Int. J. Advance Soft Compu. Appl, Vol. 12, No. 3, November 2020
ISSN 2074-8523; Copyright © ICSRS Publication, 2020
www.i-csrs.org

Reversible Data Hiding Scheme based on General Difference Expansion Cluster

Ahmad Juniar Ilham, Tohari Ahmad

Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.
e-mail: juniar.18051@mhs.its.ac.id, tohari@if.its.ac.id

Received 20 June 2020; Accepted 28 September 2020

Abstract

Technological development leads to raising the number of digital image manipulation, especially in data security. One of the popular methods that can be used to maintain privacy is steganography. It is a technique used to hide secret messages into media such a text, image, audio, or video. Among the methods in steganography, Difference Expansion (DE), Reduced Difference Expansion (RDE), and Generalized Difference Expansion (GDE) are favored to implement. There are two main obstacles when applying this method; the first is the similarity between stego and the cover media; the second is the data capacity that can be accommodated by the cover media. Therefore, further development of these methods is needed. This research proposes applying pixel clusters based on General Difference Expansion Cluster (GDEC) to maintain the stego image quality and increase the embedding capacity. In terms of these evaluated parameters, the experiment results have proved that the proposed method can increase the average quality of each image by more than 4 dB and produces 4.00 bits per pixel (bpp).

Keywords: *data hiding, data security, information security, reversible data hiding, steganography.*

1 Introduction

The sophisticated growth of network technology, such as the internet, has a significant impact on cybercrime. Several studies have noted that more than 50% of people worldwide have been victims of cybercrime, especially in data and information stealing [1]. Confidential data, such as patient health records, need to be protected so that it must be ensured that only authorized parties know the files' contents. Security techniques should be developed to protect critical information and maintain privacy to avoid illegal access [2]. For this purpose, encryption and steganography can be the principal means to secure data secrecy and privacy [3].

Moreover, Reversible Data Hiding (RDH) has become an essential technique for attracting more attention for the last few years [4]. In RDH, secret information must be embedded in an unseen way without losing the meaning of the information revealed by the image [5]. One of the popular methods in RDH is the Difference Expansion (DE) technique that many studies have been presented by extending this method. However, some of them only focused on one issue, either capacity or quality. The initial DE technique [6] works on pairs between adjacent pixels in the form of quad (2×2) blocks to embed two bits. In [5], a scheme of data hiding according to significance-bit-difference expansion is also shown. That proposed method divides the pixels in the original image into two sections, namely, the Higher Significant Bits (HSB) and the Least Significant Bits (LSB), then counts the adjacent pixels to find the HSB difference. In further research, [7] also applied DE to look for the difference d between pixel pairs with 2×1 blocks size to increase the size of the hidden messages by combining it with the modulus function. The research in [8] enhanced that in [7] by utilizing a modulus function for low difference pixel values of the neighboring pixel in a 2×1 blocks size.

However, these methods still have some challenges, namely the similarity between the original and stego media and data capacity that can be inserted into the cover medium. Significant changes can easily attract the attention of attackers and evil attacks done by unauthorized people. The research in [6]–[9] used pairs of adjacent pixels and looked for their difference. It leads to the reduction of the image quality because it enables to produce a significant difference. This paper proposes a new scheme for finding pixel differences by using pixel clusters to solve the issue. The proposed method works by placing image pixels into cluster levels and process it to obtain new pixel values. By using clusters, each pixel image should produce a small difference. Furthermore, underflow and overflow can also be avoided. Here, underflow and overflow mean that the reconstructed pixels' value is less than 0 and more than 255, respectively.

This study is organized as follows. Section 1 explains the introduction of the paper. Section 2 presents the studies dealing with the suggested scheme. The suggested methods are explained in Section 3, and the result of the experiment is provided in Section 4. Finally, the conclusion is given in Section 5.

2 Related Work

In general, RDE consists of some stages, as in the following. RDE performs the reduction to have z' before embedding a bit using Equation 1, where z is the difference between adjacent pixels.

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

To restore its original cover, each pixel block that has been reduced is marked on the location map (LM). This map is made to ensure that the data has been processed correctly, and the method is reversible. The provisions of the location map are given in Equation 2. A pixel block that has been reduced successfully is given a value of 1; otherwise, it is 0.

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

In the RDE implementation [9], it is noticed the underflow and overflow to avoid stego images from significant deterioration. In addition, it implemented multi-layer embedding to increase the embedding capacity. The experimental results show that their method can embed vast amounts of messages while also maintain the quality of the image; however, pixel pairs containing negative differences are not processed.

In 2017, Arham et al. [10] suggested an RDE-based method that focused on refining both the quality of visual and capacity by minimizing the difference value in pixels with Improved Reduced Difference Expansion (IRDE). Next, [11] suggested a method that uses pixel relevance to develop the method proposed by [10] by changing the scanning mode and resizing the block to increase the capacity of the embedded image from 3 to 8 bits. Their method has also been modified for multi-layer insertion by changing permutations from four to nine to prevent overflow/underflow and massive distortion in the image. Furthermore, [12] combined the RDE method with fuzzy logic to apply a multi-layer data hiding scheme. It can increase the embedding capacity up to four times more than a single layer while still relatively maintaining the image quality.

Generalized Difference Expansion extends DE methods by increasing the embedding capacity [6][13]. The DE can only embed 1 bit into each pixel pair, whereas the GDE method can embed three or more bits at once. The GDE method in [6] works by taking every four pixels into a block. Let $u = (u_0, u_1, u_2, u_3)$ be pixels in a block. The difference between pixels in a block j can be calculated as in Equation 3.

$$\begin{cases} j_0 = \left\lfloor \frac{u_0 + u_1 + u_2 + u_3}{4} \right\rfloor \\ j_1 = u_1 - u_0 \\ j_2 = u_2 - u_1 \\ j_3 = u_3 - u_2 \end{cases} \quad (3)$$

Data embedding is processed using either Equation 4 or Equation 5.

$$\begin{cases} j_1' = 2 \times j_1 + b_1 \\ j_2' = 2 \times j_2 + b_2 \\ j_3' = 2 \times j_3 + b_3 \end{cases} \quad (4)$$

$$\begin{cases} j_1' = 2 \times \left\lfloor \frac{j_1}{2} \right\rfloor + b_1 \\ j_2' = 2 \times \left\lfloor \frac{j_2}{2} \right\rfloor + b_2 \\ j_3' = 2 \times \left\lfloor \frac{j_3}{2} \right\rfloor + b_3 \end{cases} \quad (5)$$

In this method, b_1, b_2 and $b_3 \in [0,1]$ are the data bits to be embedded. After embedding the data bits, then $j' = (j'_0, j'_1, j'_2, j'_3)$ will be transformed into new pixels $u' = (u'_0, u'_1, u'_2, u'_3)$ as in Equation 6.

$$\begin{cases} u_0' = j_1' - \left\lfloor \frac{u_0 + u_1 + u_2 + u_3}{4} \right\rfloor \\ u_1' = j_1' + u_0 \\ u_2' = j_2' + u_0 \\ u_3' = j_3' + u_0 \end{cases} \quad (6)$$

Research in [14] introduced the GDE method to embed the confidential information into audio by modifying [6]. This scheme does not use location maps. However, this scheme can increase the capacity of ($2N$ + 1) bit embedding messages and restore original audio with minimal distortion.

3 Proposed Method

This research proposes a reversible data hiding method based on General Difference Expansion Cluster (GDEC). In this section, the designing pixels cluster is first presented. Then, the data embedding process and its example are explained. Finally, the extraction process and its example are given.

3.1 Designing pixel cluster

Unlike the RDE and GDE methods that use pixel pairs to look for pixel differences, this method uses cluster of pixels. Afterward, the pixel differences are reduced again to produce a smaller difference value before data embedding, so that the difference between the original image and stego image is not so significant.

The most crucial thing in this research is designing the pixel clusters created using Equation 7 where K_n represents the clustered index.

$$K_n = \{[x, y] \mid y = 4n - 1,\ x = y - 3,\ n = 1,2,..64\}\ \forall x, y \in [0,255] \quad (7)$$

From Equation 7, we can specify members of the pixel cluster. For example, n = 1; $y = 4 \times 1 - 1 = 3$, $x = 3 - 3 = 0$. So, it obtained the member of $K_1 = [0,3]$. Next,

$n = 2$; $y = 4 \times 2 - 1 = 7$, $x = 7 - 3 = 4$. So, the member of $K_2 = [4,7]$, and so on. We can also find out that the image pixels are divided into 64 clusters, where each cluster has a range of 3. Each pixel cluster has the lowest limit and the uppermost limit, which will later be used as a reference to find the pixel difference.

For example, there is a pixel whose value is 12. Hence, it is included in cluster $K_4 = [11,14]$ It means pixel 11 is the lowest limit of the cluster, and pixel 14 is the uppermost limit of the cluster. The purpose of designing and implementing the cluster is to control the stego image quality so that it does not cause a major alteration to the whole image.

We have conducted a preliminary study by embedding 1 kb data into each image taken from [15]. In that case, we assume that applying the uppermost limit of the cluster can achieve a better PSNR value for images with an average pixel of more than 150. Likewise, applying the lowest limit of the cluster can produce a better PSNR value for images with an average pixel of less than 150.

3.2 Data embedding process and its example

The complete embedding steps in this study are as follows. First, read all image pixels from the cover image, then insert it into the appropriate cluster. Next, calculate the average pixel of the whole image. After that, obtain difference *h* by reducing the original pixel image value with the lowest limit or the uppermost limit of the cluster range as in Equation 8 or Equation 9. C_l and C_u are the lowest and the uppermost limit of the cluster, respectively, whereas *p* is the pixel values of the original image. Equation 8 is used when the average pixel of the whole image is below 150. Otherwise, Equation 9 is used.

$$h = p - C_l \qquad (8)$$

$$h = C_u - p \qquad (9)$$

Like the RDE method that uses location maps, this proposed method takes it to ensure that the private data are processed correctly and can guarantee the return of data to original values. The location map is 00, 10, and 11 for *h* is equal to 0, 2, and either 1 or 3, respectively. The next step records the location map according to the value of *h*. Then, reduce the value of *h* using Equation 10 to produce the value of *h'*.

$$h' = \begin{cases} 0 & \text{,if } h \le 2 \\ h - 2^{[Log_2 h]} & \text{,if } h = 3 \end{cases} \qquad (10)$$

After getting the value of *h*', then insert the personal data *b* using Equation 11 to get the value of *h*". If the pixel is embedded with secret data, then the status is set; otherwise, it is reset.

Lastly, look for a new pixel value *p*' that has been embedded by the confidential data by using Equation 12 or Equation 13. As in the previous case, Equation 12 is also used when the average pixel of the whole image is less than 150; otherwise, Equation 13 is taken.

$$h'' = 2 \times h + b \tag{11}$$

$$p' = h" + C_l \tag{12}$$

$$p' = C_u - h" \tag{13}$$

In more detail, we depict an example of the data embedding process. Assume there are image pixels: 230, 242, 222, 220. The average pixel of the whole image is 188, for instance, and a secret bit $b = 1$. The steps of the embedding process are as follows:

Step 1: Read all the pixels in the cover image.

Step 2: Enter these pixels according to the cluster. In this example, the first pixel 230 is in cluster K_{58} = [228,231], where pixels 228 and 231 are the lowest and the uppermost limit of the cluster range, respectively.

Step 3: Calculate the difference *h* using Equation 9 because the average pixel of the whole image value is above 150.

h = 231 – 230 = 1

Step 4: Reduce the value of *h* to get the value of *h'* by using Equation 10.

h = 1, *h'* = 0

Step 5: Embed secret data bits *b* using Equation 11 to get the value of *h"*.

h" = 2 × 0 + 1 = 1

Step 6: Look for a new pixel value using Equation 13 and record the location map value according to the value of *h*.

$p' = 231 - 1 = 230$

Then the new pixel value of the first pixel is 230 and LM = 11

3.3 Extraction process and its example

The initial extraction process is inserting each stego pixel according to a predetermined cluster as in the embedding stage. After that, the process calculates the difference value of *h* using Equation 8 or Equation 9 according to the average pixel of the whole image. The next stage is getting the data bit *b* using Equation 14. Furthermore, finding the value of *h'* using Equation 15 as in the RDE method.

$$b = LSB(h) \tag{14}$$

$$h' = \left\lfloor \frac{h}{2} \right\rfloor \tag{15}$$

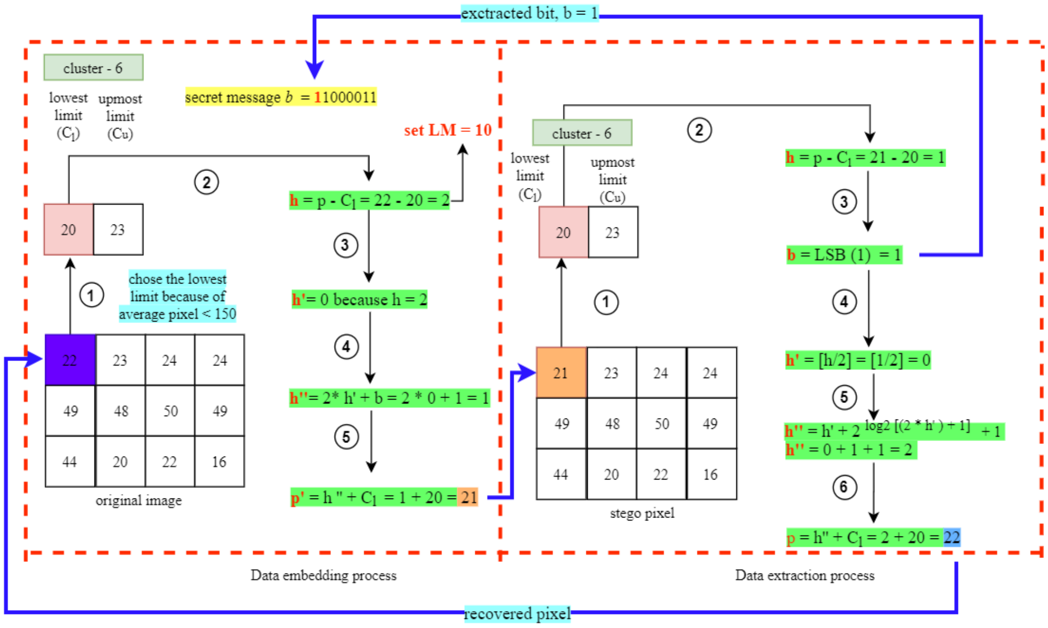


Fig. 1: The Stages of the Embedding and Extraction Process

In the extraction process, the location map is used to get *h''* by adjusting to the Equation 16. The final process is returning stego pixels to its original pixels as Equation 12 or Equation 13.

$$h'' = \begin{cases} 0 & ,\text{if } LM = 00 \\ h' + 2^{\lfloor log_2 [(2 \times h') + 1] \rfloor} + 1 & ,\text{if } LM = 10 \\ h' + 2^{\lfloor log_2 [(2 \times h') + 1] \rfloor} & ,\text{if } LM = 11 \end{cases} \quad (16)$$

We have known the first pixel of a stego image has a value of 230 and LM=11. According to this LM value, the stego pixel value will be extracted to get the confidential data and return them to the original image. The extraction stages are as follows:

Step 1: Read all pixels from a stego image.

Step 2: Enter the stego pixel into its cluster as in the embedding process.

Step 3: Calculate *h* using Equation (9).

h = 231 - 230

Step 4: Extract the secret data bits using Equation 14.

$b = LSB(h) = 1$

Step 5: calculate *h'* using Equation 15.

$$h' = \left\lfloor \frac{1}{2} \right\rfloor = 0$$

Step 6: look for *h''* using Equation 16 by adjusting the value of the location map.

$h'' = h' + 2^{log_2\lfloor(2\times h')+1\rfloor} = 0 + 1 = 1$

Step 7: calculate the original pixels using Equation 13.

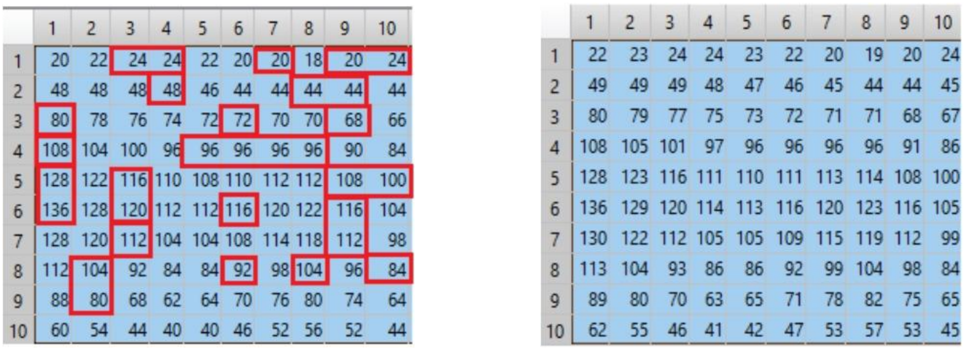
p' = 231 – 1 = 230

From these stages, we know that the stego image was successfully returned to the original image. Also, we illustrate how the embedding and extraction steps are performed, as depicted in Fig. 1.

4 Experimental Results

The quality level of the stego image is measured using PSNR, while the cover capacity is calculated based on the ability of the image to hold data. In both evaluations, the higher is the better. In this experiment, we use several sizes of payload: 10 kb, 20 kb, 30 kb, 40 kb, 50 kb, 60 kb, and 70 kb. Eight testing images with 512 × 512 pixels in size taken from [15]–[17] are used.

Before comparing the proposed with other methods, we examine the pixel difference between the stego and the cover pixels to know how far they are. As an experiment, we use an abdominal image [15] to embed 10 kb of random data, and we took the size of 10 × 10 as a model (see Fig. 2). It is found that the difference between stego pixel and the original pixels is not significant. Some pixels have the same value as indicated by the red square.



	1	2	3	4	5	6	7	8	9	10
1	20	22	24	24	22	20	20	18	20	24
2	48	48	48	48	46	44	44	44	44	44
3	80	78	76	74	72	72	70	70	68	66
4	108	104	100	96	96	96	96	96	90	84
5	128	122	116	110	108	110	112	112	108	100
6	136	128	120	112	112	116	120	122	116	104
7	128	120	112	104	104	108	114	118	112	98
8	112	104	92	84	84	92	98	104	96	84
9	88	80	68	62	64	70	76	80	74	64
10	60	54	44	40	40	46	52	56	52	44

(a) Stego pixels

	1	2	3	4	5	6	7	8	9	10
1	22	23	24	24	23	22	20	19	20	24
2	49	49	49	48	47	46	45	44	44	45
3	80	79	77	75	73	72	71	71	68	67
4	108	105	101	97	96	96	96	96	91	86
5	128	123	116	111	110	111	113	114	108	100
6	136	129	120	114	113	116	120	123	116	105
7	130	122	112	105	105	109	115	119	112	99
8	113	104	93	86	86	92	99	104	98	84
9	89	80	70	63	65	71	78	82	75	65
10	62	55	46	41	42	47	53	57	53	45

(b) Original (cover) pixels

Fig. 2: Comparison between Stego and Original Pixels.

Table 1: The maximum embedding capacity of existing and proposed methods

Cover Image	Capacity (bits)			Bit/pixel		
	[8]	[9]	Prop. Method	[8]	[9]	Prop. Method
Airplane	131072	1010463	1048576	0.500	3.854	4.00
Baboon	131061	1028764	1048576	0.499	3.924	4.00
Boat	130808	1010556	1048576	0.498	3.854	4.00
Peppers	130780	953932	1048576	0.498	3.638	4.00
Abdo-minal	129881	872825	1048576	0.495	3.329	4.00
Sandiego	131072	1014259	1048576	0.500	3.869	4.00
Fruits	128971	1025432	1048576	0.491	3.911	4.00
Car	130871	1010920	1048576	0.499	3.856	4.00

Table 2: The experimental results of proposed methods. In the column "Method", 1, 2 and 3 mean the proposed method using the lower limit, the upper limit, and the lower limit or upper limit of a cluster, respectively; and "Ave. Pixels" means the average pixel values.

Cover	Ave. Pixel	Method	PSNR (dB)						
			10 kb	20 kb	30 kb	40 kb	50 kb	60 kb	70 kb
Air-plane	179.2	1	62.06	58.98	57.21	55.97	55.02	54.55	53.57
		2	62.02	59.02	57.25	56.03	55.05	54.54	53.60
		3	62.02	59.02	57.25	56.03	55.05	54.54	53.60
Ba-boon	129.61	1	62.06	59.03	57.23	55.95	54.99	54.51	53.55
		2	61.98	58.95	57.22	55.96	54.99	54.48	53.52
		3	62.06	59.03	57.23	55.95	54.99	54.51	53.55
Boat	129.7	1	62.32	59.23	57.52	56.25	55.29	54.79	53.87
		2	62.03	59.03	57.29	56.03	55.08	54.59	53.62
		3	62.32	59.23	57.52	56.25	55.29	54.79	53.87
Pep-pers	120.21	1	62.08	59.00	57.23	55.95	54.98	54.48	53.55
		2	61.99	58.95	57.19	55.93	54.97	54.47	53.51
		3	62.08	59.00	57.23	55.95	54.98	54.48	53.55
Abdo minal	101.5	1	62.99	59.72	57.99	56.82	55.87	55.36	54.38
		2	62.03	59.05	57.36	56.18	55.25	54.78	53.85
		3	62.99	59.72	57.99	56.82	55.87	55.36	54.38
San-diego	165.78 29	1	61.96	58.95	57.20	55.94	54.95	54.46	53.52
		2	62.07	58.99	57.22	55.97	55.01	54.50	53.53
		3	62.07	58.99	57.22	55.97	55.01	54.50	53.53
Fruits	173.16 57	1	59.88	56.73	54.98	53.76	52.84	52.43	51.72
		2	63.07	60.12	58.34	57.01	56.03	55.45	54.36
		3	63.07	60.12	58.34	57.01	56.03	55.45	54.36
Car	184.73 09	1	61.92	58.69	56.80	55.73	54.85	54.39	53.43
		2	61.78	58.76	57.11	55.95	54.97	54.47	53.48
		3	61.78	58.76	57.11	55.95	54.97	54.47	53.48

We propose three embedding schemes that will be compared with [8], [9]. First, the scheme uses the lowest limit of the cluster. Second, the method uses the uppermost limit of the cluster, and third, it uses the lowest or uppermost limit following the average pixel image as explored in the preliminary study.

The evaluation of the embedding capacity and quality is provided from Table 1 to Table 4. Based on those tables, we find that in general, all proposed schemes gain a better result of PSNR and higher capacity than [8] and [9]. The proposed method can increase more than 4 dB for all images in terms of the stego image quality except in the "Abdominal" in research [9] by utilizing the pixel cluster. Regarding the embedding capacity, the proposed method can produce 4.00 bits

Table 3: The experimental results of the method proposed by [8]

Cover	Ave. pixel	PSNR (dB)						
		10 kb	20 kb	30 kb	40 kb	50 kb	60 kb	70 kb
Airplane	179.20	57.55	54.71	53.17	51.79	50.38	49.16	47.03
Baboon	129.61	45.38	43.05	41.37	40.51	39.94	39.67	39.12
Boat	129.70	53.37	49.82	47.45	45.58	44.06	43.37	42.04
Peppers	120.21	53.72	50.74	48.98	47.49	46.46	45.98	44.77
Abdominal	101.50	58.81	55.70	53.99	53.17	52.60	52.26	51.50
Sandiego	165.78	52.04	50.03	48.86	47.43	46.57	46.14	45.57
Fruits	173.16	48.63	45.33	43.60	42.26	41.31	40.88	40.05
Car	184.73	54.34	50.23	47.72	45.62	44.19	43.47	41.90

Table 4: The experimental results of the method proposed by [9]

Cover	Ave. pixel	PSNR (dB)						
		10 kb	20 kb	30 kb	40 kb	50 kb	60 kb	70 kb
Airplane	179.20	53.64	49.81	47.12	45.03	43.48	42.40	40.97
Baboon	129.61	44.48	42.36	40.88	40.24	39.83	37.83	36.27
Boat	129.70	53.83	49.99	47.37	45.34	43.87	42.73	41.25
Peppers	120.21	55.13	51.62	49.43	48.04	47.03	46.04	44.92
Abdominal	101.50	65.55	62.45	60.75	59.93	59.33	58.98	58.27
Sandiego	165.78	54.97	53.15	51.90	50.47	49.51	49.08	48.54
Fruits	173.16	45.51	41.89	40.22	39.12	38.23	37.83	37.16
Car	184.73	53.22	49.99	47.80	46.03	44.63	43.95	42.12

per pixel (bpp) on average, which is better than the method in [9] and [8], which produce 3.779 bpp and 0.345 bpp, respectively. It happens because the data embedding in the proposed scheme is processed four times (multiple layers embedding) without paying attention to the block pixel characteristics. Furthermore, the proposed method does not take the pixel pairs in the embedding process, while those existing methods ([8], [9]) used them, which can reduce the embedding capacity.

We can also deduce that embedding data by considering the average images and selecting appropriate cluster boundaries can consistently produce better PSNR values. It depicts that using the lowest or uppermost limit of the cluster while observing the average image always gains the highest value, except for the “Airplane” image when embedded data is 10 kb and 40 kb and the “Car” image when the embedded data is 10 kb.

In this experiment, the best PSNR value is achieved by the abdominal image, 69.9910 dB. Furthermore, the proposed method also produces a stable PSNR value. As in "Baboon's" image, the compared methods generate a PSNR value that significantly differs from that of the other images.

5 Conclusion

In this research, we have recommended a method to maintain image quality and increase embedding capacity by designing and implementing the cluster of pixels. Here, pixel clusters are formed by dividing image pixels into 64 clusters in which each cluster has a range of values between 0 and 3. By applying clusters, the difference between the stego image and the cover image can be reduced. We also suggested the embedding scheme to increase capacity by embedding data four times in every pixel.

The experimental results show that the proposed method obtains a higher value than the others in terms of embedding capacity and PSNR except in certain few images. The proposed method can improve image quality by more than 4 dB in each image and produce four bpp, which is higher than the two existing methods. Nevertheless, further development in increasing the PSNR value and embedding capacity is still needed in further work. It may be applied by investigating the dynamic range of value, different from this proposed static value.

References

[1] Hadlington, L.(2017). Human factors in cybersecurity; examining the link between Internet addiction, impulsivity, attitudes towards cybersecurity, and risky cybersecurity behaviours. *Heliyon*, 3(7), e00346.

[2] Agrawal, H. & Agarkar, A. A. (2019). LRSPPP: lightweight R-LWE-based secure and privacy-preserving scheme for prosumer side network in smart grid. *Heliyon*, 5(3).

[3] Guan, B. & Xu, D. (2020). An efficient high-capacity reversible data hiding scheme for encrypted images. *J. Vis. Commun. Image Represent.*, 66, 102744.

[4] Huang, D. & Wang, J. (2019). Image Communication High-capacity reversible data hiding in encrypted image based on specific encryption process. *Signal Process. Image Commun.*, 80, 115632.

[5] Wang, W., Ye, J., Wang, T., & Wang, W. (2017). Reversible data hiding scheme based on significant-bit-difference expansion. *IET Image Process.*, 11, 1002–1014.

[6] Alattar, A. M. (2004). Reversible watermark using difference expansion of quads. *IEEE Int. Conf. Acoust. Speech, Signal Process.*, 3(1), 377–380.

[7] Maniriho, P. & Ahmad, T. (2019). Information hiding scheme for digital images using difference expansion and modulus function," *J. King Saud Univ. - Comput. Inf. Sci.*, 31(3), 335–347.

[8] Muttaqi, S. R. & Ahmad, T. (2019). A New Data Hiding Method for Protecting Bigger Secret Data. In *12th Int. Conf. Inf. Commun. Technol.*

Syst., (pp. 146–151). IEEE.

[9] Lou, D., Hu, M., & Liu, J. (2009). Multiple layer data hiding scheme for medical images. *Comput. Stand. Interfaces*, 31, 329–335.

[10] Arham, A., Nugroho, H. A., & Adji, T. B. (2017). Multiple layer data hiding scheme based on difference expansion of quad. *Signal Processing*, 137.

[11] Lee, C. F., Shen, J. J., & Lai, Y. H. (2018). Data Hiding Using Multi-Pixel Difference Expansion. In *3rd International Conference on Computer and Communication Systems,* (pp. 61–66). IEEE.

[12] Ilham, A. J., Bawa, P. H., & Ahmad, T. (2019). Protecting Secret Data using RDE and Fuzzy Logic to Specify the Embedding Level. In *IEEE Int. Conf. Signal Image Process. Appl.*, (254–258). IEEE.

[13] Alattar, A. M. (2004). Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. *IEEE Trans. Image Process.*, 13(8), 1147–1156.

[14] Yang, J. & Peng, B. (2017). An Optimized Algorithm Based on Generalized Difference Expansion Method Used for HEVC Reversible Video Information Hiding. In *17th IEEE Int. Conf. Commun. Technol.*, (pp. 1668–1672). IEEE.

[15] "The USC-SIPI Image Database." URL: http://sipi.usc.edu/database/database.php?volume=misc (Accessed: 20 April 2020).

[16] "Dataset of Standard 512x512 Grayscale Test Images." URL: http://decsai.ugr.es/cvg/CG/base.htm (Accessed: 7 May 2020).

[17] "Partners Infectious Disease Images." URL: http://www.idimages.org/ (Accessed: 7 May 2020).

Notes on contributors

Ahmad Juniar Ilham finished his bachelor and master degrees from Universitas Islam Negeri Maulana Malik Ibrahim and Institut Teknologi Sepuluh Nopember, Indonesia, respectively. His research interests are mainly in data hiding and computer network.

Tohari Ahmad obtained the degree of Bachelor of Computer Science from Institut Teknologi Sepuluh Nopember, Indonesia, Master of Information Technology from Monash University, Australia, and Ph.D in Computer Science from RMIT University, Australia. His research interest includes network security, data hiding and computer network.