Hindawi
Security and Communication Networks
Volume 2021, Article ID 6610678, 15 pages
https://doi.org/10.1155/2021/6610678

Research Article

A New Method of Coding for Steganography Based on LSB Matching Revisited

Mansoor Fateh, Mohsen Rezvani, and Yasser Irani

Faculty of Computer Engineering, Shahrood University of Technology, Shahroud, Iran

Correspondence should be addressed to Mansoor Fateh; mansoor_fateh@shahroodut.ac.ir

Received 4 November 2020; Revised 19 December 2020; Accepted 23 January 2021; Published 8 February 2021

Academic Editor: Benjamin Aziz

LSB matching revisited is an LSB-based approach for image steganography. This method is a type of coding to increase the capacity of steganography. In this method, two bits of the secret message are hidden in two pixels with only one change. But this method provides no idea for hiding a message with a large number of bits. In other words, this method works only for $n = 2$, where n is the number of bits in a block of the secret message. In this paper, we propose an improved version of the LSB matching revisited approach, which works for $n > 2$. The proposed scheme contains two phases including embedding and extracting the message. In the embedding phase, we first convert the secret message into a bit-stream, and then the bit-stream is divided into a set of blocks including n bits in each block. Then we choose 2^{n-1} pixels for hiding such n bits of the secret message. In the next step, we choose the operations needed to generate such a message. Finally, we perform the obtained operations over the coefficients to hide the secret message. The proposed approach needs fewer changes than LSB MR when $n > 2$. The capacity of the proposed approach is $(((2^n - 1)/2^{n-1}) - 1) \times 100\%$ higher than the F5 method where this value for $n > 2$ is bigger than 75%. For example, the capacity of our scheme is 75% higher than the capacity of F5 for $n = 3$. The proposed method can be used in the first step of every steganography method to reduce the change in the stego image. Therefore, this method is a new coding method for steganography. Our experimental results using steganalysis show that using our method provides around 10% higher detection error for SRNet over two steganography schemes.

1. Introduction

These days, the extensive application of the Internet has made transmission of information very easy and quick. Since the data are communicated through a public network such as the Internet, the security of the data is still an important challenge. Information hiding is an applicable technique to provide security for data communication. This technique hides the secret data within a cover data to generate a stego data [1–3]. The main techniques in information hiding include steganography, watermarking, and cryptography. A steganography method aims to hide any relationship between the sender and the receiver by embedding the data within the cover text. A watermarking method aims to protect the copyright of digital content owners. A steganography technique hides both the secret data and the relationship between the sender and the receiver, while a watermarking method protects only the secret data [4]. In a cryptography method, both the sender and the receiver can be identified and the main objective is to protect the content of the secret data.

A steganography technique employs a key and a media called cover data, to hide the secret data. The main objective of a steganography method is to hide the sender, the receiver, and the content of the secret data, which makes the secret data visible only to the receiver [4, 5]. The steganography word is originally derived from two Greek words: *stego* means hidden and *grafia* means writing [6]. Steganography is a method to transfer secret data embedded within cover data through a public communication channel such as the Internet. In this method, an attacker cannot extract the secret data from the cover data [7, 8]. There are three important parameters for evaluating a secret data communication technique, namely, capacity, robustness, and

transparency. Increasing these three parameters raises the confidentiality of the data communication technique. However, increasing all these three parameters simultaneously is a very daunting task. This is because increasing the transparency and capacity decreases the robustness, and increasing the robustness of the technique against the attacks decreases the transparency and capacity of the technique. Thus, an information-hiding technique must declare its main objective to improve some of these parameters. For example, a watermarking technique often accords the highest priority to robustness, while a steganography method accords higher priority to capacity and transparency compared to robustness. In another word, transparency is the highest priority in a steganography approach followed by capacity. The cover data in steganography can be an image, voice, video, text, protocol, etc. [9–11]. Images are employed as the cover data to a great extent in steganography techniques as they provide higher capacity, suitable flexibility, and extensive utilities for end users. Hiding secret data within an image is called image steganography. The image steganography must keep the main features of the image when it embeds the secret data inside of the image.

Image steganography methods can be divided based on their context into two groups: those in the Spatial Domain and those in the Transform Domain. The spatial domain steganography methods change some bits of the pixels in the cover image. The pixels employed for hiding the secret data are selected using a simple and random method. Thus, these methods provide insufficient robustness. The transform domain steganography methods hide the secret data within the transform coefficients of the cover image. They use various transforms such as Discrete Cosine Transform (DCT), Discrete Fourier Transform, and Discrete Wavelet Transform [10]. These methods provide a high level of robustness and low capacity.

The capacity and transparency (quality) of the stego image are the most important features of an image steganography technique. The capacity is the amount of data that can be hidden in a cover-image while maintaining the imperceptibility. Increasing the capacity must protect the secret data from being detected through the HSV or other statistical recognition methods. Thus, increasing the capacity while maintaining the imperceptibility is still a challenging task in image steganography [1]. Transparency is also indicated as the security of an image-steganography technique. Transparency means that the steganography process must not generate any type of perceptual distortion or changes in the statistical features of the cover-image that lead to the perception of the process by a steganography analyzer. Making a trade-off between capacity and security is a significant challenge in designing an image steganography algorithm [12].

We can divide the LSB-based steganography methods into two categories: LSBR (LSB replacement) and LSBM (LSB matching). LSBR simply replaces each bit of the secret message into the lowest bits in the cover image. However, LSBM randomly chooses to increment or decrement some pixels in the cover image in which their lowest significant bits are different from the appropriate bits in the secret message. In both categories, the receiver can simply extract the secret message from the lowest significant bits in the stego image [12, 13].

LSB MR (LSB Matching Revisited) is another LSB-based approach proposed for image steganography. Although both LSBM and LSBR choose the pixels independently, LSB MR considers every two continuous pixels in a single group. In this method, two bits of the secret message are hidden in two pixels with only one change. In LSB, the first pixel is used to hide the first bit of the secret message, and the second bit of the message is also hidden based on the even or odd relationship between two pixels of the cover image [12]. But this method has no idea for hiding a message with a large number of bits. In other words, this method works only for $n = 2$. Therefore, it is not possible to accurately compare the capacity of the proposed method with the LSB MR method. The F5 method works for $n > 1$. Also, the F5 method is a kind of coding method and is similar to the proposed method. Therefore, the capacity of the proposed method compares with the F5 method.

To address the abovementioned challenge, in this paper, we propose a method that works for $n > 2$, called here the *Advanced LSB MR*. The proposed method is a new method of coding for steganography and can be used in the first step of every steganography method to reduce the change in the stego image. For example, in [14], secret data are embedded within the skin region of the image in which for three bits of secret data, three coefficients of DWT have to be changed. These changes are reduced to one coefficient using the proposed method in the embedding step compared to the literature where four coefficients are required. For three bits of secret data, the coefficient changes decreased from 3 to 1, and the number of coefficients increased from 3 to 4. Using the proposed method in one of the steps of every steganography method, the changes of coefficients are reduced and as a result, the security and the resistance of the steganography method against steganalysis are increased while the capacity is reduced. The suggested method proposes a novel mapping function to generate a number of 2^n different states for n bits of the secret message with only one change (additive or subtraction). This helps to hide a number of n bits in a number of 2^{n-1} coefficients only by one change. It is noted that most of the steganography approaches assume that $n < 5$ where the capacity decreases sharply with increasing value of n. For example, to hide one byte, the data are divided into two 4-bit sections, and then the embedding operation is performed, and when only 16 coefficients are needed while $n = 8$, the number of required coefficients becomes 128.

The rest of the paper is structured as follows. Section 2 provides the related work. Section 3 describes the Basic Concepts used in our image steganography scheme. Section 4 presents our novel steganography scheme. Section 5 describes our experimental result. Finally, the conclusion is drawn in Section 6.

2. Related Works

F5 is a promising approach in steganography, which provides a high level of capacity. This method employs

hamming coding to reduce the number of changes in the LSB coefficient of the cover image. Such reduction in the changes lead to a higher level of security in F5. The F5 scheme hides n bits of the message into $2^n - 1$ coefficients. Thus, by increasing the value of n , we need more coefficients, which leads to lower steganography capacity [15]. In other words, the capacity of a steganography scheme decreases as the value of n increases. Thus, it is recommended to set $n = 3$ for most real applications to avoid reducing the capacity of the steganography. The F5 algorithm is vulnerable against many statistical attacks such as histogram attack and χ^2 attack [16]. The F5 algorithm uses DCT and the generated coefficients are decimals in this transform. The technique embeds the secret message in these coefficients. It is possible that some parts of these coefficients are removed in the inverse of the transforms, and this could result in the embedded bits of the secret message being removed in such a procedure. Therefore, the F5 algorithm uses the DCT coefficient after quantization in JPEG. The quantized DCT coefficient matrix elements in JPEG are saved and the embedded secret message in the quantized DCT coefficient is extracted directly. The IDCT is used for both decoding and displaying a JPEG image [16]. In order to fairly compare our approach against F5, we converted the hamming coding used in F5 into the spatial domain before comparing the two schemes.

There are several types of research conducted to improve the capacity, processing time, and the method of finding the position to embed the message. In [1], a method based on a quantization table is proposed to raise the capacity of the F5 algorithm. This method also provides some improvement in the processing time of F5. The method proposed in [17] focuses on choosing a suitable position for embedding the message. The main contributions of this method include fewer changes in the cover image and robustness against statistical attacks compared to the original F5 algorithm.

LSB is a simple and challenging algorithm method in steganography. In this method, replacing fewer bits of LSB makes the difference between cover image and stego image very difficult to spot [10]. However, changing a large number of bits in the cover image is one of the challenges of LSB. In some cases, there is less change in the cover image when the complement of the message is hidden. Using the complement of the message for reducing the changes in the cover image makes a new version of the LSB method, called reverse LSB [18]. The PI method is proposed to improve the security level of LSB [19]. This method hides the secret message into the 24-bit color channel of the image. The PI method generates less visible corruption when the hiding rating is less than three bits. This secures the scheme against both visibility and histogram attacks. Another extension of the LSB algorithm is OPAP (Optimal Pixel Adjustment Procedure) [16]. The OPAP scheme computes the difference between pixels in the cover image and the stego image and then changes the hidden bits to improve the transparency of the steganography process [20]. Thus, OPAP provides a high value of PSNR (Peak Signal-to-Noise Ratio) for widely used images such as Baboon and Lenna [21]. Another extension of the LSB algorithm is a mixture of adaptive pixel value differentiating between PVD and LSB. The method is proposed for increasing the data hiding capacity [22].

Another method for steganography is image realization steganography, which is based on a secure key and employs image realization in spite of directly hiding the secret information. This method uses a mapping matrix for hiding the secret message and is robust against both brute-force attack and statistical attacks such as histogram attack [23]. Roy and Changder in [24] extended this method by proposing a confidential mapping technique that is based on the Longest Common Subsequence. This method is robust against the brute-force attack and also provides a promising hiding capacity.

Wu and Tsai in [25] proposed an image steganography method called PVD (pixel-value differencing). This method first divides the cover image into no-overlapping blocks. Then, the difference between every two pixels in each block is computed and all these differences are grouped into some ranges. After that, the differences are replaced by some new values to embed some bits of the secret message. The number of embedded bits in a pair of pixels is determined based on the size of their difference range. A combination of LSB and PVD can improve the capacity and transparency of the cover image [26]. This method first computes the difference between every two continues pixels using PVD. It is to be noted that such a difference in the smooth positions is low and in the edges is high. Therefore, LSB and PVD are used for smooth positions and edges, respectively.

Raja et al. in [27] proposed a combination of LSB, DCT, and compression techniques to extend the LSB algorithm. In the first step, they use LSB to embed the secret message into the cover image. Then, the obtained image is transformed into the frequency domain using DCT. Finally, to improve the security level, they compares the image using quantization and run-length coding algorithms.

Bhardwaj and Sharma in [2] proposed an LSB-based steganography method in which the pixels are groups based on their values in second and third bits to reduce the number of changes needed during the embedding phase. After applying inverted LSB, the number of changes for each group is computed and then a group which highly changes pixels is inverted. While this method chooses only one group of pixels, our method employs all groups of pixels to reduce the number of changes. Moreover, our approach increases the number of groups and grouping is based on the second, third, and fourth bits of each pixel.

Valandar et al. in [28] proposed a transform domain steganography scheme based on integer wavelet transform (IWT) for digital images, and it also used a chaotic map. They also proposed a steganography technique based on 3D sine chaotic map. This map is used in embedding and extracting processes to increase the security of the steganography scheme [29]. Gutub and Al-Ghamdi in [30] proposed a multimedia image steganography for counting-based secret sharing. The proposed method in this article is different from the above methods as it extends the LSB encoding. Moreover, we believe that the extended LSB encoding proposed in this paper can be used for improving the aforementioned research.

Reversible data hiding (RDH) has received great attention recently in the field of information hiding

[22, 31–34]. A significant challenge in RDH-based techniques is increasing the data hiding capacity along with image quality [31]. Singh in [22] proposed a new data hiding scheme using a mixture of adaptive pixel value differencing (PVD) [35] and LSB in order to increase the data hiding capacity. Kaur et al. in [32] proposed an extension of the existing pixel value ordering methods to improve the quality of the stego image. Kumar et al. in [33] employed Lempel Ziv Welch (LZW) encoding to improve the hiding capacity of the RDH. Other compression methods, such as AMBTC image compression, can be used for improving the quality of the stego image [36]. Sahu and Swain in [37] combined LSB and PVD to propose a novel data hiding scheme. They also leveraged the idea of pixel overlapping to improve the capacity and PSNR of an image steganography scheme [38]. In another work, they proposed a n-rightmost bit replacement image steganography scheme [39]. Sahu et al. in [40] proposed several dual-layered-based RDH methods using modified LSB. An interesting research issue is to investigate the possibility of applying our encoding method for improving the hiding capacity of RDH that we leave for future work.

3. Basic Concepts

In this section, we introduce several concepts used in our image steganography scheme.

3.1. Steganography. For the first time, steganography was performed by a Greek king who was held captive by King Darius. This Greek king formed steganography by carving a message on his slave's head. Soon after that, a message were inscribed directly on a wooden tablet before a wax was applied, and also invisible writing was employed by lemon juice in Roman's ancient era. Also, steganography was performed at World War II to use in Null cipher message Text [41, 42]. The objective of steganography is to hide the nature of secret messages from an unauthorized person, so that an invader cannot discover transmitted messages in the background of a general communication [7]. In cryptography, the sender, the receiver, and the encrypted message are obvious, but in steganography, hiding the sender, the receiver, and the secret is the main issue.

We can mention industrial steganography, linguistic steganography, and digital steganography as the main methods in this field. In industrial steganography, we use different sciences, such as engineering, physics, etc., to hide information. In the field of linguistic steganography, information hiding is done through writing. Also, the definition of digital steganography is to hide the messages inside digital media, such as voice, image video, and text.

In 1983, Simmons et al. represented a system based on steganography. In this system, two prisoners, named Alice and Bob, intended to plan an escape. To sketch an escape plan, Alice wants to send a message to Bob. The communication between Alice and Bob is checked by Willi, who is a prison officer. Alice should send her message to Bob in the form of a hidden message in the regular message so that Willie does not suspect and Bob will be able to fully understand the message [15]. This basic system is misused on various platforms.

Today, due to the wide variety of images, the image is used as a safe enclosure to conceal a message and is referred to as a cover. The image, which is produced after the inclusion of confidential data, is called stego. In steganography methods, we often use a key to clutter the message so that the content of the message won't be discovered clearly if the secret message is compromised or is being exposed. The general Schema of steganography with the key is shown in Figure 1.

Spatial methods are also steganography methods in the image field. In these methods, some pixels of our image are changed directly to hide secret data. In these methods, RGB or brightness intensity is used in latent data information. LSB is one of the image steganography methods in the spatial domain that will be described in detail next section.

3.2. LSB MR Method. Mielikainen in [43] proposed the LSB MR (LSB Matching Revisited) method in 2006. While both LSBR and LSBM independently consider each pixel, the LSB MR method pairs pixels in the cover image. Thus, it simultaneously embeds the secret message into a pair of two pixels. Although LSB MR reduces the expected changes per each bit in the message compared to both LSBR and LSBM from 0.5 to 0.375, it has no improved hiding capacity. Moreover, LSB MR generates less distortion in the image and thus makes the identification of the changes harder [44].

Assume that the brightness of two continuous pixels is indicated as x_i and x_{i+1}, and we aim to hide two bits of the secret message into the brightness of these two pixels. After embedding the message, the first bit of the message is equal to the lowest significant bit of x_i. The second bit of the message is obtained using a function of both pixels in the cover image. In this method, when the first bit of the message is equal to the lowest significant bit of x_i, there would be no change in x_i. Thus, in order to obtain the position of embedding the second bit of the message, the function must return different values for the incremented and decremented values of the first pixel.

$$f(x_i - 1, x_{i+1}) \neq f(x_i + 1, x_{i+1}). \tag{1}$$

In case the first bit of the message is equal to the lowest significant bit of x_i, there would be no change in such a bit in x_i. Therefore, we need to change the lowest significant bit of x_{i+1} in order to hide the second bit of the message. In fact, the function must hold an additional property in which it must return different values when the second argument of the function changes as $x_{i+1} + 1$ and $x_{i+1} - 1$.

$$f(x_i, x_{i+1}) \neq f(x_i, x_{i+1} \pm 1). \tag{2}$$

According to the above two properties, we employ function $f(x_i, x_{i+1}) = \mathrm{LSB}(\lfloor x_i/2 \rfloor + x_{i+1})$ where both properties are provided. Thus, embedding of two bits of the message (m_i, m_{i+1}) into two pixels (x_i, x_{i+1}) is executed as follows:

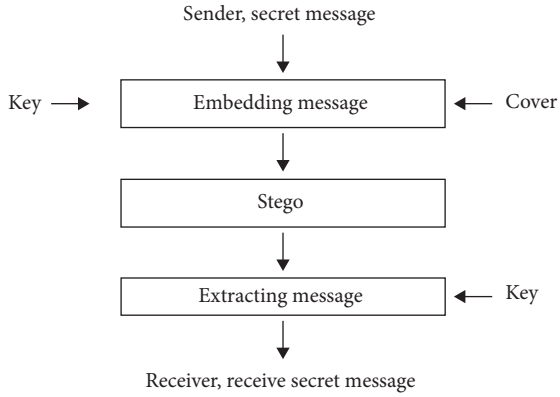


Figure 1: The general schema of steganography with the key.

$$y_i = \begin{cases} x_i, & \text{if } m_i = \text{LSB}(x_i), \quad m_{i+1} = f(x_i, x_{i+1}), \\ x_i, & \text{if } m_i = \text{LSB}(x_i), \quad m_{i+1} \neq f(x_i, x_{i+1}), \\ x_i - 1, & \text{if } m_i \neq \text{LSB}(x_i), \quad m_{i+1} = f(x_i - 1, x_{i+1}), \\ x_i + 1, & \text{if } m_i \neq \text{LSB}(x_i), \quad m_{i+1} \neq f(x_i - 1, x_{i+1}), \end{cases} \tag{3}$$

$$y_{i+1} = \begin{cases} x_{i+1}, & \text{if } m_i = \text{LSB}(x_i), \quad m_{i+1} = f(x_i, x_{i+1}), \\ x_{i+1} \pm 1, & \text{if } m_i = \text{LSB}(x_i), \quad m_{i+1} \neq f(x_i, x_{i+1}), \\ x_{i+1}, & \text{if } m_i \neq \text{LSB}(x_i), \quad m_{i+1} = f(x_i - 1, x_{i+1}), \\ x_{i+1}, & \text{if } m_i \neq \text{LSB}(x_i), \quad m_{i+1} \neq f(x_i - 1, x_{i+1}), \end{cases} \tag{4}$$

$$y_{i+1} = \begin{cases} x_{i+1}, & \text{if } m_{i+1} = f(x_i.x_{i+1}), \\ x_{i+1} \pm 1, & \text{else.} \end{cases} \tag{5}$$

To extract the message, we only need to obtain m_i from the LSB of the first pixel and m_{i+1} from the function introduced for the embedding phase [43].

In LSB MR, every two bits of the message are hidden into the brightness of two continuous pixels with only one change. This method can be applied only for hiding two bits. In this paper, we extend LSB MR to hide three bits with only one change. Such reduction in the number of changes leads to increasing the PSNR metric, and it, therefore, consolidates our method against attacks for discovering the secret message.

The LSB-based approaches randomly apply their changes without considering the statistics of the image. Huang and Ouyang in [45] proposed a method to select the suitable positions for embedding in LSB. They called regions of the image as smooth regions, redundant image regions, and regions with redundant pixels as the fragile regions. A little change in the fragile regions of an image makes the image detectable. Thus, the authors proposed to protect the fragile regions by not embedding messages within such regions.

Lou et al. in [46] reported that the statistical properties of the edges in an image are more complicated than the surface regions. After that, they proposed a steganography scheme to better keep such statistical properties. They proposed an adaptive steganography scheme based on LSB MR, called EAMR. In this method, they employed the difference between the brightness of two continuous pixels to identify the embedding regions. Huang et al. [44] proposed a method to improve the robustness of EAMR against the identification attack by changing the pixel selection algorithm. In this method, the image is divided into nonoverlapping blocks with 3×3 pixels. In spite of choosing the continuous pixels, this method randomly chooses every paired pixel from each block (a block contains four pairs of pixels) in the vertical, horizontal, or convex directions. This method is a promising method to improve the security of EAMR against the identification attacks.

In this paper, we extend LSB MR to hide a number of n bits of the secret message in 2^{n-1} pixels or coefficients with only one change. Such reduction in the number of changes leads to increasing the PSNR metric, and it, therefore, consolidates our method against attacks for discovering the secret message.

4. The Proposed Method

As mentioned, F5 is a promising method in image steganography in which the Hamming coding is employed. Using such coding decreases the number of changes in the LSB coefficients of the cover image, and consequently, it raises the security level of the steganography scheme. This method hides n bits within $2^n - 1$ coefficients. Therefore, increasing the value of n needs more coefficients and consequently decreases the capacity of the steganography. Thus, in most of the real applications, we let $n = 3$ to prevent declining capacity.

For $n = 2$, there are three coefficients in the F5 algorithm while the LSB MR needs only two coefficients. In other words, the capacity in LSB MR increases by 50% more than F5. Thus, using the LSB MR method for $n > 2$ helps us to increase the security level of the steganography. This is because we can hide a message with fewer changes in the cover image.

In this paper, we propose a novel mapping function to generate 2^n states with only one change (additive or subtraction). This helps us to hide n bits in 2^{n-1} coefficients only by one change. In this paper, for $n = 3$ (three bits of the secret message), four coefficients and choosing a suitable function, 8 different states can be obtained by modifying, increasing, or decreasing, the values of the coefficients. In general, the number of 2^n states can be obtained for the number of 2^{n-1} coefficients.

Considering function $f(x_i, x_{i+1}) = \text{LSB}(\lfloor x_i/2 \rfloor + x_{i+1})$ generates 7 states for three coefficients. This function was proposed in LSB MR and we aim to employ this function to generate 2^n states for 2^{n-1} coefficients. In the first step, we employ this function to generate 7 states for 3 coefficients. For three coefficients and using add and subtraction over the least bit of each coefficient, we obtain the following function:

$$f(x_i, x_{i+1}, x_{i+2}) = \mathrm{LSB}\left(x_i + \lfloor \frac{x_{i+1}}{2} \rfloor\right), \mathrm{LSB}\left(x_{i+1} + \lfloor \frac{x_{i+2}}{2} \rfloor\right), \mathrm{LSB}\left(x_{i+2} + \lfloor \frac{x_i}{2} \rfloor\right). \tag{6}$$

For example, for coefficient values $(8, 14, 21)$, we obtain three values $x_i = 8, x_{i+1} = 14, x_{i+2} = 21$. Now, by manipulating one bit in the coefficients, we obtain 7 different values (shown in Table 1). As one can see in Table 1, the secret message with value of 2 is not generated for any coefficient changes. This is because of the fact that there are at most 7 states for a message with three bits. It is to be noted that one state must be excluded according to the coefficients' values.

As one can see in Table 1, the function leads to a change in one bit of the message due to the change in a coefficient, while another bit of the message stays steady for the same operation of the coefficient and changes for another operation of the same coefficient.

For example, the first bit of the message (MSB) is changed for manipulating the first coefficient, and the third bit of the message (LSB) is only changed for decreasing the coefficient while there is no change in the bit for increasing the coefficient. This generates two new states. Thus, by defining a suitable function, we can generate two new states for the change in each coefficient. More specifically, we can generate 6 new states for three coefficients and 8 new states for four coefficients.

It is to be noted that these 7 states cannot hide three bits. This is because it is needed to generate 8 states for hiding three bits with only one change in the coefficients. We can hide three bits in four coefficients as it can generate 9 states by only one change in each coefficient. Thus, we can embed 8 different states of a three-bit secret message using an appropriate mapping function.

The significant point in these states is that changing only one coefficient leads to generating such states. In other words, there is no need to change more than one coefficient for having such a number of states. To this end, we first investigate the various states of the message. Assume that using the proposed function and without any change in the message, a three-bit message is generated. Such a message in bitstream format is in the range of 000–111. We aim to generate the appropriate message by changing only one of the coefficients. To generate a new message, we must change the bitstream value of the initial message from one to three. We divide such changes into categories containing two items to simulate the changes using the proposed function. In these categories, only one bit must be different. For example, for a three-bit message, we categorize the various states of the message as shown in Table 2. As one can see in this table, there are eight states for this example. For a three-bit secret message, we have four categories, for example, the first category contains 000 and 001, the second category contains 010 and 011, the third category contains 100 and 101, and the forth category contains 110 and 111. As one can see, the two messages inside of each category are distinguished at only one bit.

It is possible to design the proposed function using Table 2. There are three states for each category in this table. Each bit in the new message compared to the initial message is either constant, changed, or constant for a state and changed for another state. In the case of constant, there is no need for any coefficient in the function. In the case of changed, there is one coefficient in the function, and for the third case we consider $\lfloor \text{coefficient}/2 \rfloor$ in the function.

For example, in order to achieve the first category, we propose function $f(x_i, x_{i+1}, x_{i+2}, x_{i+3}) = \mathrm{LSB}(\lfloor x_i/2 \rfloor)$ that shows the first bit of the message. Thus, if the coefficient x_i is even, increasing the coefficient remains the message constant, while decreasing the coefficient changes the message. In the case of an odd coefficient, we have a reverse situation. For example, if $x_i = 2$, both increasing and keeping the coefficient constant set the first bit of the message to one, while decreasing such coefficient sets the first bit of the message to zero. So, by changing the coefficient with one value and keeping the coefficient constant, we can obtain a message in the first category.

For achieving a message in the second category, we propose equation (7) that shows the first and second bits of the message. Thus, by changing this coefficient, the second bit is changed and the first bit is changed in a state and remains constant for another state.

$$f(x_i, x_{i+1}, x_{i+2}, x_{i+3}) = \mathrm{LSB}\left(\lfloor \frac{x_i}{2} \rfloor + \lfloor \frac{x_{i+1}}{2} \rfloor\right), \mathrm{LSB}(x_{i+1}). \tag{7}$$

Moreover, to obtain a message in the third category, we propose equation (8) that shows the first three bits of the message. Thus, by changing this coefficient, the third bit is changed and the first bit is changed in a state and remains constant for another state.

$$f(x_i, x_{i+1}, x_{i+2}, x_{i+3}) = \mathrm{LSB}\left(\lfloor \frac{x_i}{2} \rfloor + \lfloor \frac{x_{i+1}}{2} \rfloor + \lfloor \frac{x_{i+2}}{2} \rfloor\right), \mathrm{LSB}(x_{i+1}), \mathrm{LSB}(x_{i+2}). \tag{8}$$

Moreover, to obtain a message in the fourth category, we propose equation (9) that shows the first three bits of the message. Thus, by changing this coefficient, the second and third bits are changed and the first bit (MSB) is changed in a state and remains constant for another state. The following example provides more details about the proposed approach.

$$f(x_i, x_{i+1}, x_{i+2}, x_{i+3}) = \mathrm{LSB}\left(\lfloor \frac{x_i}{2} \rfloor + \lfloor \frac{x_{i+1}}{2} \rfloor + \lfloor \frac{x_{i+2}}{2} \rfloor + \lfloor \frac{x_{i+3}}{2} \rfloor\right), \mathrm{LSB}(x_{i+1} + x_{i+3}), \mathrm{LSB}(x_{i+2} + x_{i+3}) \tag{9}$$

For example, for numbers $(8, 14, 21, 29)$ we obtain values $x_i = 8$, $x_{i+1} = 14$, $x_{i+2} = 21$, and $x_{i+3} = 29$. Clearly, we can generate different values by increasing/decreasing each of these coefficients. Table 3 shows these different values. This table shows that for $n = 3$ (a numerical message in the range of $[0, 7]$) and four coefficients, we need only one change in

Table 1: The coefficient values for LSB MR function with three coefficients.

Operation performed	Created coefficients	The binary value; resulting from apply mapping	The decimal value; resulting from apply mapping
x_i, x_{i+1}, x_{i+2}	8, 14, 21	101	5
$x_i + 1, x_{i+1}, x_{i+2}$	9, 14, 21	001	1
$x_i - 1, x_{i+1}, x_{i+2}$	7, 14, 21	000	0
$x_i, x_{i+1} + 1, x_{i+2}$	8, 15, 21	111	7
$x_i, x_{i+1} - 1, x_{i+2}$	8, 13, 21	011	3
$x_i, x_{i+1}, x_{i+2} + 1$	8, 14, 22	110	6
$x_i, x_{i+1}, x_{i+2} - 1$	8, 14, 20	100	4

Table 2: Categorizing the various states of the message for a three-bit message.

Category	Variable bit in the category	State of bits compared to the initial value of the message
First category	First bit	Constant, constant, constant; variable, constant, constant
Second category	First bit	Constant, variable, constant; variable, variable, constant
Third category	First bit	Constant, constant, variable; variable, constant, variable
Fourth category	First bit	Constant, variable, variable; variable, variable, variable

Table 3: The coefficient values for the proposed function with four coefficients.

Operation performed	Created coefficients	The binary value; resulting from apply mapping	The decimal value; resulting from apply mapping
$x_i, x_{i+1}, x_{i+2}, x_{i+3}$	8, 14, 21, 29	110	6
$x_i + 1, x_{i+1}, x_{i+2}, x_{i+3}$	9, 14, 21, 29	110	6
$x_i - 1, x_{i+1}, x_{i+2}, x_{i+3}$	7, 14, 21, 29	010	2
$x_i, x_{i+1} + 1, x_{i+2}, x_{i+3}$	8, 15, 21, 29	100	4
$x_i, x_{i+1} - 1, x_{i+2}, x_{i+3}$	8, 13, 21, 29	000	0
$x_i, x_{i+1}, x_{i+2} + 1, x_{i+3}$	8, 14, 22, 29	011	3
$x_i, x_{i+1}, x_{i+2} - 1, x_{i+3}$	8, 14, 20, 29	111	7
$x_i, x_{i+1}, x_{i+2}, x_{i+3} + 1$	8, 14, 21, 30	001	1
$x_i, x_{i+1}, x_{i+2}, x_{i+3} - 1$	8, 14, 21, 28	101	5

the coefficient to generate the message. Thus, by using the proposed mapping function, we can hide the secret message only by one change. It is to be noted that Table 3 shows 9 numbers for 9 different states in which one of the numbers is replicated.

For the n-bit message, we have to categorize the various states of the message like Table 2. Then, we have to propose a function similar to $n = 3$.

Our proposed mapping function is inspired by LSB MR while it can embed n bits into 2^{n-1} coefficients. Moreover, the proposed function needs fewer changes compared to LSB MR when $n > 2$. Also, the capacity of the proposed function is 75% higher than the F5 method for $n = 3$ and that is over 75% higher than the F5 method for $n > 3$. Now we leverage our mapping function to propose an image steganography scheme. The proposed scheme contains two phases including embedding messages and extracting the message. We explain these two phases in more detail in the next sections. Embedding message and extracting the message perform for $n = 3$ (a numerical message in the range of $[0, 7]$) and four coefficients. For the n-bit message ($n > 3$), we have to propose the function similar to $n = 3$ and embedding message and extracting the message perform similar to $n = 3$.

4.1. Embedding Phase. The main steps in the embedding phase of our steganography approach are shown in Figure 2. In the embedding phase, we perform the following steps:

(i) We first convert the secret message into binary and thus we obtain a bitstream as the result of this step.

(ii) We divide the obtained bitstream into a set of groups with three bits in each group. To this end, from the least significant bit, we group every three continuous bits in a group.

(iii) In this step, a set of pixels from the cover image is selected based on the key to embedding the secret message. It is to be noted that the number of pixels selected in this step is 1.33× length of the secret message. This is because every three bits in the secret message are hidden into four pixels of the cover image.

(iv) Now we generate Table 3 for every four coefficients. Note that by only one change in each coefficient we have a number in the range of $[0, 7]$.

(v) Based on the bits in the secret message, we choose the operations needed to generate such a message.

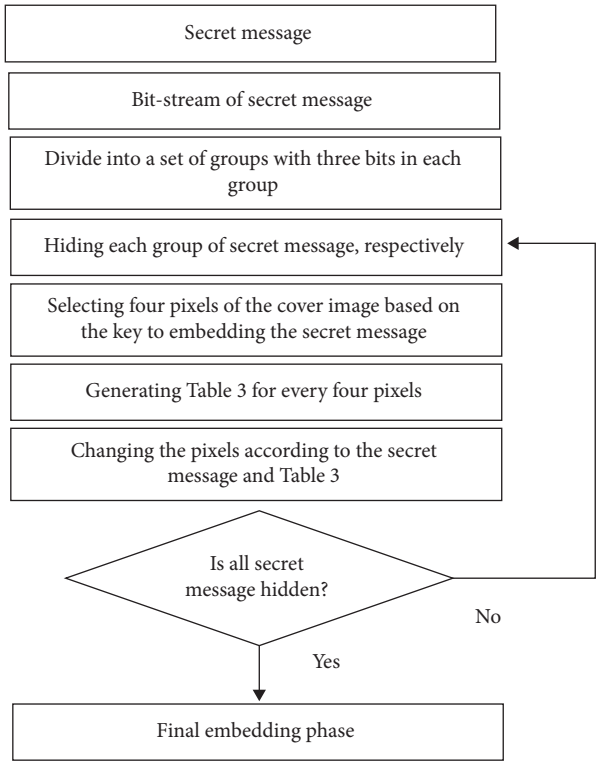


Figure 2: The embedding phase.

(vi) Now we perform the obtained operations over the coefficients to hide the secret message.

4.2. Extracting Phase. The main steps in the extracting phase of our steganography approach are shown in Figure 3.

In the extracting phase, we only need to perform the mapping function over each group of four coefficients from the received image. As a result, we obtain a secret message. For example, for coefficients $x_i = 8, x_{i+1} = 14, x_{i+2} = 20$, and $x_{i+3} = 29$ we obtain the secret message $m = 111$.

In the extracting phase, we perform the following steps:

(i) We first extract pixels of the cover image based on the key in the embedding phase

(ii) We divide the pixels into a set of groups with four pixels in each group

(iii) We extract a secret message according to the four pixels and equation (9) for each group

(iv) We add the secret messages to each other and thus we obtain a bit-stream of secret messages as the result of this step

5. Experiments

In this section, we evaluate the performance of our approach.

5.1. Experimental Environment. We evaluate our method using several public datasets. We also measure the performance of our approach based on PSNR, MSE, capacity, and SSIM metrics. We also provide a comprehensive steganalysis

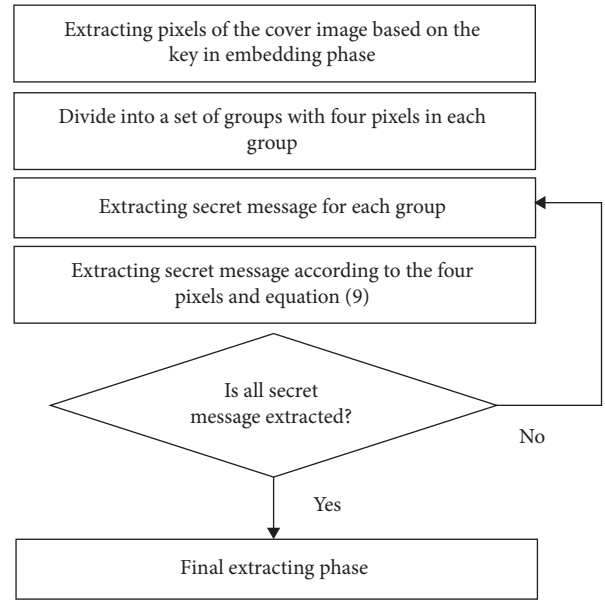


Figure 3: The extracting phase.

to study the detection of hidden messages in our steganography scheme. All the experiments were conducted on a PC with 2.00 GHz Intel Core 2 Duo processor and 8 GB RAM running Windows 10. The program code was written in Matlab 2018a.

5.2. Evaluation Metrics. To evaluate the performance of the stego image, we employed two metrics related to the quality of the image: Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). The MSE metric indicates the pixel-by-pixel difference between the cover image and stego image, which is calculated as follows:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [C(i,j) - S(i,j)]^2, \tag{10}$$

where $C(i, j)$ and $S(i, j)$ are the illumination severities in the cover image and stego image, respectively. Moreover, M and N are the dimensions of the cover image and the stego image, respectively. The lower value of MSE means less difference between the two images. Moreover, lower MSE indicates a higher quality of the stego image, higher transparency, and security.

Another metrics for evaluating the stego image is PSNR, defined as follows:

$$\mathrm{PSNR} = 10 \times \log \frac{p^2}{\mathrm{MSE}}, \tag{11}$$

$$P = \max(C(i,j), S(i,j)). \tag{12}$$

A larger value of PSNR indicates that the steganography scheme highly protects the quality of the image.

Another performance measurement is capacity. Capacity is the amount of data that can be hidden within the cover

image, which shows that the max message size can be inserted into an image and is calculated as follows [47]:

$$\text{Capacity} = \frac{\text{Total number of bits embedded into image}}{\text{Total number of pixels}}. \tag{13}$$

5.3. PSNR and Capacity Analysis. In this section, we evaluate the performance of the proposed scheme over three popular images in steganography including Cameraman, Coins, and Football as the secret message. Using such images generates a very large secret message for our scheme. This helps us to employ the whole capacity of the steganography scheme and fairly evaluates the capacity of the steganography scheme. The secret message is hidden within six images as the cover images including Lenna, peppers, Baboon, Autumn, Kids, and Office. For example, we embedded the Cameraman image within the Lenna image. Figure 4 shows these two images before and after steganography.

To evaluate the effectiveness, we measure the MSE and PSNR for our approach and four other recent approaches: LSBR, Inverted LSB [2] (called ILSB), LSB MR [43] (called LSB MR), and F5 [16]. Note that the F5 scheme is in the frequency domain while our proposed approach is in the spatial domain. In order to fairly compare our approach against F5, we need a synchronization phase. Thus, we converted the hamming coding used in F5 into the spatial domain and we then compare two schemes. It is to be noted that using the Hamming coding into the spatial domain increases the capacity of the steganography approach. This confirms that our comparative evaluation of the capacity of two schemes is fair.

Table 4 reports PSNR of Advanced LSB MR and other four approaches to the mentioned images. Figures 5–7 show the PSNR values for different capacity values of Advanced LSB MR and four other approaches. As one can see in the results, our approach outperforms other schemes for PSNR. Consequently, Advanced LSB MR provides a higher level of confidentiality. Moreover, the PSNR for ILSB is higher than the basic LSB model (LSBR). This is because the ILSB reverses only the least significant bit for a group of pixels based on a predefined priority. This applied fewer changes in the cover image compared to the LSBR model.

The results in Table 4 show that the LSB MR presents higher PSNR compared to both LSBR and ILSB. This is due to the fact that the LSB MR embeds every two bits of the secret message by changing the maximum one pixel of the cover image. Note that the capacities for all three methods LSBR, LSB MR, and ILSB are identical.

In the image steganography based on Hamming coding, such as F5, we can consider various values for the number of bits in the secret message (value of n). In our experiments, we first set $n = 3$ and observed that none of the secret images fit in the cover image. However, for $n = 2$, we observed that all three secret images could fit within the cover images. Thus, Table 4 reports only the results of experiments for $n = 2$ for the F5 scheme where every two bits of the secret message is embedded within three pixels of the cover image with maximum one change in each pixel. Thus, the number of changes in F5 is less than both LSBR and ILSB while it is very close to the number of changes in LSB MR. Similarly, F5 presents higher PSNR compared to both LSBR and ILSB, and its PSNR is slightly close to PSNR in LSB MR. It is also to be noted that F5 needs three pixels of the cover image to hide two bits of the secret message, while the other three schemes (LSBR, ILSB, and LSB MR) need only three pixels to hide two bits. This fact makes the capacity of F5 around 33% less than the capacity of the other three schemes. Such a decrease in capacity is obtained from the following equation:

$$\text{RCChtoCc} = \frac{C_H - C_C}{C_C} \times 100 = \frac{(2/3) - (2/2)}{2/2} \times 100 = -33\%, \tag{14}$$

where C_H is capacity of the steganography scheme based on hamming coding, C_C indicates the capacity of the other steganography schemes (LSBR, ILSB, and LSB MR), and RCChtoCc is the change ratio of capacity of the steganography scheme based on hamming coding to the other steganography schemes.

The results presented in Table 4 show that the increase in PSNR leads to a decrease in the capacity of a steganography scheme. In steganography, security is a significant metric. Since there is a considerable difference between the PSNR in F5 and two other schemes, LSBR and ILSB, the difference between their capacities is negligible. On the other side, F5 and LSB MR present approximately similar PSNR and we can similarly rank the performance of these two schemes.

For $n = 3$, in the proposed scheme, we embed three bits of the secret message into four pixels of the cover image with the maximum of one change in each pixel. Thus, the number of changes needed in our proposed method (and consequently PSNR) is considerably less than the other four schemes. It is to be reminded that the other three schemes including, LSBR, ILSB, and LSB MR, need four pixels for embedding four bits. Therefore, the capacity of the proposed method is 25% less than these three schemes. Such a decrease in capacity is obtained from the following equation:

$$\text{RCCptoCc} = \frac{C_P - C_C}{C_C} \times 100 = \frac{(3/4) - (4/4)}{4/4} \times 100 = -25\%,$$

$$\text{NC}_{\text{proposed},4} = 1,$$

$$\text{NC}_{\text{LSBMR},4} = 2, \tag{15}$$

where $\text{NC}_{\text{proposed},4}$ and $\text{NC}_{\text{LSBMR},4}$ indicate the number of changes for the proposed method and LSB MR in four pixels, respectively. Also, C_P is the capacity of the proposed scheme, C_C indicates the capacity of the other three steganography schemes, and RCCptoCc is the change ratio of capacity of the proposed scheme to the other steganography schemes. One can also see in Table 4 that our scheme provides higher capacity and PSNR compared to F5. The capacity of our scheme for $n = 3$ is 12.5% higher than the capacity of F5 for $n = 2$. Such a difference between the capacity of our scheme and F5 is obtained from the following equation:

(a) (b) (c) (d)

Figure 4: The results of the proposed method. (a) Secret message. (b) Cover image. (c) Extracted message. (d) Stego image.

Table 4: Comparison of Advanced LSB MR with other methods.

Cover image	Secret image	Proposed method		LSBR [48]		Method [2]		LSB MR [43]		F5 [16] ($n = 2$)	
		PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Lenna 512 ∗ 512	Cameraman 64 ∗ 64	63.3454	0.0301	60.1559	0.0627	60.2061	0.062	61.4246	0.0468	61.8454	0.0425
Lenna 512 ∗ 512	Coins 64 ∗ 79	62.3319	0.038	59.2896	0.0766	59.2808	0.0767	60.4781	0.0582	60.8286	0.0537
Lenna 512 ∗ 512	Football 64 ∗ 80	62.3132	0.0382	59.182	0.0785	59.2316	0.0776	60.4466	0.0587	60.9579	0.0522
Peppers 384 ∗ 512	Cameraman 64 ∗ 64	62.0479	0.0406	58.9338	0.0831	58.9348	0.0831	60.1452	0.0629	60.6226	0.0563
Peppers 384 ∗ 512	Coins 64 ∗ 79	61.0759	0.0508	58.0187	0.1026	58.0159	0.1027	59.2012	0.0782	59.5826	0.0716
Peppers 384 ∗ 512	Football 64 ∗ 80	61.1017	0.0505	57.9516	0.1042	57.9665	0.1039	59.2296	0.0776	59.6517	0.0705
Baboon 512 ∗ 512	Cameraman 64 ∗ 64	63.2896	0.0305	60.1709	0.0625	60.1943	0.0622	61.4327	0.0468	61.8873	0.0421
Baboon 512 ∗ 512	Coins 64 ∗ 79	62.3275	0.038	59.2683	0.077	59.2707	0.0769	60.4283	0.0589	60.8065	0.0540
Baboon 512 ∗ 512	Football 64 ∗ 80	62.3876	0.0375	59.2139	0.0779	59.2225	0.778	60.4036	0.0593	60.9363	0.0524
Autumn 345 ∗ 206	Cameraman 64 ∗ 64	57.6325	0.1122	54.5149	0.2299	54.5354	0.2288	55.6271	0.178	56.2428	0.1545
Autumn 345 ∗ 206	Coins 64 ∗ 79	56.6477	0.1407	53.5761	0.2854	53.6522	0.2805	54.7455	0.218	55.2091	0.1960
Autumn 345 ∗ 206	Football 64 ∗ 80	56.6213	0.1416	53.5644	0.2862	53.5289	0.2885	54.6937	0.2207	55.2476	0.1942
Kids 318 ∗ 400	Cameraman 64 ∗ 64	59.5063	0.0729	57.0642	0.1278	57.1522	0.1253	58.3622	0.0948	58.9518	0.0828
Kids 318 ∗ 400	Coins 64 ∗ 79	58.5506	0.0908	56.1865	0.1565	56.1869	0.1565	57.4035	0.1182	58.0010	0.1030
Kids 318 ∗ 400	Football 64 ∗ 80	58.5638	0.0905	56.0809	0.1603	56.0516	0.1614	57.4368	0.1173	58.0386	0.1021
Office 903 ∗ 600	Cameraman 64 ∗ 64	66.4486	0.0147	63.3465	0.0301	63.3625	0.03	64.6074	0.0225	65.0132	0.0205
Office 903 ∗ 600	Coins 64 ∗ 79	65.4766	0.0184	62.4273	0.0372	62.4457	0.037	63.5732	0.0286	64.0365	0.0257
Office 903 ∗ 600	Football 64 ∗ 80	65.5089	0.0183	62.3763	0.0376	62.3521	0.0378	63.6246	0.0282	64.0852	0.0254
Min		56.6213	0.0147	53.5644	0.0301	53.5289	0.03	54.6937	0.0225	55.2091	0.0205
Max		66.4486	0.1416	63.3465	0.2862	63.3625	0.778	64.6074	0.2207	65.0132	0.196
Average		61.398	0.0585	58.406	0.1153	58.421	0.1538	59.6257	0.08742	60.1080	0.0777

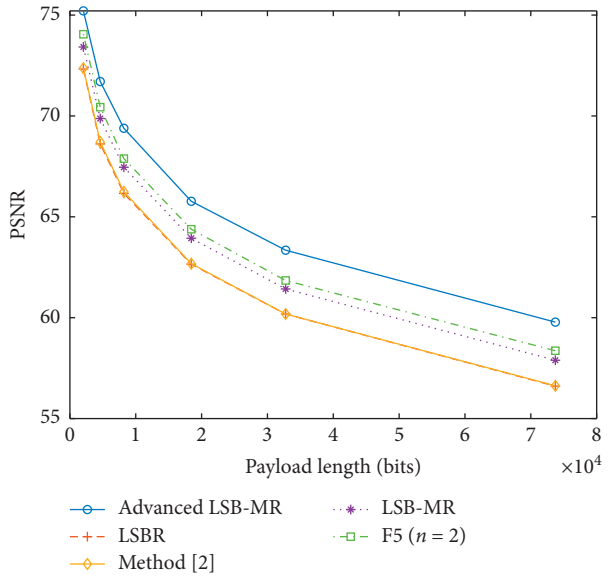


Figure 5: Comparison of Advanced LSB MR with other methods for different capacity values and Lenna cover image.

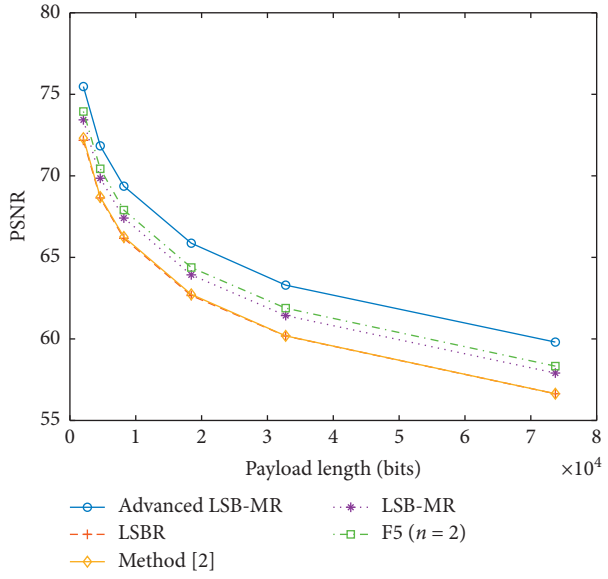


Figure 6: Comparison of Advanced LSB MR with other methods for different capacity values and peppers cover image.

$$\text{RCCptoCh} = \frac{C_P - C_H}{C_H} \times 100 = \frac{(3/4) - (2/3)}{2/3} \times 100 = 12.5\%,$$

$$\text{NC}_{\text{proposed},12} = 3,$$

$$\text{NC}_{\text{F5},12} = 4, \tag{16}$$

where $\text{NC}_{\text{proposed},12}$ and $\text{NC}_{\text{F5},12}$ indicate the number of changes for the proposed method and F5 in 12 pixels, respectively. Also, C_P is the capacity of the proposed scheme

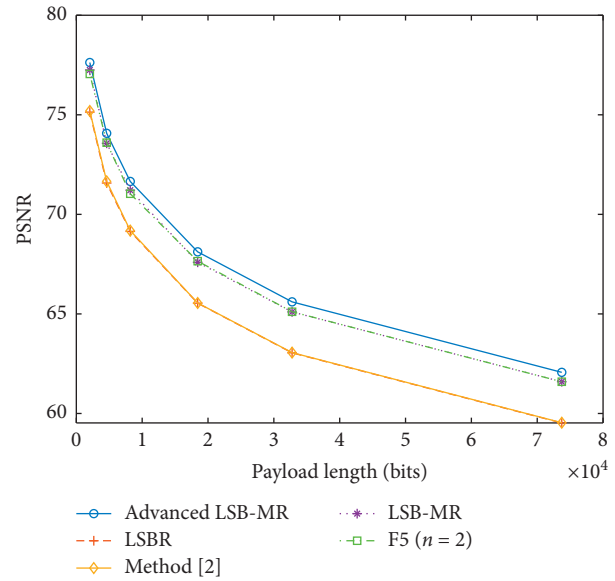


Figure 7: Comparison of Advanced LSB MR with other methods for different capacity values and Babbon cover image.

for $n = 3$, and C_H indicates the capacity of the F5 scheme for $n = 2$, and RCCptoCh is the change ratio of capacity of the proposed scheme to the steganography scheme based on hamming coding. For $n = 3$, our scheme provides a higher capacity compared to F5. The capacity of our scheme is 75% higher than the capacity of F5. Such a difference between the capacity of our scheme and F5 is obtained from the following equation:

$$\text{RCCptoCh} = \frac{C_P - C_H}{C_H} \times 100 = \frac{(3/4) - (3/7)}{3/7} \times 100 = 75\%, \tag{17}$$

where C_P is the capacity of the proposed scheme for $n = 3$ and C_H indicates the capacity of the F5 scheme for $n = 3$.

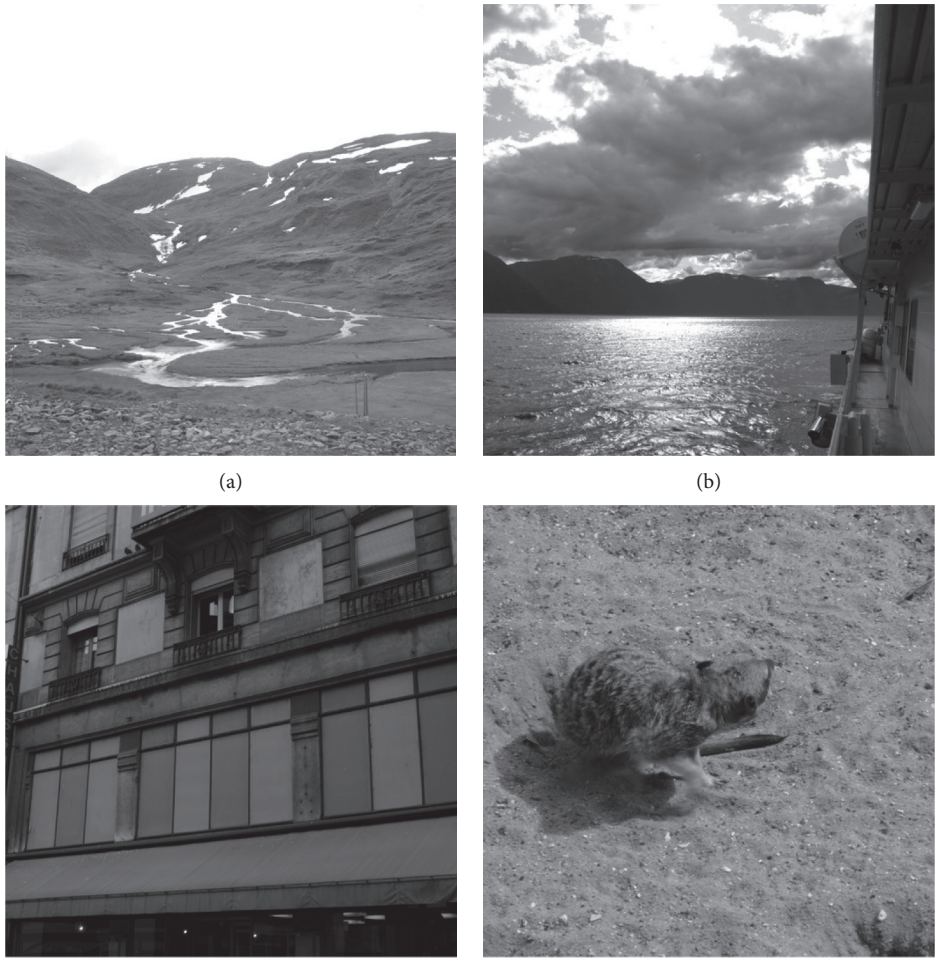
For $n > 2$, the capacity of the proposed scheme is $(((2^n - 1)/2^{n-1}) - 1) \times 100\%$ higher than the F5 method. Such a difference between the capacity of our scheme and F5 is obtained from the following equation:

$$\begin{aligned} \text{RCCptoCh} &= \frac{C_P - C_H}{C_H} \times 100 \\ &= \frac{\left(n/2^{n-1}\right) - \left(n/\left(2^n - 1\right)\right)}{\left(n/\left(2^n - 1\right)\right)} \times 100 \\ &= \left(\frac{2^n - 1}{2^{n-1}} - 1\right) \times 100\%. \end{aligned} \tag{18}$$

5.4. SSIM Analysis. In this experiment, we aim to evaluate our approach based on the structural index similarity (SSIM) metric, which is a tool used to measure image quality [49]. Especially in the image steganography, SSIM and PSNR are used to measure the quality of imperceptibility. SSIM is designed based on brightness, contrast, and structure to

Table 5: The SSIM measurement results (max payload) of image with the LSBR, ILSB, LSB MR, and Advanced LSB MR methods.

Image	Proposed method	LSBR	Method [2]	LSB MR
Lenna 512 × 512	0.9988	0.9980	0.9981	0.9983
Peppers 384 × 512	0.9982	0.9973	0.9974	0.9975
Baboon 512 × 512	0.9996	0.9994	0.9994	0.9994
Autumn 345 × 206	0.9995	0.9992	0.9992	0.9993
Kids 318 × 400	0.9981	0.9965	0.9968	0.9971
Office 903 × 600	0.9988	0.9981	0.9982	0.9983



(a) (b) (c) (d)

Figure 8: Some samples of the images of BOSSbase.

better suit the performance of the human visual system [49]. SSIM can be defined as follows:

$$\mathrm{SSIM} = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{19}$$

where μ_x and μ_y are the sample standard deviations of x and y, respectively; μ_{xy} is the sample correlation coefficient between x and y; and μ_x and μ_y are the sample means of x and y, respectively. The constants C_1 and C_2 are used to stabilize the algorithm when the denominators approach to zero. These statistics are calculated within a local window.

Table 5 reports SSIM of Advanced LSB MR and other three approaches to the mentioned images. The results show that Advanced LSB MR presents higher SSIM compared to LSBR, ILSB, and LSB MR. This is due to the fact that the Advanced LSB MR embeds every three bits of the secret message by changing the maximum of one pixel of the cover image. For $n = 3$, the capacity of the proposed method is 25% less than these three schemes.

5.5. Steganalysis. Steganalysis is the study of detecting messages hidden using steganography. In this section, we perform security analysis to show the resistance of the proposed work against steganalysis. The proposed method is a new method of coding for steganography and can be used in the embedding step of every method of steganography to

Table 6: Detection error M_E for SRNet for two payloads in bpp and four spatial domains embedding algorithms.

	0.2	0.4
S-UNI	0.2081	0.1017
S-UNI with Advanced LSB MR	0.3081	0.1516
HILL	0.2341	0.1402
HILL with Advanced LSB MR	0.3458	0.2051

reduce the change in the stego image. Therefore, to evaluate the performance of the Advanced LSB MR, we used Advanced LSB MR in the embedding step of HILL [50] and S-UNIWARD [51] for two payloads of 0.2 and 0.4 bpp (bits per pixel), respectively. We use the Advanced LSB MR with $n = 3$ (bits of the secret message) for embedding secret messages in coefficients selected by each steganography method. Also, SRNet [52] is used for steganalysis.

The S-UNIWARD method can be applied for embedding in an arbitrary domain. The embedding distortion is computed as a sum of relative changes of coefficients in a directional filter bank decomposition of the cover image. The directionality forces the embedding changes to such parts of the cover object such as textures or noisy regions while avoiding clean edges or smooth regions [51]. The HILL method presents a cost function for spatial image steganography. The cost function is designed by using a high-pass filter to locate the less predictable parts in an image, and then using two low-pass filters to make the low-cost values more clustered [50].

The SRNet method is a deep residual architecture designed to minimize the use of heuristics and externally enforced elements. This method provides state-of-the-art detection accuracy for both spatial-domain and JPEG steganography. We used BOSSbase 1.01 database [53] to evaluate the results of the proposed method. The specifications of the BOSSbase images are as follows:

(i) The resolution of these images is 512×512

(ii) Images are taken by eight different cameras

(iii) These images are uncompressed

(iv) The number of the image is 10000

(v) The image format is PGM

(vi) The images have different properties regarding textures and smooth areas

Some samples of images of the BOSSbase database are shown in Figure 8. The database is randomly split into two batches of 5000 images. The 5000 images have been used for SRNet training and the 1000 images have been used randomly for testing. The detector accuracy was reported using the minimal total detection error on the testing set under equal prior $M_E = \min M_{\text{FA}}\,(1/2)\,(M_{\text{FA}} + M_{\text{MD}})$, where M_{MD} and M_{FA} are the missed-detection and false-alarm probabilities. The detection error M_E for HILL [50] and S-UNIWARD [51] with and without Advanced LSB MR for two payloads is shown in Table 6.

As one can see in Table 6, the detection error for SRNet for different payloads using Advanced LSB MR in the embedding step of HILL and S-UNIWARD is increased in both steganography techniques. Therefore, Advanced LSB MR can be used in the embedding step of every method of steganography to reduce the change in the stego image.

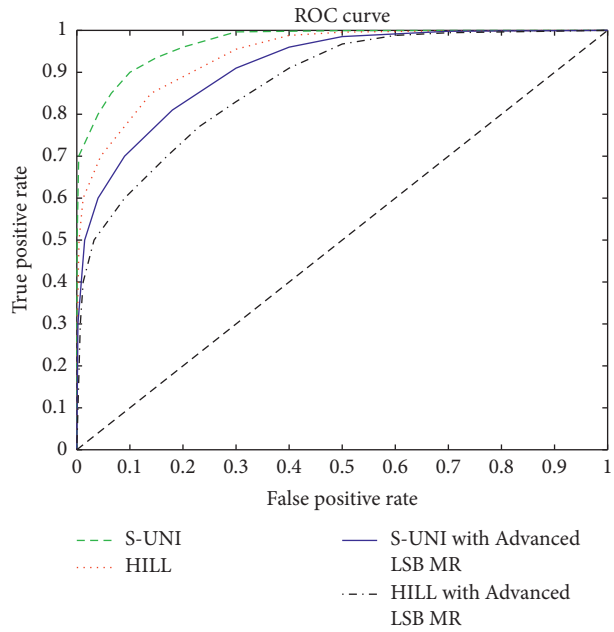


Figure 9: ROC curves of SRNet for S-UNIWARD, S-UNI with Advanced LSB MR, HILL, and HILL with Advanced LSB MR for payload 0.4 together with two detection performance measures: M_{FA} for $M_D = 0.5$ and $M_D = 0.3$.

Table 7: ROC curves of SRNet for S-UNIWARD, S-UNI with Advanced LSB MR, HILL, and HILL with Advanced LSB MR for payload 0.4 together with two detection performance measures: M_{FA} for $M_D = 0.5$ and $M_D = 0.3$.

Embedding	M_{FA} for $M_D = 0.5$	M_{FA} for $M_D = 0.3$
S-UNI	0.0006	0.0002
S-UNI with Advanced LSB MR	0.0148	0.0021
HILL	0.0042	0.0010
HILL with Advanced LSB MR	0.0325	0.0066

The ROC curve is a graph of the true-positive rate to the false-positive rate. The ROC curve for a complete detector reaches a point at FPR = 0 and TPR = 1. ROC curves, independent of their decision thresholds, allow a reliable evaluation of steganalysis detectors [54]. In this paper, the ROC curves are shown the false-alarm rates for stego-image detection probability $M_D = 1 - M_{\text{MD}} = 0.5$ and 0.3.

Figure 9 shows four ROC curves including SRNet for S-UNIWARD, S-UNI with Advanced LSB MR, HILL and HILL with Advanced LSB MR for payload 0.4, and the false alarm rates M_{FA} for two test powers. For the payload 0.4 bpp, $M_D = 1 - M_M D = 0.5$ can be achieved with $M_{\text{FA}} = 6 \times 10^{-4}$ for S-UNIWARD, 14.8×10^{-3} for S-UNI with Advanced LSB MR, 4.2×10^{-3} for HILL and 32.5×10^{-3} for HILL with Advanced LSB MR (see Table 7).

As shown in Figure 9, the area below the ROC curve for SRNet using Advanced LSB MR in the embedding step of HILL and S-UNIWARD is decreased in both steganography

techniques. Therefore, the change in the stego image has been decreased by Advanced LSB MR and the detection error for SRNet has been increased.

6. Conclusions

In this paper, we proposed a novel and LSB-based approach for image steganography. In the embedding phase of our steganography scheme, we first divide the secret message into groups of n bit, and then we choose 2^{n-1} pixels of the input image to hide each of these groups. Our experiment results show that our proposed scheme needs fewer changes compared to the basic LSB method. Moreover, the capacity of the proposed method is $(((2^n - 1)/2^{n-1}) - 1) \times 100\%$ higher than the F5 approach. Also, our scheme provides a high PSNR value of hiding secret message bits in the image compared to the state of the art in the image steganography community, which decreases the chance of detecting the secret message from the stego image.

The coding methods can be used in the first step of each steganography method to reduce the change in the stego image. The proposed method is a new method of coding for steganography and this method can be used in the first step of every method of steganography. For future work, we plan to apply our idea for providing a video, audio, and text steganography scheme. For example, three bits of the secret message is hidden in four coefficients with only one change. The coefficients can be Spatial Domain and Transform Domain (DCT, DWT, or others depending on the steganography method). They can also be further upgraded by examining other coding methods and their disadvantages.

Data Availability

No data were used to support this study

Disclosure

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Jiang, Y. Pang, and S. Xiong, "A high capacity steganographic method based on quantization table modification and f5 algorithm," *Circuits, Systems, and Signal Processing*, vol. 33, no. 5, pp. 1611–1626, 2014.

[2] R. Bhardwaj and V. Sharma, "Image steganography based on complemented message and inverted bit LSB substitution," *Procedia Computer Science*, vol. 93, pp. 832–838, 2016.

[3] S. L. Farrag and W. Alexan, "A high capacity geometrical domain based 3D image steganography scheme," in *Proceedings of the 2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–7, Rabat, Morocco, 2019.

[4] P. Rai, S. Gurung, and M. K. Ghose, "Analysis of image steganography techniques: a survey," *International Journal of Computer Applications*, vol. 114, no. 1, pp. 11–17, 2015.

[5] M. Khan, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "Image steganography using uncorrelated color space and its application for security of visual contents in online social networks," *Future Generation Computer Systems*, vol. 86, pp. 951–960, 2018.

[6] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, 1998.

[7] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[8] I. Hafi, M. Noman, M. Gohar et al., "An adaptive hybrid fuzzy-wavelet approach for image steganography using bit reduction and pixel adjustment," *Soft Computing*, vol. 22, no. 5, pp. 1555–1567, 2018.

[9] M. Fateh and M. Rezvani, "An email-based high capacity text steganography using repeating characters," *International Journal of Computers and Applications*, pp. 1–7, 2018.

[10] K. J. Devi, *A secure image steganography using lsb technique and pseudo random encoding technique*, Ph.D. thesis, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India, 2013.

[11] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: a survey," *Signal Processing: Image Communication*, vol. 65, no. 1, pp. 46–66, 2018.

[12] G. Liu, Z. Zhang, Y. Dai, and S. Lian, "Improved LSB-matching steganography for preserving second-order statistics," *Journal of Multimedia*, vol. 5, pp. 458–463, 2010.

[13] D. R. I. M. Setiadi and J. Jumanto, "An enhanced LSB-image steganography using the hybrid Canny-Sobel edge detection," *Cybernetics and Information Technologies*, vol. 18, no. 2, pp. 74–88, 2018.

[14] A. A. Shejul and U. Kulkarni, "A DWT based approach for steganography using biometrics," in *Proceedings of the 2010 International Conference on Data Storage and Data Engineering*, Bangalore, India, 2010.

[15] R. Roy, S. Changder, A. Sarkar, and N. C. Debnath, "Evaluating image steganography techniques: future research challenges," in *Proceedings of the 2013 International Conference on Computing, Management and Telecommunications (ComManTel)*, pp. 309–314, Ho Chi Minh City, Vietnam, 2013.

[16] A. Westfeld, "F5—a steganographic algorithm," in *Information Hiding*, I. S. Moskowitz, Ed., pp. 289–302, Springer, Berlin, Germany, 2001.

[17] G. Chen, M. Cao, D. Fu, and Q. Ma, "Research on an steganographic algorithm based on image edge," in *Proceedings of the 2011 International Conference on Internet Technology and Applications (iTAP)*, pp. 1–4, Wuhan, China, 2011.

[18] T. Morkel, J. H. Eloff, and M. S. Olivier, "An overview of image steganography," in *Proceedings of the 5th Annual Information Security South Africa Conference*, Sandton, South Africa, 2005.

[19] K. Raja, C. Chowdary, K. Venugopal, and L. Patnaik, "Pixel indicator high capacity technique for RGB image based steganography," in *Proceedings of the 5th IEEE International Workshop on Signal Processing and Its Applications (WoSPA2008)*, pp. 1–4, Sharjah, UAE, 2010.

[20] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[21] P. Sallee, "Model-based steganography," in *Proceedings of the 2nd International Workshop on Digital Watermarking*, pp. 154–167, Seoul, Republic of Korea, 2003.

[22] S. Singh, "Adaptive PVD and LSB based high capacity data hiding scheme," *Multimedia Tools and Applications*, vol. 79, pp. 18815–18837, 2020.

[23] S. Samima, R. Roy, and S. Changder, "Secure key based image realization steganography," in *Proceedings of the 2013 IEEE 2nd International Conference on Image Information Processing (ICIIP)*, pp. 377–382, Shimla, India, 2013.

[24] R. Roy and S. Changder, "Image realization steganography with LCS based mapping," in *Proceedings of the 2014 7th International Conference on Contemporary Computing (IC3)*, pp. 218–223, Noida, India, 2014.

[25] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[26] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proceedings—Vision, Image, and Signal Processing*, vol. 152, no. 5, pp. 611–615, 2005.

[27] K. Raja, C. Chowdary, K. Venugopal, and L. Patnaik, "A secure image steganography using LSB, DCT and compression techniques on raw images," in *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing*, pp. 170–176, Bangalore, India, 2005.

[28] M. Y. Valandar, P. Ayubi, and M. J. Barani, "A new transform domain steganography based on modified logistic chaotic map for color images," *Journal of Information Security and Applications*, vol. 34, pp. 142–151, 2017.

[29] M. Y. Valandar, M. J. Barani, P. Ayubi, and M. Aghazadeh, "An integer wavelet transform image steganography method based on 3D sine chaotic map," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 9971–9989, 2019.

[30] A. Gutub and M. Al-Ghamdi, "Hiding shares by multimedia image steganography for optimized counting-based secret sharing," *Multimedia Tools and Applications*, vol. 79, pp. 7951–7985, 2020.

[31] G. Kaur, S. Singh, R. Rani, and R. Kumar, "A comprehensive study of reversible data hiding (RDH) schemes based on pixel value ordering (PVO)," *Archives of Computational Methods in Engineering*, pp. 1–52, 2020.

[32] G. Kaur, S. Singh, and R. Rani, "A high capacity reversible data hiding technique based on pixel value ordering using interlock partitioning," in *Proceedings of the 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 727–732, Noida, India, 2020.

[33] R. Kumar, S. Chand, and S. Singh, "An optimal high capacity reversible data hiding scheme using move to front coding for LZW codes," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 22977–23001, 2019.

[34] G. Swain, "A steganographic method combining LSB substitution and PVD in a block," *Procedia Computer Science*, vol. 85, pp. 39–44, 2016.

[35] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13541–13556, 2016.

[36] R. Kumar, S. Singh, and K.-H. Jung, "Human visual system based enhanced ambtc for color image compression using interpolation," in *Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 903–907, Noida, India, 2019.

[37] A. K. Sahu and G. Swain, "Data hiding using adaptive LSB and PVD technique resisting PDH and RS analysis," *International Journal of Electronic Security and Digital Forensics*, vol. 11, no. 4, pp. 458–476, 2019.

[38] A. K. Sahu and G. Swain, "Pixel overlapping image steganography using PVD and modulus function," *3D Research*, vol. 9, no. 3, p. 40, 2018.

[39] A. K. Sahu and G. Swain, "A novel n-rightmost bit replacement image steganography technique," *3D Research*, vol. 10, no. 1, p. 2, 2019.

[40] A. Sahu, G. Swain, and G. Swain, "Dual stego-imaging based reversible data hiding using improved LSB matching," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 5, pp. 63–73, 2019.

[41] A. Siper, F. Roger, and L. Craig, "The rise of steganography," in *Proceedings of Student/Faculty Research Day, CSIS*, pp. 1–7, New York, NY, USA, 2005.

[42] G. Ashish and K. Vijay, "Comprehensive survey of 3D image steganography techniques," *IET Image Processing*, vol. 12, no. 1, pp. 1–10, 2017.

[43] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, 2006.

[44] F. Huang, Y. Zhong, and J. Huang, "Improved algorithm of edge adaptive image steganography based on LSB matching revisited algorithm," in *Proceedings of the 2013 International Workshop on Digital Watermarking*, pp. 19–31, Auckland, New Zealand, 2013.

[45] Q. Huang and W. Ouyang, "Protect fragile regions in steganography LSB embedding," in *Proceedings of the 2010 3rd International Symposium on Knowledge Acquisition and Modeling (KAM)*, pp. 175–178, Wuhan, China, 2010.

[46] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 201–214, 2010.

[47] M. H. Mohamed and L. M. Mohamed, "High capacity image steganography technique based on LSB substitution method," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, pp. 259–266, 2016.

[48] A. D. Ker, "Improved detection of LSB steganography in grayscale images," in *Proceedings of the 2004 International Workshop on Information Hiding*, pp. 97–115, Toronto, Canada, 2004.

[49] D. R. I. M. Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, p. 1, 2020.

[50] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, Paris, France, 2014.

[51] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, 2014.

[52] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[53] P. Bas, T. Filler, and T. Pevnỳ, "Break our steganographic system: the ins and outs of organizing BOSS," in *Proceedings of the 2011 International Workshop on Information Hiding*, pp. 59–70, Prague, Czech Republic, 2011.

[54] A. Westfeld, "ROC curves for steganalysts," in *Proceedings of the 3rd WAVILA Challenge*, pp. 39–45, Saint-Malo, France, 2007.