




Article

High-Capacity Image Steganography with Minimum Modified Bits Based on Data Mapping and LSB Substitution

Abdul Alif Zakaria ^{1,2,*}, Mehdi Hussain ³, Ainuddin Wahid Abdul Wahab ^{1,*} ,
Mohd Yamani Idna Idris ¹, Norli Anida Abdullah ⁴  and Ki-Hyun Jung ⁵ 

¹ Department of Computer System & Technology, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia; yamani@um.edu.my

² Department of Cryptography Development, CyberSecurity Malaysia, Selangor 43300, Malaysia

³ School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan; mehdi.hussain@seecs.edu.pk

⁴ Centre for Foundation Studies in Sciences, University of Malaya, Kuala Lumpur 50603, Malaysia; norlie@um.edu.my

⁵ Department of Cyber Security, Kyungil University, Gyeongbuk 38428, Korea; khannyjung@gmail.com

* Correspondence: alif@cybersecurity.my (A.A.Z.); ainuddin@um.edu.my (A.W.A.W.);
Tel.: +60-3-89926918 (A.A.Z.); +60-3-79676383 (A.W.A.W.)

Received: 30 August 2018; Accepted: 19 September 2018; Published: 9 November 2018



Abstract: Steganography is the art and practice of communication using hidden messages. The least significant bits (LSB) based method is the well-known type of steganography in the spatial domain. Usually, achieving the larger embedding capacity in LSB-based methods requires a large number of LSB bits modification which indirectly reduces the visual quality of stego-image and increases the risk of steganalysis detection attacks. In this study, we propose a novel steganography method with data mapping strategy which can reduce the number of bits modification per pixel. In the proposed method, four secret data bits are mapped with the four most significant bits of a cover pixel. Furthermore, the only two LSBs of a pixel are modified to indicate the mapping strategy. Experimental results show that the proposed method is able to achieve 3.48% larger embedding capacity while enhancing the visual quality (i.e., peak signal to noise ratio (PSNR) 3.73 dB) and reducing the modification of 0.76 bits per pixel. Moreover, the proposed method provides security against basic Regular and Singular groups (RS) steganalysis and histogram steganalysis detection attacks.

Keywords: steganography; mapping data hiding; information hiding; spatial domain; least significant bits; most significant bits

1. Introduction

With the evolution of the Internet, one of the major concerns is to secure communicated information. Generally, information security is achieved by encryption (cryptography) and information hiding (steganography). In cryptography, information is encoded such that only authorized users can access it. In steganography, the information is concealed in a digital medium such that no one can notice whether the digital medium has the information inside it or not [1].

In recent years, steganography techniques have been employed in various digital mediums, i.e., image, audio, video, and network protocols to achieve secret communication. Similarly, it is employed in various applications; Yesilyurt and Yalman used steganography to secure stored data elements in the cloud [2]. Wu et al. [3] employed steganography for secret sharing and authentication purposes. Xiang et al. used steganography for sending data from a mobile to the cloud and vice

versa [4]. Tondwalkar et al. [5] used steganography process to secure localization of nodes in wireless sensor networks.

Usually, the objectives of image steganography consist of listed key factors, i.e., visual imperceptibility, embedding capacity, robustness, and security. Visual imperceptibility is considered as the resistance of the human visual system to identify the changes in the stego-image that can be measured by the peak signal to noise ratio (PSNR). Embedding capacity refers to the amount of concealed information inside a cover image. Robustness means extracting the original concealed information from stego-image regardless of various types of processing, i.e., cropping, scaling and filtering. Moreover, security against steganalysis and the integrity of secret data can be added to these objectives [1]. Ideally, a steganography algorithm should be able to achieve larger embedding capacity with high visual imperceptibility while maintaining the security and integrity of secret data.

In this study, we will discuss the well-known image-based steganography techniques that deal with the least significant bit (LSB) substitution of the cover image pixels to hide secret data. LSB-based image steganography methods embed secret data in the least significant bits of the image pixel value. Usually, achieving the larger embedding capacity in LSB methods requires a large number of bits modification that can degrade the visual quality. Moreover, it relies on the fact that replacing one or more of the last 1–4 bits of cover image's pixels is imperceptible to the human visual system, but some statistical tests could detect their replacement in appropriate locations [6]. Therefore, one needs to design an embedding method requiring a lower number of bits modifications per pixel during the embedding process; the method should achieve a larger embedding capacity. In this study, the proposed method successfully achieves these objectives.

The organization of the paper is as follows. The literature review of LSB-based steganography methods is discussed in Section 2. Section 3 presents the proposed steganography method with discrete algorithm steps and explanations of embedding and extraction procedure. Section 4 is dedicated to performance analysis (embedding capacity, visual imperceptibility, and statistical security) with a comparison of existing LSB-based steganography techniques. Finally, the conclusions and future direction of the research are discussed in Section 5.

2. Related Work

This section is intended to give a brief literature review of LSB-based image steganography techniques since the proposed method is based on LSB substitution. Usually, in LSB-based methods, the secret data bits are substituted in k -least significant bits of each pixel. In the literature, various types of LSB-based techniques have been proposed [7–23] targeted at high capacity, visual quality and security.

One of the earliest adaptive LSB-based methods was introduced by Yang et al. [7]. This method exploited the brightness, edges, and texture of the cover image to determine the number of k LSBs for data embedding. The higher value of k depends on the noise non-sensitive regions. Furthermore, an optimal pixel adjustment process is employed to enhance the visual quality. To obtain a larger embedding capacity, Tseng et al. [8] proposed a method employing an adaptive LSB substitution for data embedding based on edge computation. A maximum of 4 LSBs are utilized during the embedding process. This method improved the embedding capacity but suffered from low visual imperceptibility (PSNR < 35 dB). Jung and Yoo [9] proposed a semi-reversible data hiding technique involving interpolation and LSB substitution. The interpolation methods are used to scale up and scale down the cover image, where the LSB substitution (up to 3 LSBs) is employed for embedding. This method provides the larger embedding capacity while maintaining acceptable visual quality. However, this method did not discuss any security evaluation. Mohamed and Mohamed [10] also employed the LSB substitution technique by dividing the image into two parts, one for embedding data and the other for indicating the embedding changes. This method increases the embedding capacity and improves the visual quality of the stego image, while the number of LSB substitutions is adaptive,

ranging from 1–5. The visual quality of this method directly depends on the number of utilized LSBs. As the number of LSB substitutions increases, the visual quality of the stego-image decreases.

To further enhance the embedding capacity, researchers combined the other existing methods with the LSB technique. For example, Liao et al. [11] proposed a larger embedding capacity based on LSB with the pixel value difference (PVD) [12] based method. The average difference value of the four-pixel block is used to classify the high and low texture pixels block. Furthermore, the secret data are embedded using k -bit modified LSB substitution. Similarly, Swain et al. [13] integrated the LSB substitution with a PVD method. This method divides the image into 2×2 pixel blocks and applies k -bit LSB in the upper left pixel of the block. Next, a PVD embedding process is applied to the base (upper-left) pixel with the remaining pixels of the block. This method provides the larger embedding capacity with the improved PSNR value. However, the k -bits ranges are from 1 to 3, indirectly indicating that up to 3 LSBs of a pixel can be modified during the embedding process. In [14], Hussain et al. proposed a recursive hybrid approach of LSB with other existing PVD and modification of prediction error (MPE) methods. First, it classified the lower and higher texture of an image; then it employed the LSB substitution and PVD methods. Next, to enhance the embedding capacity it recursively employed the PVD shift and MPE. This method provides larger capacity with acceptable PSNR. However, up to 4 LSBs are utilized for the embedding process. Similarly, Khodaei et al. [15] introduced adaptive LSB embedding by utilizing the PVD characteristics. This method divides the cover image into two-pixel blocks, further computing the difference in value between the two pixels. The numbers of secret bits are estimated based on the computed difference value. Next, the adaptive LSB scheme is employed to embed the number of secret data bits inside the selected pixel blocks. This method also introduced a readjustment process to keep intact the stego-pixels with respective ranges; it successfully achieved the larger embedding capacity by retaining high visual imperceptibility. However, the maximum numbers of 4 LSBs are utilized during the embedding process. Another novel LSB-based technique [16] proposed is bit inversion, aimed at improving the PSNR value. In this inversion technique, certain LSBs of the cover image pixels are changed if they are matched with a particular pattern. Thus, this improved the visual quality but suffered from the low payload.

Recently, another texture-adaptive (i.e., edge-based) LSB method was proposed [17]. In this method, the cover pixels are classified into edge areas and non-edge areas. The edge information is determined by the 3 most significant bits (MSBs) of the cover pixels, where the rest of 5 LSBs are adaptively employed by the LSB substitution. This method obtained high embedding capacity while retaining acceptable visual quality. However, a maximum of 5 LSBs are employed in the embedding process. Similarly, Lee [18] proposed an adaptive LSB method for color images on smartphones. For all RGB channels, it proposed various LSB replacement strategies i.e., all 4 bits in each RGB channel and similarly 4 bits in R, two-2 bits in G, B channels and so on. This method achieved an average 2.8 bits per channel while it retains the high value of PSNR 43.7 dB. Its 4-bit LSB substitution strategy may employ up to 4 LSBs of each channel for embedding.

On the other hand, to ensure data security, various methods employed the compression and encryption techniques to reduce the data size and increase the data security before the embedding process. For example, Kuo et al. [19] employed run-length encoding (RLE) to compress the secret data bits. Furthermore, it efficiently utilized the multi-bit generalized exploiting modification direction for embedding that indirectly reduces the number of modified pixels of a stego-image. This method maintained high visual quality and capacity (due to compression) but it effectively provided low embedding capacity. Similarly, Sethi and Kapoor provide better data security along with imperceptibility and capacity [20]. First, the secret data is encrypted using the Advanced Encryption System (AES) algorithm and converted into four-bits fixed blocks. The LSB substitution is then performed on the basis of four bits. Although the security of data is improved the computational cost of encryption seems to be a major concern. Recently, Muhammad et al. [21] proposed a channel indicator-based lightweight encrypted embedding method. It provides multi-level encryption for

secret data and as well as stego-key. This method retains high imperceptibility and security against detectability but suffers from low embedding capacity.

Most of the aforementioned recent methods are summarized in Table 1. This shows the clear picture of embedding capacity versus visual quality of existing methods. All the statistics are directly taken from the studies, where the embedding capacity depicts the average number of embedding bits per pixel (bpp). Similarly, visual quality measured in peak signal to noise ratio (PSNR) and security of undetectability is shown in the 3rd and 4th column. Finally, the maximum no. of LSBs (or bits) that would be modified during the embedding process of each technique is shown in the last column of the table.

Table 1. General performance comparison of existing methods, PSNR: peak signal to noise ratio.

Approaches	Embedding Capacity (bpp)	Visual Quality (PSNR)	Undetectability Security	Maximum Used LSBs/byte
Tseng et al. 2014 [8]	3.16	33.40	-	4 bits LSB
Jung and Yoo 2015 [9]	2.55	37.56	-	3 bits LSB
Mohamed and Mohamed 2016 [10]	3.19	38.87	-	4 bits LSB
Swain et al. 2016 [13]	3.17	39.29	RS analysis	3 bits LSB
Khodaei et al. 2016 [15]	3.13	38.42	-	4 bits LSB
Hussain et al. 2018 [14]	3.11	38.40	RS analysis	3 bits LSB
Bai et al. 2018 [17]	4.05	30.10	-	1–5 bits LSB
Lee et al. 2018 [18]	2.83	43.7	-	All 4 bits LSB or Two 3 bits one-2 bits RGB LSB
Wien. 2018 [22]	~1.0	32.27	-	1 bits
Setiadi and Jumanto. 2018 [23]	1.13	47.83	-	1 bits + 2 bits + 3 bits RGB LSB

As can be observed in all the aforementioned LSB-based techniques; these were mainly focused on following steganography objectives, i.e., to increase the embedding capacity [8,10,13–15], to enhance visual imperceptibility [13,15,18,23] and to improve security against detectability and hidden data [13,14,20,21]. However, we found tradeoffs among all these objectives because these are correlated to one another. As we noticed (see Table 1), most of the larger capacity-based LSB methods [8,10,13–15,17] employed the maximum LSBs or modify the maximum number of bits in each pixel (i.e., 3–5 bits). This maximum bits modification in pixels increases the visual distortion between cover and stego-pixels (that can be seen as low PSNR value 30 to 33 dB). Consequently, it reduces the stego-image visual quality and increases the risk of steganalysis detection attacks. (However, most of the existing techniques did not perform any steganalysis.) Therefore, a novel steganography approach is required that must achieve larger embedding capacity while retaining the lower number of bits changed per pixel or lesser modification in pixel bits. Furthermore, it must maintain the maximum visual imperceptibility and security against steganalysis detection attacks. In the next section, we will discuss the proposed method.

3. The Proposed Method

In this section, we will present the proposed LSB-based steganography method that aimed at enhancing embedding capacity, achieving good visual quality, and security against steganalysis detection attacks. The proposed method extends the concept of LSB substitution. This employed the mapping of secret data bits with cover pixel bits, where the record of mapping is maintained by the LSB substitution in cover pixels. For data security, the proposed method also applies the permutation on secret data bits using a shared key.

In the embedding process, first, the secret data bits are grouped into two secret bits pairs, i.e., $M_1(10)$, $M_2(10)$, $M_3(11)$, ... as shown in Figure 1a. Similarly, all the cover pixels bits are also categorized as pairs, i.e., Left Pair ($C_{L,pixel}$) and Right Pair ($C_{R,pixel}$) depicted in Figure 1b. $C_{L,pixel}$ denotes the 7th and 8th MSBs of the cover pixel, and similarly, $C_{R,pixel}$ denotes the 5th and 6th MSBs

of the cover pixel. Both $C_{L,pixel}$ and $C_{R,pixel}$ bits would be used for mapping with secret data bits pair. In the same way, the 1st and 2nd LSBs of cover pixel also denoted with Right Bit ($B_{R,pixel}$) and Left Bit ($B_{L,pixel}$) and used as an indicator for mapping of pixels pair. $B_{L,pixel}$ and $B_{R,pixel}$ are closely coupled with $C_{L,pixel}$ and $C_{R,pixel}$ in terms of indicator bits, respectively. However, the mapping strategies consist of the following cases.

- Case 1: If first secret bit pair (i.e., M_1) is matched with the $C_{L,pixel}$ of a cover pixel then $B_{L,pixel}$ is replaced with '1' otherwise replaced it with '0'.
- Case 2: Similarly, if the second secret bit pair (i.e., M_2) is matched with $C_{R,pixel}$, it substitutes $B_{R,pixel}$ with '1' otherwise replaces it with '0'.
- Case 3: If either $B_{L,pixel}$ or $B_{R,pixel}$ is '0', then the 2-LSBs of very next cover pixel would be replaced with the previous unmatched secret bit pair (i.e., M_1 or M_2).
- Case 4: If both $B_{L,pixel}$ and $B_{R,pixel}$ are '0', it would indicate the skipped mapped block (which does not contain any mapping).

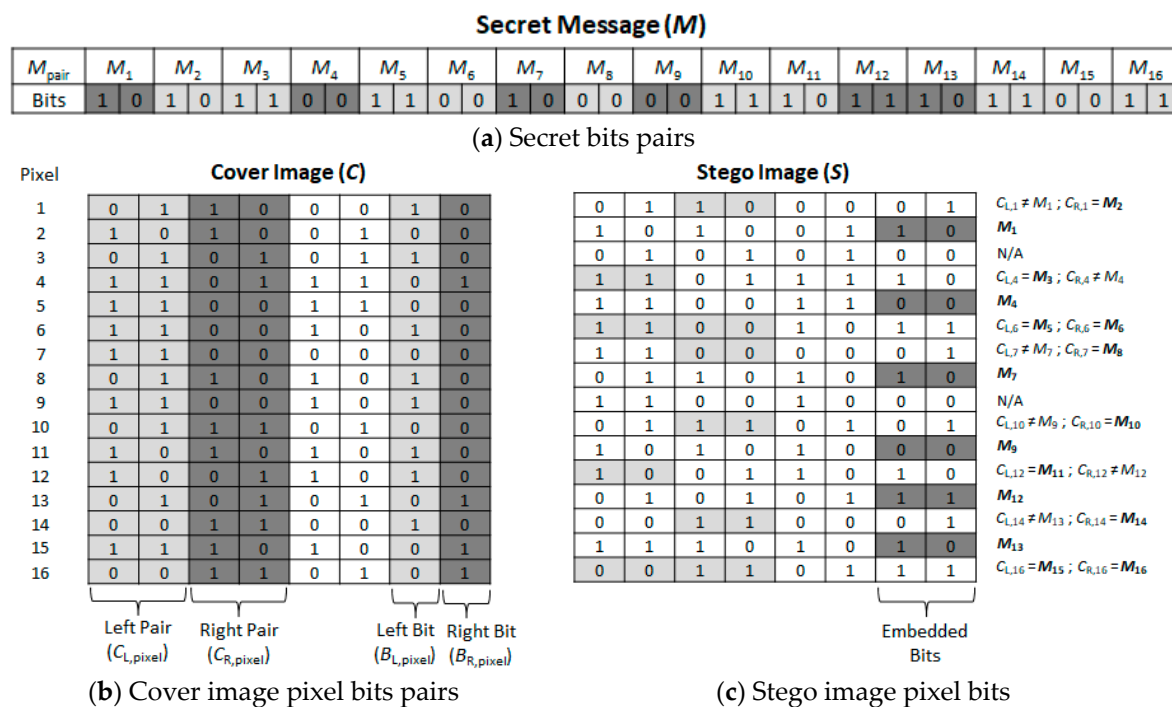


Figure 1. A secret message, cover image, and stego image pixels bits.

For example, from Figure 1, the first pixel of cover image has the following values, $C_{L,1} = 01$, $C_{R,1} = 10$, $B_{L,1} = 1$, $B_{R,1} = 0$ and the secret bits pair are $M_1 = 10$, $M_2 = 10$. From case 1, $C_{L,1} \neq M_1$, i.e., $01 \neq 10$, so it replaces the ($B_{L,1}$) 1 with 0. Similarly, from case 2, $C_{R,1} = M_2$, so it substitutes the 1 in $B_{R,1}$ location. Next, the case 3 conditioned would be satisfied, i.e., if $B_{L,1} = 0$ or $B_{R,1} = 0$, then embed the remaining unmatched secret pair that was $M_1 = 10$. These $M_1 = 10$ are substituted in the 2-LSBs of the next pixel, i.e., $B_{L,2} = 1$ and $B_{R,2} = 0$, this can be shown in the second pixel. Similarly, this process can be extended for rest of the pixels. Table 2 describes the complete embedding steps.

Table 2. The proposed embedding steps.

Input	Stego key (K), secret message (m), and cover image (C)
Output	Stego image (S)
Step 1	Convert K into Sum value.
	Find values of K components $P_1, P_2, P_3, \dots, P_k$ where k is the length of K using ASCII table. (e.g., $K = \text{Alif2s}$)
(a)	i. Refer ASCII table to search values of K components. $P_1 = A = 65, P_2 = l = 108, P_3 = i = 105, P_4 = f = 102, P_5 = 2 = 50, P_6 = s = 115$
	Compute Sum value using the following formula
(b)	i. $Sum = \sum_{i=1}^k (P_i \times i) \bmod 256$ $= [(65 \times 1) + (108 \times 2) + (105 \times 3) + (102 \times 4) + (50 \times 5) + (115 \times 6)] \bmod 256$
Step 2	Compute M .
(a)	Set the bits in m as $m_{bit} = m_1, m_2, m_3, \dots, m_n$ where n is the length of m .
(b)	Set the permuted bits in m as $g_{bit} = g_1, g_2, g_3, \dots, g_n$.
(c)	Set the bits in M as $M_{bit} = M_{b1}, M_{b2}, M_{b3}, \dots, M_{bn}$.
(d)	Set the permuted bit position of m_{bit} as $pos_{bit} = pos_1, pos_2, pos_3, \dots, pos_n$.
	Find M by permuting m_{bit} using following formula.
(e)	i. For $1 \leq i \leq n$, compute $pos_i = (i + Sum) \bmod (n+1)$ ii. For $1 \leq i \leq n$, $g_i = m_{pos_i}$ iii. For $1 \leq i \leq n$, set $M_{bi} = g_i$ iv. Combine all M_{bit} as M .
Step 3	Group M bits into pairs (e.g., total number of M bits = 32). $M_{pair} = M_1, M_2, M_3, \dots, M_{16}$.
Step 4	Group four MSBs in every pixel of C into two pairs.
(a)	Two MSBs are grouped in a pair. $C_{L,pixel} = C_{L,1}, C_{L,2}, C_{L,3}, \dots, C_{L,16}$.
(b)	The remaining two bits are grouped in a pair. $C_{R,pixel} = C_{R,1}, C_{R,2}, C_{R,3}, \dots, C_{R,16}$.
Step 5	Set the two LSBs in every pixel as $B_{L,pixel}$ and $B_{R,pixel}$.
(a)	The LSB is assigned as $B_{R,pixel} = B_{R,1}, B_{R,2}, B_{R,3}, \dots, B_{R,16}$.
(b)	The remaining bit is assigned as $B_{L,pixel} = B_{L,1}, B_{L,2}, B_{L,3}, \dots, B_{L,16}$.
Step 6	Embedding process
(a)	Compare M_1 with $C_{L,1}$. If the bits are equal, replace $B_{L,1}$ with 1. Otherwise, replace with 0
(b)	Compare M_2 with $C_{R,1}$. If the bits are equal, replace $B_{R,1}$ with 1. Otherwise, replace with 0 If $[B_{L,1}; B_{R,1}] = [1; 0]$
(c)	i. Replace $[B_{L,2}; B_{R,2}]$ with M_2 . ii. Repeat Step 6 with pixel number 3 of C . Continue with M_3 and M_4 .
	If $[B_{L,1}; B_{R,1}] = [0; 1]$
	i. Replace $[B_{L,2}; B_{R,2}]$ with M_1 . ii. Repeat Step 6 with pixel number 3 of C . Continue with M_3 and M_4 .
	If $[B_{L,1}; B_{R,1}] = [0; 0]$
	i. Repeat Step 6 with pixel number 2 of C . Continue with M_1 and M_2 .
	If $[B_{L,1}; B_{R,1}] = [1; 1]$
	i. Repeat Step 6 with pixel number 2 of C . Continue with M_3 and M_4 .
Step 7	Stop if all M_{pair} have been used.

In the extraction phase, it follows the similar embedding group pairing process of stego-pixels, i.e., Left and Right Pair with Left and Right Bit, etc. The secret data bits are extracted by referring to the indicator bits (i.e., $B_{L,pixel}$ or $B_{R,pixel}$) that were substituted by the two LSBs in each pixel. The extraction cases are as follows.

Case 1: If $B_{L,pixel}$ is '1' and $B_{R,pixel}$ is '0', restore $C_{L,pixel}$ as a first secret pair (M_1) and restore the 2-LSBs of the next pixel as a second secret pair (M_2).

Case 2: If $B_{L,pixel}$ is '0' and $B_{R,pixel}$ is '1', restore $C_{R,pixel}$ as a second secret pair (M_2) and restore the 2-LSBs of the next pixel with a first secret pair (M_1).

Case 3: If both $B_{L,pixel}$ and $B_{R,pixel}$ are '1', then restore the first secret pair (M_1) from $C_{L,pixel}$ and similarly, restore the second secret pair (M_2) from the $C_{R,pixel}$ bits.

Case 4: If both $B_{L,pixel}$ and $B_{R,pixel}$ are '0', it would indicate the unmapped block (which does not contain any mapping).

For example, as shown in Figure 2, the first pixels of stego-image bits are (01 10 00 01). This satisfied the case 2, the stego-pixel's $B_{L,pixel}$ is '0' and $B_{R,pixel}$ is '1', this indicates extraction of $C_{R,1} = 10$ as the second secret pair, i.e., M_2 . Furthermore, extract the 2 LSBs (i.e., 10) from the 2nd pixel of stego-image M_1 . For 3rd stego-pixel, case 4 is satisfied because both $B_{L,pixel}$ and $B_{R,pixel}$ are '0', which means this is a skipped unmatched block. Similarly, this process can be extended for the rest of the pixels to extract the secret data bits from the stego-image. Table 3 describes the complete extraction steps.

Table 3. The proposed extraction steps.

Input	Stego image (S) and stego key (K)
Output	Secret message (m)
Step 1	Extraction process
(a)	If $[B_{L,1}; B_{R,1}] = [1; 0]$ i. Save $C_{L,1}$ as M_1 . ii. Save $[B_{L,2}; B_{R,2}]$ as M_2 . iii. Repeat Step 1 with pixel number 3 of S . Continue with M_3 and M_4 .
	If $[B_{L,1}; B_{R,1}] = [0; 1]$ i. Save $[B_{L,2}; B_{R,2}]$ as M_1 . ii. Save $C_{R,1}$ as M_2 . iii. Repeat Step 1 with pixel number 3 of S . Continue with M_3 and M_4 .
	If $[B_{L,1}; B_{R,1}] = [0; 0]$ i. Repeat Step 1 with pixel number 2 of S . Continue with M_1 and M_2 .
	If $[B_{L,1}; B_{R,1}] = [1; 1]$ i. Save $C_{L,1}$ as M_1 . ii. Save $C_{R,1}$ as M_2 . iii. Repeat Step 1 with pixel number 3 of S . Continue with M_3 and M_4 .
Step 2	Continue step 1 until all M_{pair} have been extracted.
Step 3	Combine all extracted M_{pair} as M .
Step 4	Convert K into Sum value.
(a)	Find values of K components $P_1, P_2, P_3, \dots, P_k$ where k is the length of K using ASCII table. (e.g., $K = \text{Alif2s}$) i. Refer ASCII table to search values of K components. $P_1 = A = 65, P_2 = l = 108, P_3 = i = 105, P_4 = f = 102, P_5 = 2 = 50, P_6 = s = 115$
	Compute Sum value using the following formula (b) i. $Sum = \sum_{i=1}^k (P_i \times i) \text{ mod } 256$ $= [(65 \times 1) + (108 \times 2) + (105 \times 3) + (102 \times 4) + (50 \times 5) + (115 \times 6)] \text{ mod } 256$
Step 5	Compute m .
(a)	Set the bits in M as $M_{bit} = M_{b1}, M_{b2}, M_{b3}, \dots, M_{bn}$ where n is the length of M .
(b)	Set the permuted bits in M as $f_{bit} = f_1, f_2, f_3, \dots, f_n$
(c)	Set the bits in m as $m_{bit} = m_1, m_2, m_3, \dots, m_n$
(d)	Set the permuted bit position of M_{bit} as $pos_{bit} = pos_1, pos_2, pos_3, \dots, pos_n$.
(e)	Find m by permuting M_{bit} using following formula. i. For $1 \leq i \leq n$, compute $pos_i = (i - Sum) \text{ mod } (n + 1)$ ii. For $1 \leq i \leq n$, $f_i = M_{bpos_i}$ iii. For $1 \leq i \leq n$, set $m_i = f_i$ iv. Combine all m_{bit} as m .

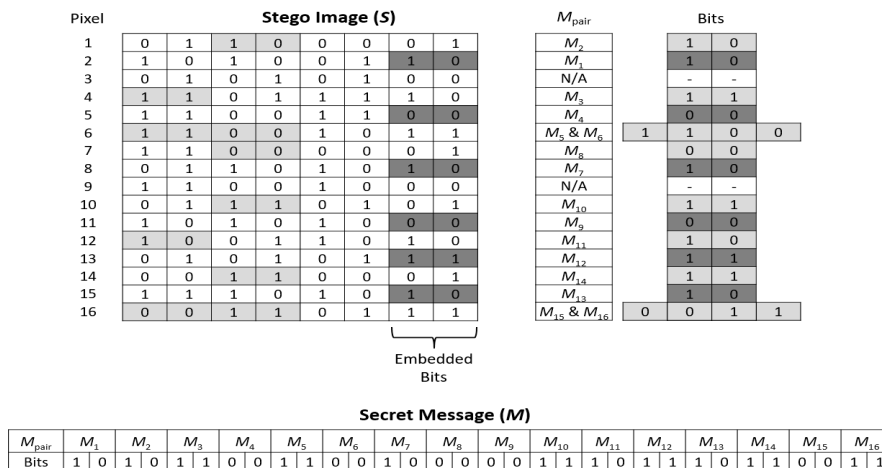


Figure 2. Stego image and a secret message.

4. Experimental Results

The proposed method was implemented using C++ on an Intel(R) Core(TM) i7 2.70 GHz CPU with 8 GB RAM on Windows 10. For performance evaluation, we employed a well-known standard image dataset by Signal and Image Processing Institute from University of Southern California namely USC-SIPI [24] with grayscale (8 bits per pixel) based images, i.e., Aerial, Airplane, APC, Boat, Car, Couple, Gray21, Motion, Ruler, Stream, Tank, and Truck as shown in Figure 3. For secret data embedding, a random function from Microsoft Visual Studio 2008 (Version 9.0.21022.8 RTM, Microsoft, Washington, United States) was used to generate the bits stream for secret data.

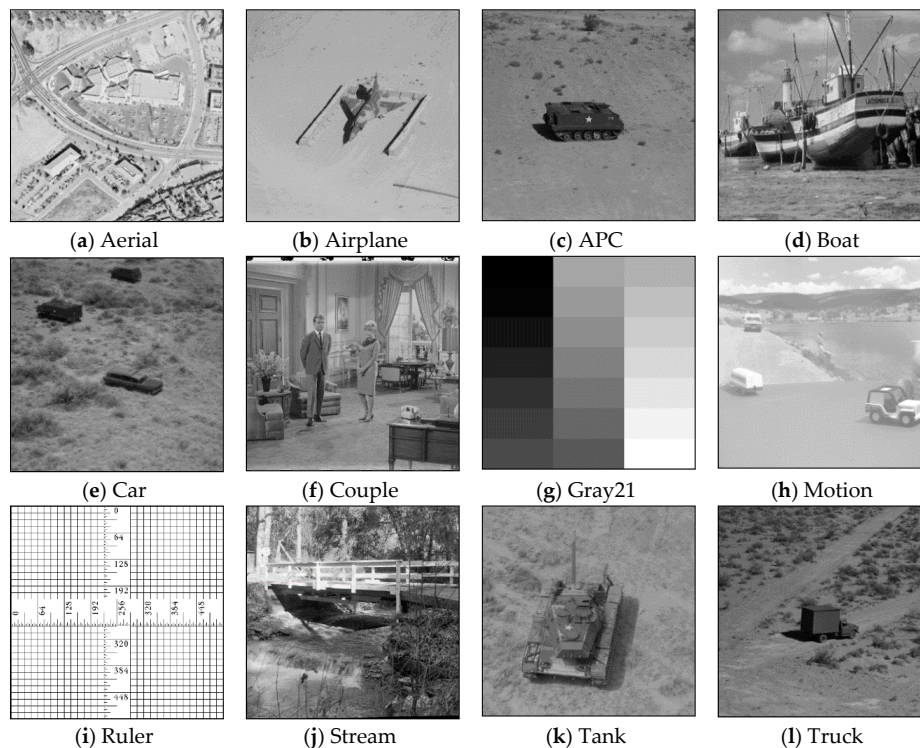


Figure 3. Cover images from USC-SIPI [24] dataset for experiments.

For a comprehensive performance analysis, we categorized the evaluations into the following sub-sections. Firstly, Section 4.1 consists of the maximum obtained embedding capacity with various visual quality evaluation parameters, i.e., PSNR, bpp, and modified bits per pixels. Secondly,

in Section 4.2, we evaluated the performance of the visual quality of the proposed method on various embedding rates as compared to existing techniques. Similarly, for security aspects of the proposed method against existing methods, Sections 4.3 and 4.4 discussed the well-known detection attacks, i.e., histogram analysis and RS analysis. However, in all of the above evaluation categories, we employed the similar evaluation procedures (i.e., image dataset and PSNR) on proposed and existing Yang et al., Liao et al. and Khodaei et al. [7,11,15] methods.

4.1. Embedding Capacity and Visual Quality Analysis

In this section, the performance of embedding capacity and visual quality is compared in three ways. First, the performance of the proposed method with Yang et al. [7], Liao et al. [11], and Khodaei et al. [15] are shown in Table 4. Secondly, the embedding capacity vs visual quality performance over the complete USC-SIPI dataset is shown in Table 5. Finally, the performance of the proposed method is compared with the most recent LSB-based methods (in Figure 4).

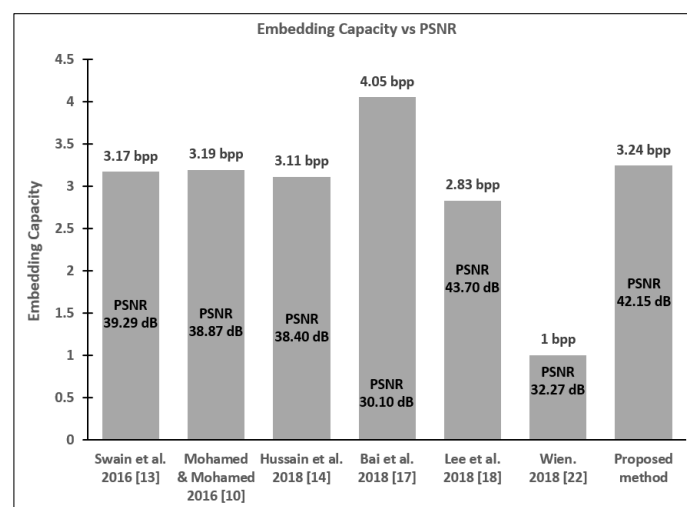


Figure 4. Performance comparison of proposed and recent data embedding methods.

The embedding capacity is calculated as the maximum number of secret data bits that can be embedded into the cover image. Bits per pixel are determined using Equation (1), where W and H are the width and height of the cover image respectively. PSNR is adopted to evaluate the visual quality or evaluate the distortion of the cover images after embedding secret data. The PSNR is computed as shown in Equation (2). A high value of PSNR illustrates a low degree of image distortion. Conversely, a small PSNR value indicates a larger distortion between the cover image and the stego image.

$$bpp = \frac{\text{Embedding capacity}}{W \times H} \tag{1}$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{2}$$

Mean square error (MSE) is the measure of an average of the squares of the difference between the pixel intensities of the stego and the cover images. That is defined in Equation (3), where p'_i and p_i are the pixel values of the stego and cover images, respectively. The smaller the MSE value, the better the image quality; the higher the MSE value, the greater the distortion. MSE is inversely proportional to the PSNR, which demonstrates that the higher the value of PSNR, the lower the MSE value produced. So, a higher PSNR would be considered as good or an indication of low image distortion.

$$MSE = \sum_{i=1}^{W \times H} \frac{(p'_i - p_i)^2}{W \times H} \tag{3}$$

Table 4. Comparison of capacity (bpp), image quality (PSNR), and modified bits with existing methods using USC-SIPI [24] dataset images.

Method	Yang et al. 2009 [7]				Liao et al. 2011 [11]				Khodaei et al. 2016 [15]				Proposed Method			
	Capacity (bits)	Bits/Pixel (bpp)	PSNR (dB)	Modified Bits/Pixel	Capacity (bits)	Bits/Pixel (bpp)	PSNR (dB)	Modified Bits/Pixel	Capacity (bits)	Bits/Pixel (bpp)	PSNR (dB)	Modified Bits/Pixel	Capacity (bits)	Bits/Pixel (bpp)	PSNR (dB)	Modified Bits/Pixel
Aerial	758,163	2.89	39.94	2.27	801,143	3.06	39.25	2.35	812,592	3.10	38.42	2.57	825,175	3.15	42.14	1.86
Airplane	734,642	2.80	39.81	2.36	825,662	3.15	39.61	2.19	826,164	3.15	37.53	2.65	856,628	3.27	41.85	1.89
APC	769,075	2.93	40.14	2.32	803,537	3.07	39.26	2.22	805,530	3.07	38.74	2.32	867,990	3.31	42.35	1.91
Boat	778,648	2.97	39.58	2.09	842,486	3.21	39.56	2.31	822,492	3.14	38.51	2.79	833,482	3.18	41.75	1.94
Car	797,597	3.04	39.59	2.12	816,695	3.12	38.91	2.46	810,783	3.09	38.95	2.46	855,914	3.27	42.47	1.79
Couple	785,760	3.00	40.61	2.25	806,096	3.08	39.71	2.23	822,729	3.14	37.95	2.71	831,462	3.17	42.66	1.80
Gray21	767,179	2.93	39.87	2.43	820,781	3.13	39.41	2.51	841,186	3.21	38.56	2.46	849,714	3.24	41.74	1.77
Motion	725,028	2.77	39.68	2.21	808,820	3.09	39.06	2.08	812,757	3.10	37.91	2.51	862,853	3.29	41.59	1.95
Ruler	768,931	2.93	39.66	2.20	813,329	3.10	38.61	2.22	830,549	3.17	38.46	2.74	872,136	3.33	42.32	1.69
Stream	745,624	2.84	40.23	2.24	827,024	3.15	39.82	2.34	813,358	3.10	38.67	2.88	848,472	3.24	42.49	1.93
Tank	794,153	3.03	39.78	2.08	804,163	3.07	39.59	2.47	836,372	3.19	38.41	2.73	864,835	3.30	42.15	1.87
Truck	756,642	2.89	39.75	2.41	818,452	3.12	38.73	2.29	814,381	3.11	38.88	2.69	823,499	3.14	42.31	1.98
Average	765,120	2.92	39.89	2.25	815,682	3.11	39.29	2.31	820,741	3.13	38.42	2.63	849,347	3.24	42.15	1.87

Table 5. Performance comparison using complete USC-SIPI [24] dataset of the proposed and previous methods.

Method	Volume 1			Volume 2			Volume 3			Volume 4		
	Resolution	Average Capacity	Average PSNR	Resolution	Average Capacity	Average PSNR	Resolution	Average Capacity	Average PSNR	Resolution	Average Capacity	Average PSNR
Yang et al. [7]												
Liao et al. [11]							256 × 256	192,554	40.60	256 × 256	192,293	39.88
Khodaei et al. [15]	N/A	N/A	N/A	N/A	N/A	N/A	(14 images)	204,952	38.56	(59 images)	204,527	39.28
Proposed Method								205,545	38.59		206,082	38.52
								213,044	41.86		213,432	42.11
Yang et al. [7]		765,075	39.80		764,130	40.12		766,814	39.88		760,609	39.79
Liao et al. [11]	512 × 512	815,366	39.23	512 × 512	815,284	39.37	512 × 512	815,863	39.37	512 × 512	814,808	39.29
Khodaei et al. [15]	(130 images)	820,537	38.50	(12 images)	820,752	38.31	(22 images)	821,206	38.39	(10 images)	819,487	38.41
Proposed Method		848,971	42.22		848,324	42.02		848,925	42.25		850,975	42.04
Yang et al. [7]		3,068,218	39.87		3,067,924	39.88		3,060,506	38.84			
Liao et al. [11]	1024 × 1024	3,266,064	39.37	1024 × 1024	3,266,786	39.22	1024 × 1024	3,260,171	39.31	N/A	N/A	N/A
Khodaei et al. [15]	(25 images)	3,285,862	38.28	(25 images)	3,288,377	38.53	(3 images)	3,282,716	38.27			
Proposed Method		3,394,497	42.19		3,386,517	42.31		3,407,548	42.23			
Yang et al. [7]					14,803,939	39.73						
Liao et al. [11]				2250 × 2250	15,703,642	39.12						
Khodaei et al. [15]	N/A	N/A	N/A	(1 image)	15,913,361	38.34	N/A	N/A	N/A	N/A	N/A	N/A
Proposed Method					16,403,103	42.09						

From the experiments as shown in Table 4, the average embedding capacity obtained from the proposed method is higher than in Yang et al. [7], Liao et al. [11], and Khodaei et al. [15]. The proposed method is able to embed a payload from 823,499 bits (or 805 KB) to 872,136 bits (or 852 KB). This achieved the embedding bits per pixel ranging from 3.14 to 3.33 bpp with PSNR values from 41.59 to 42.66 dB. However, the Khodaei et al. and Liao et al. methods had the lower average payload of 820,741 bits (or 802 KB) and 815,682 bits (or 797 KB), respectively. The Yang et al. method resulted in an average payload of approximately 765,120 bits (or 748 KB). From Table 4, it was proved that the proposed method obtained the highest average embedding capacity of 849,347 bits (or 830 KB) and average 3.24 bpp. We can say the proposed method is able to embed 28,606, 33,664, and 84,227 more bits than the Khodaei et al., Liao et al., and Yang et al. methods, respectively.

As seen in Table 4, the proposed method can also perform better to produce less distorted stego-images than existing methods in terms of higher PSNR and a lower number of modified bits per pixel statistics. The proposed method achieved the average modified bits per pixel of 1.87 bits (ranging from 1.69 to 1.98 bits) while existing methods had higher modified bits per pixel, i.e., Yang et al. with 2.25, Liao et al. with 2.31, and Khodaei et al. with 2.63. Generally, the lower modification bits per pixel value produces a less distorted stego-image which indirectly helps to enhance the PSNR value. Hence, it is proved, and as shown in Table 4 statistics, that the proposed method yielded higher average PSNR of 42.15 dB against the compared methods. The proposed method has a PSNR value 2.27 dB higher than achieved by Yang et al., 2.86 dB higher than Liao et al., and 3.74 dB higher than the Khodaei et al. method.

Finally, the performance of embedding capacity and PSNR are compared for the complete 301 image samples in the USC-SIPI dataset, categorized in four volumes, i.e., Textures, Aerials, Miscellaneous, and Sequences as shown in Table 5. The experimental results show that the proposed method's average embedding capacity is higher than that of the existing methods as highlighted in bold font. Similarly, the PSNR of the proposed method is the highest among others in all tested image resolutions. The overall results demonstrate that the proposed method has better embedding capacity while maintaining higher visual quality and requires a lower number of bits per pixel modification as compared to existing methods. The highest capacity achieved in a proposed method is based on its internal data bits mapping and indicator strategy. It can be seen that the proposed method indirectly embeds up to 4 secret bits per pixel while limiting the modification of bits per pixel up to 2.

The statistics in Figure 4 show the comparisons of proposed and recent data embedding approaches. The embedding capacity of the proposed method is far better than that of all the other compared methods except Bai et al. [17], where the PSNR value of Bai et al. [20] is too low (−13 dB) as compared to the proposed method. Similarly, the proposed method achieved the higher PSNR value while retaining the lower number of (only 2) LSBs used. However, Wien [22] utilized only 1 LSB, but the embedding capacity and visual quality statistics are quite low as compared to the proposed method. In conclusion, we can say the proposed method outperformed all the compared methods in terms of embedding capacity, visual quality and maintaining the minimum no. of LSBs employed for the embedding process.

4.2. Peak Signal to Noise Ratio (PSNR) Analysis on Various Embedding Levels

In this section, we evaluated the visual quality of the proposed method at different embedding rates to verify whether the proposed method has consistent performance of visual quality at various embedding levels. The graphical representation of PSNR values of images for different sizes of secret bits are shown in Figure 5a–f. The methods developed by Yang et al. [7], Liao et al. [11], and Khodaei et al. [15] are compared with the proposed method to observe the differences in performance. For the Gray21 image, the capacity vs PSNR graph is as shown in Figure 5a. Once the embedding capacity is high, i.e., 500,000 bits, the proposed method obtained the highest PSNR of 43.17 dB, while the compared methods achieved lower PSNR, i.e., 41.70 dB (Yang et al.), 40.20 dB (Liao et al.), and 39.30 dB (Khodaei et al.). Similarly, if the embedding bits are 250,000, again the proposed method

achieved a higher PSNR (44.40 dB) while the others obtained 42.50, 41.90, and 40.23 dB, respectively. Furthermore, if the embedding capacity is around 50,000 bits, the proposed method has far better PSNR up to 48.65 dB as compared to existing methods.

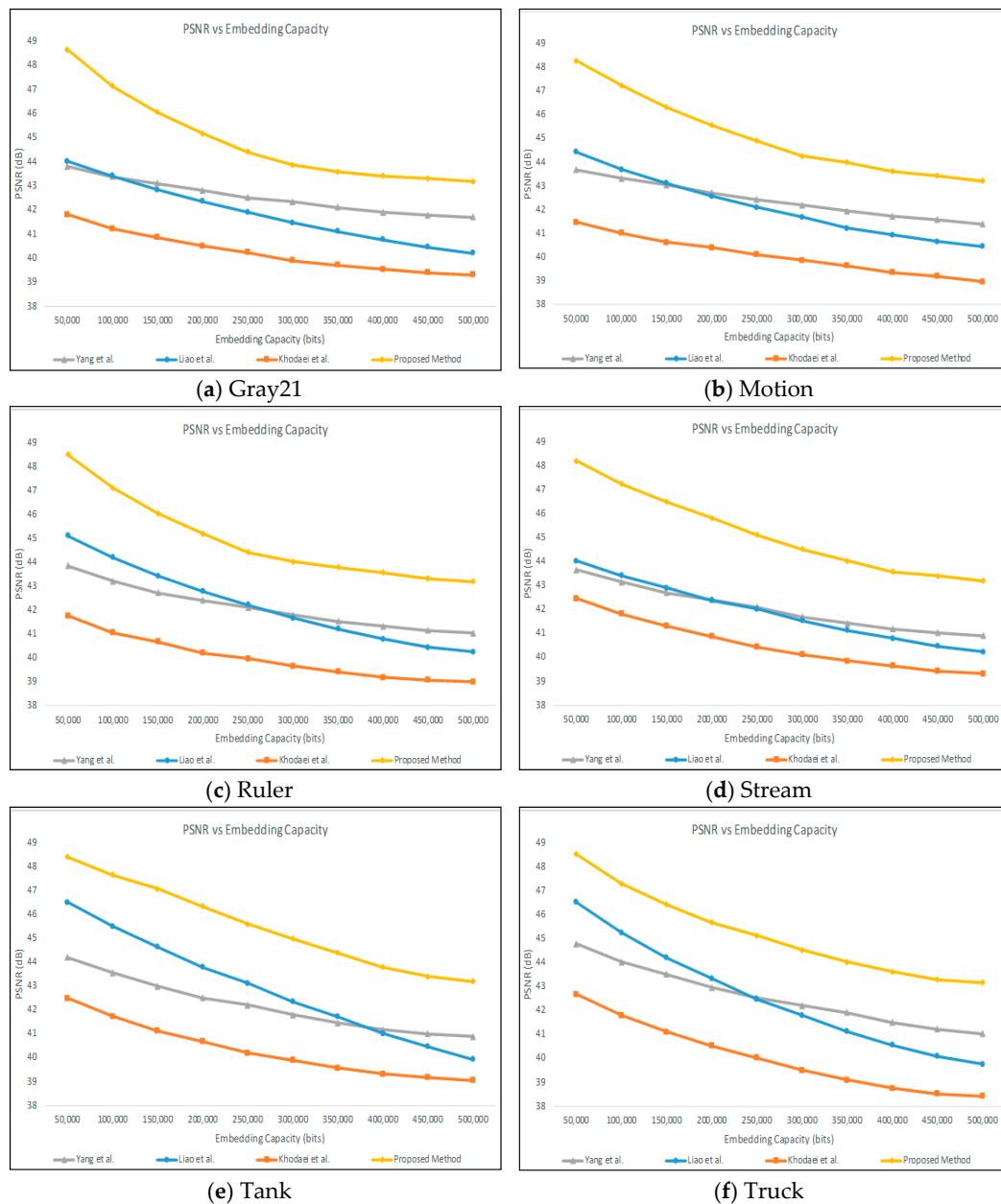


Figure 5. Comparison graphs for PSNR versus various embedding capacity from the USC-SIPI [24] dataset. The figures are (a) Gray21, (b) Motion, (c) Ruler, (d) Stream, (e) Tank, and (f) Truck.

From Figure 5b of the Motion image, the proposed method obtained the maximum PSNR 43.21 dB at 500,000 bits embedding. However, the compared methods achieved low PSNR as compared to the proposed method, i.e., [7] 41.39 dB, [11] 40.45 dB, and [15] 38.96 dB at the same 500,000 bits. Similarly, the proposed method achieved the maximum PSNR (48.50 dB) at 50,000 bits for Ruler image in Figure 5c and 48.20 dB for Stream image in Figure 5d.

In the same way, from Figure 5e, the proposed method obtained the highest PSNR 43.19 dB at 500,000 bits while the compared methods recorded low PSNR, i.e., 40.89 dB [7], 39.93 dB [11], and 39.05 dB [15] for the Tank image. In Figure 5f, the highest PSNR achieved by previous methods is 44.78 dB for Yang et al., 46.52 dB for Liao et al., and 42.67 dB for Khodaei et al., while the proposed

method obtained the highest PSNR of 48.53 dB for the Truck image. On average, PSNR results at the lowest embedding capacity are obtained by the methods: 43.99 dB [7], 45.10 dB [11], 42.10 dB [15], and the proposed method with 48.43 dB. At 500,000 bits, the proposed method recorded the highest PSNR with 43.19 dB followed by Yang et al. with 41.16 dB, Liao et al. with 40.14 dB, and Khodaei et al. with 39.01 dB. Therefore, the proposed method yields the highest PSNR at different embedding capacities against Yang et al., Liao et al., and Khodaei et al.

Consequently, from experiments, the proposed method outperformed the compared methods; it was proven that the proposed method retains high visual imperceptibility irrespective of embedding rates. The reason behind achieving the better results depends on the maximum number of bits mapping strategy while reducing the number of bits modification into the pixel.

4.3. Histogram Analysis

For measuring the robustness against common statistical attacks, the histogram analysis [25] between the cover and stego images of the proposed method are presented. The histograms are generated from the embedding process using Aerial, Airplane, Boat and Couple images as presented in Figure 6. Generally, by embedding the secret bits in the cover image, the frequency of pixels values are changed and can become visible in the histogram. Figure 6c,g,k,o shows the frequency histogram of the cover images. Meanwhile, Figure 6d,h,l,p shows the frequency histogram of the stego images. As observed from histograms, the disparities between the constructed histograms are comparatively less for all tested images. The results indicate that distortions produced from embedding process are unnoticeable to human visual perception. The analysis results show that there is no significant difference found in histograms of the cover and stego images. The embedded information is not easily detected by the difference histogram analysis. Thus, histogram-based steganalysis attacks are not possible for the proposed method. The security of hidden secret is well protected while keeping the advantage of low visual distortion introduced by the proposed method.

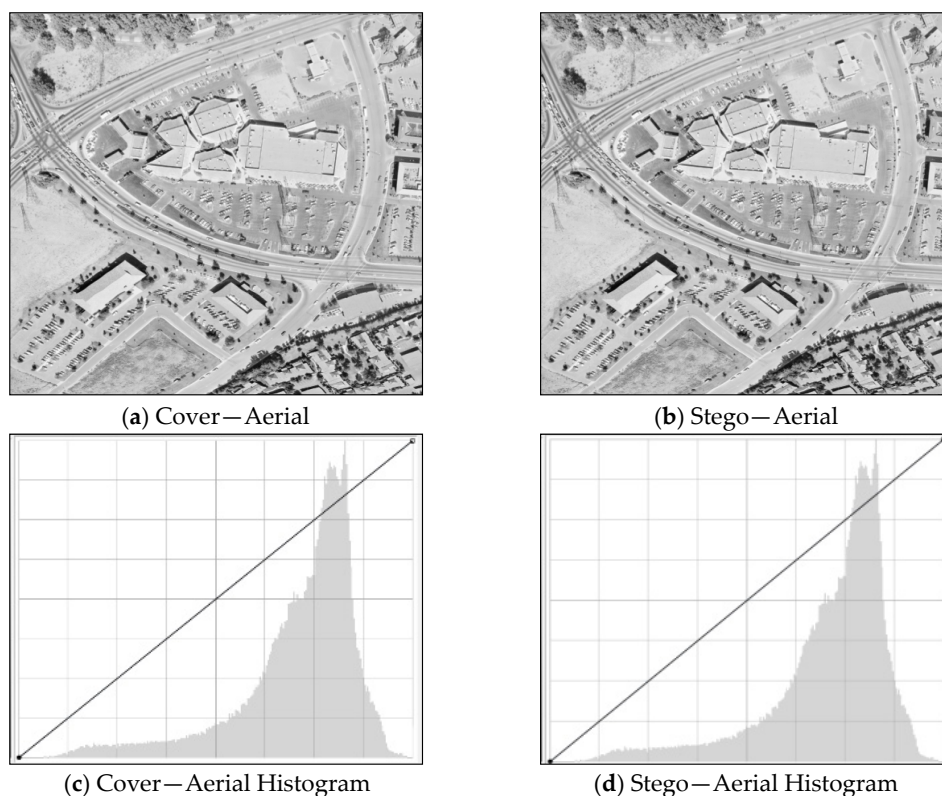


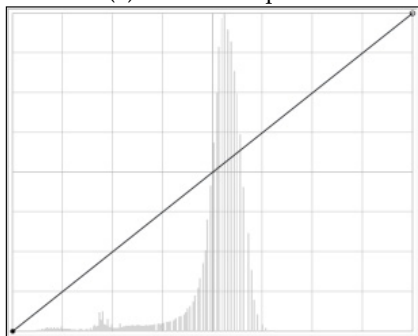
Figure 6. Cont.



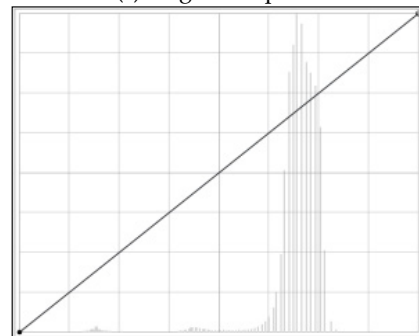
(e) Cover—Airplane



(f) Stego—Airplane



(g) Cover—Airplane Histogram



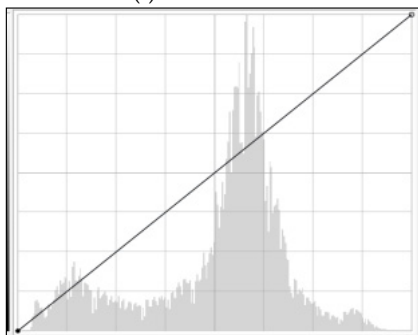
(h) Stego—Airplane Histogram



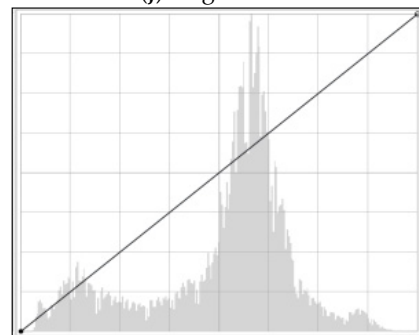
(i) Cover—Boat



(j) Stego—Boat



(k) Cover—Boat Histogram



(l) Stego—Boat Histogram

Figure 6. Cont.

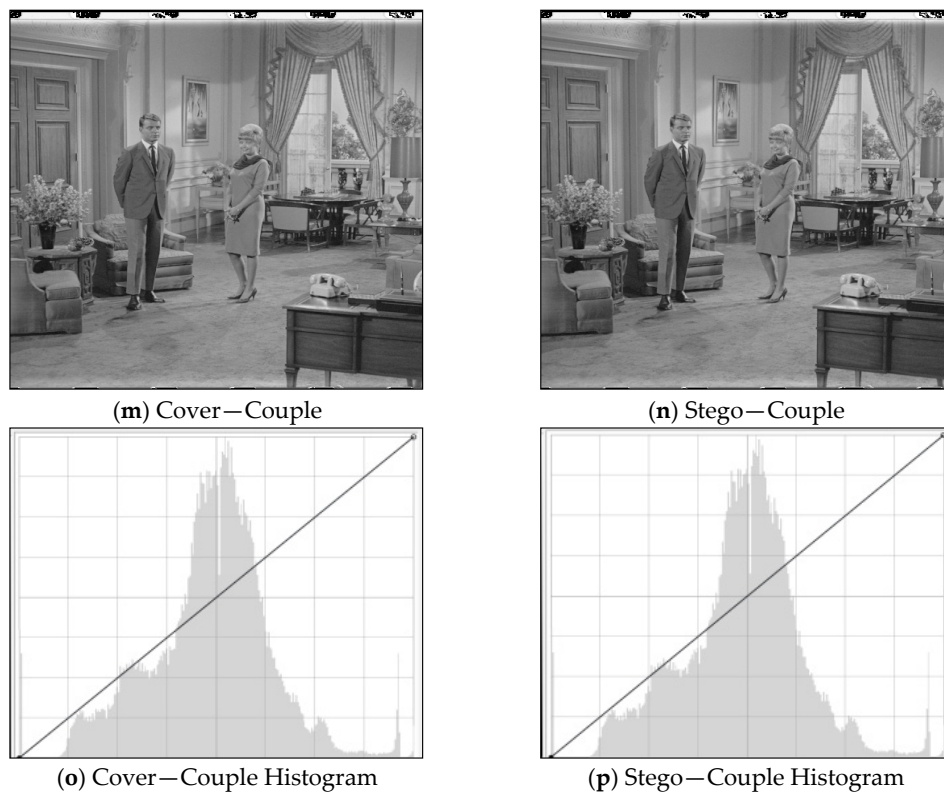


Figure 6. Comparison of histograms for images from USC-SIPI [24] dataset. The figures are (a) Aerial, (e) Airplane, (i) Boat, and (m) Couple.

4.4. Security against RS Steganalysis

The security of the proposed method against the statistical RS (Regular and Singular groups) steganalysis method [6] is depicted in Figure 7. The RS steganalysis is based on discrimination function (DF) with M and $-M$ as flipping masks. Furthermore, $R_M, R_{-M}, S_M,$ and S_{-M} parameters are used to find the magnitude of the pixel block using DF function. If the parameters satisfy $R_M \approx R_{-M} > S_M \approx S_{-M}$, then no hidden data are in the respective image. When an image has hidden data in its least significant bits, R_{-M} and S_M increases, whereas R_M and S_{-M} decreases and is exposed by RS analysis.

In the RS analysis, we used $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$ as M and $-M$, respectively. The x -axis on the RS analysis graph represents the percentage of hidden data in the stego image, and the y -axis denotes the relative percentages of regular (R_M, R_{-M}) and singular (S_M, S_{-M}) groups with the application of the above masks ($M, -M$).

The RS detection results can be seen in Figure 7 with the proposed method for the Aerial, Airplane, APC, Boat, Car, and Couple images. In Figure 7a, the differences between singular and regular parameters remain constant and close to each other even when the hiding capacity increases. As shown in Figure 7d, the RS detection differences values of the proposed method are very close to each other between R_{-M} and R_M , and between S_{-M} and S_M curves. For Figure 7b,c,e,f images, the differences between regular and singular groups are maintained close to each other via the proposed method. These results indicate that the proposed method is secured against RS steganalysis detection attacks.

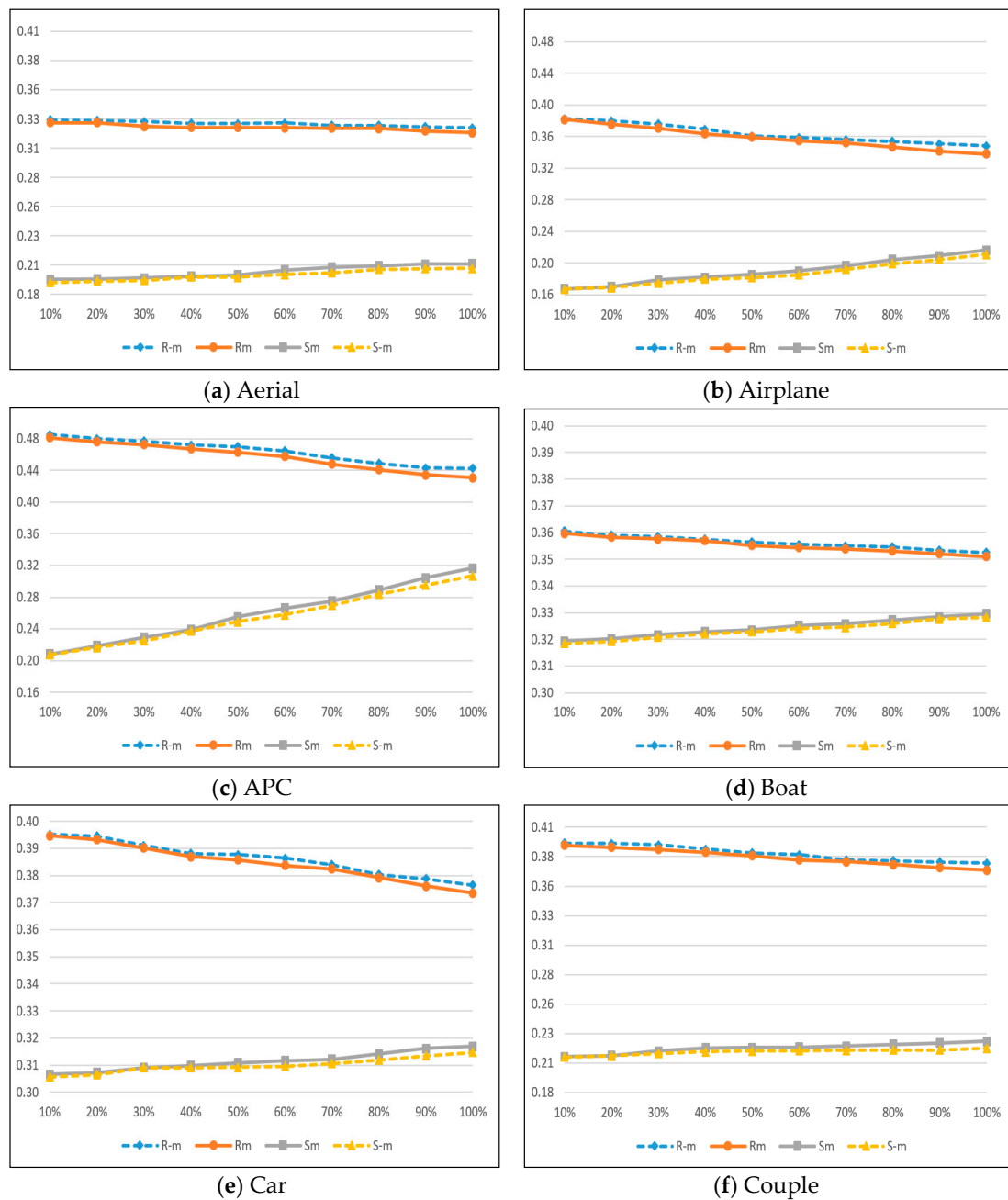


Figure 7. Comparison graphs for RS analysis from USC-SIPI [24] dataset. The figures are (a) Aerial, (b) Airplane, (c) APC, (d) Boat, (e) Car, and (f) Couple.

The maximum differences of detection parameters, i.e., $R_M - R_{-M}$, and $S_M - S_{-M}$ for the compared methods [7,11,15] are shown in Table 6. From the statistical analysis, it is evident that the proposed method has the smallest average difference in both regular and singular groups, i.e., 0.1853% and 0.1451%, respectively. However, Yang et al.'s method has a slightly higher average difference for both regular singular groups (0.1947%, 0.1542%). Similarly, Khodaei et al. and Liao et al. methods have higher average difference results for both regular groups (0.2160% and 0.2358%) and singular groups (0.1697% and 0.1934%). In short, the proposed method has the smallest differences against all the compared methods, thus indicating the fewer detection artifacts in stego-images. Moreover, we can say that the proposed method has more capability to resist the RS steganalysis detection attacks.

Table 6. Comparison of maximum differences in RS-steganalysis detection parameters with previous methods using USC-SIPI [24] dataset images.

Method	Yang et al. [7]			Liao et al. [11]			Khodaei et al. [15]			Proposed Method		
Images	Capacity (bits)	$ R_M - R_{-M} $	$ S_M - S_{-M} $	Capacity (bits)	$ R_M - R_{-M} $	$ S_M - S_{-M} $	Capacity (bits)	$ R_M - R_{-M} $	$ S_M - S_{-M} $	Capacity (bits)	$ R_M - R_{-M} $	$ S_M - S_{-M} $
Aerial	758,163	0.1661	0.1490	801,143	0.2161	0.1704	812,592	0.1868	0.1540	825,175	0.1615	0.1373
Airplane	734,642	0.2272	0.1779	825,662	0.2618	0.2156	826,164	0.2535	0.1883	856,628	0.2141	0.1632
APC	769,075	0.1958	0.1491	803,537	0.2230	0.1812	805,530	0.2017	0.1637	867,990	0.1879	0.1431
Boat	778,648	0.1549	0.1248	842,486	0.1996	0.1541	822,492	0.1796	0.1356	833,482	0.1522	0.1160
Car	797,597	0.2394	0.1824	816,695	0.2713	0.2320	810,783	0.2553	0.1985	855,914	0.2183	0.1698
Couple	785,760	0.2483	0.1938	806,096	0.2924	0.2645	822,729	0.2728	0.2301	831,462	0.2320	0.1801
Gray21	767,179	0.2142	0.1661	820,781	0.2585	0.2035	841,186	0.2440	0.1831	849,714	0.2027	0.1527
Motion	725,028	0.1401	0.1135	808,820	0.1872	0.1415	812,757	0.1609	0.1229	862,853	0.1354	0.1071
Ruler	768,931	0.1626	0.1377	813,329	0.2097	0.1687	830,549	0.1819	0.1426	872,136	0.1566	0.1349
Stream	745,624	0.2437	0.1850	827,024	0.2859	0.2453	813,358	0.2682	0.2148	848,472	0.2238	0.1755
Tank	794,153	0.1425	0.1192	804,163	0.1942	0.1463	836,372	0.1622	0.1272	864,835	0.1401	0.1114
Truck	756,642	0.2010	0.1513	818,452	0.2301	0.1974	814,381	0.2254	0.1759	823,499	0.1992	0.1505
Average	765,120	0.1947	0.1542	815,682	0.2358	0.1934	820,741	0.2160	0.1697	849,347	0.1853	0.1451

5. Conclusions

In this paper, we have presented a novel data hiding approach based on LSB substitution using a mapping bits strategy. In the proposed method, first cover pixel bits and secret data bits were logically grouped into the number of pairs. Next, in the embedding process, these logical groups of four secret data bits were mapped with the (4-MSBs of) cover pixel bits. To maintain the mapping status between the cover and secret data, the method employed the (2-bit) LSB substitution. The contributions of the proposed method are high embedding capacity, less distortion quality, and robustness. First, the proposed method achieved high embedding capacity (up to 3.24 bpp) due to the maximum (4 bits) mapping between the cover and secret data bits. Second, the proposed method maintained a high (42.15 dB) PSNR value due to limiting the number of bits modification per pixel up to two bits. Third, the proposed method was able to resist the RS and histogram steganalysis attacks due to low visual distortion. In the future, we plan to take into consideration an open challenge to achieve adaptable mapping between the cover and secret data bits.

Author Contributions: All authors discussed the contents of the manuscript and contributed to its preparation. A.A.Z. and M.H. designed and implemented the proposed scheme; M.Y.I.I. and A.W.A.W. contributed input for this research and reviewed the manuscript; N.A.A. provided the experimental setup. K.H.J. designed and conducted the literature review.

Funding: The APC was funded by CyberSecurity Malaysia. The work was supported by the Special Assistance Research Grant of University of Malaya (BKS107-2017) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1A09081842).

Acknowledgments: We thank the anonymous reviewers for their valuable suggestions that improved the clarity of this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussain, M.; Wahab, A.W.A.; Idris, Y.I.B.; Ho, A.T.; Jung, K.H. Image steganography in spatial domain: A survey. *Signal Proc. Image Commun.* **2018**, *65*, 46–66. [[CrossRef](#)]
2. Yesilyurt, M.; Yalman, Y. New approach for cloud computing security: Using data hiding methods. *Sadhana* **2016**, *41*, 1289–1298.
3. Liu, Y.N.; Zhong, Q.; Xie, M.; Chen, Z.B. A novel multiple-level secret image sharing scheme. *Multimed. Tools Appl.* **2018**, *77*, 6017–6031. [[CrossRef](#)]
4. Xiang, T.; Hu, J.; Sun, J. Outsourcing chaotic selective image encryption to the cloud with steganography. *Digit. Signal Proc.* **2015**, *43*, 28–37. [[CrossRef](#)]
5. Tondwalkar, A.; Jani, P.V. Secure localisation of wireless devices with application to sensor networks using steganography. *Proc. Comput. Sci.* **2016**, *78*, 610–616. [[CrossRef](#)]
6. Fridrich, J.; Goljan, M.; Du, R. Reliable detection of LSB steganography in color and grayscale images, New Challenges. In Proceedings of the 2001 Workshop on Multimedia and Security, Ottawa, ON, Canada, 5 October 2001; pp. 27–30.
7. Yang, H.; Sun, X.; Sun, G. A high-capacity image data hiding scheme using adaptive LSB substitution. *Radio Eng.* **2009**, *18*, 509–516.
8. Tseng, H.W.; Leng, H.S. High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion. *IET Image Process.* **2014**, *8*, 647–654. [[CrossRef](#)]
9. Jung, K.H.; Yoo, K.Y. Steganographic method based on interpolation and LSB substitution of digital images. *Multimed. Tools Appl.* **2015**, *74*, 2143–2155. [[CrossRef](#)]
10. Mohamed, M.H.; Mohamed, L.M. High capacity image steganography technique based on LSB substitution method. *Appl. Math. Inf. Sci.* **2016**, *10*, 259–266. [[CrossRef](#)]
11. Liao, X.; Wen, Q.Y.; Zhang, J. A steganographic method for digital images with four-pixel differencing and modified LSB substitution. *J. Vis. Commun. Image Represent.* **2011**, *22*, 1–8. [[CrossRef](#)]
12. Wu, D.C.; Tsai, W.H. A steganographic method for images by pixel value differencing. *Pattern Recognit. Lett.* **2003**, *24*, 1613–1626. [[CrossRef](#)]

13. Swain, G. A steganographic method combining LSB substitution and PVD in a block. *Proc. Comput. Sci.* **2016**, *85*, 39–44. [[CrossRef](#)]
14. Hussain, M.; Wahab, A.W.A.; Javed, N.; Jung, K.H. Recursive information hiding scheme through LSB, PVD shift, and MPE. *IETE Technol. Rev.* **2018**, *35*, 53–63. [[CrossRef](#)]
15. Khodaei, M.; Bigham, B.S.; Faez, K. Adaptive data hiding, using pixel-value-differencing and LSB substitution. *Cybern. Syst.* **2016**, *47*, 617–628. [[CrossRef](#)]
16. Akhtar, N. An LSB substitution with bit inversion steganography method. In Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics, New Delhi, India, 25–27 February 2016; pp. 515–521.
17. Bai, J.; Chang, C.C.; Nguyen, T.S.; Zhu, C.; Liu, Y. A high payload steganographic algorithm based on edge detection. *Displays* **2017**, *46*, 42–51. [[CrossRef](#)]
18. Lee, H. Data hiding in spatial color images on smartphones by adaptive RGB LSB replacement. *IEICE Trans. Informat. Syst.* **2018**, *101*, 2163–2167. [[CrossRef](#)]
19. Kuo, W.C.; Kuo, S.H.; Wu, L.C. Multi-bit data hiding scheme for compressing secret messages. *Appl. Sci.* **2015**, *5*, 1033–1049. [[CrossRef](#)]
20. Sethi, P.; Kapoor, V. A proposed novel architecture for information hiding in image steganography by using genetic algorithm and cryptography. *Proc. Comput. Sci.* **2016**, *87*, 61–66. [[CrossRef](#)]
21. Muhammad, K.; Ahmad, J.; Rehman, N.U.; Jan, Z.; Sajjad, M. CISSKA-LSB: Color image steganography using stego key-directed adaptive LSB substitution method. *Multimed. Tools Appl.* **2016**, *76*, 8597–8626. [[CrossRef](#)]
22. Hong, W. Efficient data hiding based on block truncation coding using pixel pair matching technique. *Symmetry* **2018**, *10*, 36. [[CrossRef](#)]
23. Setiadi, D.R.I.M.; Jumanto, J. An enhanced LSB-image steganography using the hybrid Canny-Sobel edge detection. *Cybern. Informat. Technol.* **2018**, *18*, 74–88. [[CrossRef](#)]
24. The USC-SIPI Image Database. Available online: <http://sipi.usc.edu/database/> (accessed on 2 April 2018).
25. Swain, G. Steganography in digital images using maximum difference of neighboring pixel values. *Int. J. Secur. Appl.* **2013**, *7*, 284–294. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).