Received: June 12, 2019


International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Improving the Quality of Stego Image Using Prediction Error and Histogram Modification

**Chaidir Chalaf Islamy[1]*** **Tohari Ahmad[1]**

*Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia*
* Corresponding author's Email: chaidir31@gmail.com tohari@if.its.ac.id

**Abstract:** The transmitted information on the internet are prone to security concerns. To overcome this issue, some security methods have been developed in the last few decades. One of the reliable techniques to secure data is called data hiding. This method can be used in various multimedia data, like image, video, text, and audio. Histogram shifting (HS) is a reliable data hiding method used in digital images. It can preserve the similarity of the stego and original image; although, the amount of the protected data depends on the frequency of the most amount pixel in the original image. To solve this problem, we combine HS with prediction error (PE). By using the histogram generated by PE, the size of data that can be embedded increases. In the embedding phase, we do not shift the histogram, different from the existing methods. It is to maintain its original form as high as possible. Moreover, each PE value can contain more data. Based on the experimental result, the average increase in PSNR is 24.8075 dB for general images and 21.5625dB for medical images; while the average improvement of SSIM is 0.026823 and 0.022863 for general and medical images, respectively.

**Keywords:** Data hiding, Prediction error, Histogram shifting, Data protection.

## 1. Introduction

Multimedia data that circulates on the internet have become the target of copyright attacks. This problem is increasing dramatically because of the freedom of data exchange that occurs in the internet network. Two data security methods can be relied on for several years: cryptography and steganography. Cryptography aims to protect data by encrypting data before sending it. The existence of data that can be used to determine suspicion and risk of attack. The vulnerability of encrypted data to be attacked means that the security relies on the strength of the cryptography method. Unlike cryptography, steganography hides data into the media or covers in the form of text, images, video, and audio. We can conceal information data into the cover media without arousing suspicion because the hidden data are only known by the sender and recipient. The steganography method is often applied in the critical field that requires high confidentiality, such as crime, medic and copyright protection.

In digital image steganography, data are hidden into the cover image by minimizing the level of distortion. The size of embedded data in the cover image affects the level of deformation of the stego image. In general, the quality of the stego image is inversely proportional to the amount of data hidden in the cover [1]. The characteristics of a suitable steganography method are represented by its capability to balance between the amount of data that can be hidden and the quality of stego images [2]. During the data extraction process, not all cover images must be returned to their original state. But in some conditions, the cover image must be returned to the state before the data are embedded. Therefore the method is called reversible data hiding (RDH).

Many techniques have been developed in the last few decades. Broadly speaking, the method of steganography on spatial domains can be divided into compress-and-append, expansion (EB) and

histogram shifting (HS) [3]. The compress-and-append method is widely used in the early days of the RDH method. The EB-based approach is first proposed by Tian [4], the technique is called difference expansion (DE). The DE method utilizes differences from adjacent pixels. Prediction error expansion (PEE) is another method based on the EB approach; it is developed by Thodi and Rodriguez [5]. The HS-based method is introduced by Ni et al. [6]. In HS, data are inserted in the free space by modifying histogram. Empty space is created by shifting pixels on the image histogram. The HS scheme produces images with a low distortion rate compared to the EB approach. The disadvantage of the HS method is the capacity of data that can be embedded is relatively smaller than the EB scheme.

A reversible data hiding scheme using PE and HS is presented in this research. The purpose of this research is to enhance the quality of stego images. We use histogram generated by PE to embed data. The differences between existing methods and the proposed method are, first, we do not use the shifting process before inserting data. It is to reduce unnecessary modification in the histogram. Because of it, secret data are not embedded in the empty space resulted from the shifting process. To restore the embedded data, the location map is generated during the embedding process. Moreover, to further minimize the modification of histogram, we categorize secret bits to be embedded in selected PE value. This not only reduces the change but also improves the embedding room.

The next section of this paper is organized as follows. Section 2 presents the related works. The proposed method expounded in Section 3. In Section 4, the experimental results are provided, and Section 5 is the conclusion of this work.

## 2. Related works

The idea of hiding data by utilizing histogram modification is first introduced by Ni et al. [7]. The idea of this method is to use the most frequent color in an image. Generally, data embedding is carried out in three steps. The first step, find the most frequent pixel or peak pixel and least frequent pixel or minimum pixel in cover image. Then provide a place for the embedded data by shifting pixels between peak pixel and minimum pixel. Data embedding is done by modifying the peak pixels, which are moved to the space provided after the shifting process. The pixel shifting process can be illustrated in Eq. (1) while the process of embedding data can be described in Eq. (2). In those equations, the peak pixel is $X$, and the minimum pixel is $Z$; $I$ is the pixel before being shifted; $I'$ is the pixel that has been shifted, $i$ and $j$ are the pixel location. Then $s(n)$ is secret bits where $n$ is an index of obscure bits.

$$I'_{ij} = \begin{cases} I_{ij} + 1 & if\ X + 1 \le I_{ij} \le Z - 1 \\ & and\ X < Z \\ I_{ij} - 1 & if\ Z + 1 \le I_{ij} \le X - 1 \\ & and\ X > Z \end{cases} \quad (1)$$

$$I''_{ij} = \begin{cases} I'_{ij} + 1\ if\ I'_{ij} = X\ and\ s(n) = 1, X < Z \\ I'_{ij} - 1\ if\ I'_{ij} = X\ and\ s(n) = 1, X > Z \\ I'_{ij} \qquad if\ I'_{ij} = X\ and\ s(n) = 0 \end{cases} \quad (2)$$

We can see from those two equations above, the shortcoming of this scheme is the total capacity relies on the frequency of the peak pixel ($X$). Therefore, if the rate of $X$ is low, it will impact the hiding capacity.

There are other reliable schemes, such as interpolation [7–9]. The essence of the interpolation scheme is to utilize the enlargement of the image resolution. Other than the spatial domain, data embedding can also be carried out in the transform domain [10–12].

Hong et al. [13] combine HS schemes with EB. They use prediction error (PE) to produce a histogram, where the data insertion process is carried out. The resulting stego image has better quality than the EB and HS schemes. The main weakness in the HS scheme can be eliminated because the PE histogram is able to produce a histogram with a higher number of frequencies. This scheme is further developed in [14–16].

Rad et al. [3] then produce a checkerboard predictor (CBP). It is able to generate more accurate PE histogram than MED. Despite the higher accuracy on CBP, the frequency of non-smooth pixel (pixel other than 0) is not as much as MED. Therefore, if we want to utilize non-smooth pixel on the histogram, the hiding capacity is lower than MED.

Yi et al. [17] propose a data hiding scheme called block-level prediction-error expansion (BLPEE), which is capable of enhancing the capacity of data that can be accommodated whose impact on the quality can still be tolerated. The predictor predicts pixels in 2×2 block. The process of predicting can be described in Eqs. (3) and (4), where $I$ is the pixel that will be predicted, $I_r$, $I_c$and $I_d$ are pixel positioned in the same row, column and

diagonal side of $I$, while $C_r$, $C_c$ and $C_d$ are the weight coefficients. In [17] weight coefficients are set $C_r$= $C_c$=0.4 and $C_d$=0.2 The embedding method can provide a balance between hiding capacity and distortion level. However, the level of distortion produced in the stego image still needs improvement.

$$\hat{I}_{i,j} = C_r I_r + C_c I_c + C_d I_d \tag{3}$$

$$R'_{i,j} = \begin{cases} R_{i,j} - 1 & if\ R_{i,j} < T_l \\ R_{i,j} - s(n) & if\ R_{i,j} = T_l \\ R_{i,j} + s(n) & if\ R_{i,j} = T_r \\ R_{i,j} + 1 & if\ R_{i,j} > T_r \\ R'_{i,j} & otherwise \end{cases} \tag{4}$$

In this formula, $R'_{i,j}$ is the PE after hiding data, $T_l$ and $T_r$ are the capacity parameters and defined by Eq. (5) and Eq. (6) where $h$ is the number of occurrences when prediction-error values in the sequence are equal to $R_{i,j}$.

$$T_l = \min\{\arg\max\{h(R_{i,j})\}\} \tag{5}$$

$$T_r = \max\{\arg\max\{h(R_{i,j})\}\} \tag{6}$$

To expand the hidden data space, Kumar and Agrawal [18] develop a reversible data hiding method using PEE. They use the adjacent PE value to hide data. They predict the value of even columns using odd columns; then secret bits are embedded on the even. Their method can provide a large capacity due to the use of even columns on the cover image. While this method [18] can offer a massive amount of embedding capacity, the quality of the stego image is barely tolerable. The prediction technique is denoted in Eq. (7), where $j_e$ is the position of even column. To hide the secret data, Eq. (8) is utilized. In this equation, $\hat{I}$ is the resulting error value of the predictor.

$$\hat{I}_{i,j_e} = I_{i,j_e-1} + I_{i,j_e+1}/2 \tag{7}$$

$$R'_{i,j_e} = (R_{i,j_e} \times 2) + s(n) \tag{8}$$

## 3. Proposed method

The process of embedding data is performed in two stages: histogram generation and histogram manipulation. The flow of the data embedding process can be seen in Fig. 1. Before hiding the secret bits, the predicted image is calculated by using predictor. Then, from that operation, the PE value is produced to form the PE histogram, where we embed the secret bits.

Before embedding data on the cover image, the PE histogram is formed using MED. We use it because the produced PE histogram has an even distribution. It is suitable for the method that we propose because we use more than one PE value. We also use PE values other than 0 because it is a smooth value in the image and can affect the stego image. Illustration of predictions can be seen in Fig. 2. The prediction error is calculated using Eq. (9), where $I$ is the original pixel, $\hat{I}$ is the resulting error value used to be the embedding space, and $R$ is the PE value. The PE value is calculated using Eq. (10).

$$\hat{I}_{i,j} = \begin{cases} min(I_{i,j-1}, I_{i-1,j}) & if\ I_{i-1,j-1} \geq \\ & max(I_{i,j-1}, I_{i-1,j}) \\ max(I_{i,j-1}, I_{i-1,j}) & if\ I_{i-1,j-1} \leq \\ & min(I_{i,j-1}, I_{i-1,j}) \\ I_{i,j-1} + I_{i-1,j} - I_{i-1,j-1} & otherwise \end{cases} \tag{9}$$

$$R_{i,j} = I_{i,j} - \hat{I}_{i,j} \tag{10}$$

### 3.1 Histogram modification

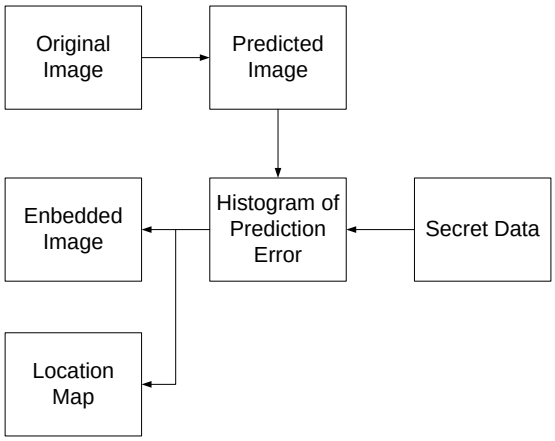


Figure. 1 The flow of data embedding process

| | | | |
|---|---|---|---|
| | $I_{(i-1,j-1)}$ | $I_{(i-1,j)}$ | |
| | $I_{(i,j-1)}$ | $I$ | |
| | | | |

Figure. 2 Illustration of predictions on pixel $I$

To minimize the modification of the histogram, we exclude the shifting process. We do not place secret data in the location provided by shifting histogram; instead, the hidden data are placed into the neighbour's PE value, and the location of those secret data are recorded in the location map. The capacity of embedded data increases if each PE value can hold more than one bit of data. Obscure bits are scanned and each of the two adjacent data bits is checked first and placed in 4 different categories. Each of these categories is embedded in different PE values. Because secret data are in the form of binary numbers, those four pairs of groups are (1, 1), (0, 0), (0, 1) and (1, 0).

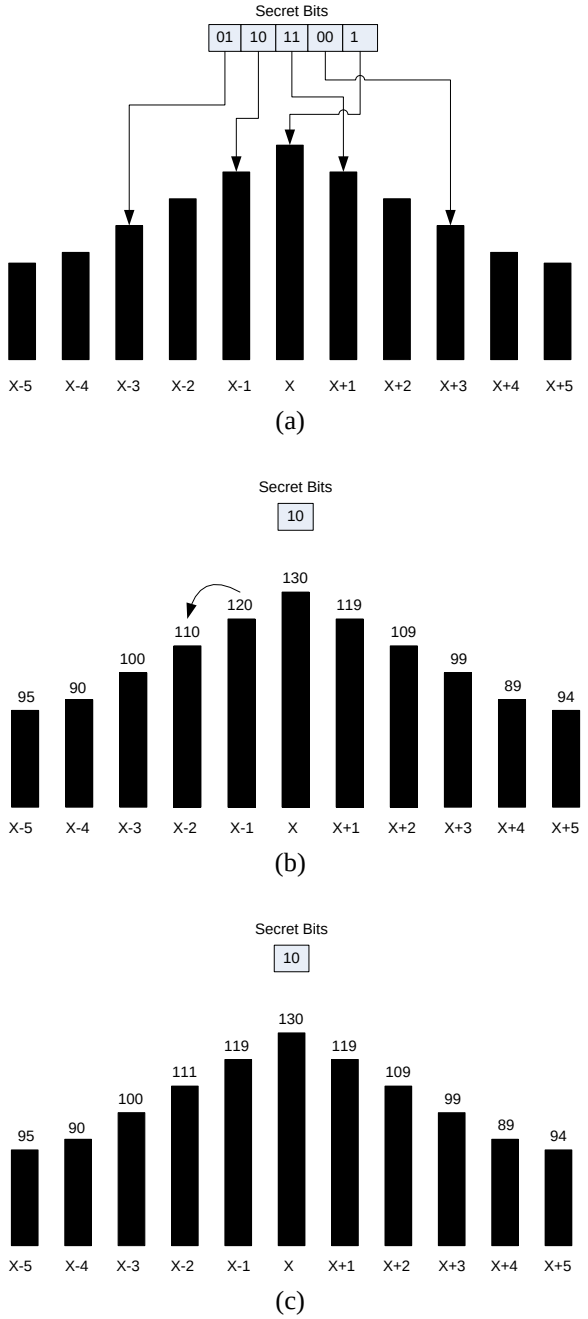


Figure. 3 (a) location of secret bits on the histogram, (b) example of embedding bits (0,1), and (c) Histogram after embedding bits (0,1)

The two adjacent bits are scanned; if they are (1, 1), then the value of PE $X+3$ is added by 1. PE value of $X+1$ is also added by one if two adjacent bits are (0, 0). If two adjacent bits are (0, 1) then the PE value $X-3$ is reduced by 1. If (1, 0) are two adjacent bits, then the reduced PE value is $X-1$ value. The next two bits are checked according to the conditions we have mentioned. The process is repeated until all bits of data have been embedded. If the amount of secret data is odd, then the last bit is embedded in the PE value of $X$. The embedding process can be described in Eq. (11). The location of embedded secret bits can be observed in Fig. 3 (a). An example of embedding of bits (0,1) is illustrated in Fig. 3 (b) and the result of the embedded histogram also illustrated in Fig. 3 (c). During the data embedding process, the location of the predicted value is stored for the secret data extraction process. The position of the $x$-axis is stored at $x(n)$ and the $y$-axis in $y(n)$, with $n$ being the index of the confidential data. The locations of the PE value are stored by using Eqs. (12) and (13).

$$R'_{i,j}=\begin{cases} R_{i,j}+1 & \begin{array}{r} if\ R_{i,j}=X+3 \\ and\ s(n)=1 \\ and\ s(n+1)=1 \\ or \\ R_{i,j}=X+1 \\ and\ s(n)=0 \\ and\ s(n+1)=0 \end{array} \\ R_{i,j}-1 & \begin{array}{r} if\ R_{i,j}=X-3 \\ and\ s(n)=0 \\ and\ s(n+1)=1 \\ or \\ R_{i,j}=X-1 \\ and\ s(n)=1 \\ and\ s(n+1)=0 \end{array} \\ R_{i,j} & otherwise \end{cases} \tag{11}$$

$$x(n)=i \tag{12}$$

$$y(n)=j \tag{13}$$

For secret data that have an odd number of bits, if the last hidden bit is 0, the PE value is reduced by one. If the last bit is 1, then the amount of PE $X$ is added by one. The process of embedding the last bit of secret data can be described by using Eq. (14). The location of PE value is also stored using Eq. (12) and Eq. (13).

$$R'_{i,j} = \begin{cases} R_{i,j} + 1 \;\; if \; R_{i,j} = X \\ \qquad and \; s(n) = 1 \\ R_{i,j} - 0 \;\; if \; R_{i,j} = X \\ \qquad and \; s(n) = 0 \end{cases} \tag{14}$$

After modification of the PE histogram, PE values are modified by using Eq. (15).

$$I'_{i,j} = \hat{I}_{i,j} - \; R'_{i,j} \tag{15}$$

### 3.2 Extraction and image recovery

The first step in the process of extracting data is changing the stego image to the PE values that contain secret data by using Eq. (16).

$$R'_{i,j} = \hat{I}_{i,j} - I'_{i,j} \tag{16}$$

It needs to scan the PE values. If we find the PE value $X \; + \; 4$ and the location of the PE value match $x \; (n)$ and $y \; (n)$, then return the PE value to the state before the modification by reducing the PE value by 1. The extracted secret bits $s \; (n)$ and $s( \; n \; + \; 1)$ is (1,1). The extracted secret bits depend on the found PE value. If it is $X \; + \; 2$, then the extracted secret bits are (0,0); if it is $X - 2$ are (1, 0) and $X - 4$ are (0, 1). The description of the step can be seen in Eqs. (17), (18) and (19).

If the total amount of bits are odd, the location of the last bit for bit 0 is at $X - 1$ or $X \; + \; 1$ for bit 1. To return the PE value to the original condition, use Eq. (20), while secret bit extraction is described on Eq. (21). After that, the cover image is obtained using Eq. (22).

To emphasize the position of the proposed method and related method, we compare our method with other relevant ones in Table 1.

Table 1. Comparison of the proposed method with related methods

| **Image** | **Proposed Method** | **Yi et al. [17]** | **Kumar and Agrawal** [18] |
|---|---|---|---|
| Predictor | MED | BLPEE | Prediction employed by using even columns and adjacent pixels |
| Number of bits that can be embedded in a PE value | 2 | 1 | 1 |
| Embedding space location | $X + 3$, $X + 1$, $X - 1$, $X - 3$ | Block size of 2×2 | Predicted even columns |

$$R_{i,j} = \begin{cases} R'_{i,j} - 1 \;\; if \; R'_{i,j} = X + 4 \; or \\ \qquad R'_{i,j} = X + 2 \\ R'_{i,j} + 1 \;\; if \; R'_{i,j} = X - 4 \; or \\ \qquad R'_{i,j} = X - 2 \end{cases} \tag{17}$$

$$s(n) = \begin{cases} 1 \;\; if \; R'_{i,j} = X + 2 \; and \; x(n) = i \\ \qquad and \; y(n) = j \\ \qquad or \\ \qquad R'_{i,j} = X - 2 \; and \; x(n) = i \\ \qquad and \; y(n) = j \\ 0 \;\; if \; R'_{i,j} = X + 4 \; and \; x(n) = i \\ \qquad and \; y(n) = j \\ \qquad or \\ \qquad R'_{i,j} = X - 4 \; and \; x(n) = i \\ \qquad and \; y(n) = j \end{cases} \tag{18}$$

$$s(n+1) = \begin{cases} 1 \;\; if \; R'_{i,j} = X + 2 \\ \quad and \; x(n) = i \\ \quad and \; y(n) = j \\ \quad or \\ \quad R'_{i,j} = X - 4 \\ \quad and \; x(n) = i \\ \quad and \; y(n) = j \\ 0 \;\; if \; R'_{i,j} = X + 4 \\ \quad and \; x(n) = i \\ \quad and \; y(n) = j \\ \quad or \\ \quad R'_{i,j} = X - 2 \\ \quad and \; x(n) = i \\ \quad and \; y(n) = j \end{cases} \tag{19}$$

$$R_{i,j} = \begin{cases} R'_{i,j} - 1 & if \; R'_{i,j} = X + 1 \\ R'_{i,j} + 1 & if \; R'_{i,j} = X - 1 \end{cases} \tag{20}$$

$$s(n) = \begin{cases} 1 \ if \ R'_{i,j} = X + 1 \\ \quad and \ x(n) = i \\ \quad and \ y(n) = j \\ 0 \ if \ R'_{i,j} = X - 1 \\ \quad and \ x(n) = i \\ \quad and \ y(n) = j \end{cases} \quad (21)$$

$$I_{i,j} = \hat{I}_{i,j} + R_{i,j} \quad (22)$$

## 4. Results and discussions

After the extraction process, it is important to evaluate the proposed method. We measure the distortion level in the stego image and how many bits can be embedded. We utilize the peak signal-to-noise ratio (PSNR) as the measurement of the distortion level of the stego image. We use PSNR as the benchmark to determine the degradation of the quality of the image after the embedding process. The lower the degradation of PSNR value, the better the quality of the embedded image. Additionally, we use structural similarity index (SSIM) to evaluate the similarity of the stego image. To test the performance of the method, we use ten grayscale images of size 512x512 pixels that we obtain from [19][20]. Those are "Baboon", “Lena”, “Pepper”, “Elaine”, “Boat”, “Abdominal”, "Hand", "Chest", “Head” and “Leg”. In the experiment, two different sizes of secret bits are used in the embedding phase; they are 10 Kb and 20 Kb of data.

The PSNR value is calculated using Eq. (23), where MSE is the mean square error and calculated using Eq. (24); $I_{MAX}$ is the pixel with the highest value, while $L$ and $P$ are the height and width of the image.

$$PSNR = 10log_{10}\frac{(I_{MAX})^2}{MSE} \quad (23)$$

$$MSE = \left(\frac{1}{LP}\right)\sum_{i=1}^{L}\sum_{j=1}^{P}\left(I_{ij} - I'_{i,j}\right)^2 \quad (24)$$

The total data that can be inserted is measured by how many bits per pixel ($BPP$) can be stored in the image. To calculate the $BPP$, the amount of bits that can be hidden inside the stego image is divided by the length and width of the image. Calculation of $BPP$ can be seen in Eq. (25), where $T$ is the total bits that can be embedded.

$$BPP = \frac{T}{LP} \quad (25)$$

In order to find the position of the proposed method, we use Yi et al. [17] and Kumar and Agrawal [18] approach as a comparison. We implement both methods and measure the performance and use the same secret messages and test images for the entire experiment.

The result of the similarity comparison is presented in Tables 2 and 3. As shown in those tables, the proposed scheme has higher PSNR and SSIM than [17] and [18]. The improvement in PSNR and SSIM is resulted by the proposed system which does not shift the histogram in the embedding phase. Also, it can embed more than one bit in each PE value; therefore, it also decreases the number of PE value that has to be modified. The visualization of PSNR average of general and medical images are depicted in Fig. 4 and Fig. 5 respectively. From those figures, we find that the proposed scheme achieves higher PSNR average than [17] and [18] in each secret data for both general and medical images. Despite [17] has lower PSNR average than our scheme, it has smaller PSNR decline than both the proposed method and [18].

To observe the changes produced by the stego image, Fig. 6 presents the original Pepper image and original Hand medical image along with the stego image that has been inserted 20 Kb of data. We can see in Fig. 6 the difference between the original and the embedded images is relatively small. It means that our scheme is capable of maintaining the quality of the produced stego image.

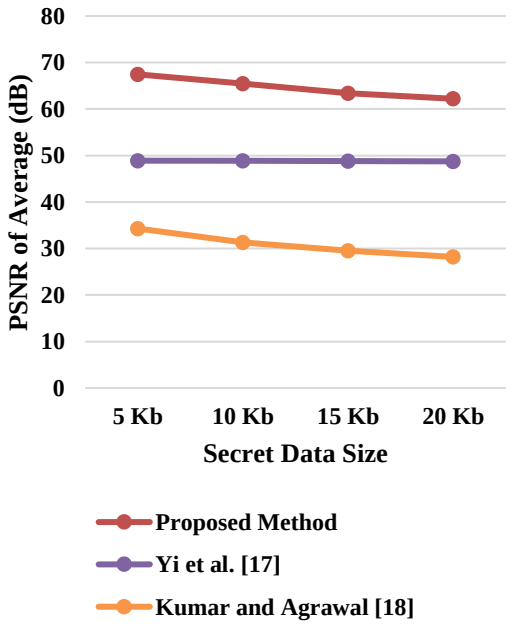


Figure. 4 The overall average of PSNR for general images

Table 2. Comparison of similarity on general grayscale images

| Image | Proposed method | | | Yi et al. [17] | | | Kumar and Agrawal [18] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Payload (Kb) | PSNR (dB) | SSIM | Payload (Kb) | PSNR (dB) | SSIM | Payload (Kb) | PSNR (dB) | SSIM |
| Baboon | 5 | 68.34 | 0.9999 | 5 | 48.48 | 0.9979 | 5 | 33.94 | 0.9941 |
| | 10 | 65.33 | 0.9999 | 10 | 48.44 | 0.9979 | 10 | 31.50 | 0.9883 |
| | 15 | 63.57 | 0.9999 | 15 | 48.39 | 0.9978 | 15 | 30.18 | 0.9818 |
| | 20 | 62.32 | 0.9998 | 20 | 48,35 | 0.9978 | 20 | 28.94 | 0.9756 |
| Lena | 5 | 67.26 | 0.9999 | 5 | 49.29 | 0.9947 | 5 | 36.97 | 0.9761 |
| | 10 | 64.98 | 0.9998 | 10 | 49.23 | 0.9947 | 10 | 33.91 | 0.9546 |
| | 15 | 62.69 | 0.9997 | 15 | 49.18 | 0.9946 | 15 | 31.96 | 0.9327 |
| | 20 | 61.72 | 0.9996 | 20 | 49.13 | 0.9946 | 20 | 30.42 | 0.9125 |
| Pepper | 5 | 67.66 | 0.9999 | 5 | 49.04 | 0.9945 | 5 | 33.28 | 0.9814 |
| | 10 | 65.40 | 0.9998 | 10 | 48.99 | 0.9944 | 10 | 30.41 | 0.9609 |
| | 15 | 64.13 | 0.9998 | 15 | 48.94 | 0.9944 | 15 | 28.73 | 0.9395 |
| | 20 | 62.77 | 0.9997 | 20 | 48.89 | 0.9943 | 20 | 27.73 | 0.9201 |
| Elaine | 5 | 67.00 | 0.9999 | 5 | 48.83 | 0.9953 | 5 | 32.75 | 0.9738 |
| | 10 | 65.78 | 0.9998 | 10 | 48.78 | 0.9952 | 10 | 29.80 | 0.9497 |
| | 15 | 63.53 | 0.9998 | 15 | 48.74 | 0.9952 | 15 | 28.14 | 0.9264 |
| | 20 | 62.22 | 0.9997 | 20 | 48.69 | 0.9952 | 20 | 27.04 | 0.9039 |
| Boat | 5 | 67,03 | 0.9999 | 5 | 48.84 | 0.9957 | 5 | 34.55 | 0.9733 |
| | 10 | 65,79 | 0.9998 | 10 | 48.80 | 0.9957 | 10 | 30.99 | 0.9479 |
| | 15 | 63.26 | 0.9997 | 15 | 48.75 | 0.9956 | 15 | 28.68 | 0.9216 |
| | 20 | 62.02 | 0.9997 | 20 | 48.70 | 0.9956 | 20 | 26.90 | 0.8938 |

Table 3. Comparison of similarity on medical grayscale images

| Image | Proposed method | | | Yi et al. [17] | | | Kumar and Agrawal [18] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Payload (Kb) | PSNR (dB) | SSIM | Payload (Kb) | PSNR (dB) | SSIM | Payload (Kb) | PSNR (dB) | SSIM |
| Abdominal | 4 | 69.75 | 0.9999 | 4 | 51.76 | 0.9954 | 4 | 43.23 | 0.9839 |
| | 8 | 66.60 | 0.9999 | 8 | 51.69 | 0.9951 | 8 | 38.78 | 0.9685 |
| | 12 | 64.76 | 0.9998 | 12 | 51.62 | 0.9947 | 12 | 36.60 | 0.9522 |
| | 16 | 62.47 | 0.9998 | 16 | 51.54 | 0.9943 | 16 | 35.30 | 0.9362 |
| Hand | 4 | 69.31 | 0.9999 | 4 | 51.81 | 0.9972 | 4 | 41.82 | 0.9878 |
| | 8 | 66.30 | 0.9999 | 8 | 51.73 | 0.9972 | 8 | 37.70 | 0.9777 |
| | 12 | 64.53 | 0.9998 | 12 | 51.66 | 0.9971 | 12 | 36.63 | 0.9612 |
| | 16 | 63.29 | 0.9997 | 16 | 51.59 | 0.9971 | 16 | 35.78 | 0.9444 |
| Chest | 4 | 69.42 | 0.9999 | 4 | 52.81 | 0.9959 | 4 | 40.83 | 0.9876 |
| | 8 | 66.41 | 0.9999 | 8 | 52.72 | 0.9955 | 8 | 37.00 | 0.9662 |
| | 12 | 64.72 | 0.9998 | 12 | 52.62 | 0.9950 | 12 | 33.85 | 0.9422 |
| | 16 | 63.47 | 0.9998 | 16 | 52.53 | 0.9946 | 16 | 31.81 | 0.9176 |
| Head | 4 | 69.83 | 0.9999 | 4 | 51.94 | 0.9955 | 4 | 43.26 | 0.9801 |
| | 8 | 66.74 | 0.9999 | 8 | 51.86 | 0.9950 | 8 | 38.66 | 0.9640 |
| | 12 | 64.96 | 0.9999 | 12 | 51.79 | 0.9946 | 12 | 36.48 | 0.9446 |
| | 16 | 63.68 | 0.9998 | 16 | 51.71 | 0.9942 | 16 | 35.14 | 0.9253 |
| Leg | 4 | 69.33 | 0.9999 | 4 | 51.81 | 0.9964 | 4 | 37.03 | 0.9834 |
| | 8 | 66.31 | 0.9999 | 8 | 51.74 | 0.9963 | 8 | 34.64 | 0.9662 |
| | 12 | 64.55 | 0.9998 | 12 | 51.66 | 0.9963 | 12 | 32.88 | 0.9477 |
| | 16 | 63.40 | 0.9998 | 16 | 51.59 | 0.9962 | 16 | 31.56 | 0.9291 |

The capacity comparison can be found in Tables 4 and 5. Here, the proposed scheme has less capacity than [17] and [18]. In [17], the embedding method can be performed in multiple round, so if in one round there are still secret bits left, the next round is performed until all obscure bits are embedded. Therefore, in Tables 4 and 5 we do not include the maximum capacity of [17], because of its adaptive nature. While in [18], the predictor utilizes even columns in the image, so the number of BPP is always 0.4980. Also, the frequency of PE value of $X-3$, $X-1$, $X+1$ and $X+1$ are lower than the number of total PE value on even column

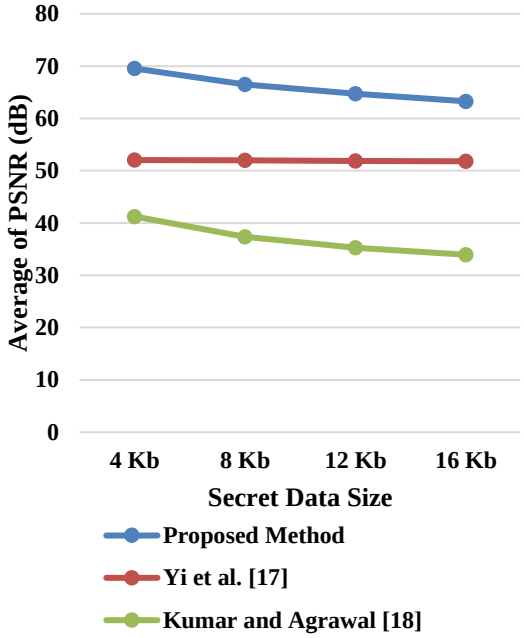


Figure. 5 The overall average of PSNR for medical images

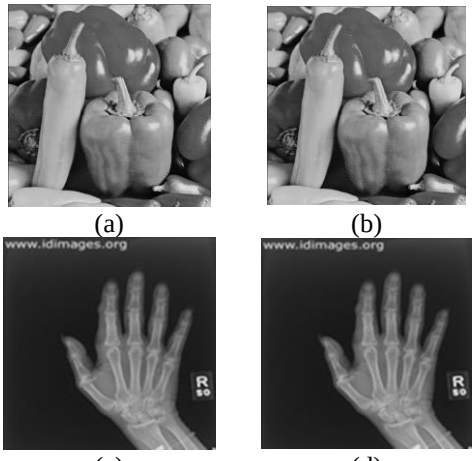


Figure. 6 Examples of the cover images [19, 20] and the stego image: (a) pepper image before embedding data, (b) pepper image after inserting 20 Kb of data, (c) hand image before inserting data, and (d) hand image after inserting 20 Kb of data

used in [18]. That is why more data can be embedded in [18]. As a result, our scheme is more suitable to hide low to medium secret data size.

## 5. Conclusion

We develop an embedding strategy based on PE and HS approaches. While in the embedding phase, we eliminate the shifting process to reduce the modification on the histogram and the secret data are categorized and embedded on the specified PE value. Since the elimination of the shifting process can make the process to obtain hidden data challenging to do, to get them, we use a location map that created during the embedding process. The distribution of categorized secret bits also reduces the number of PE value that has to be utilized to contain the data. From the outcome of the experiment, we can conclude that our scheme can provide better similarities in all test conditions. That is proven by the better quality measured by PSNR and SSIM. The average improvement of PSNR for general images is 24.8075 dB and for medical images is 21.5625 while the SSIM average increase for general images is 0.026823 and 0.022863 for medical images.

There is a room for improvement for the proposed scheme to expand further the size of data that can be embedded. It can be done by extending the number of bits that can be contained in a single PE value.

Table 4. Comparison of bit per pixel value on general grayscale images

| Image | Bit per pixel (BPP) | |
|---|---|---|
| | **Proposed method** | **Kumar and Agrawal [18]** |
| Baboon | 0.2062 | 0.4980 |
| Lena | 0.4351 | 0.4980 |
| Pepper | 0.3935 | 0.4980 |
| Elaine | 0.2689 | 0.4980 |
| Boat | 0.3254 | 0.4980 |

Table 5. Comparison of bit per pixel value on medical grayscale images

| Image | Bit per pixel (BPP) | |
|---|---|---|
| | **Proposed method** | **Kumar and Agrawal [18]** |
| Abdominal | 0.0961 | 0.4980 |
| Hand | 0.0744 | 0.4980 |
| Chest | 0.0742 | 0.4980 |
| Head | 0.1165 | 0.4980 |
| Leg | 0.0552 | 0.4980 |

## References


[1] M. Hussain, A. W. A. Wahab, Y. I. Bin Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey", *Signal Process. Image Commun.*, Vol. 65, pp. 46–66, 2018.

[2] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research", *Neurocomputing*, Vol. 335, pp. 299–326, 2019.

[3] R. M. Rad, K. S. Wong, and J. M. Guo, "Reversible data hiding by adaptive group modification on histogram of prediction errors", *Signal Processing*, Vol. 125, pp. 315–328, 2016.

[4] J. Tian, "Reversible Data Embedding Using a Difference Expansion", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 8, pp. 890–896, 2003.

[5] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking", *IEEE Trans. Image Process.*, Vol. 16, No. 3, pp. 721–730, 2007.

[6] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 16, No. 3, pp. 354–362, 2006.

[7] T.-C. Lu, "Interpolation-based hiding scheme using the modulus function and re-encoding strategy", *Signal Processing*, Vol. 142, pp. 244–259, 2018.

[8] C. N. Yang, S. C. Hsu, and C. Kim, "Improving stego image quality in image interpolation based data hiding", *Comput. Stand. Interfaces*, Vol. 50, pp. 209–215, 2017.

[9] A. Benhfid, E. bachir Ameur, and Y. Taouil, "High capacity data hiding methods based on spline interpolation", In: *Proc. of 2016 5th International Conference on Multimedia Computing and Systems*, pp. 157–162, 2016.

[10] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding", *Signal Processing*, Vol. 130, pp. 190–196, 2017.

[11] X. Wu, J. Weng, and W. Q. Yan, "Adopting secret sharing for reversible data hiding in encrypted images", *Signal Processing*, Vol. 143, pp. 269–281, 2018.

[12] C. Kim, D. Shin, L. Leng, and C.-N. Yang, "Separable reversible data hiding in encrypted halftone image", *Displays*, Vol. 55, pp. 71–79, 2018.

[13] W. Hong, T. Chen, and C. Shiu, "The Journal of Systems and Software Reversible data hiding for high quality images using modification of prediction errors", *J. Syst. Softw.*, Vol. 82, No. 11, pp. 1833–1842, 2009.

[14] H. E. Prabowo and T. Ahmad, "Adaptive Pixel Value Grouping for Protecting Secret Data in Public Computer Networks", *J. Commun.*, Vol. 13, No. 6, pp. 325–332, 2018.

[15] H. Chen, J. Ni, W. Hong, and T. Chen, "High-Fidelity Reversible Data Hiding Using Directionally Enclosed Prediction", *IEEE Signal Process. Lett.*, Vol. 24, No. 5, pp. 574–578, 2017.

[16] J. Qin and F. Huang, "Reversible Data Hiding Based on Multiple Two-Dimensional Histograms Modification", *IEEE Signal Process. Lett.*, Vol. 26, No. 6, pp. 843–847, 2019.

[17] S. Yi, Y. Zhou, and Z. Hua, "Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion", *Signal Process. Image Commun.*, Vol. 64, pp. 78–88, 2018.

[18] M. Kumar and S. Agrawal, "Reversible data hiding based on prediction error expansion using adjacent pixels", *Secur. Commun. Networks*, Vol. 9, No. 16, pp. 3703–3712, 2016.

[19] "SIPI Image Database." [Online]. Available: http://sipi.usc.edu/database/database.php?volume=misc. [Accessed: 28-Apr-2019].

[20] "eMicrobes Digital Library." [Online]. Available: https://www.idimages.org/. [Accessed: 28-Apr-2019].