Received: December 19, 2020. Revised: February 5, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancing Quality of the Stego Image by Using Histogram Partition and Prediction Error

Chaidir Chalaf Islamy[1,2] **Tohari Ahmad[1]***

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia*
[2]*Department of Informatics, Universitas 17 Agustus 1945 Surabaya, Indonesia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: The widespread use of the internet to communicate with each other can introduces data security issues. Therefore, a reliable technique to secure information is becoming mandatory. Data hiding is a method to secure data by hiding it in cover media. In this research, we present a secure data hiding method by using images as cover. This research proposed the use of prediction error (PE) of images and utilization of the histogram-based method to conceal data. The histogram-based method can conceal data and still preserve the quality of the image. However, its lower hiding space can be a drawback. To overcome this, instead of using a histogram from the image's pixel, we calculate the error value of images and create a histogram from it. After that, we divide the histogram into partitions, whose peak is as the reference point for the embedding process. We utilize more than one error value in each partition to further increase the embedding space. Based on the experimental result, with the same image dataset, the average of PSNR values improved by 17.94 dB for general images and 14.95 dB for medical images compared to previous methods. These increases are statistically significant for the visual image quality, where the final PSNR values are between 61 and 66 dB, and between 62 and 68 dB for general and medical images, respectively. It is higher than the minimum standard of stego image quality, which is 30 dB.

Keywords: Data hiding, Histogram partition, Prediction error, Information security, Stego image.

1. Introduction

These days, the internet has become a vital part of human activities, from communicating with each other to transferring essential data. The ever increasing data that constantly circulated on the internet are prone to security issues. Without some form of information security technique, important information can be tampered with or stolen.

Generally, there are two primary techniques to secure information, cryptography and information hiding [1]. The major goal of those techniques is to protect information with different mechanisms. The objective of cryptography is to convert secret information into encrypted or unreadable form. Many permutations and substitutions are involved in building a cryptography method. It is fundamental to build a robust cryptography method, as it is clear that the encrypted data contain protected or essential information. This situation can lead to any unwanted third party attempting to gain access to the encrypted information.

In certain conditions, it is necessary to deceive unwanted third parties; this is where information hiding or steganography plays an important role [2]. As the name suggests, it is a method to secure information by hiding it into a container, which is a cover medium. This method can be performed in widely used cover media like audio [3, 4], video [5, 6], and image. The benefit of this method is the secrecy of the protected information.

When using a digital image as a container, information is concealed into the cover image. The effect of this action is, it introduces distortion that causes the embedded image to differ from the original cover image. One of the determining factors of the level of distortion is the size of the embedded data. Typically, the bigger the size of embedded data, the

greater the distortion level of the stego image [2]. In that regard, a suitable steganography method aims to achieve a higher amount of embedded data while maintaining stego image quality [2]. In some cases, some methods do not change the cover images to their original form. However, in other conditions, it is necessary to return the cover image to the initial state, or in other words, it has to be reversible. Hence, it is why this particular steganography method is known as reversible data hiding (RDH).

In the last few decades, various approaches have been established to improve the quality of the embedded image or the capacity of secret data. In general, steganography on spatial domains can be divided into compress-and-append, expansion-based (EB), and histogram shifting (HS)-based [7] RDH. The compress-and-append method is widely used in the earlier RDH system. The EB-based approach is first developed by Tian [8], which is known as difference expansion (DE). It utilizes the differences of adjacent pixels. Another steganography method based on EB is prediction error expansion (PEE), established by Thodi and Rodriguez [9]. It uses differences between the original pixel and the error value of the original pixel. Next, there are other reliable schemes, such as interpolation [10–12]. The main goal of the interpolation scheme is to utilize the enlargement of the image resolution. The disadvantage of those methods is low embedded data storage and poor defense against noise attacks. Other than the spatial domain, data embedding can also be carried out in the transform domain [13–15]. The use of this domain affects the robustness of the stego images, but the payload capacity is likely to drop.

The HS-based method is introduced by Ni et al. [16]. It utilizes histograms to embed data by creating space in it. 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. Nevertheless, this HS method can only carry a smaller number of data than the EB scheme.

In this research, we utilize PE and HS as the base of the proposed method. By using those two techniques, this research aims to increase the quality of stego images further. The utilized histogram is the one formed by PE. The shifting process has made the proposed method differs from the existing ones; we remove it to avoid unnecessary histogram modification. So, confidential data are not embedded in the space resulted from the shifting process. To restore the embedded data, the location map is generated during the embedding. Moreover, to further minimize the histogram modification, we categorize secret bits to be embedded in selected PE values. It not only reduces the modification but also improves the embedding room.

The remaining sections of this paper are organized as follows. Section 2 presents previous related works. The proposed method is explained in Section 3. Experimental results are given in section 4, and finally, section 5 is the conclusion of this research.

2. Related works

The first research to utilize image histogram was initiated by Ni et al. [7]. This method's main concept is to pick the most frequent color in an image as a reference to embed data. The process of embedding data is divided into three-phase. First, search for the most frequent pixel and least frequent pixel in the cover image; this can be done quickly by creating a cover image histogram. In the histogram, the most frequent pixel is that with the highest peak or peak pixel, and the least frequent pixel is the minimum or sometimes zero pixels because that particular pixel number does not exist in the cover image. Then, shift all pixels between the peak and minimum pixels. In doing so, there is one pixel left empty. Next, fill it with pixels from the neighbors. The phase of shifting pixels is explained in Eq. (1), and the embedding data phase can be described in Eq. (2). In those equations, the peak pixel is P, and the minimum pixel is L; I is the pixel before being shifted; I' is the pixel that has been shifted, i and j are the pixel location. Then $b(n)$ is secret bits, where n is an index of secret bits.

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

$$I''_{ij} = \begin{cases} I'_{ij} + 1 & if\ I'_{ij} = P\ and \\ & b(n) = 1, P < L \\ I'_{ij} - 1 & if\ I'_{ij} = P\ and \\ & b(n) = 1, P > L \\ I'_{ij} & if\ I'_{ij} = P\ and\ b(n) = 0 \end{cases} \quad (2)$$

We can see in Eq. (2) that embedding operation can only occur as many as the number of the peak pixel. This is the drawback of this scheme because peak pixel (P) frequencies limit the total capacity. If the cover image has an even distribution of pixel color, it will impact the hiding capacity.

Another scheme developed by Hong et al. [17] also utilizes HS schemes and combines them with EB. However, before the embedding process occurs, they change the cover image into prediction error (PE), and then they insert it with the help of the histogram

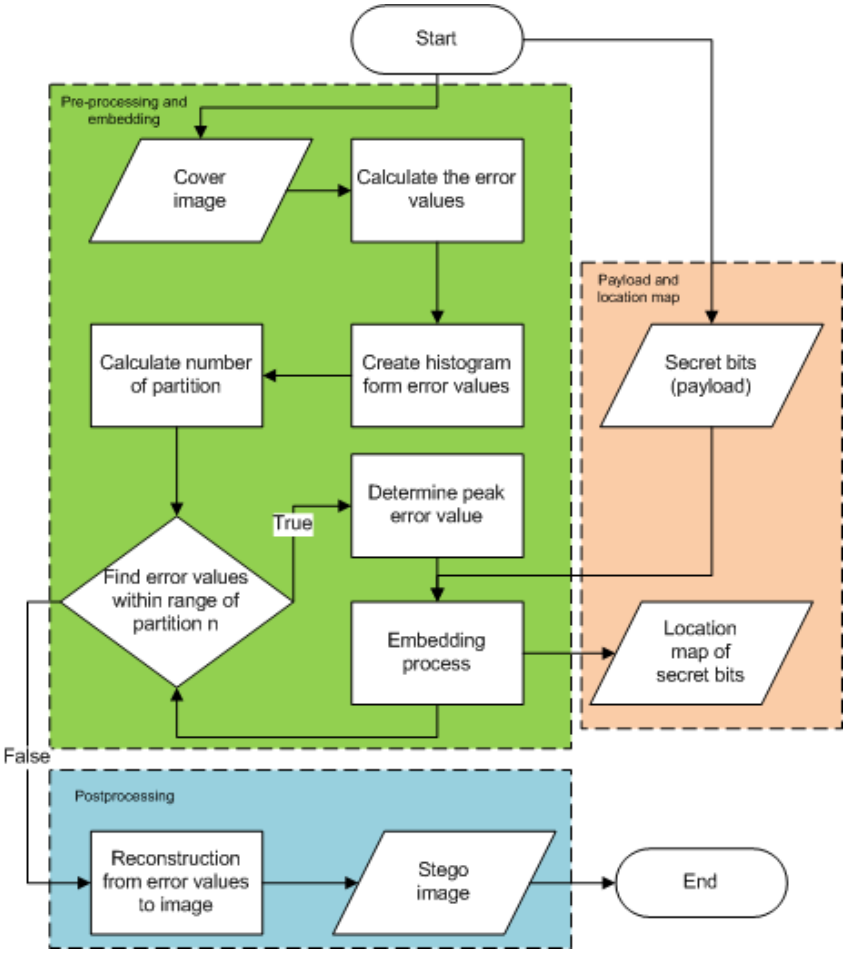


Figure. 1 The flow of the data embedding process

produced by PE. This resulting stego image has better quality than that of EB and HS schemes, and the major drawback of HS can be mitigated. A histogram is used as a reference for the embedding phase, or the PE histogram can provide a higher number of pixel frequencies. This method is also used and further refined in [18–20]. Rad et al. [21] then established a checkerboard predictor (CBP). It can generate a more accurate PE histogram than [17]. This means the frequency of smooth pixels (peak pixels) is much more than [17]. Nevertheless, problems arise if we want to utilize non-smooth pixels on the histogram. Depending on the cover image, the hiding capacity could be lower than [21].

Yi et al. [22] conducted a new approach using block-level prediction-error expansion (BLPEE). While it can improve the capacity of embedded data, the impact of the image quality is not severe and still tolerated. Before the embedding phase, the predictor calculates the PE by using a 2×2 block of pixels. It is interpreted in Eqs. (3) and (4), where I is the pixel that will be predicted, I_r, $I_c$and I_d are pixel located in the same row, column and diagonal side of I, while W_r , W_c and W_d are the weight coefficients. The weight coefficients are set as W_r = W_c = 0.4 and W_d=0.2. Based on the test results, this method gives better embedding capacity than [23, 24].

$$\hat{I}_{i,j} = W_r I_r + W_c I_c + W_d I_d \tag{3}$$

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

In Eq. (3) and Eq. (4) $E'_{i,j}$ is the PE after hiding data, T_l and T_r are the capacity parameters. Both T_l and T_r are calculated using Eq. (5) and Eq. (6), where h is the number of occurrences when prediction-error values in the sequence are equal to $E_{i,j}$.

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

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

Another method by Kumar and Agrawal [25] is also using PEE. They use the adjacent PE value to

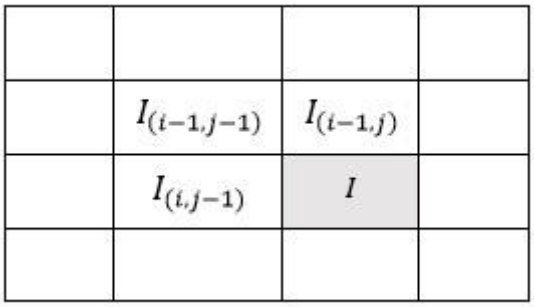


Figure. 2 Illustration of prediction using MED

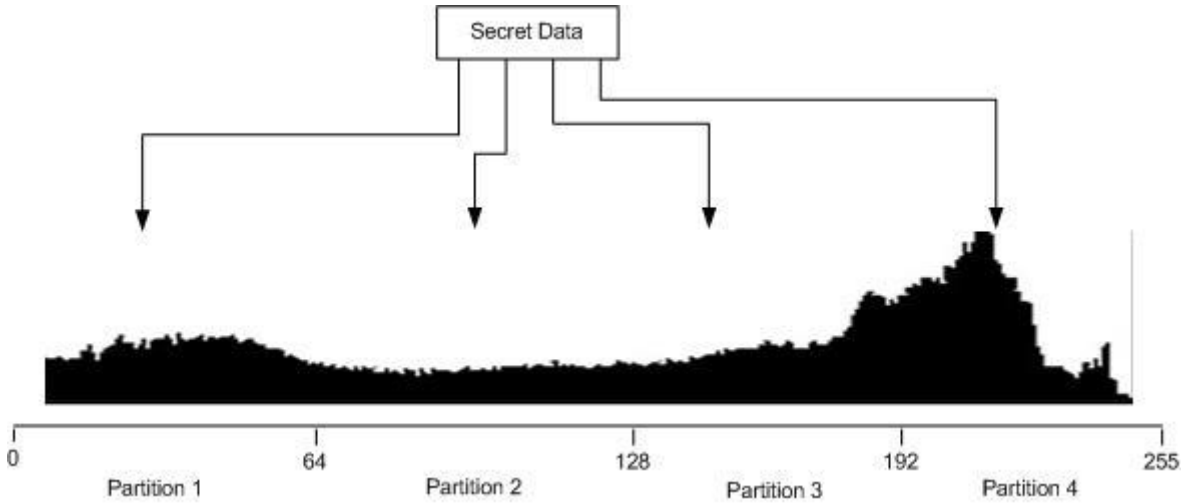


Figure. 3 Illustration of data embedding in a partitioned histogram

hide data. In their research, Kumar and Agrawal predict the error value of even columns by using odd columns, and then, secret data are embedded on the even. This leads to more data that can be embedded due to the use of even columns. While it can provide a massive amount of embedding capacity, it sacrificed much quality. The error value prediction process is denoted in Eq. (7), where j_e is the position of the even column. The embedding process is described in Eq. (8). In this equation, $\hat{I}$ is the error value after the prediction process.

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

$$E'_{i,j_e} = (E_{i,j_e} \times 2) + b(n) \quad (8)$$

Kamal and Islam [26] presented an idea to distribute pixel values and separate them into the pixel group based on image histogram. They implement prediction errors to improve payload capacity. Then they implement prediction to the last calculated absolute-valued errors. This can yield more data that can be put in the image, but it introduces massive image quality degradation.

3. Proposed method

In the previous section, we have focused on the various methods using HS, PEE, or both. The advantage of using HS is that it is more tolerant to change, but its capacity is limited. This weakness can be mitigated using a histogram made using PEE, but there are some drawbacks in quality depending on the corresponding PEE method. Our proposed method's main goal is to increase the secret data capacity and stego image quality by utilizing both methods.

As it is known, in steganography, there are two processes, i.e., the process of embedding and extracting data. The flow of the data embedding process can be seen in Fig. 1. Before hiding the secret data, the PE is calculated using a predictor; then, we embed the secret data by using the histogram of PE as the reference.

3.1 Data embedding

MED is used as the predictor to generate PE because it can produce a more even distributed histogram from PE. To increase the embedding capacity, we divide the PE histogram into partitions

so that each of them has an individual peak. It is advantageous, specifically if the PE histogram is distributed evenly. Also, to further increase the capacity, we do not utilize the peak of PE values because they are considered a smooth value in the image or the most frequent pixels. This characteristic has made it more sensitive to change; thus, we prefer to utilize the adjacent values. The illustration of predictions can be seen in Fig. 2. The prediction error is calculated by using Eq. (9), where I is the original pixel, $\hat{I}$ is the error value used to be the embedding location, and E is the PE value calculated using Eq. (10). To divide the histogram, we use Eqs. (11), (12), and (13), where X is the PE value range, and F is the number of the partition, C^n is the divided partition. Fig. 3 depicts the process of data embedding using the partitioned histogram.

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

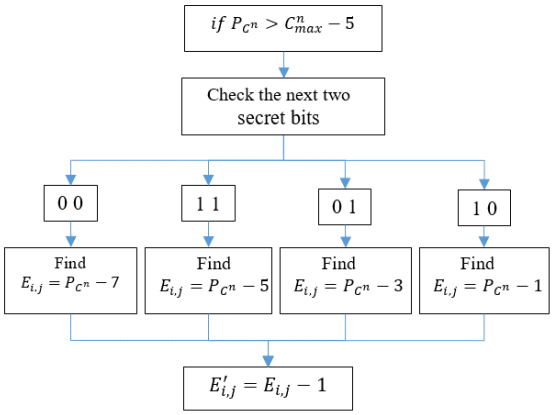
$$E_{i,j} = I_{i,j} - \hat{I}_{i,j} \quad (10)$$

$$X = P - L + 1 \quad (11)$$

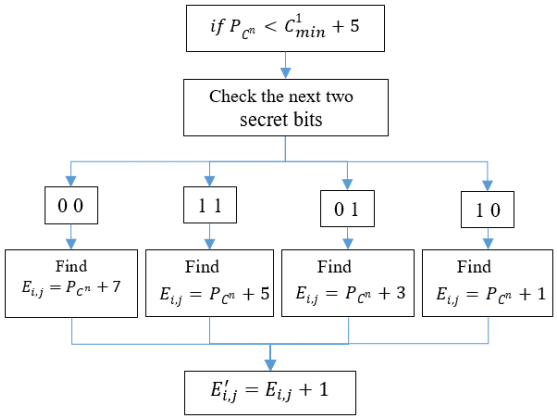
$$F = 256/2^{\lceil log_2 X \rceil} \quad (12)$$

$$C = \{C^1, C^2, C^3, \ldots, C^F\} \quad (13)$$

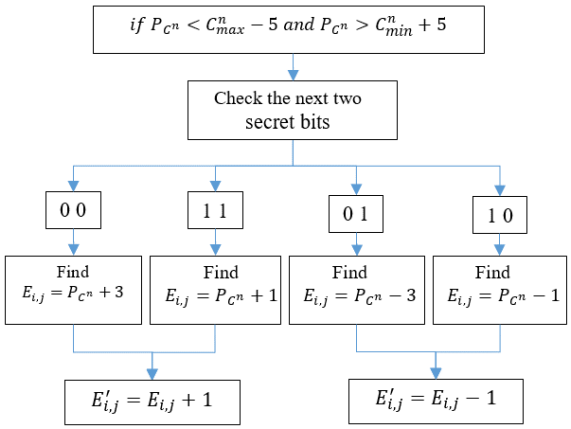
After the histogram of PE is generated and divided into partitions, the PE value is becoming a reference to embed data in each utilized partition. This leads to more data can be embedded than using only one peak value as the reference. As we know, a shifting histogram can cause difficulties if we utilize more values as a reference, especially in recovering the original image. We do not shift the histogram; instead, we simply change it to the adjacent PE value if we encounter the reference PE in the scanning process. To extract secret data, which we will discuss in detail later, we use a location map to help find the location of embedded bits. Not only enables easier data recovery, but it also reduces modification that can lead to better image quality without any kind of shifting process. To enhance the storage of embedded data, we increase the number of each PE value that can hold more than one bit of data. We made secret bits categories; each of them contains a pair of bits. Every category is embedded in different PE values reference, so certain PE values will only accept



(a)



(b)



(c)

Figure 4. Example of embedding process in a partition: (a) Embedding process where $P_{C^1} > C^1_{max} - 4$, (b) Embedding process where $P_{C^1} < C^1_{min} + 4$ and (c) Embedding process where $if\ P_{C^n} < C^n_{max} - 5\ and\ P_{C^n} > C^n_{min} + 5$

certain categories. Those four pairs of groups are (1, 1), (0, 0), (0, 1) and (1, 0).

The following explanation can be implemented for each partition. We embed secret data on partition C^1 with a PE value of 2 to 16, where C^1_{max} is 16 and C^1_{min} is 2. First, we pick the most frequent PE value or P_{C^1} to use it as a reference point. If $P_{C^1} > C^1_{max} -$

5, then scan two neighboring secret bits or $b(n)$ and $b(n+1)$; if they are (1, 1), search for the value of PE $P_{C^1} - 5$ and reduce it by 1. If two neighboring bits are (0, 0), search for the PE value of $P_{C^1} - 7$ and lower it by 1. For the neighboring bits of (0, 1), the PE value $P_{C^1} - 3$ decreases by 1. Finally, if (1, 0) are two neighboring bits, then PE value $P_{C^1} - 1$ is the one to decrease by 1. So, each pair of bits can only be embedded in this partition if those conditions are met. This process can be elaborated in Fig. 4 (a).

In the condition of $P_{C^1} < C^1_{min} + 5$, the reference position for pairs of bits are changed. Search for PE value of $P_{C^1} + 5$ if the discovered pair of bits are (1, 1) and add it by 1. If the next bits pair are (0, 0), find $P_{C^1} + 7$ and increase it by 1. Scan the next pair of bits; if they are (0, 1), add $P_{C^1} + 3$ by 1. Lastly, when the next pair is (1, 0), add PE value of $P_{C^1} + 1$ by 1. This process can be described in Fig. 4 (b).

Another situation to embed secret bits is where $P_{C^n} < C^n_{max} - 5$ and $P_{C^n} > C^n_{min} + 5$. We must check the next two secret bits; if they are (0, 0) or (1, 1), then search for $P_{C^n} + 3$ or $P_{C^n} + 1$, respectively. After that, add it by 1. If the next two secret bits are (0, 1) or (1, 0), then find $P_{C^n} - 3$ or $P_{C^n} - 1$, respectively and increase the PE value by 1. This condition can be observed in Fig. 4 (c).

Those processes are performed until no data is left to be embedded, or in this case, C^1 does not have enough space to continue to the next partition. During the data embedding process, the predicted value location is stored for the secret data extraction process. The position of x-axis is stored at $x(n)$, and y-axis in $y(n)$, with n is the index of the secret data. To store the embedded bits' positions, we use Eq. (14) and Eq. (15).

$$x(n) = i \tag{14}$$

$$y(n) = j \tag{15}$$

If the number of secret bits is odd, then the last bit is used in the PE value of *P* as the reference. If the last bit is 0, then *P* is decreased by one; if it is 1, increase *P* by one. For example, if all pair of secret bits have been embedded and leave bit 0 as the last bit, search for P_{C^1} and decrease it, as implemented using Eq. (16). The position of embedded bits is also saved using Eq. (14) and Eq. (15).

$$E'_{i,j} = \begin{cases} E_{i,j} + 1 \;\; if \; E_{i,j} = P_{C^1} \; and \; b(n) = 1 \\ E_{i,j} - 0 \;\; if \; E_{i,j} = P_{C^1} \;\; and \; b(n) = 0 \end{cases} \tag{16}$$

After that, return PE values to image form by using Eq. (17).

$$I'_{i,j} = \hat{I}_{i,j} - \; R'_{i,j} \tag{17}$$

3.2 Data extraction and image restoration

Extracting secret data is relatively simple. First, change the image to PE values by using Eq. (18); then, we use Eq. (12) to divide the histogram of the PE values into the partition.

$$E'_{i,j} = \hat{I}_{i,j} - I'_{i,j} \tag{18}$$

After that, scan the PE values in a partition. Let C^1 be an example. Just like the embedding process, if we encounter $P_{C^1} > C^1_{max} - 5$, then $P_{C^1} - \; 8$ and its location is the same as $x\,(n)$ and $y\,(n)$; add its value by one to return it to its original state. Now, $P_{C^1} - 8$ has changed into $P_{C^1} - 7$ that meets conditions where the extracted bits $b\,(n)$ and $b(\,n\,+\,1)$ is (0, 0). The bits extracted are going to depend on conditions met. The rest of the conditions for this partition are: if it is $P_{C^1} - \; 6$, the extracted bits are (0,0); if it is $P_{C^1} - \; 4$, extracted bits are (0, 1) and $P_{C^1} - \; 2$ the extracted bits are (1, 0). The complete process of extracting data can be seen in Fig. 5. Regarding the odd amount of bits, the last bit's position at $P_{C^1} - 1$ for bit 0 or $P_{C^1} \; + \; 1$ for bit 1. Overall, this step is described in Eq. (19) and (20). After all secret bits have been extracted, change the PE value to the original image pixels using Eq. (21).

$$E_{i,j} = \begin{cases} E'_{i,j} - 1 & if \; E'_{i,j} = P_{C^1} + 1 \\ E'_{i,j} + 1 & if \; E'_{i,j} = P_{C^1} - 1 \end{cases} \tag{19}$$

$$b(n) = \begin{cases} 1 & if \; E'_{i,j} = P_{C^n} + 1 \\ & and \; x(n) = i \; and \; y(n) = j \\ 0 & if \; E'_{i,j} = P_{C^n} - 1 \\ & and \; x(n) = i \;\; and \; y(n) = j \end{cases} \tag{20}$$

$$I_{i,j} = \hat{I}_{i,j} + E_{i,j} \tag{21}$$

4. Results and discussion

In this section, the results of the experiment are provided. We evaluate the proposed scheme's performance by measuring the peak signal to noise ratio (PSNR) and embedding space size. We use Eq. (22) and Eq. (23) to calculate PSNR and the corresponding mean square error (MSE),

respectively. In those equations, I_{MAX} is the pixel with the highest value, while W and L are the image's height and width. PSNR measures the distortion level of stego images and identifies any dissimilarities between the original and stego images. MSE is gained by calculating the total pixel value difference between the stego image and the original cover image. The lower MSE value implies that the quality of the stego images is degrading.

$$PSNR = 10log_{10}\frac{(I_{MAX})^2}{MSE} \quad (22)$$

$$MSE = \left(\frac{1}{WL}\right)\sum_{i=1}^{W}\sum_{j=1}^{L}\left(I_{ij} - I'_{i,j}\right)^2 \quad (23)$$

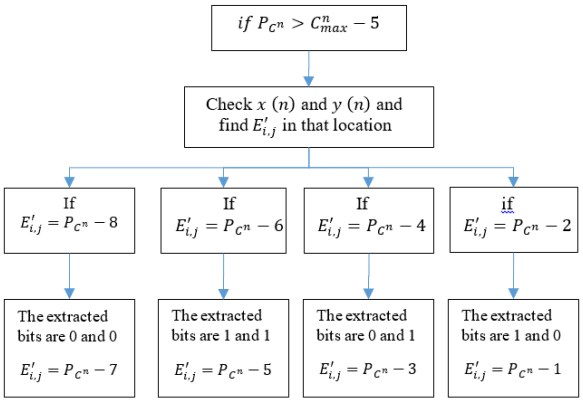
The size of the embedding space is measured by calculating the bit per pixel (BPP). It is obtained by dividing the total amount of bits that can be embedded by the total cover image pixel. The calculation process of BPP is described in Eq. (24).

$$BPP = \frac{Total\ number\ of\ embedded\ bits}{Total\ number\ of\ cover\ image\ pixels} \quad (24)$$

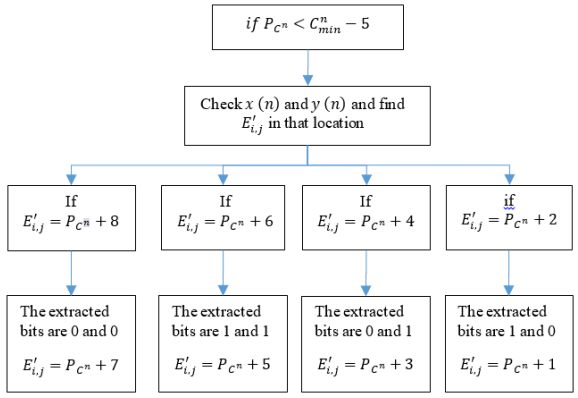
The cover image dataset used for this experiment is obtained from [27, 28]. Each of the test images is a grayscale image with a size of 512×512 pixels. Test images are divided into two categories, medical and general images, to test the performance of the proposed scheme under different circumstances. Examples of test images are depicted in Fig. 6.

Tables 1 and 2 show a comparison of PSNR values between the proposed scheme and [22, 26], without any encryption applied in test images. From those tables, we see the proposed scheme produces better PSNR than the others in every test image. On average, the proposed scheme is 14.56 dB better than [22] in medical images and 15.31 dB better in generalimages. Compared with [26], the proposed scheme also produces an average of 17.24 dB more in medical images and has 18.66 dB more on average in general images. This is due to the utilized PE histogram in the proposed scheme that does not get shifted. Partition and a bit allocation are employed to preserve the number of PE values used for embedding space, resulting in less modification of images and less distortion.

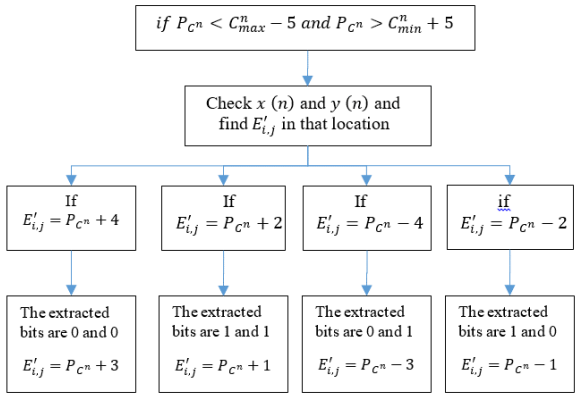
Embedding capacity is presented in Table 3 and Table 4. Here, we can see that the proposed scheme has a lower overall BPP than others, especially in medical images. It is crucial to notice that different image has different properties that can lead to varied capacity. Because general images mostly have diverse pixel colours and changed into PE, the



(a)



(b)



(c)

Figure 5. Example of extracting process in a partition: (a) Extracting process where $P_{C^1} > C^1_{max} - 4$, (b) Extracting process where $P_{C^1} < C^1_{min} + 4$ and (c) Extracting process where $if\ P_{C^n} < C^n_{max} - 5\ and\ P_{C^n} > C^n_{min} + 5$

distribution of PE value is more even, which helps toprovide more embedding space as the proposed scheme will take advantage of it. From those tables, we can see that the embedding space provided in [26] is higher than the proposed method. It is because they implement prediction to the last calculated absolute-value errors and embed more than one bit in one

Table 1. Comparison of similarity results from the proposed method and previous methods on general grayscale images

Image	**Proposed method**		**Yi et al.** [22]		**Kamal and Islam** [26]	
	Payload (Kb)	**PSNR (dB)**	**Payload (Kb)**	**PSNR (dB)**	**Payload (Kb)**	**PSNR (dB)**
Baboon	5	66.34	5	48.48	5	33.65
	10	65.31	10	48.44	10	31.53
	15	64.67	15	48.39	15	30.26
	20	63.42	20	48,35	20	29.76
Lena	5	66.25	5	49.29	5	35.27
	10	65.18	10	49.23	10	34.56
	15	63.56	15	49.18	15	33.29
	20	62.67	20	49.13	20	32.25
Pepper	5	66.43	5	49.04	5	33.48
	10	64.36	10	48.99	10	32.67
	15	63.25	15	48.94	15	31.23
	20	62.68	20	48.89	20	30.68
Elaine	5	66.91	5	48.83	5	32.75
	10	65.21	10	48.78	10	31.80
	15	63.16	15	48.74	15	30.14
	20	61.19	20	48.69	20	29.04
Boat	5	65,29	5	48.84	5	34.57
	10	64,95	10	48.80	10	33.79
	15	62.77	15	48.75	15	32.86
	20	61.89	20	48.70	20	31.30

Table 2. Comparison of similarity results from the proposed method and previous methods on medical grayscale images

Image	**Proposed method**		**Yi et al.** [22]		**Kamal and Islam** [26]	
	Payload (Kb)	**PSNR (dB)**	**Payload (Kb)**	**PSNR (dB)**	**Payload (Kb)**	**PSNR (dB)**
Abdominal	4	68.56	4	51.76	4	34.62
	8	65.49	8	51.69	8	33.37
	12	64.71	12	51.62	12	31.92
	16	63.22	16	51.54	16	30.87
Hand	4	68.31	4	51.81	4	34.64
	8	66.50	8	51.73	8	33.61
	12	64.25	12	51.66	12	32.53
	16	63.19	16	51.59	16	31.21
Chest	4	68.38	4	52.81	4	34.68
	8	66.42	8	52.72	8	32.11
	12	64.83	12	52.62	12	33.24
	16	62.58	16	52.53	16	30.96
Head	4	68.72	4	51.94	4	34.62
	8	66.65	8	51.86	8	33.42
	12	64.87	12	51.79	12	32.49
	16	62.56	16	51.71	16	31.23
Leg	4	67.24	4	51.81	4	34.06
	8	66.45	8	51.74	8	32.93
	12	64.63	12	51.66	12	32.01
	16	63.24	16	51.59	16	30.96

image block; those can lead to higher secret bits capacity and lower PSNR quality. In [22], the embedding capacity is dependent on how much we are willing to destroy the embedded image as the embedding process is performed until all available secret bits are embedded. For measuring the payload capacity, the PSNR value of the embedded image should not be less than 30 dB to maintain good imperceptibility [2]. For this reason, in the experiment, we continue to embed the secret while still maintaining the PSNR value not lower than about 30 dB.

Table 3. Comparison of bit per pixel value between the proposed and previous methods on general grayscale images

Image	Bit per pixel (BPP)		
	Proposed method	**Yi et al.** [22]	**Kamal and Islam** [26]
Baboon	0.3126	0.3815	0.6560
Lena	0.4351	0.3815	0.5982
Pepper	0.3935	0.3815	0.4387
Elaine	0.3689	0.3815	0.6683
Boat	0.3524	0.3815	0.5962

Table 4. Comparison of bit per pixel value between the proposed and previous methods on medical grayscale images

Image	Bit per pixel (BPP)		
	Proposed method	**Yi et al.** [22]	**Kamal and Islam** [26]
Abdominal	0.0983	0.3052	0.4376
Hand	0.0756	0.3052	0.5688
Chest	0.0802	0.3052	0.6370
Head	0.1245	0.3052	0.6917
Leg	0.0611	0.3052	0.5209

5. Conclusion

This research is inspired by previous works that utilized the HS and PEE methods. The HS method has the advantage of better overall image quality, but it has a lower capacity due to the reliance on the highest pixel frequency in the original cover image. PEE helps to mitigate these issues by providing a histogram with higher peak frequencies.

The proposed scheme increases the quality of stego images by using both the PEE and histogram-based methods. Furthermore, we implement histogram partition and secret bit distribution to reduce original cover image modification, which can cause less distortion introduced in stego images. The experimental results show that the proposed method can improve the quality of the stego images when compared to the existing studies. Using the same image dataset, the average of PSNR values increases by 17.94 dB for general images and 14.95 dB for medical images.

However, there is still a drawback to be resolved in future works. That is, the proposed scheme has limited number of bits that can be embedded in the original cover image, which in certain cases, is lower than existing research; so that, it can be performed as long as the distortion is not too severe. Hence, the proposed scheme is appropriate to use for various numbers of secret bits, from relatively small to medium.

Conflicts of Interest

The authors declare that they have no competing interests.

Author Contributions

The conceptualization, research methodology, implementation and experiments, writing—original draft preparation and funding are provided by CCI. The supervision, writing—review and editing and administration of research are provided by TA.

Acknowledgments

This work was supported and funded by the Universitas 17 Agustus 1945 Surabaya.

References

[1] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods”, *Signal Processing*, 2010.

[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] M. M. Amrulloh and T. Ahmad, “Utilizing Fuzzy Logic in Developing Reversible Data Hiding Method”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 327–336, 2020.

[4] S. Jiang, D. Ye, J. Huang, Y. Shang, and Z. Zheng, “SmartSteganogaphy: Light-weight generative audio steganography model for smart embedding application”, *J. Netw. Comput. Appl.*, Vol. 165, p. 102689, 2020.

[5] M. Ramalingam, N. A. M. Isa, and R. Puviarasi, “A secured data hiding using affine transformation in video steganography”, *Procedia Comput. Sci.*, Vol. 171, pp. 1147–1156, 2020.

[6] A. Fatnassi, H. Gharsellaoui, and S. Bouamama, "Towards Novel Video Steganography Approach for Information Security", *Procedia Comput. Sci.*, Vol. 159, pp. 953–962, 2019.

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

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

[9] 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.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] X. Niu, Z. Yin, X. Zhang, J. Tang, and B. Luo, "Reversible Data Hiding in Encrypted AMBTC Compressed Images BT - Digital Forensics and Watermarking", pp. 436–445, 2017.

[24] Z. Yin, A. Abel, J. Tang, X. Zhang, and B. Luo, "Reversible data hiding in encrypted images based on multi-level encryption and block histogram modification", *Multimed. Tools Appl.*, Vol. 76, No. 3, pp. 3899–3920, 2017.

[25] 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.

[26] A. H. M. Kamal and M. M. Islam, "A prediction error based histogram association and mapping technique for data embedment", *J. Inf. Secur. Appl.*, Vol. 48, p. 102368, 2019.

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

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