Bull. Soc. Photogr. Imag. Japan. (2023) Vol. 33 No. 1: 1–6

Original Paper

Security Enhancement for Reversible Data Hiding for Palette-Based Images

Daichi NAKAYA[a], Shoko IMAIZUMI[b]

Abstract: In this paper, we propose an extended reversible data hiding method for palette-based images that enhances payload security. In a previous method, multiple index colors appear in duplicate in a color palette as a result of the data hiding process. This leads to the locations where payload bits have been embedded being disclosed, so the security for the payload is weakened. In contrast, the proposed method configures a new approximate color for each target color without reducing the total number of index colors. Consequently, this method can derive high-quality marked images while maintaining the security of payload bits. Experimental results show the effectiveness of the proposed method in terms of image quality using peak signal to noise ratio (PSNR)/structural similarity (SSIM) and the number of reduced colors.

Key words: Reversible data hiding, palette-based image, hiding capacity, enhanced safety, image quality

1. Introduction

With the development of internet services such as social networking services and cloud services, many images have been exchanged over the Internet. In addition, unauthorized secondary use of images that infringes on copyrights is seriously increasing. As one solution for copyright protection, data hiding methods have been proposed [1)–8)]. With these methods, we can directly embed copyright/authorship information and other arbitrary data into an image. The main advantage of data hiding is that a payload can be secretly hidden without increasing the data amount of the image. Data hiding methods can be broadly classified into reversible and irreversible methods. In particular, reversible data hiding (RDH) methods can fully recover the original image after extracting the payload from the marked image. This feature of RDH is strongly requested in the field of medical, military, and satellite images.

RDH methods have been actively studied in recent years [9)–15)]. Most of these studies have focused on grayscale or full-color images. In contrast, palette-based images are still popularly used on social networking services and in electronic papers. A palette-based image is a color image composed of a limited number colors, e.g., 256 colors. The construction of a palette-based image is shown in Fig. 1. Each pixel holds an index number. As shown in Fig. 1 (a), the color and indices are stored in a color palette. The colors in the palette are called index colors. By limiting the colors, we can greatly reduce the data amount of an image compared with a full-color image. A palette-based image of Lena is shown in Fig. 1 (b). However, there are few RDH studies on palette-based images [16)–20)].

One RDH method for palette-based images is based on histogram analysis [16),17)]. This method has a high capacity of 0.5 bpp for images with large homochromatic regions. The hiding capacity is, however, variable depending on the texture of each target image and seriously small in some cases. Additionally, this method is effective only when there exist unused colors in the color palette. Wu et al. extended the above method to increase the number of effective cases [18)]. Their method first performs preprocessing to newly produce unused colors in the color palette before the embedding process. Then, some particular colors are duplicated for multiple indices. A payload is embedded in the duplicated colors. Thus, a third party can easily find the location of the embedded payload and extract it. Consequently, the quality of the marked images is relatively high, but the security of the payload is sacrificed. As another approach to RDH for palette-based images, two methods in the encrypted domain have been proposed [19),20)]. In these methods, the original palette-based images are first encrypted. A payload is then embedded into the encrypted images.

Index number		R	G	B
1		72	9	57
2		79	9	53
3		96	10	63
...	...	...	...	...
255		250	217	173
256		243	220	197

(a) Color palette

(b) Image

Fig. 1. Structure of palette-based image (Lena)

In these methods, the decryption process and the data extraction process are fully separable. Consequently, there exist three types of

Received 20th December, 2022. Accepted 20th January, 2023. Published 2nd March, 2023.

[a] Graduate School of Science and Engineering, Chiba University
1-33 Yayoicho, Inage-ku, Chiba-shi, Chiba 263-8522, Japan
[b] Graduate School of Engineering, Chiba University
1-33 Yayoicho, Inage-ku, Chiba-shi, Chiba 263-8522, Japan

privilege for receivers:(i) data extraction only, (ii) decryption only, and (iii) data extraction and decryption. The RDH algorithms of these methods have been developed for encrypted images. Thus, it is difficult to directly apply them to the RDH methods for the plain domain.

Another method [21)] can be applied to both plain and encrypted domains. We first rearrange index colors within a color palette so that analogous colors get close to each other. A payload is then embedded using an arbitrary RDH method for grayscale or full-color images. The main contribution of this method is that the rearrangement process alleviates the distortion caused by applying the traditional RDH method to palette-based images. However, compared with the RDH methods for palette-based images [16)-18)], this method generally produces marked images with lower quality.

In this paper, we focus on the issue with one of the previous methods [18)] and propose an extended RDH method to improve the payload security. Instead of reducing the number of index colors, some index colors are updated with new approximate colors of the target colors. Approximate colors are colors that are not included in the original color palette and have a small color difference from the target colors. A payload is embedded in the approximate colors. Accordingly, the number of index colors is not reduced but preserved. The proposed method needs to store essential data for restoration, i.e., data on the original index colors before palette updating. We call such data additional data and embed the data together with the payload. The main contribution of this paper is the improvement of payload security without decreased marked-image quality. We demonstrate the usefulness of our method in terms of marked-image quality, hiding capacity, and the number of index color reductions.

2. Previous Method

One of the RDH methods for palette-based images generates a marked image with high quality by reducing the number of index colors [18)]. Specifically, an index color with the lowest frequency is merged with another index color that is similar to the color to create an empty index. In contrast, an index color with the highest frequency is a target color for data hiding and duplicated to the empty index. A payload bit is embedded into each pixel with the target color. In the case that the payload bit is 0, the index of the target pixel is unchanged. Otherwise, the index of the target pixel is shifted to the other index where the index color has been duplicated. Consequently, no distortion should be caused by the data hiding process, while the preprocessing leads to some distortion through the integration of two index colors. Figure 2 shows the flow of the previous method. We describe the preprocessing, the embedding process, and the data extraction process and summarize the features of this method.

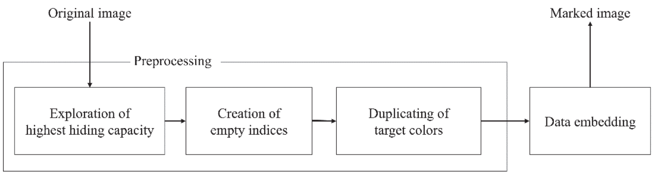

Fig. 2. Flow of previous method

2.1 Preprocessing

The preprocessing consists of the following three processes. We describe each process in detail.

2.1.1 Exploration of Highest Hiding Capacity

The previous method creates empty indices by integrating index colors with the lowest frequency into their closest colors. Closest color means another index color in the color palette with the smallest color difference from the target color. First, the number of empty indices is determined. Following this number, a single pattern, where the highest hiding capacity is attained, is explored from a pattern table.

Table 1 shows an example of a pattern table for an image, Airplane, with 512 × 512 pixels. In this case, there are six patterns, and ten index colors with the highest frequencies are listed. The number of embedded bits for each of the index colors is shown in the table. In the case that multiple bits would be embedded into a single index color, we need to duplicate the index colors for multiple empty indices. For instance, when we embed 2-bit data into an index color, we previously duplicate the index color for three of the empty indices. In this table, we assume that ten empty indices are created.

2.1.2 Creation of Empty Indices

Here, an empty index is created by integrating an index color with the lowest frequency into its closest color. This process is repeated to generate the required number of empty indices. Note that we cannot choose the closest color from the other target colors and the index colors that have been already merged with another index color.

2.1.3 Duplicating of Colors to be Embedded

The target colors are duplicated using the empty indices on the basis of the embedding pattern. For reversibility, the restoration process requires the original RGB values of the index colors, which are located in the empty indices. Thus, we embed these values as addi-

Table 1. Payload amount for each embedding pattern (Airplane)

Pattern number	Number of payload bits for top 10 colors with high frequency										Payload amount [bits]
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	
1	1	1	1	1	1	1	1	1	1	1	109,502
2	2	1	1	1	1	1	1	1	1	0	117,955
3	2	2	1	1	1	1	0	0	0	0	115,902
4	2	2	2	1	0	0	0	0	0	0	109,502
5	3	1	1	1	0	0	0	0	0	0	102,472
6	3	2	0	0	0	0	0	0	0	0	92,864

tional data along with the pure payload.

2.2 Data Embedding Process

The payload is embedded on the basis of the embedding pattern. In Pattern 2 of Table 1, for example, 2-bit data is embedded into a single index color with the highest frequency. This color is located in four indices, so the index of each pixel with this color is replaced with one of the four indices. Even when the index is replaced with another index, the index color itself is never changed in this method.

2.3 Restoration Process

This section describes data extraction and image restoration. The indices where the same index color is assigned are first explored from the color palette. The embedded bits can be extracted from the indices, and then the indices are restored to their originals. Finally, the index colors eliminated by the preprocessing are restored using the additional data so that the original image is reconstructed.

2.4 Features of Previous Method

The target color is duplicated in the preprocessing. Therefore, the distortion in the previous method is caused only by integrating index colors with low frequencies into their similar colors. The embedding process never produces any distortion.

In terms of the security of the payload, however, there exists a serious issue. Generally, each index has a unique color in palette-based images. In contrast, multiple indices have an equal color in the previous method. Consequently, a third party can easily detect where the payload has been embedded. In the next section, we propose an extended method to tackle this issue.

3. Proposed Method

We propose an RDH method that avoids severe deterioration in image quality from data embedding. Compared with the above-mentioned previous method [18], our method considers the security of the payload by concealing the embedded field. Figure 3 shows the flow of the proposed method. The procedure of the proposed method can be roughly divided into two processes: pre-processing and data embedding processes. The preprocessing further contains two steps: creation of empty indices and definition of approximate colors for target colors. Each process and step are described in detail below.

3.1 Preprocessing

Here, we create empty indices and then define the approximate colors for target colors.

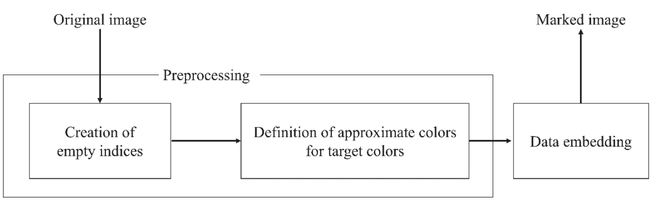

Fig. 3. Flow of proposed method

3.1.1 Creation of Empty Indices

We first define the number of empty indices E according to the payload amount. For all index colors, we explore the closest index color to each one in the color palette. The total frequencies of each color pair are then calculated. We extract the E of the color pairs in ascending order of the total frequency. In each of the extracted pairs, the index color with the lower frequency is merged into the other one so that the empty indices are prepared.

3.1.2 Definition of Approximate Colors for Target Colors

For each empty index, an approximate color for a target color is derived. The approximate color should be located between the target and closest colors; that is, the approximate color should be closer to the target color than the present closest-color and will be a unique color in the color palette. We describe the algorithm in detail below.

For the target color with the highest frequency, the closest color to the target one is explored in the color palette. We create a new color called an approximate color using the target and closest colors. The absolute R, G, and B differences between the target and closest colors is calculated. On the basis of these difference values, the R, G, and B values of the approximate color are determined using pseudo-random numbers. Note that the possible ranges of the RGB components in the approximate color are slightly changed depending on which component has the largest absolute color difference. The color component with the largest absolute color difference between the target and closest colors is called the largest component in this paper. The largest component of the approximate color should be determined so as not to have the same value as that of the target color. If the largest component is R, the possible ranges for each component of the approximate color are given by

$$\begin{cases} R_1 < R_{upd} \leq R_1 + \frac{R_2 - R_1}{2}, \\ G_1 \leq G_{upd} \leq G_1 + \frac{G_2 - G_1}{2}, \\ B_1 \leq B_{upd} \leq B_1 + \frac{B_2 - B_1}{2}, \end{cases} \tag{1}$$

where R_1, G_1, and B_1 are the color components of the target color, R_2, G_2, and B_2 are those of the closest color, and R_{upd}, G_{upd}, and B_{upd} are those of the approximate color. Similarly, in the case that the *largest* component is G or B, the possible ranges are given by Eqs.(2) and (3), respectively:

$$\begin{cases} R_1 \leq R_{upd} \leq R_1 + \frac{R_2 - R_1}{2}, \\ G_1 < G_{upd} \leq G_1 + \frac{G_2 - G_1}{2}, \\ B_1 \leq B_{upd} \leq B_1 + \frac{B_2 - B_1}{2}, \end{cases} \tag{2}$$

$$\begin{cases} R_1 \leq R_{upd} \leq R_1 + \frac{R_2 - R_1}{2}, \\ G_1 \leq G_{upd} \leq G_1 + \frac{G_2 - G_1}{2}, \\ B_1 < B_{upd} \leq B_1 + \frac{B_2 - B_1}{2}. \end{cases} \tag{3}$$

If there are multiple color components with the largest absolute color difference, we give priority in the order of R, G, and B. For example, when the R and G components have the largest absolute color difference, the largest component is assigned to R, and Eq.(1) is applied. After deriving the approximate color, we assign the color to an empty index. The series of the procedure is repeated E times, and the color palette is updated.

3.2 Data Embedding Process

Each payload bit is embedded into a single pixel with a target color in raster scan order. In the case that a payload bit is 0, a pixel value with a target color remains invariant. In contrast, when the payload bit is 1, the pixel value is changed to an index of the approximate color. The embedding process is repeated E times.

A payload consists of a pure payload and additional data for reversibility. The additional data is located at the head of the payload followed by the pure payload. Specifically, the index of the first target color, which is 8-bit data, is embedded into the least significant bits (LSBs) of the first eight pixels in the top row. The pixels are excluded from both the preprocessing and embedding process. The original LSBs should be included into the additional data and embedded into the image.

3.3 Restoration Process

We extract the data of the target color, where the payload bits have been first embedded, from the LSBs of the first eight pixels in the top row. The closest color to the target color is explored in the color palette. This closest color is the approximate color in the embedding process, so we call this color the approximate color. We extract the payload bits from all the pixels with the target or approximate color in raster scan order. Note that the first eight bits are excluded from the restoration process. The indices of the pixels with the approximate color are then turned back to the index of the target color. We repeat the series of the procedure E times. Finally, the index colors and pixels, which have been modified in the preprocessing, are restored using the additional data so that we reconstruct the original image.

3.4 Advantages of Proposed Method

The preprocessing in our method newly defines a color approximate to each target color. The approximate color is closest to the target color and does not exist in the original color palette. Therefore, the distortion caused in the marked image can be efficiently suppressed in the embedding process.

The main distortion is caused by integrating pairs of index colors to create empty indices. In the integration, the distortion can be reduced by setting a threshold for the color difference. Practically, when the color difference exceeds the threshold, the pair is excluded from the integration process.

In contrast to the previous method, the proposed method never produces identical colors in the color palette. Consequently, it is difficult for a third party to identify the embedded field, so our method improves the security of the payload.

4. Simulation

We confirmed the effectiveness of the proposed method by comparing it with the previous method in terms of marked image quality and hiding capacity [18]. For the image quality assessment, we adopted the peak signal to noise ratio (PSNR) and structural similarity (SSIM) under the same amount of payload (0.31 bpp on average). In our simulation, we used ten images with 512 × 512 pixels from the USCSIPI image database. The number of colors in each image was reduced to 256 by using GNU Image Manipulation Program (GIMP) 2.10. In the proposed method, the threshold of color difference in the integration of two index colors was defined as 25.

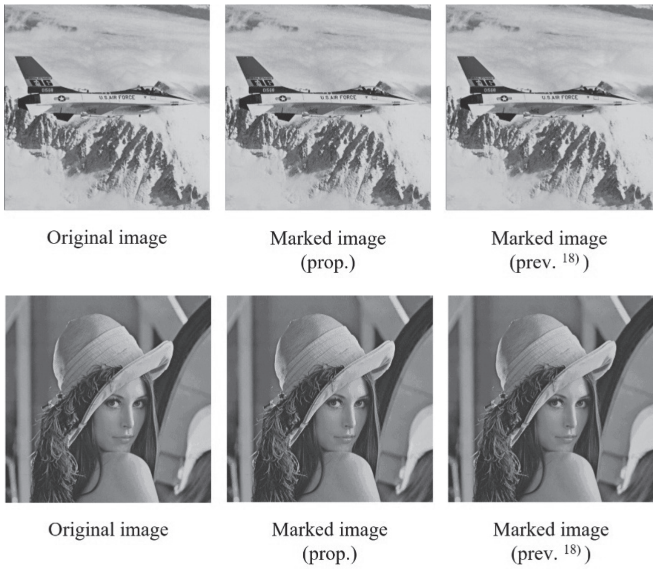

Fig. 4. Comparison of marked images

4.1 Evaluation of Image Quality and Number of Reduced Colors

Figure 4 shows two examples of an original and two marked images; one was obtained by the proposed method, the other was derived by the previous method. The payload amounts of the marked images were equalized to the hiding capacity of one of the two methods and resulted in 0.31 bpp on average. As shown in Fig. 4, the marked images from both methods have high quality without any artifacts.

Table 2 summarizes the quantitative evaluation results using PSNR and SSIM. In this table, the bold numbers indicate that the proposed method achieved higher quality than the previous method. In five out of the ten test images, the proposed method outperformed the previous method. In regard to the average, the average PSNR value for the proposed method was higher than that for the previous method. Additionally, the proposed method never reduced the number of index colors by data embedding, while the previous method eliminated 39.9 colors on average.

4.2 Evaluation of Hiding Capacity and Number of Reduced Colors

Table 3 shows the hiding capacity for each test image. The hiding capacity of the previous method exceeded that of the proposed method in eight out of ten test images. In particular, the previous method achieved 1.37 bpp for the Airplane image. Let us analyze in more detail the result. The histogram distribution of Airplane is highly skewed. In the previous method, there exist multiple embedding patterns, and the most efficient pattern should be explored. When the histogram distribution is highly skewed, we can increase the capacity by embedding multiple bits into a couple of the index colors in decreasing order of frequency. In the case of Lena, contrarily, the hiding capacity of the proposed method surpassed that of the

Table 2. Image quality and number of reduced colors

	PSNR[dB]		SSIM		Payload amount [bpp]		Number of reduced colors	
	Prop.	Prev.[18]	Prop.	Prev.[18]	Prop.	Prev.[18]	Prop.	Prev.[18]
Airplane	40.57	50.43	0.9952	0.9999	0.64	0.63	**0**	18
Lena	**41.27**	35.80	**0.9961**	0.9872	0.25	0.25	**0**	56
Peppers	**35.58**	34.62	**0.9873**	0.9872	0.27	0.28	**0**	56
Mandirll	34.07	34.08	0.9901	0.9918	0.13	0.14	**0**	29
Earth	**44.88**	42.60	**0.9976**	0.9966	0.24	0.24	**0**	43
Girl	**39.34**	30.62	**0.9923**	0.9876	0.37	0.36	**0**	63
Splash	38.78	38.79	0.9910	0.9944	0.39	0.39	**0**	49
Aerial	**37.21**	34.05	**0.9935**	0.9916	0.21	0.22	**0**	41
House	35.69	36.69	0.9907	0.9934	0.35	0.34	**0**	30
Lake	35.69	37.74	0.9913	0.9950	0.29	0.29	**0**	25
Average	**38.30**	37.54	0.9925	0.9925	0.31	0.31	**0**	39.9

Table 3. Hiding capacity and number of reduced colors

	Hiding capacity [bpp]		Number of reduced colors	
	Prop.	Prev.[18]	Prop.	Prev.[18]
Airplane	0.64	1.37	**0**	124
Lena	**0.27**	0.25	**0**	56
Peppers	0.27	0.31	**0**	62
Mandirll	0.13	0.14	**0**	34
Earth	0.24	0.24	**0**	43
Girl	0.37	0.39	**0**	71
Splash	0.39	0.45	**0**	75
Aerial	0.21	0.22	**0**	41
House	0.35	0.48	**0**	92
Lake	0.29	0.38	**0**	71
Average	0.32	0.42	**0**	66.9

previous method. This is attributable to the amount of additional data. To vacate an index, we need to integrate two colors in the color palette. In the proposed method, we explore the closest index color to each target color and calculate the total frequencies of the color pairs. When the total frequency is large, a large amount of additional data should be stored. Thus, we choose the color pairs to be integrated in order of increasing total frequency. Consequently, we can reduce the amount of additional data. In the case of Lena, this algorithm is significantly effective for reducing additional data. Therefore, the pure payload of the proposed method was larger than that of the previous method.

With respect to the number of color reductions, the proposed method constantly preserved the number of index colors, while the previous method detracted 66.9 colors on average. As described in 3.4, this leads to security enhancement for the embedded payload in the proposed method. Consequently, we proved that the proposed method never sacrifices the number of colors without a severe decrease in hiding capacity.

5. Conclusion

In this paper, we proposed an extended RDH method for palette-based images. The proposed method not only improves the security of payload bits but also preserves the high quality of marked images. The previous method reduces the number of index colors so that the marked images have high quality. This leads to sacrificing payload security. In contrast, our method maintains the number of index colors and makes it difficult to maliciously extract the payload. By using PSNR and SSIM, we confirmed that the proposed is comparable in marked-image quality to the previous method. On the other hand, the hiding capacity of the proposed method was inferior to that of the previous method for eight out of ten test images. In the future, we will expand the hiding capacity to 1 bpp without deteriorating the marked-image quality.

References

1) W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Systems Journal, **35**, 3.4(1996).
2) Yu-Chee Tseng, Yu-Yuan Chen, Hsiang-Kuang Pan, A secure data hiding scheme for binary images, IEEE Trans. Commun., **50**, 8(2002).
3) J. Fridrich, A new steganographic method for palette-based images, Proc. PICS, 285-289(1999).
4) C. -H. Tzeng, Z. -F. Yang, W. -H. Tsai, Adaptive data hiding in palette images by color ordering and mapping with security protection, IEEE Trans. Commun., **52**, 5(2004).
5) S. Imaizumi, K. Ozawa, Palette-based image steganography for high capacity embedding, Bull. Soc. Photogr. Imag. Japan, **25**, 1(2015).
6) Y. Ueda, T. Koga, N. Suetake, E. Uchino, Selectively embeddable multibit steganographic method for palette-based images, IEEJ Trans. Electron. Inf. Syst., **138**, 12(2018).
7) J. Lin, Z. Qian, Z. Wang, X. Zhang, G. Feng, A new steganography method for dynamic GIF images based on palette sort, Wirel. Commun. Mob. Comput., **2020** (2020).
8) H. -Y. Wu, L. -H. Chen, Y. -T. Ching, Block-based steganography method using optimal selection to reach high efficiency and capacity for palette images, Appl. Sci., **10**, 21(2020).
9) Z. Ni, Y. -Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol., **16**, 3(2006).
10) X. Zhang, Reversible data hiding in encrypted image, IEEE Signal Process. Lett., **18**, 4(2011).
11) X. Li, B. Li, B. Yang, T. Zeng, General Framework to Histogram-Shifting Based Reversible Data Hiding, IEEE Trans. Image Process., **22**,

6(2013).
12) W. Tai, C. Yeh, C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circuits Syst. Video Technol., **19**, 6(2009).
13) X. Zhang, Separable reversible data hiding in encrypted Image, IEEE Trans. Inf. Forensics Security, **7**, 2(2012).
14) B. Ou, X. Li, Y. Zhao, R. Ni, Y. -Q. Shi, Pairwise prediction error expansion for efficient reversible data hiding, IEEE Trans. Image Process., **22**, 12(2013).
15) F. Huang, X. Qu, H. J. Kim and J. Huang, Reversible data hiding in JPEG images, IEEE Trans. Circuits Syst. Video Technol., **26**, 9(2016).
16) N. A. Saleh, H. N. Boghdady, S. I. Shaheen, A. M. Darwish, An efficient lossless data hiding technique for palette-based images with capacity optimization, Proc. 2007 IWSSIP and EC-SIPMCS, 241-244(2007).
17) N. A. Saleh, H. N. Boghdady, S. I. Shaheen, A. M. Darwish, High capacity lossless data embedding technique for palette images based on histogram analysis, Digit. Signal Process., **20**, 6(2010).
18) M. -Y. Wu, J. -H. Lee, A reversible data-hiding method for palette-based images with capacity optimization, J. Chin. Inst. Eng., **35**, 8(2012).
19) H. Wu, Y. Shi, H. Wang and L. Zhou, Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification, IEEE Trans. Circuits Syst. Video Technol., **27**, 8(2017).
20) M. Sneha Wakke and V. Navale, Reversible data hiding using color partitioning technique for encrypted palette images, Int. Res. J. Mod. Eng. Technol. Sci., **2**, 5 (2020).
21) M. Yu, H. yao, C. Qin, X. Zheng, Reversible data hiding in palette images, IEEE Trans. Circuits Syst. Video Technol.(2022).