Improving the Capability of Reduced Difference Expansion based Digital Image Data Hiding

Mohammed Hatem Ali Al-Hooti, *Member, IAENG,* Tohari Ahmad, *Member, IAENG,* Supeno Djanali

***Abstract*—Difference Expansion along with its variants such as Reduced Difference Expansion (RDE) is a popular approach in the field of data hiding. Nevertheless, current works have not yet achieved the optimal performance, in terms of the capacity of the secret data and the quality of the stego media. In this paper, we propose a two layer-based algorithm in order to specifically improve the capability of RDE method regarding to the capacity and the quality of the stego image. This is done in the following manners. The first is enhancing the difference transformation scheme by taking the difference between the two closest pixels; and correspondingly, a new pixel construction scheme is also adjusted. The second is optimizing the size of the pixel blocks. The third stage is expanding the algorithm for both grayscale and RGB color images. Fourth, the proposed method improves the use of non-changeable blocks to hide the secret data. Finally, every block is processed once again to embed more secret bits. It occurs by replacing the Least Significant Bit (LSB) of each pixel. The experimental results show that this method has made a significant improvement in terms of the capacity since around 1.85 bits are carried by each pixel where the quality is slightly higher than that of the presented by previous research. Overall, by using the same block size, this proposed method delivers higher performance for both capacity and quality; furthermore, by using a bigger block size, this method is able to obtain a significant increase of the capacity with only slightly decrease of the quality.**

***Index Terms*—Data Hiding, Steganography, Information Security, Network Security, Reduced Difference Expansion.**

I. Introduction

THE information technology has grown significantly for some decades which brings many advantages to the public. For example, people can transfer files or messages to each other all over the world within a few seconds by using smart-phones [1]. This message can be an image, text, video, or audio [2]–[4]. This fast and easy transmission has saved time and money. This transmission, however, may have a severe problem that anyone can see and modify this processed file or message easily. Therefore, it is needed to protect either the computer network or the file itself along with their related systems against security threats [5]. This protection can be carried out by using either cryptography [1], [6], watermarking [7], [8], or steganography [6] which is also known as data hiding [9].

In more details, data hiding is about concealing secret data inside another media [10] such as text [11], audio [4], video [12], [13], image [14]–[19] in order to mislead public while transmitting secret data via the public network [20], [21]. In different words, data hiding is a mechanism of embedding secret messages into a carrier medium (e.g., an image) [22]. In the rest of this paper, we assume that an image is to be used as the cover, whose simple description is provided in Fig. 1.

Fig. 1: A general example of data hiding schemes using images.

In general, data hiding methods can be grouped into spatial and transform domains [23]–[25]. In the spatial domain [10], [21], [26], [27], the hiding can be performed directly to the cover image. For example, it is done by replacing the redundant bits of the cover image. On the other hand, in the transform domain [24], [28], we need to convert the cover image into another domain where the embedding is applied. These two different methods have advantages and disadvantages. However, the spatial domain is often preferred to the transformed one due to its simplicity and capability of embedding high payload [4], [24].

Generally, Difference Expansion (DE) [29] is a well-known reversible data hiding method which initially uses digital images as the cover. As a spatial domain-based method, DE considers the difference value between two neighboring pixels where the secret bit is put. A straightforward idea, [30] has proposed a method that extends the size of the blocks by 2×2 called Quads. This kind of blocks is able to accommodate three bits, which is higher than the previous method. In addition, Reduced Difference Expansion (RDE) method [23], [31] has been proposed to reduce the amount of the ignored blocks due to the overflow/underflow problem. This happens by reducing the difference value before the embedding process. This method has significantly improved the visual quality and increased the embedding capacity. Ahmad et al. [32] have proposed a method based on Difference Expansion of Quads [30]. Finally, Al-Huti et al. [10] have proposed a scheme that mainly has considered enlarging the pixel block size to 4×4 instead of 2×2 which was used by [32]. All the methods mentioned earlier are able to enhance the visual quality of the stego image as well as the

Manuscript received July 13, 2018; revised May 25, 2019. Part of this works was supported by the Ministry of Research, Technology and Higher Education, The Republic of Indonesia.

Mohammed Hatem Ali Al-Hooti is with Faculty of Computer & Information Technology, Sanaa University, Sanaa, Yemen (corresponding author, e-mail: moh_hat84@yahoo.com).

Tohari Ahmad is with Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia; (email: tohari@if.its.ac.id).

Supeno Djanali is with Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia; (email: supeno@its.ac.id).

secret message capacity. However, the amount of the capacity and the quality has not yet reached the optimal amount. Therefore, there is still an opportunity to increase both the amount of the embedded secret bits and the stego quality by, for example, varying the size of the pixel blocks. In this paper, we pay more attention to this issue as an improvement over our previous research [10], [32]. This enhancement is as the following points:

- Obtaining an optimal size of the pixel blocks and analyzing the corresponding methods.
- Extending the type of the cover image, for both grayscale and RGB ones.
- Providing flexibility concerning the block size, e.g., the size along with pixel LSB bit substitution enable this scheme to embed 7 or 31 bits per pixel block by using 2×2 or 4×4, respectively.
- Employing non-changeable blocks without causing either overflow or underflow problem, which is an improvement of [10] and [32]. It is because we replace the second LSB bit of each pixel except the first pixel from each non-changeable block with secret bits.
- Substituting the LSB [33]–[35] of each stego pixel by a secret binary digit is done in order to increase the amount of capacity.

The rest of this paper is organized as follows. Section 2 provides the related works. Section 3 explains our proposed method. The experimental results and the conclusion are in sections 4 and 5, respectively.

II. Related Studies

A. Reduced Difference Expansion

Reduced Difference Expansion (RDE) [23] improves the performance of Difference Expansion (DE) as in its implementation [29] by using (1). It is intended to reduce the expandable difference value d. The authors consider raising the possibility of using more non-changeable blocks which were ignored in DE.

$$d' = \begin{cases} d, & if \quad d < 2 \\ d - 2^{\lfloor log_2(d) \rfloor - 1}, & if \quad d \geq 2 \end{cases} \tag{1}$$

This method has also utilized a location map LM which is represented in (2). It is used in order to locate the pixels which have been reduced. Later, the payload will be stored in that reduced difference value d'. In general, this method performs better than DE in terms of both the capacity and quality. However, this only works on the difference whose value is greater than 2.

$$LM = \begin{cases} 0, & if \quad 2^{\lfloor log_2(d') \rfloor} = 2^{\lfloor log_2(d) \rfloor} \quad or \quad d' = d \\ 1, & if \quad 2^{\lfloor log_2(d') \rfloor} \neq 2^{\lfloor log_2(d) \rfloor} \end{cases} \tag{2}$$

B. Difference Expansion of Quads

Difference expansion of quads [30] extends DE scheme to have a larger block size. The block is a 1 × 4 vector $u = (u_0, u_1, u_2, u_3)$, which should not overlap each other. By constructing blocks of quad, this method is able to hide 3 bits in each block. The difference expansion transformation is an integer transformation $d = (d_0, d_1, d_2, d_3)$ as specified in (3). Here, "$\lfloor . \rfloor$" means the floor operation that brings the least integer value.

$$\begin{aligned} d_0 &= \left\lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \right\rfloor \\ d_1 &= u_1 - u_0 \\ d_2 &= u_2 - u_1 \\ d_3 &= u_3 - u_2 \end{aligned} \tag{3}$$

The new image is constructed by using (4), which is actually the inverse integer transformation of (3).

$$\begin{aligned} u_0 &= d_0 - \left\lfloor \tfrac{d_1+d_2+d_3}{4} \right\rfloor \\ u_1 &= d_1 + u_0 \\ u_2 &= d_2 + u_1 \\ u_3 &= d_3 + u_2 \end{aligned} \tag{4}$$

As in DE [29], there are three types of blocks in this method: expandable, changeable, and non-changeable. Suppose that s_1, s_2, s_3 are the secret data whose value can be either 0 or 1. The embedding will be done using (5) if the block is expandable, or (6) if the block is changeable. Otherwise, the block is ignored and it is considered as non-changeable.

$$\begin{aligned} d'_1 &= 2 \times d_1 + s_1 \\ d'_2 &= 2 \times d_2 + s_2 \\ d'_3 &= 2 \times d_3 + s_3 \end{aligned} \tag{5}$$

$$\begin{aligned} d'_1 &= \left\lfloor \tfrac{d_1}{2} \right\rfloor + s_1 \\ d'_2 &= \left\lfloor \tfrac{d_2}{2} \right\rfloor + s_2 \\ d'_3 &= \left\lfloor \tfrac{d_3}{2} \right\rfloor + s_3 \end{aligned} \tag{6}$$

In addition, this method has made a location map to locate each block and to indicate it as expandable, changeable, or non-changeable. However, this method pays no attention to a large amount of the ignored non-changeable blocks.

C. An Improved Quads and RDE-based Medical Data Hiding

This method [32] is based on quads and RDE. It is proposed to increase the capacity and to improve the quality. This scheme is similar to [30] in terms of the embedding and extraction processes. However, there are some dissimilarities, for example, the value of the first difference d_0 is 0; also, this method has made use of RDE [23] which is applied only on the expandable blocks with some changes. For instance, RDE considers only the difference which is higher than 2, while this method uses both positive and negative differences which are not in the interval [1, -1]. Therefore, they implement (7) and (8) to process both positive and negative differences, respectively.

$$d'_n = d_n - 2^{\lfloor log_2(|d_n|) \rfloor}, \quad if \quad d_n > 1 \tag{7}$$

$$d'_n = d_n + 2^{\lfloor log_2(|d_n|) \rfloor}, \quad if \quad d_n < -1 \tag{8}$$

Slightly different from [30], the new image is built by using (9). Furthermore, this method uses a location map that allows it to locate and determine the type of the block either it is expandable (RDE or non-RDE), changeable, or non-changeable.

$$\begin{aligned} u_0' &= u_0 \\ u_1' &= d_1' + u_0 \\ u_2' &= d_2' + u_1 \\ u_3' &= d_3' + u_2 \end{aligned} \tag{9}$$

According to their results, the research has delivered higher capacity and quality. Nevertheless, it is still not able to make all pixels carrying the secret data because there are many non-changeable blocks in the cover.

D. Multi-bit Reversible Data Hiding with Prediction and Difference Alteration

As one of the digital rights management applications, the scheme in [36] aims to improve the capacity for prediction-based reversible data hiding. This is done by altering the difference values between the original image and the predicted one. Based on that, they can hide multiple bits in a block. For this process, they consider every two bits at a time, as follows:

- If the two bits are 00 then the original value is not modified.
- If the two bits are 01 then the original value is added by 1.
- If the two bits are 10 then the original value is added by 2.
- If the two bits are 11 then the original value is added by 3.

Based on that, the authors has guaranteed the capacity which is higher than 1 bpp. On the other hand, the quality is degraded. It needs to balance the quality, capacity, and reversibility.

E. Robust Reversible Data Hiding Algorithm For Color Image Based on 2D-DCT

As provided in [37] a reversible data hiding method base on 2-dimensional discrete cosine transform is proposed. They aim to solve the low interpretation of anti-rotation attacks. In addition, authors also consider controlling the trade-off that occurs between the capacity and the stego image quality. A vector-based process is used for the embedding in this scheme. The three RGB components and the correlation among pixel integral are joined together while the embedding so that all R, G, and B channels are employed. However, this algorithm chooses only a low-frequency region for embedding. It is claimed that low-frequency regions can resist against attack. This work has paid much attention on the robustness and considered less for capacity and security.

III. Improving the Capability of RDE

In order to improve the capacity of the secret data and the quality of the stego image, we work by using [10], [32], and [34] as our base research. This designed method is able to work on different sizes of blocks as depicted in Fig. 2. In more details, instead of using 2×2 block pixel which is only able to accommodate 3 bits, this proposed method is able to embed 7 or 31 bits per pixel block by applying two layers of embedding and using 2×2 or 4×4 pixels, respectively.

Different from [10] and [32], this work is able to use non-changeable blocks without causing overflow or underflow problems. This happens by replacing the second LSB bit of $(n-1)$ pixels in each non-changeable block with the secret bits. We also apply the second embedding process on the stego pixels of each block by replacing the LSB bit with a secret message and this substituted pixel bit is saved into the location map file. This second process helps to hide more bits while maintaining the recovery of the embedded bits by using RDE of the same block. It is worth to note that a large block size reduces the number of location maps LM and raises the capacity. As a trade-off, the quality of the embedded image is affected.

Because of enlarging the block size, the possibility of the pixel value after the embedding to change is higher. For this reason, we also work on the difference transformation and its corresponding inverse integer transformation for obtaining new pixel values. As shown in Fig. 2, the corresponding pixels are represented by red arrows. In more details, the process of embedding and extraction are described in the following subsections.

A. Embedding Process

The overall embedding process is provided in Fig. 3 whose steps can be described as follows:

1: The secret message is generated and reshaped to be a 2D matrix $s(i, w)$ where i represents the number of blocks, and w indicates the number of embedded bits in each block. For example, in 2×2 and 4×4 the value of w is 7 and 31, respectively. It is worth to note that L bits are embedded using improved reduced difference expansion method and the other $w - L$ bits are hidden using LSB replacement for every pixel in the same block.

2: The location map file $LM(i, Q)$ is initiated based on the block size. Here, i is the number of blocks within the cover image, and $Q = 2 \times N$ is either 9 or 33 based on the block size 2×2 or 4×4, respectively. The reason is that the first two location maps are used to indicate either the block is RDE-expandable, nonRDE-expandable, changeable, or non-changeable. The next L location maps are used to help the recovery process from RDE scheme.

3: The cover image is divided into non-overlapped blocks, which are constructed from left to right and from top to bottom.

4: Each block is treated as a vector $u = (u_0, u_1, u_2, ., u_{BL})$ where u_i and BL represent the value of i^{th} pixel and the number of elements in the block, respectively (e.g. if the block size is 2×2 or 4×4, then BL is 4 or 16). Here, the last BL location maps are used to save and transfer the substituted LSB bits of every block.

5: Different from [10], [30], [32], we provide (10) as an integer transformation to get a difference value between two pixels in the same block. Here, j is the length or width of the blocks, and L is the number of the block elements minus one $(L = BL - 1)$. It also means the number of secret bits in each block (for example, if the block size is 2×2 or 4×4 the value of

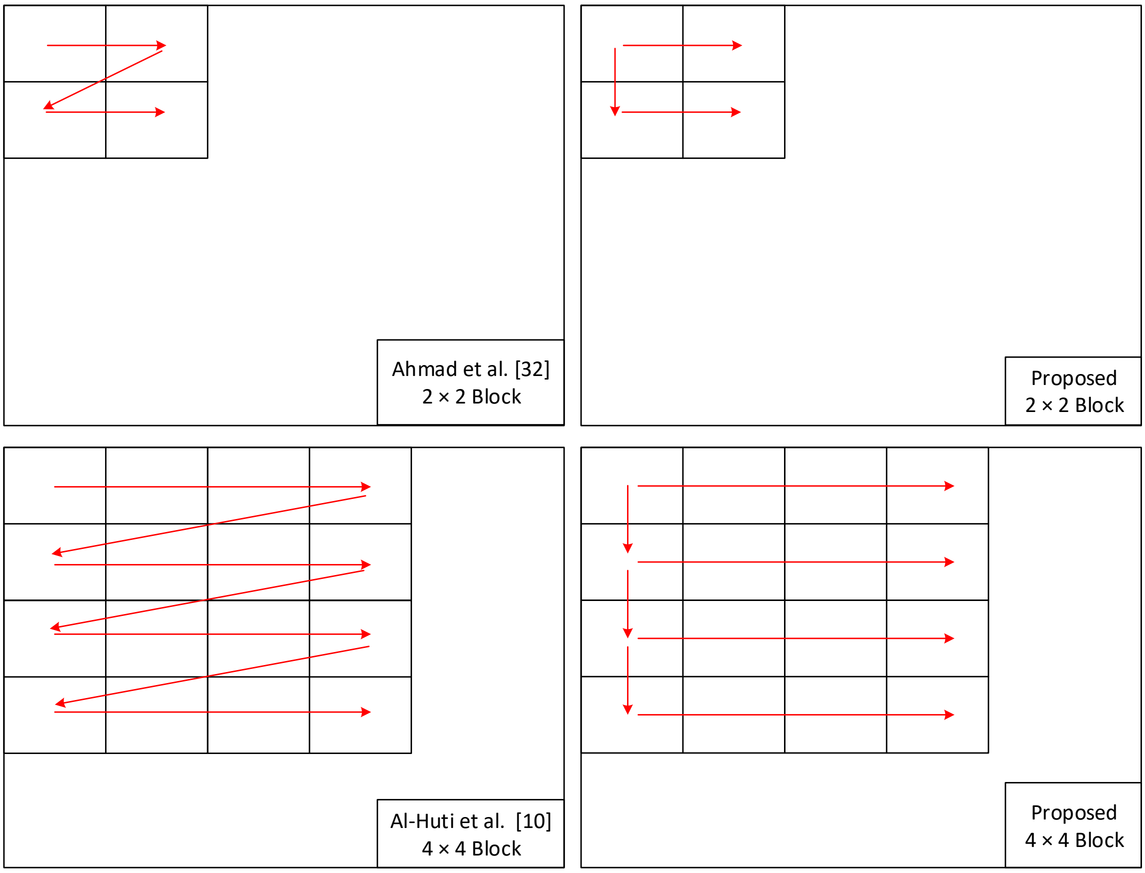


Fig. 2: The variation between the proposed method and the previous works in terms of obtaining the difference value [10], [32].

L is 3 or 15, respectively). Furthermore, the iteration which is carried out from 0 to L represents the block size minus one. This is because the first pixel in each block is always not used for embedding. This means the first difference d_0 is always 0.

$$d_i = \begin{cases} 0, & if \quad i = 0 \\ u_{i+1} - u_{i-j}, & if \quad mod(i,j) = 0 \\ u_{i+1} - u_i, & if \quad otherwise \end{cases} \tag{10}$$

6: The type of the blocks is determined. Similar to [10], [32], a block of pixels is either expandable, changeable or non-changeable. Differently, we define those categories as follows.
(i) Expandable blocks. These types of blocks consist of RDE-expandable and nonRDE-expandable. The RDE reduces the difference values before the embedding process targeting to minimize the overflow or underflow problem. Improved RDE [10] scheme helps to further minimizing the difference values. This can increase the capacity and maintain the quality. This method handles the positive or negative values, which are processed by (11) and (12), respectively. Here, d_n and $d_n^{'}$ are the original difference and the reduced difference values before the embedding, respectively. After that, these reduced values are embedded with the secret bits using (13). In terms of nonRDE-expandable, the secret bits are directly embedded using the same method (13), where s_i is the secret bit, and $d_1^{''}, d_2^{''}, ..., d_L^{''}$ are the difference values after carrying the secret bit.

$$d_n^{'} = d_n - 2^{\lfloor log_2(d_n) \rfloor - 1} + \lfloor log_2(d_n) \rfloor - 1, \quad if \quad d_n > 1 \tag{11}$$

$$d_n^{'} = d_n + 2^{\lfloor log_2(|d_n|) \rfloor - 1} + \lfloor log_2(|d_n|) \rfloor - 1, \quad if \quad d_n < -1 \tag{12}$$

$$\begin{aligned} d_1^{''} &= 2 \times d_1 + s_1 \\ d_2^{''} &= 2 \times d_2 + s_2 \\ \dots \quad & \dots\dots \quad \dots \quad \dots\dots \\ d_L^{''} &= 2 \times d_L + s_L \end{aligned} \tag{13}$$

(ii) Changeable blocks. These blocks are used to reduce the expandable values to avoid overflow or underflow problem. The embedding occurs by using the difference values divided by two as in (14) in order to embed the secret data.
(iii) Non-changeable. Different from [10], [32], these blocks are also used for embedding. This happens by substituting the second least significant bit of each pixel excluding the first pixel by the secret bits. These blocks can carry bits same as the other types in a

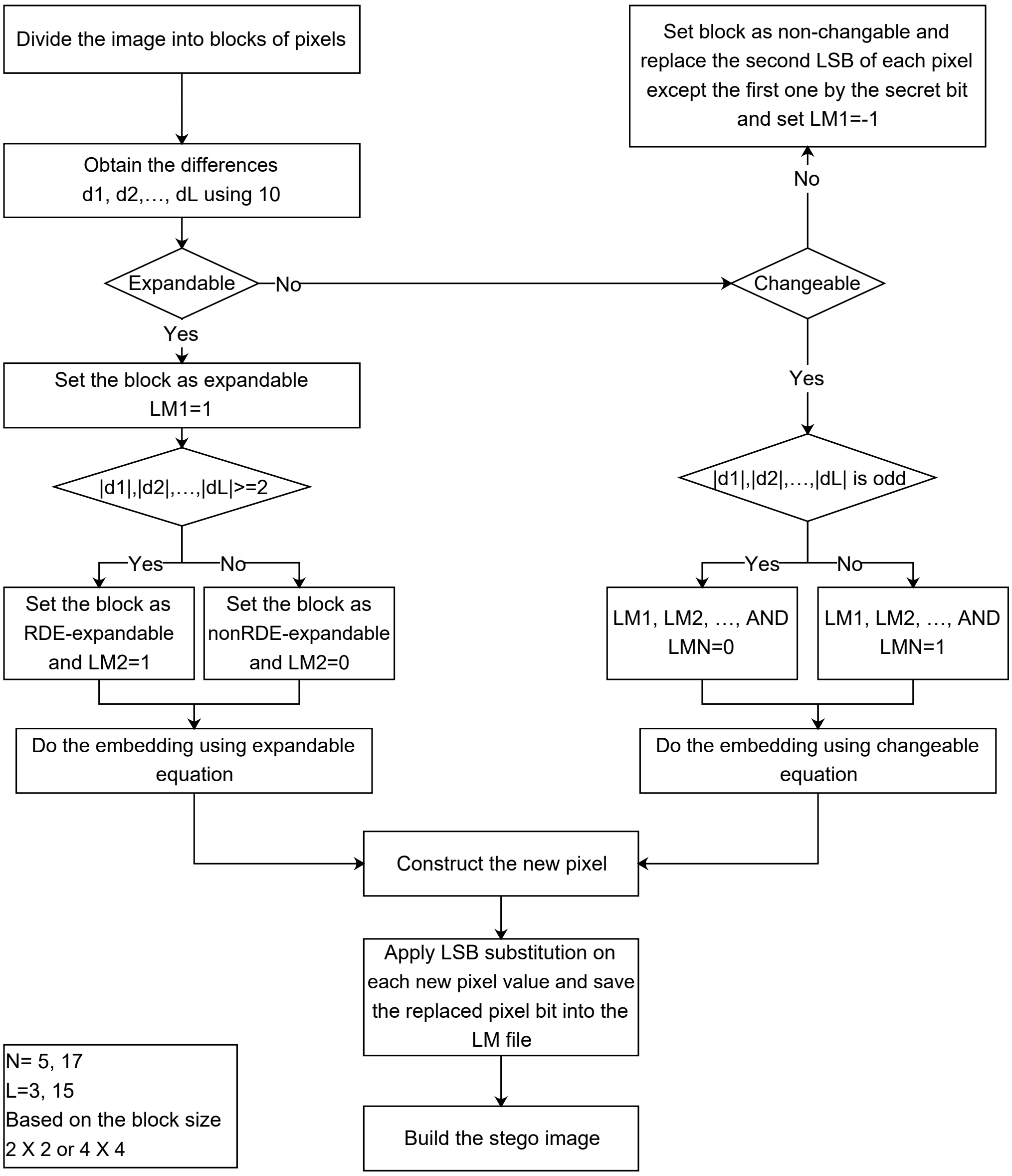


Fig. 3: The embedding process.

different manner.

$$
\begin{aligned}
d_1^{''} &= \left\lfloor \frac{d_1}{2} \right\rfloor + s_1 \\
d_2^{''} &= \left\lfloor \frac{d_2}{2} \right\rfloor + s_2 \\
\ldots & \quad \ldots\ldots \quad \ldots\ldots \\
d_L^{''} &= \left\lfloor \frac{d_L}{2} \right\rfloor + s_L
\end{aligned} \tag{14}
$$

6: The new stego pixel based RDE technique can be made using (15) which is the inverse integer transformation of (10).

$$
u_i^{'} = \begin{cases} u_0^{'} = u_0 \quad if \quad i = 0 \\ d_{i+1} + u_{i-j}, \quad if \quad mod(i,j) = 0 \\ d_{i+1} + u_i, \quad if \quad otherwise \end{cases} \tag{15}
$$

7: Finally, the same block is processed using LSB technique [33]–[35]. As shown in (16) the least significant bit of each pixel is saved into the location map file. Since this file is used in the embedding using RDE, it is extended to hold these bits starting from the index BL+2 where BL represents the number of pixels in the block.

$$
\begin{aligned}
&LM_{BL+2} = LSB(u_1^{'}) \\
&LM_{BL+3} = LSB(u_2^{'}) \\
&\ldots\ldots\ldots\ldots \quad \ldots\ldots \\
&LM_Q = LSB(u_{BL}^{'})
\end{aligned} \tag{16}
$$

Here, Q means the number of location maps, and $u^{'}$ represents the stego pixel before being processed by LSB. As presented in (17), $u^{''}$ means the stego pixel that is used to construct the stego image.

$$
\begin{aligned}
u_1^{''} &= u_1^{'} \\
u_2^{''} &= u_1^{'} \\
\ldots & \quad \ldots\ldots \\
u_{BL}^{''} &= u_{BL}^{'}
\end{aligned} \tag{17}
$$

According to (18) the LSB of each pixel are replaced by the secret message that starts from the bit BL and ends up with w. This indicates the maximum capacity.

$$\begin{array}{l} LSB(u_1^{''}) = s_{BL} \\ LSB(u_2^{''}) = s_{BL+1} \\ .... \quad .... \\ LSB(u_{BL}^{''}) = s_w \end{array} \tag{18}$$

After all these steps, the difference values and the secret bits are already merged, which results in the stego image. In this algorithm, we have extended the location map for each block to be $(LM = LM_1, LM_2, ..., LM_Q)$ where Q is (2× the number of the pixels in every block) subtracted by one, all are added by two.

In more details, the use of the location map can be described as follows. The values of LM_1 can be either 1, 0, or -1 which means that the block is expandable, changeable, or non-changeable, respectively. In addition, $LM_1 = 1$ and $LM_2 = 1$ are used to locate the RDE-expandable blocks. $LM_1 = 1$ and $LM_2 = 0$ mean non-RDE expandable blocks. However, the location maps starting from LM_3 to LM_N are 1, if the block is RDE-expandable and $d_n^{'} \pm (2^{\lfloor log_2(|d_n|)\rfloor-1} + \lfloor log_2(|d_n|)\rfloor - 1)$ is equal to d_n. Moreover, the location maps starting from LM_3 till LM_N are 0 if the block is RDE-expandable and $d_n^{'} \pm (2^{\lfloor log_2(|d_n|)\rfloor} + \lfloor log_2(|d_n|)\rfloor)$ is exactly same as d_n. However, if the block is changeable and the difference values are odd, the location maps starting from LM_3 to LM_N are 0, and vice versa for a difference even values [30].

It is worth noting that, this method is also suitable to use for RGB color images as the cover image. This occurs by individually applying the embedding process to every single channel (R, G and B). Based on that, we can hide exactly three times secret bits compared to grayscale ones.

B. The Extraction Process

The hidden secret data and the cover image can be reconstructed in the following steps:

1: The stego image is partitioned into non-overlapping blocks.

2: The secret message that was hidden based on LSB substitution is obtained directly from each pixel using (19).

$$\begin{array}{l} s_{BL}^{'} = mod(u_1^{''}, 2) \\ s_{BL+1}^{'} = mod(u_2^{''}, 2) \\ ...... \quad .... \\ s_w^{'} = mod(u_{BL}^{''}, 2) \end{array} \tag{19}$$

3: The modification that has occurred by LSB replacement is recovered by using the values that are in the location maps.

4: The difference values are obtained using (10).

5: Determine the type of the pixel block based on the location maps values. For instance, we can obtain the RDE-expandable blocks if both LM_1 and LM_2 are 1. Moreover, we can obtain the nonRDE-expandable blocks if LM_1 is 1 and LM_2 is 0. However, if LM_1 is 0 then the obtained block is changeable; the obtained block is non-changeable if LM_1 is -1.

6: The secret message can be extracted based on the type of the pixel blocks. So, for the RDE-expandable or non-RDE, and the changeable blocks, the secret bits can be obtained by gaining the LSB of each $d_n^{''}$. Concerning non-changeable blocks, the secret bits are the second LSB bit of each pixel excluding the first one.

7: The recovery of the original image is based on the type of the pixel block as the following:
(i) In the previous embedding process, the difference of RDE-expandable blocks is reduced. Thus, it needs to recover it to its original value. This happens if the difference $d_n^{''} > 1$ and the value of the location maps from LM_3 to LM_N is 0. Then, the first part of (20) is applied to obtain the original difference. On the other hand, if the difference value $d_n^{''} > 1$ and the corresponding value of location maps from LM_3 to LM_N is 1, then the second part of (20) is applied.

$$d_n = \begin{cases} d_n^{''} + 2^{\lfloor log_2(|d_n^{''}|)\rfloor-1} + \lfloor log_2(|d_n^{''}|)\rfloor - 1, \\ \qquad if \quad LM = 0 \\ d_n^{''} + 2^{\lfloor log_2(|d_n^{''}|)\rfloor} + \lfloor log_2(|d_n^{''}|)\rfloor, \\ \qquad if \quad LM = 1 \end{cases} \tag{20}$$

Similar to [30], we recover the secret using (21) and (22). This corresponds to the reduction which has been done in the previous embedding process. It needs to restore the reduced difference value to its original value since this difference value is required to reconstruct the new image after the extraction process. Therefore, if the difference value $d_n^{''} < -1$ and the value of location maps from LM_3 to LM_N is 1, then (21) is used. Note that, $d_n^{''}$ means a difference value that is still carrying the secret bits.

$$d_n = d_n^{''} - 2^{\lfloor log_2(|d_n^{''}|)\rfloor-1} + \lfloor log_2(|d_n^{''}|)\rfloor - 1 \tag{21}$$

However, if the difference value $d_n^{''} < -1$ and the location maps from LM_3 to LM_N are 0 then (22) is utilized.

$$d_n = d_n^{''} - 2^{\lfloor log_2(|d_n^{''}|)\rfloor} + \lfloor log_2(|d_n^{''}|)\rfloor \tag{22}$$

(ii) The recovery process of the nonRDE-expandable pixel block uses (23) which is also used by [30].

$$d_n = \left\lfloor \frac{d_n^{''}}{2} \right\rfloor \tag{23}$$

(iii) In processing the changeable blocks, the original difference can be gained based on the values of the location maps LM_3 to LM_N. If they are 0 and the difference value is odd then (24) is employed.

$$d_n = 2 \times \left\lfloor \frac{d_n^{''}}{2} \right\rfloor - 1 \tag{24}$$

However, if the location maps LM_3 to LM_N are 1 and the difference value is even, then (25) is used. After that, the new pixels can be processed using (14).

$$d_n = 2 \times \left\lfloor \frac{d_n^{''}}{2} \right\rfloor + 1 \tag{25}$$

Finally, if LM_1 =-1, then the block is non-changeable, and we can recover the second least significant bit excluding the first pixel of each block.

TABLE I: Differences between the proposed method and its related previous works [10], [32].

Parameter	The proposed method	Related studies [10], [32]
Cover	grayscale and RGB images	grayscale
Block size	2×2 and 4×4	2×2 [10], 4×4 [32]
Base	combining RDE and LSB techniques	improving RDE technique
Bit per block	7 bits for 2×2	3 bits for 2×2 [10]
	31 bits for 4×4	15 bits for 4×4 [32]
Difference	The difference is obtained from the two closest pixels as shown in Fig. 2 $d_i = \begin{cases} 0, & if \quad i = 0 \\ u_{i+1} - u_{i-j}, & if \quad mod(i,j) = 0 \\ u_{i+1} - u_i, & if \quad otherwise \end{cases}$	The difference is obtained from two consecutive pixels as shown in Fig. 2 $d_0 = 0$ $d_1 = u_1 - u_0$ $d_2 = u_2 - u_1$ $d_3 = u_3 - u_2$
Stego pixel	$u_i^{'} = \begin{cases} u_0^{'} = u_0 & if \quad i = 0 \\ d_{i+1} + u_{i-j}, & if \quad mod(i,j) = 0 \\ d_{i+1} + u_i, & if \quad otherwise \end{cases}$	$u_0' = u_0$ $u_1' = d_1' + u_0$ $u_2' = d_2' + u_1$ $u_3' = d_3' + u_2$

In terms of the RGB images, the extraction procedure is individually done on each R, G and B channels. Table I demonstrates the main differences between this proposed work and its related previous studies [10], [32].

IV. The Experimental Results

This experiment compares the performance of the proposed method with the previous works [10], [32] using Matlab 2017a. The used dataset is as follows:

1) The secret message is generated randomly using Randi function.
2) The cover images are obtained from [38], [39], which are eleven (medical and standard 512 × 512) grayscale images and five 512 × 512 RGB color images. In addition, ”Lungs”, ”Barbara”, and ”Lena” are chosen to represent grayscale and color images to evaluate the suitability of this method.

There are many data hiding evaluation ways and parameters. All the previous works [10], [32] and the proposed method, are evaluated based on the secret messages capacity and the stego image quality. We use some evaluation aspects to measure the performance of this proposed method as follows:

A. Capacity

The secret message capacity is measured using (26) by counting the number of bits that can be inserted into the stego image. It is worth to note that this formula is only applicable to count the capacity of the proposed method. Here, ρ indicates the capacity of the cover. The symbols ∂, $\mathcal{E}$, $\mathcal{C}$, and $\mathcal{N}$ represent the amount of RDE-expandable, nonRDE-expandable, changeable, and non-changeable blocks, respectively; while w depicts the number secret bit that can be embedded into each block. It is shown that the payload means the number of bits in each block multiplied by the number of blocks within the whole image. It is common that, the greater the capacity, the better the performance of the data hiding method, and vice versa.

$$\rho = ((\partial + \mathcal{E} + \mathcal{C} + \mathcal{N}) * w) \tag{26}$$

Tables II and III show that the number of block of each category differs each other, referring to the color intensity of the cover. According to Table II, this proposed method is able to hide 7 bits per 2×2 block, whereas from Table III we find that the method carries 31 bits per 4×4 block. In terms of the color image, the total amount of payload in each cover is also ensured to be three times the grayscale. It is because all RGB channels are fully embedded. In summary, the amount of embedded secret bits in any cover image is almost twice the number of pixels within that particular image.

B. Imperceptibility

The subjective and objective measurement is performed as follows.

1) Subjective Evaluation: This can be seen in Fig. 4, which depicts that by using human eye it is relatively hard to distinguish whether the image has been embedded or not. For example, Fig. 4(a) is the Aerial cover image before the embedding, while Fig. 4(b) is the stego file of the same image after being embedded.

2) Objective Measurement: Peak Signal-to-Noise Ratio (PSNR) is a well-known evaluation scheme that is used to measure the quality [10], [40], [41]. It is presented in (27) and (28). This PSNR considers measuring the similarity between the cover and the stego image. The higher the PSNR value the better the quality of the stego image, and vice versa. Here, MSE represents Mean Square Error, while $x_{i,j}$ and $x'_{i,j}$ are the original and stego pixel values, respectively; whereas $M \times N$ indicates the image size. Note that, PSNR is adapted not only to measure the quality of grayscale but also measuring that of RGB images. The difference is that in RGB images the columns N are extended to three times more than the size of the gray ones. For instance, if the image size is 512×512, then the size of the color image is 512×1536. This enables the MSE scheme to go through all the pixel values within all RGB channels.

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE} (dB) \tag{27}$$

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{i,j} - x'_{i,j})^2 \tag{28}$$

TABLE II: The number of blocks of each category and capacity within each type of block based 2×2 blocks using grayscale images.

The image name	Number of blocks				Amount of capacity in each type of block (bits)			
	RDE -expandable	nonRDE -expandable	Changeable	Non -changeable	RDE -expandable	nonRDE -expandable	Changeable	Non -changeable
Truck and APCs	46561	18937	33	5	325927	132559	231	35
Stream and bridge	41224	23508	624	180	288568	164556	4368	1260
Aerial	37348	27603	357	228	261436	193221	2499	1596
Lungs	35351	29747	248	190	247457	208229	1736	1330
Tank	46913	18549	74	0	328391	129843	518	0
Baboon	51665	13199	574	98	361655	92393	4018	686
Peppers	35197	30008	293	38	246379	210056	2051	266
Barbara	38849	26405	208	74	271943	184835	1456	518
Couple	37643	27373	454	66	263501	191611	3178	462
Boat	39605	25805	85	41	277235	180635	595	287
Lena	30677	34808	35	16	214739	243656	245	112

TABLE III: The number of blocks of each category and capacity within each type of block based 4×4 blocks using grayscale images.

The image name	Number of blocks				Amount of capacity in each type of block (bits)			
	RDE -expandable	nonRDE -expandable	Changeable	Non -changeable	RDE -expandable	nonRDE -expandable	Changeable	Non -changeable
Truck and APCs	2995	12981	316	92	92845	402411	9796	2852
Stream and bridge	2525	11012	1764	1083	78275	341372	54684	33573
Aerial	2388	11101	1180	1715	74028	344131	36580	53165
Lungs	2611	12765	377	631	80941	395715	11687	19561
Tank	2958	13040	360	26	91698	404240	11160	806
Baboon	5635	7992	825	1932	174685	247752	25575	59892
Peppers	1356	13936	895	197	42036	432016	27745	6107
Barbara	3789	11366	685	544	117459	352346	21235	16864
Couple	1670	13178	1136	400	51770	408518	35216	12400
Boat	2097	13218	690	379	65007	409785	21390	11749
Lena	1443	14347	429	165	44733	444757	13299	5115

We also investigate the effect of the payload size on the quality. For this purpose, we determine the payload to 7kb, 35kb, 175kb, and 350k. This is according to the number of bits which can be embedded in each block. Also, this size represents possible amount of bits often used by users. This experimental result is provided in Tables IV and V which use 2×2 and 4×4, respectively for grayscale images. It is shown that increasing the payload size reduces the quality. Nevertheless, the quality reduction is going lower for higher payload size. This trend also happens to color images as presented in Fig. 5.

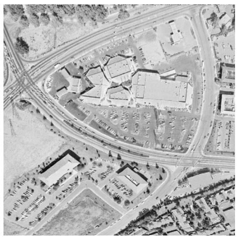
According to Tables IV and V, it is found that smaller blocks (i.e., 2×2) produce better quality of the stego images than the bigger ones (i.e., 4×4). On average, the difference ranges from 3.75dB to 4.5dB, where rising the payload may also rise this difference. Nevertheless, increasing the payload from 175kb to 350kb shows slightly dissimilar trends, where the difference is about 0.052dB lower than that of from 35kb to 175kb. This condition depicts that increasing payload size causes more decreases in quality of the stego images containing bigger blocks than that of smaller ones. Nevertheless, in a certain payload size, bigger blocks may be more appropriate as long as it does not exceed the maximum capacity that the cover can carry.

So, in general, it can be inferred that both bigger and smaller blocks (i.e., 4×4 and 2×2, respectively) are suitable to use, depending on the available environment. Additionally, it is worth noting that smaller blocks hold fewer payload but their number goes up. As the trade-off, the quality increases.

C. Histogram Analysis

Histogram analyzes the difference between the cover and stego images. Based on histograms, a small change in the stego image can be detected. It is due to noise in the pixels which appears in the stego histogram. Fig. 6 which indicates the histogram comparison between the cover and stego images of Lena grayscale. Sub-figure (b) proves that the stego image is almost similar to the original image whereas this image is carrying capacity from 1.75 to 1.93 bits per pixel. Lena color image histogram is plotted in Fig. 7 that shows the difference between the original color image and the embedded image. This also proves the noise that occurs in the stego file is very small compared to the amount of the embedded payload where all RGB channels of this image are fully embedded. To sum up, the histogram analysis shows that this work is able to keep the shape of the stego image

(a)



(b)

(c)

(d)

(e)

(f)

Fig. 4: Examples of the cover and its stego. (a) Original Aerial. (b) Stego Aerial. (c) Original Barbara. (d) Stego Barbara. (e) Original Airplane (e) Stego Airplane (F-16) [39].

TABLE IV: The quality of the stego grayscale images by using 2×2 blocks.

The image name	PSNR (dB) by using 2×2 blocks for respective payload size			
	7 kb	35 kb	175 kb	350 kb
Truck and APCs	55.07	48.63	41.55	38.6
Stream and bridge	52.24	45.62	39.23	36.76
Aerial	52.91	47.78	41.48	38.49
Lungs	62.08	51.66	42.56	38.7
Tank	56.79	49.77	42.28	38.98
Baboon	51.05	44.68	38.54	36.36
Peppers	57.46	50.36	43.37	40.21
Barbara	56.13	49.79	42.38	38.04
Couple	57.18	50.24	41.85	38.64
Boat	57.83	50.56	42.25	38.82
Lena	58.71	52.4	43.83	40.36

histogram since it is almost identical to the original cover histogram.

D. The Secret Message Reversibility

In this section, we investigate the recovery of the secret message. Normally, there is an original secret message s_i and the recovered secret message represented by $s_i^{'}$ where the recovered secret message should be 100% similar to the original one. Exclusive OR (XOR) function [34], [35] is used for this evaluation where its values are zeros if both the original and recovered messages are same. However, if any recovered bit is not similar to the original one then the resulted XOR value of that bit will be one. According to the obtained results, this proposed work is able to fully recover the secret message.

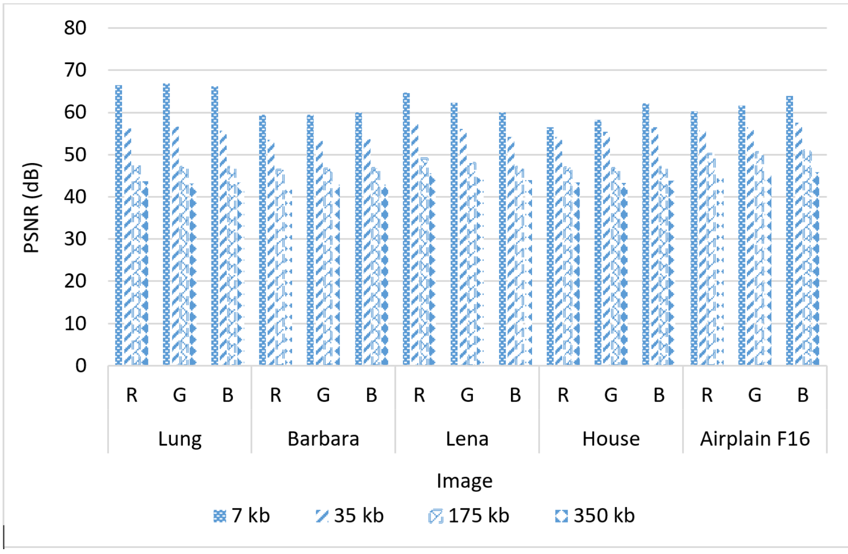
E. Comparison Study and Discussion

We use 2×2 and 4×4 block sizes, so each of them corresponds to the block size in the previous works [10] and [32], respectively. The obtained capacity based on grayscale images is shown in Table VI which shows that by implementing the proposed method, the cover is capable to carry the secret more than twice the capacity of the previous works [10], [32]. This occurs by not only using the non-changeable block to be embedded but also the second embedding bits replacement.

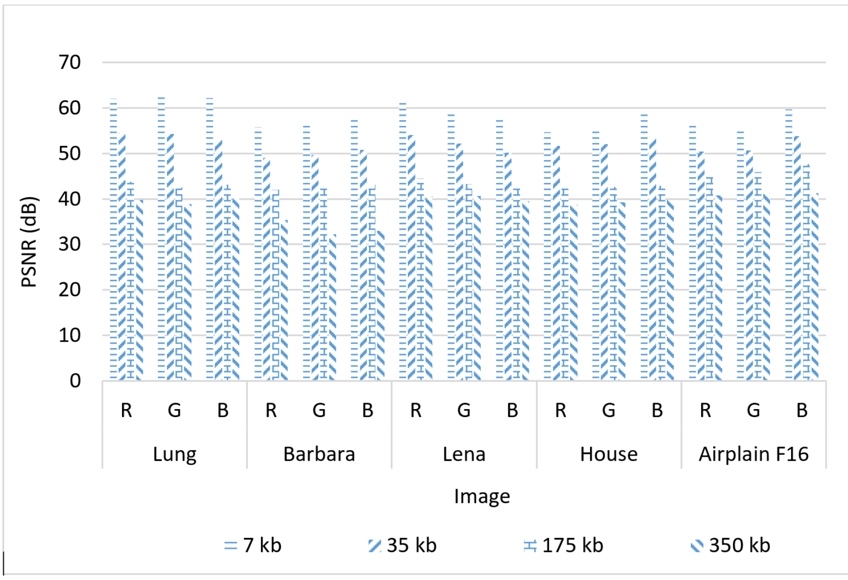
Table VII indicates the achieved quality of the proposed work compared with the previous works using fully embedded stego images [10], [32]. It is depicted that the proposed method has increased the PSNR values for about 2dB and 0.5dB for 2×2 and 4×4 block sizes, respectively.

TABLE V: The quality of the stego grayscale images by using 4×4 blocks.

The image name	PSNR (dB) by using 4×4 blocks for respective payload size			
	7 kb	35 kb	175 kb	350 kb
Truck and APCs	48.74	42.25	35.64	32.61
Stream and bridge	49.39	42.24	35.86	33.27
Aerial	52.13	43.31	36.47	33.64
Lungs	58.32	49.59	37.75	33.8
Tank	50.7	43.18	36.05	33.14
Baboon	49.88	43.17	36.46	33.26
Peppers	54.66	45.9	38.79	35.69
Barbara	50.98	44.31	37.44	34.27
Couple	50.99	44.42	37.05	34.13
Boat	54.01	46.6	38.38	34.54
Lena	56.3	48.89	39.04	35.79



(a)



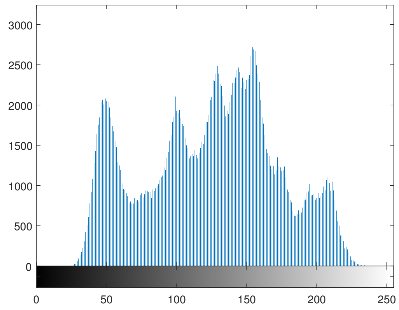
(b)

Fig. 5: The quality of some stego RGB images. (a) Using 2×2 blocks. (b) Using 4×4 blocks.

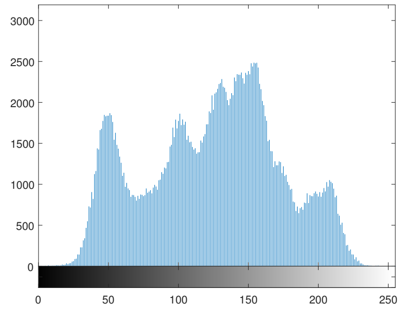
Moreover, in certain covers, this PSNR value sharply rises for about 5dB. This quality enhancement is the result of improving the difference transformation scheme and its corresponding new pixel construction method. Both are modified to consider the difference between the two closest pixels. This is because the closer the pixel values, the higher the possibility of the color similarity. All come out with better capacity and quality. It is noted that the larger the pixel block

TABLE VI: The obtained capacity in bits using grayscale images.

The image name	The previous works (bits)		The proposed method (bits)	
	Block 2×2 [32]	Block 4×4 [10]	Block 2×2	Block 4×4
Truck and APCs	196608	245520	458752	507904
Stream and bridge	196254	241920	458752	507904
Aerial	196242	240465	458752	507904
Lungs	196107	242010	458752	507904
Tank	196608	245685	458752	507904
Baboon	196587	241170	458752	507904
Peppers	196605	244095	458752	507904
Barbara	196608	241335	458752	507904
Couple	196497	242700	458752	507904
Boat	196602	242895	458752	507904
Lena	196608	244605	458752	507904

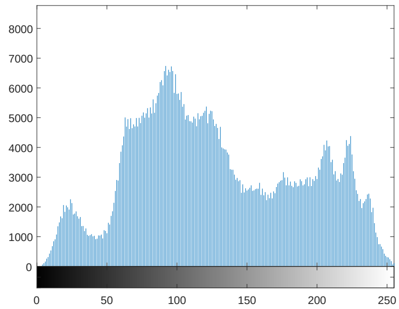


(a)

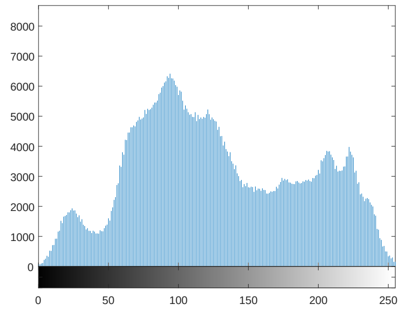


(b)

Fig. 6: Histogram analysis based on Lena grayscale image (a) The original image (b) The stego image.



(a)

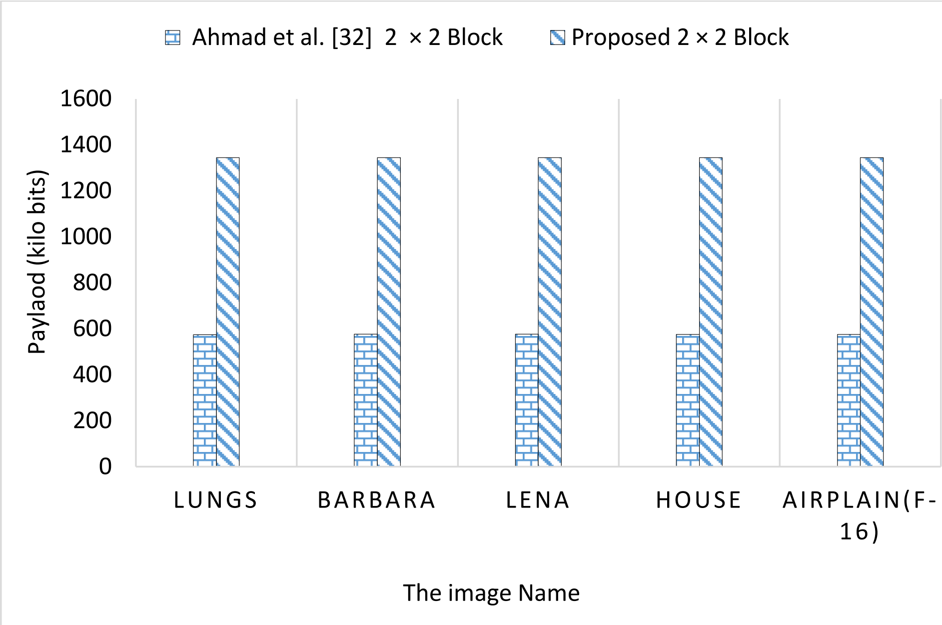


(b)

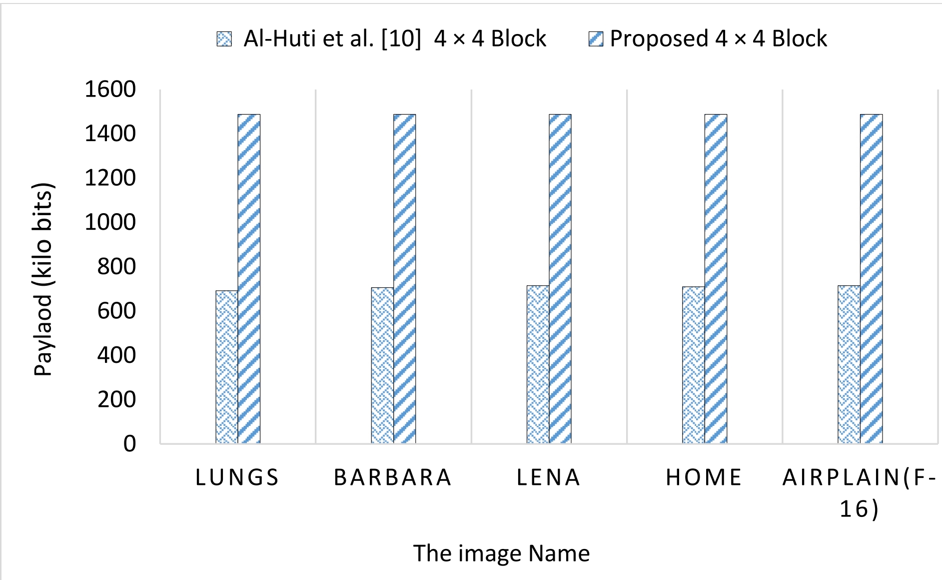
Fig. 7: Histogram analysis based on Lena color image (a) The original image (b) The stego image.

TABLE VII: The quality of the stego image (grayscale).

The image name	The previous works (dB)		The proposed method (dB)	
	Block 2×2 [32]	Block 4×4 [10]	Block 2×2	Block 4×4
Truck and APCs	36.41	31.17	37.62	31.33
Stream and bridge	35.12	31.69	35.89	31.84
Aerial	35.79	32.07	37.18	32.34
Lungs	37.05	32.42	37.96	32.68
Tank	36.96	31.52	37.89	31.62
Baboon	33.87	31.32	35.42	31.68
Peppers	38.73	34.16	39.19	34.31
Barbara	31.82	32.39	36.96	32.95
Couple	35.93	32.74	37.84	32.84
Boat	37.09	32.53	37.86	32.61
Lena	38.97	34.00	39.28	34.23



(a)



(b)

Fig. 8: The capacity comparison between the proposed method and the previous works [10], [32] for RGB images. (a) The capacity based on 2×2 blocks. (b) The capacity based on 4×4 blocks.

size, the lesser the value of PSNR. On the other hand, the average of the payload increase is relatively high, compared with the decrease of the stego quality .

On the othe hands, a color image consists of R, G, and B channels; so, every single channel is embedded with the maximum payload. RGB image allows hiding almost three times payload the grayscale images.

Table VIII shows the evaluation of the capacity by using RGB images as the cover. This indicates that this method can perform superior to both the previous works [10], [32]. It is good to note that, the more the use of each R, G, or B in the color image, the greater the opportunity for the secret data to be embedded within that channel, and vice versa. So, according to Table VIII, the amount of the capacity varies from one channel to the others, depending on the color intensity of that channel.

According to Fig. 8, which summarizes the amount of payload within all R, G, and B channels, this method has achieved higher embedding capacity than both previous RDE algorithms [32] and [10]. Moreover, the color intensity affects the embedding capacity in both the previous works whereas it has a slight impact on the proposed method only in the quality.

Table IX presents PSNR values by implementing RGB images. These values are obtained after embedding all the RGB channels. This RGB-based method performs better than the previous grayscale-based method because a stego color image carries payload three times the grayscale ones.

TABLE VIII: The obtained capacity in bits using RGB images.

The image name		The previous works (bits)		The proposed method (bits)	
		Block 2×2 [32]	Block 4×4 [10]	Block 2×2	Block 4×4
Lungs	Red	195438	230985	458752	507904
	Green	196050	240540	458752	507904
	Blue	195933	237330	458752	507904
Barbara	Red	196608	241650	458752	507904
	Green	196608	241440	458752	507904
	Blue	196596	239910	458752	507904
Lena	Red	196536	244095	458752	507904
	Green	196608	242250	458752	507904
	Blue	196608	245475	458752	507904
House	Red	196602	243675	458752	507904
	Green	196230	241065	458752	507904
	Blue	196374	242175	458752	507904
Airplane (F-16)	Red	196608	244620	458752	507904
	Green	196377	242670	458752	507904
	Blue	196608	245535	458752	507904

TABLE IX: The obtained quality in dB using RGB images.

The image name		The previous works (dB)		The proposed method (dB)	
		Block 2×2 [32]	Block 4×4 [10]	Block 2×2	Block 4×4
Lungs	Red	42.19	38.07	42.88	38.38
	Green	41.49	37.39	42.51	37.73
	Blue	41.77	38.58	42.72	39.04
Barbara	Red	40.93	37.29	41.83	37.73
	Green	41.15	37.71	42.03	38.21
	Blue	41.11	38.06	42.01	38.47
Lena	Red	44.25	38.78	44.45	38.86
	Green	43.18	39.08	43.61	39.34
	Blue	42.64	37.87	42.92	38.04
House	Red	41.54	37.38	42.72	37.64
	Green	41.65	37.84	42.60	38.02
	Blue	42.26	38.41	43.11	38.56
Airplane (F-16)	Red	43.62	39.92	44.30	38.12
	Green	43.28	39.21	44.06	39.45
	Blue	44.43	39.41	44.86	39.45

Moreover, the color stego image quality is almost similar to that of the grayscale. However, if we use only one or two channel, the quality of the stego image may increase between about 1dB and 3dB.

V. CONCLUSIONS

Commonly, research on data hiding considers two evaluation criteria: capacity and quality. This paper proposes a method that improves the capability of RDE scheme by considering those two parameters. This enhancement is accomplished by modifying the pixel block size and adjusting the transformation method to take the difference between the two closest pixels in the same block. Correspondingly, the new pixel construction scheme is also modified. We can also use non-changeable blocks and each block is embedded once again using LSB replacement method. This work is flexible to be conducted in different sizes of blocks where not only 2×2 or 4×4 but also other blocks sizes.

Based on grayscale and RGB images, the experimental results show that this method is able to enhance the capacity to be more than twice the previous works in a well-maintained quality. Note that, the bigger the size of the pixels block, the greater the capacity. Correspondingly, the quality slightly decreases, and vice versa. The overflow and underflow problems are considered in this method; so, the embedding does not cause much noise in the pixels values.

The analysis proves that color images are more useful than grayscale ones. It is because its capacity is three times bigger than the grayscale; whereas, the achieved quality is almost the same. In facts, color stego image quality is better than that of the grayscale by around 1dB-3dB if we use only one or two channels. It is noted that the color intensity in each RGB channels can affect the amount of payload in the same channel. However, this method is able to overcome this problem.

For the future work, we aim to further enhance the hiding efficiency of the proposed method. As a straightforward idea, it is better to investigate the first pixel in each block since they are not used to be embedded. Moreover, the trade-off between capacity and distortion when extending the block size needs to be investigated.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Institut Teknologi Sepuluh Nopember (ITS) for their kind support. We also thank anyone who has helped us and could not be mentioned here.

REFERENCES

[1] L. Bao, and Y. Zhou, "Image encryption: Generating visually meaningful encrypted images," *IEEE Spectrum*, vol. 324, pp. 197-207, 2015.

[2] D. Hou, H. Wang, W. Zhang, and N. Yu, "Reversible data hiding in JPEG image based on DCT frequency and block selection," *Signal Processing*, vol. 148, pp. 41-47, 2018.

[3] M. Holil, and T. Ahmad, "Secret data hiding by optimizing general smoothness difference expansion-based method," *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 2, pp. 155-163, 2015.

[4] M. H. A. Al-Hooti, S. Djanali, and T. Ahmad, "Audio data hiding based on sample value modification using modulus function," *Journal of Information Processing Systems*, vol. 12, no. 3, pp. 525-537, 2016.

[5] C. M. Wang, N. I. Wu, C. S. Tsai, and M. S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *Journal of Systems and Software*, vol. 81, no. 1, pp. 150-158, 2008.

[6] S. Marwan, A. Shawish, and K. Nagaty, "DNA-based cryptographic methods for data hiding in DNA media," *Biosystems*, vol. 150, pp. 110-118, 2016.

[7] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences*, vol. 279, pp. 251-272, 2014.

[8] D. Bouslimi, G. Coatrieux, M. Cozic, and C. Roux, "Data hiding in encrypted images based on predefined watermark embedding before encryption process," *Signal Processing: Image Communication*, vol. 47, pp. 263-270, 2016.

[9] Z. L. Liu, and C. M. Pun, "Reversible data-hiding in encrypted images by redundant space transfer," *Information Sciences*, vol. 433-434, pp. 188-203, 2018.

[10] M. H. A. Al-Huti, T. Ahmad, and S. Djanali, "Increasing the capacity of the secret data using DEpixels blocks and adjusted RDE-based on grayscale images," in *International Conference on Information & Communication Technology and Systems (ICTS)*, pp. 225-230, 2015.

[11] Y. Yang, M. Lei, J. Wang, and B. Liu, "A SVM based text steganalysis algorithm for spacing coding," *China Communications*, vol. 11, no. 13, pp. 108-113, 2014

[12] Y. Liu, L. Ju, M. Hu, H. Zhao, S. Jia, and Z. Jia, "A new data hiding method for H. 264 based on secret sharing," *Neurocomputing*, vol. 188, pp. 113-119, 2016.

[13] Y. Liu, L. Chen, M. Hu, Z. Jia, S. Jia, and H. Zhao, "A reversible data hiding method for H. 264 with Shamirs (t, n)-threshold secret sharing," *Neurocomputing*, vol. 188, pp. 63-70, 2016.

[14] H. Chen, J. Ni, W. Hong, and T. S. Chen, "Reversible data hiding with contrast enhancement using adaptive histogram shifting and pixel value ordering," *Signal Processing: Image Communication*, vol. 46, pp. 1-16, 2016.

[15] S. Weng, J. S. Pan, and J. Deng, "Invariability of remainder based reversible watermarking," *Journal of Network Intelligence*, vol. 1, pp. 16-22, 2016.

[16] Y. Qiu, H. He, Z. Qian, S. Li, and X. Zhang, "Lossless data hiding in JPEG bit stream using alternative embedding," *Journal of Visual Communication and Image Representation*, vol. 52, pp. 86-91, 2018.

[17] M. B. Andra, T. Ahmad, and T. Usagawa, "Medical record protection with improved GRDE data hiding method on audio files," *Engineering Letters*, vol. 25, no. 2, pp. 112-124, 2017.

[18] P-C. Chen, Y-S. Chen, and W-H. Hsu, "A communication system model for digital image watermarking problems," in *International Conference on Information Systems Analysis and Synthesis*, pp. 29-35, 2007.

[19] G. R. Tadiparthi and T. Sueyoshi, "StegAnim-A novel information hiding technique using animations," *Engineering Letters*, vol. 13, no. 4, pp. 225-235, 2006.

[20] P. R. Anurenjan and S. SanthoshKumar, "A novel steganographic system using continued fraction," in *International Conference on Communications and Signal Processing (ICCSP)*, pp. 23-27, 2011.

[21] W. Zhao, Z. Jie, L. Xin, and W. Qiaoyan, "Data embedding based on pixel value differencing and modulus function using indeterminate equation," *The Journal of China Universities of Posts and Telecommunications*, vol. 22, pp. 95-100, 2015.

[22] C. C. Chang, P. Y. Pai, C. M. Yeh, and Y. K. Chan, "A high payload frequency-based reversible image hiding method," *Information Sciences*, vol. 180, pp. 2286-2298, 2010.

[23] D. C. Lou, M. C. Hu, and J. L. Liu, "Multiple layer data hiding scheme for medical images," *Computer Standards & Interfaces*, vol. 31, pp. 329-335, 2009.

[24] S. A. El_Rahman, "A comparative analysis of image steganography based on DCT algorithm and steganography tool to hide nuclear reactors confidential information," *Computers & Electrical Engineering*, vol. 70, 2018.

[25] F. Y. Shih and X. Zhong, "High-capacity multiple regions of interest watermarking for medical images," *Information Sciences*, vol. 367, pp. 648-659, 2016.

[26] H. Al-Dmour and A. Al-Ani, "Quality optimized medical image information hiding algorithm that employs edge detection and data coding," *Computer Methods and Programs in Biomedicine*, vol. 127, pp. 24-43, 2016.

[27] D. Cavagnino, M. Lucenteforte, and M. Grangetto, "High capacity reversible data hiding and content protection for radiographic images," *Signal Processing*, vol. 117, pp. 258-269, 2015.

[28] S. W. Jung, "Adaptive post-filtering of JPEG compressed images considering compressed domain lossless data hiding," *Information Sciences*, vol. 281, pp. 355-364, 2014.

[29] C. C. Chang, Y. P. Pai, C. M. Yeh, and Y. K. Chan, "Reversible data embedding using a difference expansion" *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, pp. 890-896, 2003.

[30] A. M. Alattar, "Reversible watermark using difference expansion of quads," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'04)*, p. III-377, 2004.

[31] C. L. Liu, D. C. Lou, and C. C. Lee, "Reversible data embedding using reduced difference expansion," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 433-436, 2007.

[32] T. Ahmad, M. Holil, W. Wibisono, and I. R. Muslim, "An improved Quad and RDE-based medical data hiding method," in *IEEE International Conference on Computational Intelligence and Cybernetics (CYBERNETICSCOM)*, pp. 141-145, 2013.

[33] J. M. Blackledge and A. I Al-Rawi, "Steganography using stochastic diffusion for the covert communication of digital images," *International Journal of Applied Mathematics*, vol. 41, no. 4, 2011.

[34] C. Arun and S. Murugan, "Design of image steganography using LSB XOR substitution method," in *IEEE International Conference on Communication and Signal Processing*, pp. 0674-0677, 2017.

[35] M. H. A. Al-Hooti, T. Ahmad, S. Djanali, "Developing audio data hiding scheme using random sample bits with logical operators," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, 2019.

[36] H. C. Huang, F. C. Chang, and Y. Y. Lu, "Multi-bit reversible data hiding with prediction and difference alteration," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, pp. 435-444, 2017.

[37] Q. Y. Zhang, Q. Y. Dou, R. H. Dong, and Y. Yan, "Robust reversible data hiding algorithm for color image based on 2d-dct," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, pp. 392-403, 2017.

[38] "Partners Infectious Disesase Images," 2017. [on-line]. Available: http://www.idimages.org/images/.

[39] The USC-SIPI Image Database. 2017. [on-line]. Available: http://sipi.usc.edu/database/.

[40] D. Vaishnavi and T. S. Subashini, "Robust and invisible image watermarking in RGB color space using SVD," *Procedia Computer Science*, vol. 46, pp. 1770-1777, 2015.

[41] H. T. Wu, S. Tang, J. Huang, and Y. Q.Shi, "A novel reversible data hiding method with image contrast enhancement," *Signal Processing: Image Communication*, vol. 62, pp. 64-73, 2018.