Bulletin of Electrical Engineering and Informatics
Vol. 8, No. 3, September 2019, pp. 1128~1134
ISSN: 2302-9285, DOI: 10.11591/eei.v8i3.1321

Histogram-based multilayer reversible data hiding method for securing secret data

Chaidir Chalaf Islamy, Tohari Ahmad
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Sep 3, 2018
Revised Jan 1, 2019
Accepted Apr 5, 2019

Keywords:

Data hiding
Data protection
Histogram
Information security

ABSTRACT

In this modern age, data can be easily transferred within networks. This condition has brought the data vulnerable; so they need protection at all times. To minimize this threat, data hiding appears as one of the potential methods to secure data. This protection is done by embedding the secret into various types of data, such as an image. In this case, histogram shifting has been proposed; however, the amount of secret and the respective stego image are still challenging. In this research, we offer a method to improve its performance by performing some steps, for example removing the shifting process and employing multilayer embedding. Here, the embedding is done directly to the peak of the histogram which has been generated by the cover. The experimental results show that this proposed method has a better quality of stego image than existing ones. So, it can be one of possible solutions to protect sensitive data.

Corresponding Author:

Chaidir Chalaf Islamy,
Department of Informatics,
Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.
Email: chaidir31@gmail.com

1. INTRODUCTION

The use of digital media such as video, image, and audio plays a vital role in communication and data transmission on the internet. However, this rapid development results in various problems, such as insecurity of the transmitted data. The application of data protection is critical in many fields such as military and medical data. In order to protect the data, some techniques have been presented. One of them is cryptography that the protection is done by converting the data into an unrecognizable form. The implementation of this method can be found in [1, 2]. Another method to secure data is steganography or data hiding. In this algorithm, we can keep the data confidential without arousing suspicion that there is a secret in it. Data hiding can be applied to various media, one of which is an image.

A suitable data hiding method should produce a stego image which is able to hold a large payload and has a low distortion level [3-5]. The common problem in this research is how to balance those two factors. According to [6], the more data being embedded in an image leads to decreasing the quality of the stego data, which means that it has been much distorted. On the contrary, a stego image with less payload produces less distortion. In this research, we would like to work on these problems by exploring the histogram of the cover image. The bases and research of data hiding are provided in Section 2, while the proposed method is in Section 3. Next, the experimental results and the respective analysis are presented in Section 4; it is followed by the conclusion in Section 5.

2. DATA PROTECTION METHODS

Difference Expansion (DE) is one of the protection algorithms which is firstly proposed by Tian et al. [7]. This method is the forerunner of the RDE (Reduced Difference Expansion) method [8]. Al Huti et

al. [9] improve the capacity of payload data by investigating the size of the blocks of RDE. They expand the size of pixel blocks to 4x4, intending to decrease the difference between pixels which brings higher capacity. The experimental results show that there is an improvement regarding the capacity. Angreni and Ahmad [10] seek to improve the quality of the image by avoiding the overflow and underflow problems in the RDE method. As in other RDE-based methods, this process requires the difference of pixel values; however, their algorithm employs a random value instead of a pixel for the paired in a block. Similarly, this difference holds a bit for each virtual block. According to the experimental results, it is shown that their performance is superior.

Another method that can be used to perform data embedding on the cover image is the interpolation. In general, this method increases the size of the image resolution. For example, the method which is proposed by Benhfid et al. [11] which develops bicubic interpolation methods, nearest neighbors, and bilinear. It has been proved that their method is able to raise the capacity of data to store. In further research, the data hiding method can be performed in the encryption domain. Data embedding is performed after the original image has been encrypted, as shown in [4, 12, 13].

Another reliable approach is histogram shifting. The benefit of this method is that it has low image distortion, so the quality of the stego image is relatively high. Firstly introduced by Ni et al. [14], the histogram shifting method embeds secret data into the original image by manipulating the histogram. Here, the pixel value in the histogram is shifted so that it creates a space that can be used as a place to embed the secret. The process is done by scanning the pixel value in the original image to find the highest (the peak) frequency and the lowest (the zero) pixel value. This condition determines the direction of the shifting: left and right. If it is to the left, then the respective value is reduced; otherwise, it is added. Let the peak and the zero points be X and Y, respectively; and the secret bit is $z(n)$. The pixel value of the image before shifting is P and after shifting is P' with the position (i, j) is P_{ij} and P'_{ij} (see (1).

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

After the original image has been shifted, the process of embedding secret data starts. Secret data are inserted in a space or pixel whose frequency is zero as a result of the shifting process. We take an example that the peak pixel value is 174 and the zero value is 3. The algorithm scans the image to find pixels whose value is 174. If the secret bit $z(n)$ is 1, then the value of the respective pixel is reduced to 173. This value results in filling the space of 173 which previously is empty. However, if the secret bit $z(n)$ is 0, then the pixel value does not change and the process continues to find the next 174. This operation can be illustrated in (2).

$$P'_{ij} = \begin{cases} P_{ij} + 1 & if\ P_{ij} = X\ and\ z(n) = 1, X < Y \\ P_{ij} - 1 & if\ P_{ij} = X\ and\ z(n) = 1, X > Y \\ P_{ij} & if\ P_{ij} = X\ and\ z(n) = 0 \end{cases} \quad (2)$$

Pan et al. [15] improve the method in [14] by increasing the space to hide the secret data. They use the peak point as the reference without the minimum point. Similarly, the process of shifting is done two ways, left and right. Once the peak pixel value is found, the neighboring pixels of the left and right of the peak are shifted. The shifting process produces two empty spaces to embed the secret data. Suppose the peak is X, the left neighbor of X is $X - 1$ and the right neighbor of X is $X + 1$. We shift the left neighbor of $X - 1$ or $X - 2$ to the left and right neighbor of $X + 1$ or $X + 2$ to the right. After that, the process scans the image. If the pixel value is from $X + 2$ to 254, then the respective value is added by one. If the pixel value is $1\ X - 2$ then it is reduced by one as depicted in (3).

$$P'_{ij} = \begin{cases} P_{ij} + 1 & if\ X + 2 \le P_{ij} \le 254 \\ P_{ij} - 1 & if\ X - 2 \ge P_{ij} \ge 1 \end{cases} \quad (3)$$

In the process of hiding the secret data, each bit is embedded in both empty spaces, $X - 2$ and $X + 2$. Again, scan the image, if a pixel with the value of $X + 1$ is found and the secret bit is 1 then the pixel value of $X + 1$ is added by one. If the pixel with the value of $X - 1$ is found and the secret bit is 1, then the pixel value is reduced by one as provided in (4). As in the previous example, if in the scanning process we find the pixel with the value 175, then that pixel is added by one and the value has become 176. If the value is 173 then it is reduced by one to have 172.

$$P'_{ij} = \begin{cases} P_{ij} + 1 & if\ X + 1\ and\ z(n) = 1 \\ P_{ij} - 1 & if\ X - 1\ and\ z(n) = 1 \\ P_{ij} & if\ z(n) = 0 \end{cases} \quad (4)$$

Wu et al. [16] increase the brightness level of the original image by modifying its histogram. This enhancement aims to improve both the quality of the stego and provide more spaces to hold the secret. Chen et al. [17] revise the methods of Wu et al. [9] by determining the process of histogram shifting adaptive to the distribution of histogram characteristics. Furthermore, they explore the pixel value ordering for increasing the data capacity. Next, Prabowo and Ahmad [18] use Adaptive Pixel Value Grouping to protect the secret data. Their method is also in the field of histogram-based embedding. Another example in applying histogram modification using the residual histogram is shown in [19].

Those previously discussed histogram-based methods employ a shifting process to generate empty space for embedding. In facts, this shifting process has an impact on the quality of the stego image. In this paper, we enhance the stego image quality by further exploring the shifting of the histogram. Here, the shifting is not done before the embedding process. So, the secret is directly inserted to additional preprocessing of the cover image. In addition, we use multilayer embedding in the whole process.

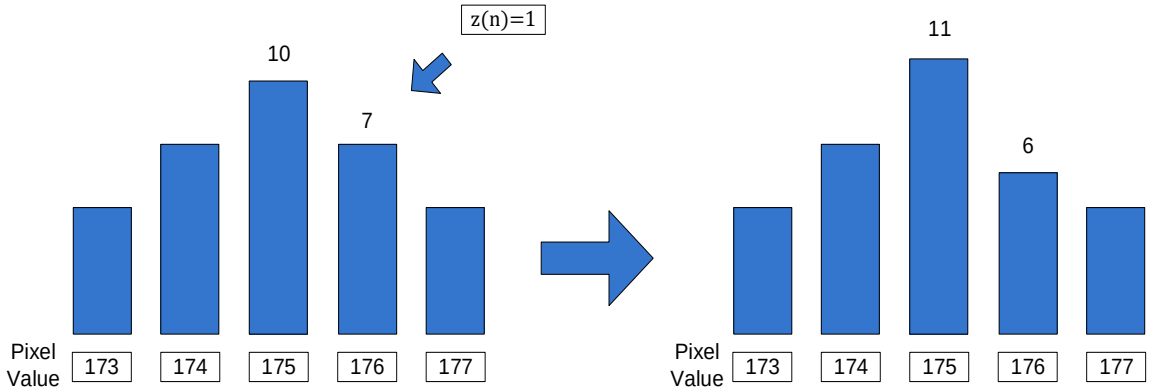
3. RESEARCH METHOD

To protect the secret data, we use the same principle as the method of Ni et al. [14], that the embedding is done in the specified histogram. In contrast to that algorithm, where secret data are embedded in empty spaces, we embed them in the peak, without shifting pixel values. By assuming that the peak pixel value of the original image is X, and the right neighbor of X is $X + 1$, we take an example as follows. We have X is 175; so, $X + 1$ is 176. Then, we scan the cover image pixel by pixel. If the pixel with the value of $M + 1$ is found and the secret $z(n)$ to be embedded is 1, then the pixel value of $X + 1$ is reduced by one. The value of $X + 1$ does not change if the secret data $z(n)$ is 0. The illustration of data insertion is provided in Figure 1 and the embedding process is elaborated in (5).

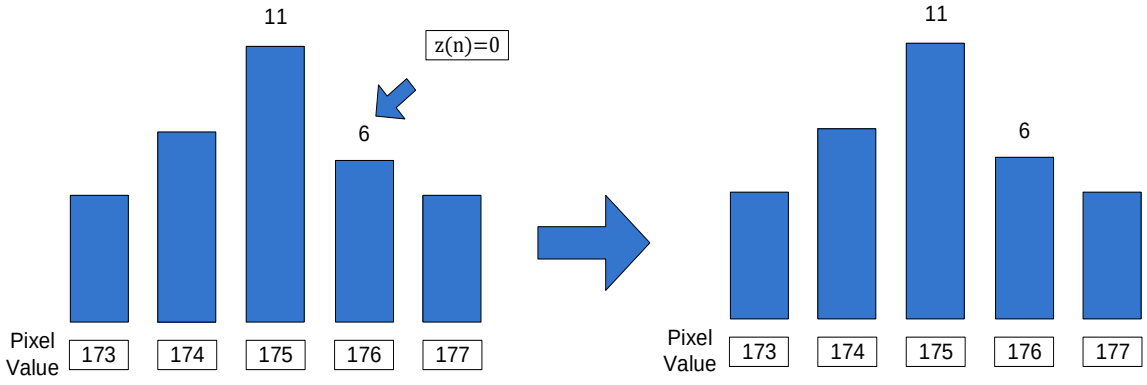
$$P'_{ij} = \begin{cases} P_{ij} - 1 & if\ P_{ij} = X + 1\ and\ z(n) = 1 \\ P_{ij} & if\ P_{ij} = X + 1\ and\ z(n) = 0 \end{cases} \quad (5)$$

In order to extract the secret bits, we need information of the frequency of the original pixels. It is because this proposed method does not provide a specific space for bit 1 as in [14] and [15]. The number of bit 1 of the secret can be found by subtracting the frequency of the peak of the whole cover after being embedded (F') by that before embedding (F). So, if the number of bit 1 of the secret is G, then $G = F' - F$. In the scanning process for extracting the secret, if we find a pixel M while $G > 0$, then extract the secret of bit 1; add the pixel value by 1, and substract G by 1. In case we find $M + 1$ in the scanning, bit 0 of the secret is extracted. This process is provided in (6).

$$P_{ij} = \begin{cases} P_{ij} + 1\ and\ z(n) = 1\ and\ G = G - 1 & if\ P'_{ij} = X\ and\ G > 0 \\ P_{ij}\ and\ z(n) = 0 & if\ P'_{ij} = X + 1 \end{cases} \quad (6)$$



(a)



(b)

Figure 1. An example of the hiding process, (a) Embedding with $z(n)$=1, (b) Embedding with $z(n)$=0

3.1. Multilayer embedding

To increase the size of data that can be hidden in the original image we implement the concept of multilayer embedding. Here, the resulted stego image of the embedding process (layer) is split into non-overlapping blocks. This method can secure the embedded image, so the public may not be aware of the resulted stego image [15].

In the first layer, we divide the original image into blocks of $s \times s$, with s is the half of the width and height of the image. The data embedding process is applied to each block using (7) where B_{ij} is the image of each block and B'_{ij} is the block of the embedded secret data. Then, the blocks are merged to obtain an image with the original size. In the next layer, we separate the image that has been embedded in the first layer. In this layer, the block size is $s \times (2s)$. Each block is also embedded with secret data like the process in the first layer and then merged to have one image. In the third layer, the same process is carried out as in the previous layers; however, the block size is $(2s) \times s$. Overall, this multilayer embedding is shown in Figure 2.

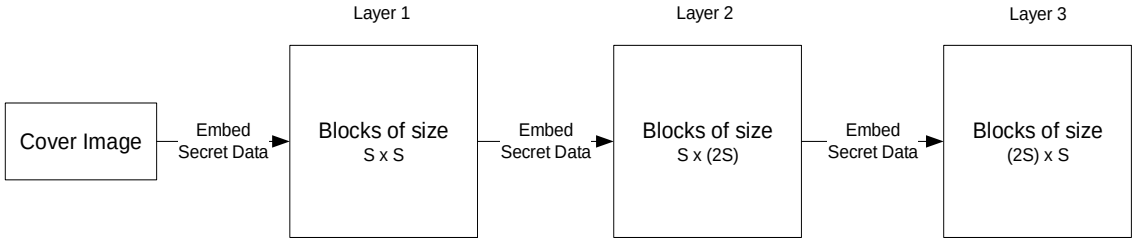


Figure 2. Multilayer embedding

3.2. Embedding process

Because we use multilayer embedding, we split each layer into non-overlapping blocks, where the process of embedding data is done in those blocks. We do not need the minimum point because no pixel value should be shifted. Let the peak value of a block be Xb and its right neighbor be $Xb + 1$. If in the scanning process a pixel whose value is $Xb + 1$ is found and the secret bit $z(n)$ is 0, then the value of the respective pixel does not change. On the contrary, if $z(n) = 1$ then it is reduced by 1. By assuming that B_{ij} is the pixel value of the original image B at position (i, j) is and B'_{ij} is the pixel value of the stego image B' at position (i, j), that process can be described in (7).

$$B'_{ij} = \begin{cases} B_{ij} - 1 & if\ B_{ij} = Mb + 1\ and\ z(n) = 1 \\ B_{ij} & if\ B_{ij} = Mb + 1\ and\ z(n) = 0 \end{cases} \quad (7)$$

3.3. Extraction process on image blocks

As previously explained, before performing the data embedding, the peak value of the original image is stored for the extraction process. Supposed the peak value of a block in the cover image is BF and that of stego image is BF'. Equivalent to the processing the whole cover image, in this block level, the value GB, is obtained from the difference between BF' and BF.

In [14] and [15], the number of bit 1 of the secret is already known because both methods provide an empty space for the bit 1. In this proposed method, we do not allocate it, so the difference of F' and F is required. In general, this process can be described in (8).

$$B_{ij} = \begin{cases} B_{ij} + 1 \; and \; z(n) = 1 \; and \; GB = GB - 1 & if \; B'_{ij} = Xb \; and \; GB > 0 \\ B_{ij} \; and \; z(n) = 0 & if \; B'_{ij} = Xb + 1 \end{cases} \quad (8)$$

4. RESULTS AND ANALYSIS

For the experiment, we use the images of "Baboon", "Lena", “Pepper” which taken from [20] and medical images: “Abdominal”, “Chest” and “Hand” from [21]. All those images are 512x512 pixels in size. The hardware we use is a PC with Ryzen 5 1400 CPU, 8GB RAM, Nvidia GTX 1050Ti GPU and 1TB 7200RPM storage media.

In this evaluation, we compare our method with [15] and [14], regarding the quality of stego image and the size of secret data that can be put into the original image. In each layer, we test the amount of data which can be embedded in the original image for both our and the compared methods whose results are provided in Tables 1, 2 and 3.

Table 1. PSNR value of stego images (layer 1)

Cover Image	PSNR			Payload Size
	Proposed	Ni et al. [14]	Pan et al. [15]	
Pepper	67,5668	50,6892	48,2243	2 KB
Baboon	68,8118	52,3366	48,2006	2 KB
Lena	66,7737	50,1295	48,2438	2 KB
Abdominal	66,0796	59,6239	49,7144	10 KB
Chest	65,2988	51,1278	50,3845	10 KB
Hand	62,9465	48,8758	48,5629	20 KB

Table 2. PSNR value of stego images (layer 2)

Cover Image	PSNR			Payload Size
	Proposed	Ni et al. [14]	Pan et al. [15]	
Pepper	66,3338	45,8150	43,0364	4 KB
Baboon	65,4134	46,3737	42,2979	4 KB
Lena	65,7189	46,0483	43,7989	4 KB
Abdominal	49,1067	48,7216	42,6768	10 KB
Chest	50,5732	45,0127	43,3652	10 KB
Hand	48,9795	42,3795	41,8346	20 KB

Table 3. PSNR value of stego images (layer 3)

Cover Image	PSNR			Payload Size
	Proposed	Ni et al. [14]	Pan et al. [15]	
Pepper	66,5355	42,8267	40,0172	4 KB
Baboon	67,0826	43,7545	39,8353	4 KB
Lena	64,8506	42,9889	40,2543	4 KB
Abdominal	49,0925	48,6078	39,7364	10 KB
Chest	50,5479	42,6293	40,2899	10 KB
Hand	48,9387	39,2916	38,7742	20 KB

4.1. Quality of stego image

The results of the experiment in Tables 1, 2 and 3 show the respective quality of the stego images that is measured by Peak Signal to Noise Ratio (PSNR) from Layer 1 to Layer 3, respectively. As predicted, increasing the number of layers results in decreasing the quality. However, the number of bits raises since more layers hold more data. It is found that in our proposed method, the drop is not as significant as that of Ni et al. [14] and Pan et al. [15], especially for general images.

For medical images, the decrease of ours is slightly higher than the other two methods. The comparison of the embedded medical image shows that our proposed method is still superior in preserving stego image quality. However, our method has decreased dramatically in the second layer. Then, in the third layer, the quality of stego images of our method has decreased but not as significant as that from layer one to layer two. In general, our generated stego images have higher PSNR value than others. This experiment shows that the method we propose is able to keep the quality of the stego image effectively.

In all three layers, our proposed method is still able to achieve more than 60 dB of PSNR for general images (Baboon, Lena and Pepper) and more than 45 dB on medical images. In Ni et al. [14] and Pan et al. [15], before the data embedding, there is a process of shifting the pixel values of the original image. This process has a significant impact on stego image quality. As previously discussed, in this proposed method, we do not shift the original pixels. An example of the cover image and its stego images are provided in Figure 3.

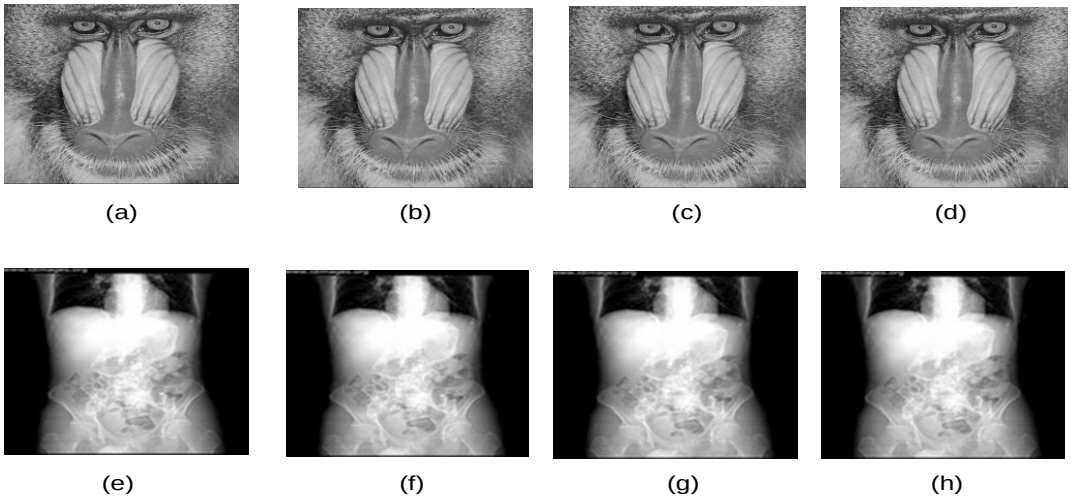


Figure 3. Cover and stego images, (a) Original Baboon image, (b) Layer 3 embedding with the proposed method, (c) Layer 3 embedding with Ni et al. [14], (d) Layer 3 embedding with Pan et al. [15], (e) Original Abdominal Image, (f) Layer 3 embedding with the proposed method, (g) Layer 3 embedding with Ni et al. [14] (h) Layer 3 embedding with Pan et al. [15]

4.2. Capacity of stego image

The data capacity that can be accommodated in the original image is shown in Table 4. The results we obtained at the maximum capacity test show that the method we proposed has the lowest capacity in all of the original images we tested. In Pepper, Baboon and Lena, the method of Pan et al. [15] have the highest capacity. It is because their method uses the peak value of the left and right neighbors of the image as a space for embedding secret data. Whereas in medical images, the method of Ni et al. [14] is able to hold the most secret data. In the medical image, the value of the peak is relatively high and its difference from the neighboring pixel is significant. On the other hand, our method and Pan et al. [15] do not use the peak pixel value as the location to store the secret data. So, the capacity that can be accommodated is lower.

Table 4. Total Capacity of the embedded secret data from layer 1 to layer 3

Cover Image	Capacity (Bit(s))		
	Proposed	Ni et al. [14]	Pan et al. [15]
Pepper	6037	10295	13920
Baboon	6440	8519	13942
Lena	7047	11953	18642
Abdominal	9966	198906	13040
Chest	11302	276511	16090
Hand	30314	39959	54839

Multilayer embedding plays a primary role in increasing the capacity of secret data. Apparently, the increase in capacity also followed by some quality lost. However, this trade-off is still worthy because the decrease is still in the tolerable range.

5. CONCLUSION

In this paper, we have proposed a variation of data hiding methods base on the histogram shifting. Different from the previous research, we do not shift pixel values; instead, we directly embed the secret. It is shown that this method is able to raise the quality of the stego image. In order to compensate the capacity, we explore the multilayer embedding. Overall, there is a trade-off between the capacity and the quality. In facts, we may determine whether the focus is on the capacity or quality. Additionally, we may specify the balancing point between those two factors.mIn the future, this research can be further extended to have a higher quality level. Applying histogram shifting only in the specific value, is a possible way to do. Hopefully, this can maintain the capacity of the embedding space.

REFERENCES

[1] E. R. Arboleda, J. L. Balaba, and J. C. L. Espineli, "Chaotic rivest-shamir-adlerman algorithm with data encryption standard scheduling," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 6, no. 3, pp. 219-227. 2017.

[2] A. O. Mulani and P. B. Mane, "Watermarking and cryptography based image authentication on reconfigurable platform," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 6, no.2, pp. 181-187. 2017.

[3] C. Qin, W. Zhang, F. Cao, X. Zhang, and C.-C. Chang, "Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Processing*, vol. 153, pp. 109–122, 2018.

[4] 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.

[5] 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.

[6] M. Manju and V. Kavitha, "Survey on reversible data hiding techniques," *Int. J. Secur. its Appl.*, vol. 8, no. 4, pp. 297–306, 2014.

[7] Jun Tian, "Reversible data embedding using a difference expansion," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

[8] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," in *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156, Aug. 2004.

[9] M. H. A. Al Huti, T. Ahmad, and S. Djanali, "Increasing the capacity of the secret data using DEpixels blocks and adjusted RDE-based on grayscale images," *in Proceedings of 2015 International Conference on Information and Communication Technology and Systems, ICTS 2015*, pp. 225–230. 2016.

[10] D. S. Angreni and T. Ahmad, "Enhancing DE-based data hiding method by controlling the expansion," in *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 2016.

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

[12] 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.

[13] S. Agrawal and M. Kumar, "Mean value based reversible data hiding in encrypted images," *Optik (Stuttg).*, vol. 130, pp. 922–934, 2017.

[14] Zhicheng Ni, Yun-Qing Shi, N. Ansari and Wei Su, "Reversible data hiding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, March 2006.

[15] Z. Pan, S. Hu, X. Ma, and L. Wang, "Reversible data hiding based on local histogram shifting with multilayer embedding," *J. Vis. Commun. Image Represent.*, vol. 31, pp. 64–74, 2015.

[16] H. Wu, J. Dugelay and Y. Shi, "Reversible Image Data Hiding with Contrast Enhancement," in *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 81-85, Jan. 2015.

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

[18] H. Eko Prabowo and T. Ahmad, "Adaptive Pixel Value Grouping for Protecting Secret Data in Public Computer NetworksNo Title," *J. Commun.*, vol. 13, no. 6, pp. 325–332, 2018.

[19] I. C. Chang, Y. C. Hu, W. L. Chen, and C. C. Lo, "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding," *Signal Processing*, vol. 108, pp. 376–388, 2015.

[20] "SIPI Image Database." [Online]. Available: http://sipi.usc.edu/database/database.php?volume=misc. [Accessed: 28-Apr-2018].

[21] "eMicrobes Digital Library." [Online]. Available: https://www.idimages.org/. [Accessed: 28-Apr-2018].