key to steganography, has exhibited superior performance by maintaining elevated PSNR values even at smaller payload sizes. This characteristic underscore the method's proficiency in preserving image quality, especially when minimizing embedded data volume is imperative for secure image transmission.

Based on the results of this work and the state-of-the-art, there is room for improving the ability to hide more data by preserving the quality of the stego image. Future work should focus on several key areas to address these limitations. First, optimizing the computational efficiency of the edge detection and embedding processes will be essential for real-time applications. This could involve developing more efficient algorithms or leveraging hardware acceleration techniques. Second, enhancing the robustness of the method against advanced steganalysis techniques will require ongoing research and adaptation to emerging threats. This could include integrating machine learning approaches to adjust embedding strategies based on detected threats dynamically. Third, further research should explore methods to optimize the balance between embedding capacity and image quality across a broader range of image types and conditions. Developing adaptive techniques that can tailor the embedding process to the specific characteristics of each image could significantly improve the method's overall performance.

REFERENCES

[1] Simmons, G.J. (1984). The prisoners' problem and the subliminal channel. In Advances in Cryptology: Proceedings of Crypto 83, Boston, MA: Springer US, pp. 51-67. https://doi.org/10.1007/978-1-4684-4730-9_5

[2] Zhang X., Li C., Tian L. (2023). Advanced audio coding steganography algorithm with distortion minimization model based on audio beat. Computers and Electrical Engineering, 106: 108580. https://doi.org/10.1016/j.compeleceng.2023.108580

[3] Kaushik, K., Bhardwaj, A. (2021). Zero-width text steganography in cybercrime attacks. Computer Fraud & Security, 2021(12): 16-19. https://doi.org/10.1016/S1361-3723(21)00130-5

[4] Chanda D'Layla, A.W., Nevin, M., Sunardi Putra, G.G., De La Croix, N.J., Ahmad, T. (2023). Steganography in grayscale images: Improving the quality of a stego image. In 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, pp. 1-6. https://doi.org/10.1109/SMARTGENCON60755.2023.10442310

[5] Li, N., Qin, J., Xiang, X., Tan, Y. (2023). Robust coverless video steganography based on inter-frame keypoint matching. Journal of Information Security and Applications, 79: 103653. https://doi.org/10.1016/j.jisa.2023.103653

[6] Vanmathi, C., Prabu, S. (2018). Image steganography using fuzzy logic and chaotic for large payload and high imperceptibility. International Journal of Fuzzy Systems, 20: 460-473. https://doi.org/10.1007/s40815-017-0420-0

[7] Yu, C., Zhang, X., Zhang, X., Li, G., Tang, Z. (2022). Reversible data hiding with hierarchical embedding for encrypted images. IEEE Transactions on Circuits and Systems for Video Technology, 32(2): 451-466. https://doi.org/10.1109/TCSVT.2021.3062947

[8] He, W., Cai, Z. (2021). Reversible data hiding based on dual pairwise prediction-error expansion. IEEE Transactions on Image Processing, 30: 5045-5055. https://doi.org/10.1109/TIP.2021.3078088

[9] Kosuru, S.D., Pradhan, A., Basith, K.A., Sonar, R., Swain, G. (2023). Digital image steganography with error correction on extracted data. IEEE Access, 11: 80945-80957. https://doi.org/10.1109/ACCESS.2023.3300918

[10] Ren, F., Hao, Y., Pang, K., Wu, Z. (2024). Reversible data hiding scheme in encrypted images based on homomorphic encryption and pixel value ordering. Multimedia Tools and Applications, 83(14): 40607-40627. https://doi.org/10.1007/s11042-023-17242-4

[11] Ding W., Zhang H., Reulke R., Wang Y., (2022). Reversible image data hiding based on scalable difference expansion. Pattern Recognit Lett, 159: 116-124. https://doi.org/10.1016/j.patrec.2022.05.014

[12] Abdollahi, B., Harati, A., Taherinia, A. (2023). Image steganography based on smooth cycle-consistent adversarial learning. Journal of Information Security and Applications, 79: 103631. https://doi.org/10.1016/j.jisa.2023.103631

[13] De La Croix, N.J., Ahmad, T., Han, F. (2023). Enhancing secret data detection using convolutional neural networks with fuzzy edge detection. IEEE Access, 11: 131001-131016. https://doi.org/10.1109/ACCESS.2023.3334650

[14] Yu, L., Zhang, Z., Weng, S., Cao, P., Cao, G. (2024). A deep steganalysis network combining source-supervised and target-unsupervised information for cover-source mismatch. Expert Systems with Applications, 255: 124790. https://doi.org/10.1016/j.eswa.2024.124790

[15] De La Croix, N.J., Ahmad, T. (2023). Toward secret data location via fuzzy logic and convolutional neural network. Egyptian Informatics Journal, 24(3): 100385. https://doi.org/10.1016/j.eij.2023.05.010

[16] Ntivuguruzwa, J.D.L.C., Ahmad, T. (2023). A

convolutional neural network to detect possible hidden data in spatial domain images. Cybersecurity, 6(1): 23. https://doi.org/10.1186/s42400-023-00156-x

[17] Chen, M., Boroumand, M., Fridrich, J. (2018). Deep learning regressors for quantitative steganalysis. Electronic Imaging, 30: 1-7. https://doi.org/10.2352/ISSN.2470-1173.2018.07.MWSF-160

[18] Rupa, C., Shaikh, S., Chinta, M. (2021). Multimedia concealed data detection using quantitative steganalysis. International Journal of Digital Crime and Forensics, 13(5): 101-113. https://doi.org/10.4018/IJDCF.20210901.oa6

[19] Al-Jarah, A.I.H., Arjona, J.L.O. (2021). Secret key steganography: Improve the security level of LSB algorithm. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, pp. 0215-0220. https://doi.org/10.1109/UEMCON53757.2021.9666569

[20] Luo, W., Huang, F., Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security, 5(2): 201-214. https://doi.org/10.1109/TIFS.2010.2041812

[21] AlKhodaidi, T., Gutub, A. (2021). Refining image steganography distribution for higher security multimedia counting-based secret-sharing. Multimedia Tools and Applications, 80: 1143-1173. https://doi.org/10.1007/s11042-020-09720-w

[22] Gaurav, K., Ghanekar, U. (2018). Image steganography based on canny edge detection, dilation operator and hybrid coding. Journal of Information Security and Applications, 41: 41-51. https://doi.org/10.1016/j.jisa.2018.05.001

[23] Sahu, A.K., Swain, G. (2019). An optimal information hiding approach based on pixel value differencing and modulus function. Wireless Personal Communications, 108(1): 159-174. https://doi.org/10.1007/s11277-019-06393-z

[24] Liu, Q., Su, W., Ni, J., Hu, X., Huang, J. (2024). An efficient distortion cost function design for image steganography in spatial domain using quaternion representation. Signal Processing, 219: 109370. https://doi.org/10.1016/j.sigpro.2023.109370

[25] The USC-SIPI Image Database. (2023). Signal and Image Processing Institute, USCViterbi. https://sipi.usc.edu/database.

[26] Ding, H., Zhang, R., Reulke, H., Wang, Y. (2022). Reversible image data hiding based on scalable difference expansion. Pattern Recognition Letters, 159: 116-124. https://doi.org/10.1016/j.patrec.2022.05.014

[27] Chang, J., Ding, F., Li, X., Zhu, G. (2021). Hybrid prediction-based pixel-value-ordering method for reversible data hiding. Journal of Visual Communication and Image Representation, 77: 103097. https://doi.org/10.1016/j.jvcir.2021.103097