Data Hiding Methods Using Voting Strategy and Mapping Table

Hengxiao Chi[1], *Chin-Chen Chang*[1], *Chia-Chen Lin*[2*]

[1] *Department of Information Engineering and Computer Science, Feng Chia University, Taiwan*
[2] *Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan*
hx9704@gmail.com, alan3c@gmail.com, ally.cclin@ncut.edu.tw

Abstract

With advancements in technology, the study of data hiding (DH) in images has become more and more important. In this paper, we introduce a novel data hiding scheme that employs a voting strategy to predict pixels based on their neighbors, then embeds data into the predicted pixels according to a designed mapping table. To extract the information, it is only necessary to use the voting strategy to predict the pixels again, then to compare the predicted and hidden pixels to extract the secret data with the assistance of the mapping table. Our experimental results demonstrate that the proposed hiding scheme has a high embedding capacity, and preserves a satisfactory visual quality. Additionally, it attains a high peak signal-to-noise ratio (PSNR) even at high embedding capacity.

Keywords: Data hiding, Pixel value prediction, Voting strategy

1 Introduction

As communication technology and the social economy continue to advance, the demand for an information society grows, resulting in the increased popularity of computers and mobile communication devices. Today, modern communication technology is entering a new stage with the advent of the fifth generation (5G) mobile communication technology, which will provide faster speeds, lower latency, and greater connectivity. The widespread adoption and improvement of these technologies have enabled people to communicate at any time and from any place, improving the convenience and efficiency of communication. With the ability to upload and download information from the cloud at any time through these devices, the privacy and security of information transmission have become increasingly crucial. In ancient times, individuals utilized various methods for secretly transmitting information during seemingly normal communication processes, such as the use of acrostics. In the era of digital communication, the carrier for people to transmit information becomes various digital carriers. The challenge of concealing confidential or protected information within seemingly ordinary digital carriers remains a significant area of research within the field of information security. Within the digital image carriers domain, Data Hiding (DH) technology is a widely adopted solution for ensuring the protection of information privacy while facilitating its transmission. Data hiding (DH) technology plays an important role in modern society with advanced communication technology. For example, data hiding technology can be used to hide sensitive information so that it is not easily detected by third parties during transmission. This helps to protect personal privacy and business secrets and prevent information from being accessed illegally. For another example, data hiding technology can embed digital watermarks to hide copyright information or ownership marks in digital media. This helps protect intellectual property and provides attribution capabilities to identify unauthorized use and infringement. Typically, data hiding methods can be classified into two broad categories: reversible data hiding (RDH) [1-2] and irreversible data hiding (IRDH) [3-5]. This classification is based on the receiver's ability to restore the original cover image after reading the embedded secret message. In the case of RDH, the original cover image can be fully recovered after the secret message is extracted. Conversely, IRDH cannot restore the cover digital image without damage after the secret message is extracted. IRDH methods are usually used in cases where a high-precision cover digital image is not required.

To date, researchers have devised numerous efficient and user-friendly techniques for image data hiding, such as difference expansion (DE) [6], histogram shifting (HS) [7-8], prediction-based data hiding methods [9-28], pixel value ordering (PVO) [29-36], and more. DE methods incorporate secret data by increasing the difference between adjacent pixels. The HS technique first creates an image intensity histogram and then altered for data hiding purposes. The prediction-based data hiding approach comprises three primary steps: content prediction, data embedding, and data extraction. This process involves using various predictors to estimate pixel values and modifying the original or predicted pixel values based on specific data embedding rules. The recipient then extracts the data based on the predicted values computed from the received images. The PVO-based DH method requires pixels to be arranged in ascending order within an image block, then modifies the two extreme values (minimum and maximum pixel values) therein to embed the information.

Due to the characteristics of prediction-based data hiding methods, the embedding performance and degree of image distortion are usually dependent on the pixel predictor and

*Corresponding Author: Chia-Chen Lin; E-mail: ally.cclin@ncut.edu.tw
DOI: 10.53106/160792642024052503003

embedding rules used. For content prediction, we need to design a predictor that can accurately estimate the pixels to be embedded, such as median edge detector (MED), rhombus predictor [12-13], and interpolation prediction [14-28], etc... In order to improve the data embedding process, we should further optimize the data embedding method so that the distortion is low for a given payload.

Interpolation prediction is a relatively special predictor. In DH based on interpolation prediction, the data hider transmits an image with the same dimensions as the original image, then reduces that original to a quarter of its size. It keeps a quarter of the original pixels constant as seed pixels, and uses these seed pixels to interpolate to produce a cover image that is the same size as the original image. The confidential message is then only hidden in the interpolated pixels. Thus, we are unable to recover the original-sized image after the secret message is extracted. In the scenario where the data hider desires to transmit an image of a larger size than the original, they maintain the pixels of the original image as seed pixels and employ these seed pixels for interpolation, resulting in a cover image that is four times larger than the original. The confidential message is then concealed exclusively within the interpolated pixels. To restore the original image without any loss, it is necessary to simply remove the interpolated pixels after extracting the embedded data.

The focus of interpolation-based DH is on interpolation and data embedding methods. Previous works on interpolation techniques include neighborhood mean interpolation (NMI) [14-16], the interpolation by neighboring pixel (INP) [17], enhanced neighborhood mean interpolation (ENMI) [18], etc. In the year 2009, Jung and Yoo [15] first presented their innovative DH approach, which was based on the concept of neighborhood mean interpolation (NMI), to the information security field. The capacity for embedding data in an interpolated pixel is primarily determined by the discrepancy between the interpolated pixel and the seed pixel. Lee and Huang [17] later improved upon this scheme, utilizing a more effective neighborhood pixel interpolation (INP) to interpolate the image. Subsequently, Chang et al. [18] later used an improved ENMI technique, which takes into account a two-level embedding using a greater number of neighborhood pixels. A subsequent technique, proposed by [19], uses parabolic interpolation (PI) to generate interpolated images and embeds secret bits through the relationship between the interpolated pixels and the mean of a group of original pixels. Malik [20-21] further improved the modified neighbor mean interpolation (MNMI) technique, which effectively preserves image quality. In [22-23], the secret information is transformed using a Lagrangian interpolation polynomial, which was then divided and embedded into the alternating pixels of each interpolated sub-sampled image.

In this paper, we propose an effective DH scheme that utilizes a voting strategy for prediction. Our approach is similar to interpolation-based data hiding methods in that it maintains a constant set of seed pixels for prediction purposes. Unlike other prediction error-based data hiding techniques, this scheme performs the embedding operation directly on the predicted values. The predicted pixels are modified by adding or subtracting values according to the secret data. As the amount of secret data increases, the range of modification to the predicted pixels also increases. Consequently, the quality of the stego image is primarily dependent on the accuracy of the prediction, with higher accuracy leading to improved image quality. In our scheme, each predicted pixel can contain up to k bits of confidential information (where k can range from 1 to 7). Our proposed technique demonstrates a substantial increase in data hiding capacity and results in a stego image with higher visual quality compared to existing techniques, particularly at high hiding capacities. The improved data hiding capability of the proposed scheme enables the concealment of a greater amount of information without causing detectable alterations. The good visual quality of the stego image means that the stego image maintains the original characteristics of the carrier and the proposed scheme makes it more difficult to detect changes compared to other existing techniques.

The rest of this paper is organized as follows. Section 2 presents a review of related works. Section 3 provides a detailed description of the proposed data hiding scheme. The experimental results, comparisons, and discussions are given in Section 4. Section 5 presents the conclusions of this paper.

2 Related Works

2.1 Neighborhood Mean Interpolation (NMI) [15]

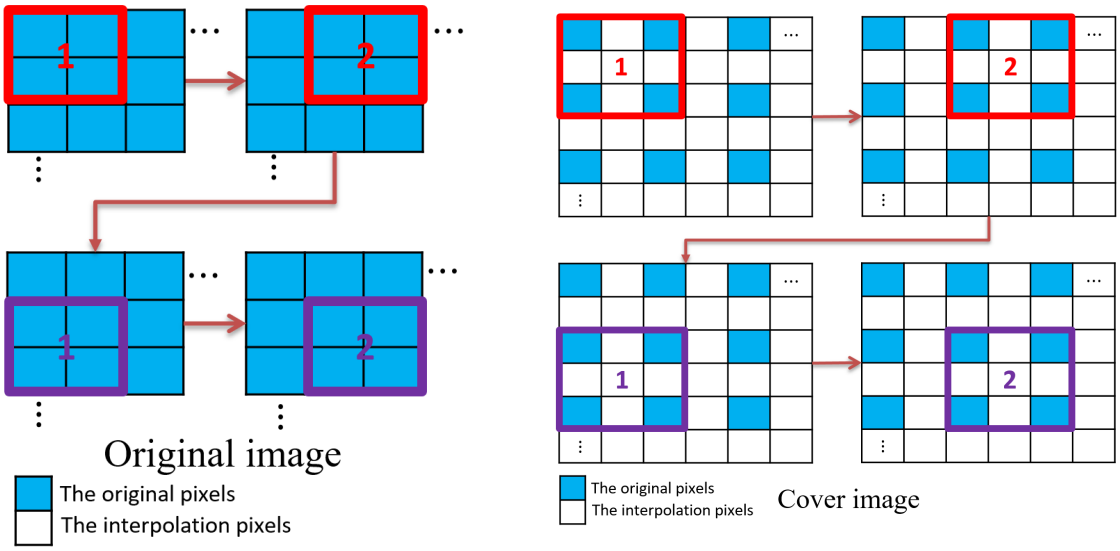
The standard process in conventional interpolation involves using the four neighboring seed pixels to calculate the interpolated value for a given pixel. The steps involved in using interpolation for data hiding are as follows: first, the original image O of a given size $M \times N$ is scaled down to a reduced image of size $\frac{M}{2} \times \frac{N}{2}$. Then, the reduced original image is divided into 2 × 2 sized blocks with overlapping rows and columns, and these blocks are processed in a raster scan order as shown in Figure 1(a). Next, each 2 × 2 block is expanded into a 3 × 3 overlay image block, and the empty pixels are filled using interpolation techniques. As shown in Figure 1(b), where the blue pixels are the seed pixels from the original image, and the white areas need to be computed as interpolated pixels.

Briefly, the NMI technique introduced by Jung and Yoo [15] predicts the interpolation using the average of the pixels adjacent to each interpolation position as shown in Equation (1).

$$\begin{aligned} C(1,0) &= \frac{C(0,0)+C(2,0)}{2} \\ C(0,1) &= \frac{C(0,0)+C(0,2)}{2}. \\ C(1,1) &= \frac{C(0,0)+C(0,2)+C(2,0)}{3} \end{aligned} \tag{1}$$

C(1,0) is predicted from the average of the horizontally adjacent pixels in the original block, while C(0,1) is

estimated from the average of the vertically adjacent pixels in the original block. After that, the value of C(1,1) is calculated by taking the average of the three seed pixels. Figure 2 shows a specific example of image interpolation implemented by NMI.



(a) Interpolation-based image division and processing sequence

(b) Block processing sequence for up-sampled images

Figure 1. Image division, up-sampling and block processing

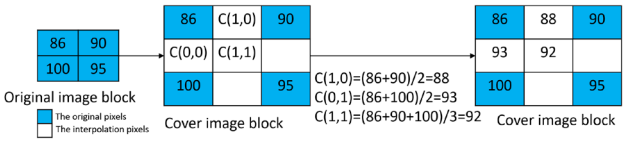


Figure 2. An example of NMI

2.2 The Interpolation by Neighboring Pixel (INP) [17]

Lee and Huang [17] proposed the Interpolated by Neighborhood Pixel (INP) method in 2012. Compared with NMI, INP increases the weight of the upper left pixel C(0,0). Then, it uses the average of the two interpolated values it has computed as the interpolated value for the third position C(1,1), as shown in Equation (2). Figure 3 shows a specific example of image interpolation implemented by INP.

$$C(1,0)=\frac{C(0,0)+(C(0,0)+C(2,0))/2}{2}$$
$$C(0,1)=\frac{C(0,0)+(C(0,0)+C(0,2))/2}{2}. \quad (2)$$
$$C(1,1)=\frac{C(1,0)+C(0,1)}{2}$$

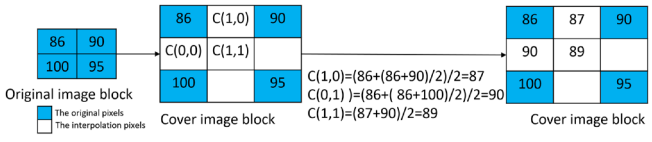


Figure 3. An example of INP

3 Proposed Scheme

This section presents a novel data hiding method that is based on the voting strategy prediction. The scheme utilizes a voting strategy for predicting the pixel values, and then the sensitive information is hidden according to the mapping table. The specifics of the pixel value prediction method are outlined in Section 3.1, while the data embedding process is described in Section 3.2, and the data extraction process is outlined in Section 3.3. Finally, in the end of Section 3, measures to prevent overflow/underflow are also provided.

3.1 The Pixel Value Prediction

First, we divide the pixels of the original image O of size $M \times N$ into two parts in the manner of a chess board as shown in Figure 4. Here, we define the white part as the seed pixels, which remain unchanged, and the gray part as the location that will conceal the secret data.

As in Figure 4, the current pixel is denoted as X and its four surrounding seed pixels are represented as ai (i = 1, 2, 3, 4), respectively. Then, based on the values of the four seed pixels, we can divide the prediction method for pixel X into five types. The formula is as follows:

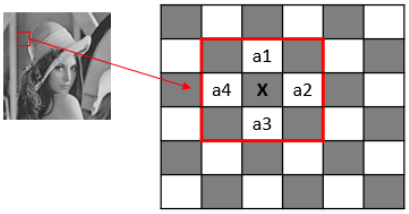


Figure 4. Example of pixel division

$$X=\begin{cases} a1, & if\ a1=a2=a3=a4 \\ (ai_1+ai_3)/2, & if\ ai_1=ai_2\neq ai_3=ai_4 \\ ai_1, & if\ \exists!\ ai_1=ai_2, \\ ai_1, & if\ ai_1=ai_2=ai_3\neq ai_4 \\ (a1+a2+a3+a4)/4, & if\ a1\neq a2\neq a3\neq a4 \end{cases} \quad (3)$$

where ai_1 is any pixel in ai, and ai_2, ai_3, and ai_4 are the three other pixels in ai. Of the five types of prediction, Type 1 is that all four seed pixels have equal values; then, the predicted value of X is the value of the seed pixel. For Type 2, the four seed pixels have two sets of identical pixels; then, the predicted value of X is the average of these two sets. Type 3 is that only two of the four seed pixels are equal, and the remaining pixels are not equal to each other; thus, based on the voting strategy, the predicted value of X is the one with the higher number. Type 4 is that only one of the four seed pixels is different from the others; thus, based on the voting strategy, the predicted value of X is the one with the higher number. Finally, Type 5 is that the four seed pixels are not equal to each other; then, the predicted value of X is the average of these four. To describe our prediction method more intuitively, Figure 5 gives specific examples of these five types of prediction.

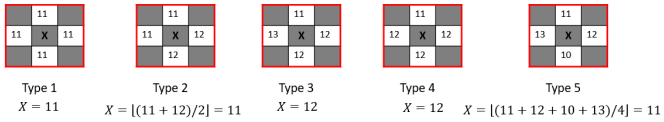


Figure 5. Examples of five types of prediction

Of course, this method differs depending on the location of the pixel in question. If the pixel to be predicted is at the edge of the image, with only three seed pixels around it, the prediction method for pixel X can be classified into 3 types. When the three seed pixels are equal, the method of predicting X is the same as Type 1; when only two of the three seed pixels are equal, the method of predicting X is the same as Type 4; and when the three seed pixels are not equal to each other, the method of predicting X is the same as Type 5. Meanwhile, if the pixel to be predicted is located at the vertex of the image and there are only two seed pixels around it, the prediction method for pixel X can be divided into 2 types. When the two seed pixels are equal, the method of predicting X is the same as Type 1; otherwise, when the two seed pixels are not equal, the method of predicting X is the same as Type 5.

3.2 The Data Embedding Phase

After deriving the prediction value for the given pixel X, we can embed our sensitive information in the pixel. First, we read the k-bit secret data $b_1\, b_2\, \ldots\, b_k\, (k = 1, 2, \ldots, 7)$ and convert it to a decimal number B. The value of k depends on the size of the secret data to be embedded, and the more secret data there is, the larger k should be so in order to embed all the secret data into the image. However, according to the proposed data embedding method, when k = 8, it is easy to have pixel value overflow, so we set the value of k to [1,7]. We then design a mapping table for embedding information; the predicted pixel X can be modified to hide the corresponding secret data according to the content of the mapping table, which is applicable to all the proposed prediction types. The design principle of this mapping table is that according to the value of the secret data to be embedded, different decimal values are assigned sequentially on a number axis, one left and one right, with 0 as the origin. This number axis is represented in the form of our mapping table, as shown in Table 1. With an embedding mapping table designed according to this principle, the modification to be made to the predicted pixel X will fall within a small range if the embedded secret data is relatively small. Because the decimal values are assigned symmetrically, one on the left and one on the right with 0 as the origin, so when additional secret data is embedded, the modifications needed for the predicted pixel X will be reduced by half.

Table 1. The mapping table for embedding data

Stego value X'	X-5	X-4	X-3	X-2	X-1	X	X+1	X+2	X+3	X+4
Decimal secret data B	9	7	5	3	1	0	2	4	6	8

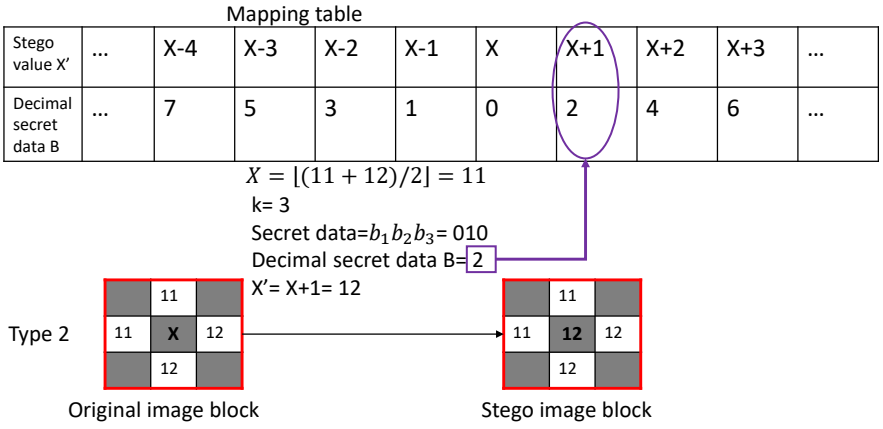


Figure 6. An example of the embedding process

Here, we illustrate our embedding with an example of prediction type 2 in Figure 6. As shown in the figure, when we want to embed a 3-bit secret data at the position of X, since the secret data is $(010)_2 = (2)_{10}$, we only need to go to the mapping table to check how pixel X is to be modified when the decimal of the secret data to be embedded is 2. The mapping table shows that to embed the decimal number 2, the operation +1 is needed for the predicted pixel X. Finally, we get the stego pixel X' = 12.

3.3 The Data Extraction and Image Recovery Phase

When we receive the stego image containing the sensitive information, we first divide the pixels of the stego image into two groups following the format of a chess board, where the gray pixels carry the sensitive information and the white pixels serve as the seed pixels. Then, we use the classification and prediction described in Section 3.1, we predict the value of the current pixel X' as X. Next, we query the mapping table to extract the corresponding decimal secret data B according to the gap between X' and X, and finally convert this decimal secret data B into a k-bit binary number $b_1 b_2 \ldots b_k (k = 1, 2, \ldots, 5)$.

As shown in Figure 7, we determine the best prediction type for the current pixel X' based on its four seed pixels and learn that we have to predict it using the Type 2 prediction method, producing the predicted value X = 11. Then, we find that the current pixel is the predicted value for the +1 operation. U sing the mapping table, we can see that the +1 operation corresponds to the embedded decimal number 2, and then convert it to k-bit binary to extract our secret data $(010)_2$.

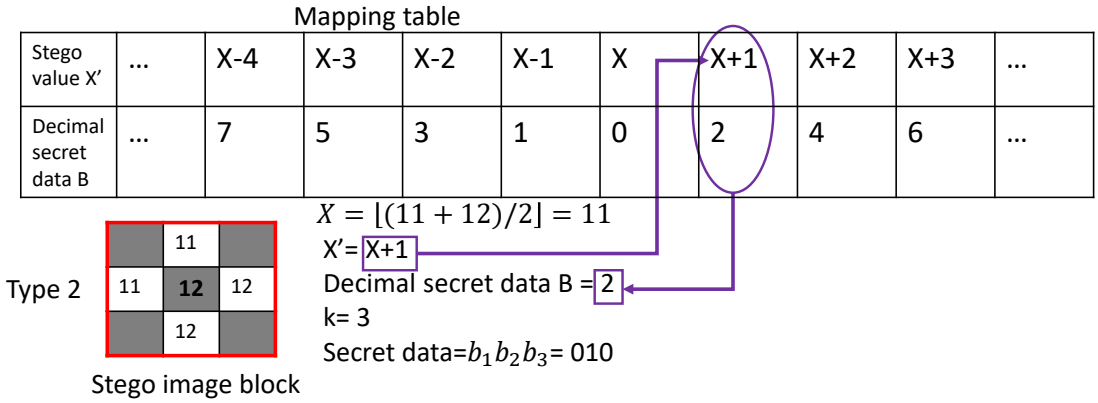


Figure 7. An example of the extraction process

3.4 The Overflow/Underflow

If the predicted value of our current pixel produces a situation greater than 255 or less than 0 after embedding multiple bits of secret data according to the mapping table, we call this situation overflow/underflow. In order to avoid the pixel value overflow/underflow, we have to perform an overflow/underflow test before embedding the secret data. We determine whether this predicted pixel will have an overflow/underflow situation when embedding data from the left and right extremes of the k-bit mapping table. If the predicted pixel is judged to have an overflow/underflow, we do not embed the secret data in the pixel and instead restore the original. If it is judged that the predicted pixel will not generate overflow/underflow, then we embed it. For example, if k = 3, then we have to determine if embedding $(111)_2 = (7)_{10}$ and $(111)_2 = (6)_{10}$, i.e. X-4 and X+3 in the mapping table will make X' greater than 255 or less than 0. If yes, then we do not embed data in X and the pixel at X is the original pixel; if not, then we embed the 3-bit secret data normally.

4 Experimental Results

In this section, we present the results of our experimental evaluation of the proposed data hiding scheme. The performance of our method is compared with other existing schemes. The experiments were performed using the 2017a version of MATLAB software on a Windows PC. The experiments were performed on seven commonly used grayscale images: "Airplane", "Baboon", "Boat", "Barbara", "Couple", "Lena", and "Peppers".

We conduct experiments to evaluate the performance of our proposed data hiding scheme based on voting strategy prediction. The binary data stream S, which is generated by a random number generator, serves as the secret information to be embedded in the cover image. The effectiveness of the proposed scheme is assessed based on the embedding capability (EC), peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and cosine similarity (CS). The PSNR, which is expressed in decibels (dB), and the SSIM are the two primary metrics used to evaluate the image distortion between the original image and the stego image. The PSNR and SSIM are calculated as follows:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (O_{i,j} - D_{i,j})^2, \tag{6}$$

$$\mathrm{PSNR} = 10 log_{10} \frac{(255)^2}{\mathrm{MSE}} (\mathrm{dB}), \tag{7}$$

$$\text{SSIM} = \frac{(2\mu_O\mu_D + c_1)(2\sigma_{OD} + c_2)}{\left[(\mu_O)^2 + (\mu_D)^2 + c_1\right]\left[(\sigma_O)^2 + (\sigma_D)^2 + c_2\right]}, \quad \textbf{(8)}$$

$$\text{CS} = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|}. \quad \textbf{(9)}$$

In Eq. (6), $M \times N$ represents the size of the cover image, $O_{i,j}$ and $D_{i,j}$ denote the pixel values of the cover image and the stego-image, respectively. A PSNR value greater than 30 dB indicates that the information embedding causes imperceptible distortion to the human eye. The higher the PSNR value, the lower the distortion caused by the hidden data. SSIM combines three different factors: luminance, contrast, and structure, as a way to assess the similarity between images. In Eq. (8), μ is the mean value, which is used as an estimate of luminance; and σ is the standard deviation, which is used as an estimate of contrast; σ_{OD} denotes the covariance between the original image O and the stego image D, which is used as a measure of structural similarity; and c_1, and c_2 are two constants close to zero. SSIM ranges from -1 to 1. A SSIM value close to 1 indicates that the stego image is similar to the original image. When the value of SSIM is equal to 1, it means that the two images used for comparison are the same. In Eq. (9), we express the original image as a vector $\vec{x}$ and the stego image as a vector $\vec{y}$ as well, and then we can calculate the cosine similarity (CS) between the two images. The closer the value of CS is to 1, the more similar the stego image is to the original image. EC is defined as the total payload in bits. The larger the EC, the more secret information can be embedded in the image.

We evaluate the performance of our proposed scheme for different values of k in Section 4.1. A comparison with other schemes is given in Section 4.2.

4.1 Performances of Our Proposed Scheme

In the proposed scheme, k denotes the number of bits of secret data that are embedded at the current predicted position each time. Table 2 presents the results of the proposed scheme for various values of k when applied to the test images.

Our experimental results reveal two distinct characteristics of the embedding capacity, as shown in Table 2. The maximum capacity can currently reach 655,361 bits when k = 5, which is already higher than the maximum capacity of most related work; thus, we present here the experimental results only up to k = 5. Additionally, the capacity exhibits a consistent growth as k increases, indicating that there is potential for further enhancement without the need for additional considerations. The second feature is the highly stable capacity for different images: embedding capacity is similar for all images with the same k value. That is, the proposed scheme is applicable to both complex and smooth images, and there is no situation where only a small amount of information can be embedded in a certain image.

Table 2 also shows the PSNR, SSIM, and CS for different k values. The PSNR values are as high as 36.0874 dB when a single bit of sensitive data is embedded in each interpolated element space. The highest value of the SSIM is 0.9609, with low image distortion. The highest value of CS is 0.9997, while its lowest value is 0.9937; both of these are very close to 1, which shows that the stego image produced by the proposed scheme is very similar to the original image.

Table 2. Performance of the proposed scheme for different k values

Image	k-bit secret data	EC	PSNR	SSIM	CS
Airplane	k=1	131,073	33.4449	0.9609	0.9993
	k=2	262,145	33.3755	0.9567	0.9993
	k=3	393,217	33.1031	0.9402	0.9986
	k=4	524,289	32.1244	0.8816	0.9976
	k=5	655,346	29.5765	0.7249	0.9937
Baboon	k=1	131,048	26.4306	0.8661	0.9968
	k=2	262,083	26.4114	0.8648	0.9968
	k=3	393,052	26.3686	0.8600	0.9969
	k=4	524,033	26.1509	0.8426	0.9968
	k=5	654,811	25.3639	0.7831	0.9964
Barbara	k=1	131,073	28.2384	0.9060	0.9989
	k=2	262,145	28.2161	0.9033	0.9989
	k=3	393,217	28.1245	0.8928	0.9989
	k=4	524,289	27.7923	0.8553	0.9985
	k=5	655,331	26.6655	0.7470	0.9970
Boat	k=1	131,073	32.4595	0.9050	0.9981
	k=2	262,145	32.4019	0.9025	0.9983
	k=3	393,217	32.1755	0.8924	0.9983
	k=4	524,177	31.3556	0.8540	0.9984
	k=5	647,081	29.1424	0.7467	0.9975
Couple	k=1	131,025	32.5836	0.9260	0.9987
	k=2	262,031	32.5269	0.9237	0.9986
	k=3	392,839	32.2945	0.9140	0.9986
	k=4	522,269	31.4717	0.8790	0.9984
	k=5	646,126	29.2476	0.7747	0.9980
Lena	k=1	131,073	36.0874	0.9440	0.9997
	k=2	262,145	35.9633	0.9403	0.9997
	k=3	393,217	35.4546	0.9252	0.9997
	k=4	524,289	33.9127	0.8720	0.9994
	k=5	655,361	30.4636	0.7223	0.9968
Peppers	k=1	131,069	34.0482	0.9075	0.9997
	k=2	262,139	33.9599	0.9038	0.9997
	k=3	393,118	33.6425	0.8903	0.9997
	k=4	521,709	32.5597	0.8423	0.9993
	k=5	634,751	29.8975	0.7098	0.9973

In order to further illustrate the capabilities of our proposed scheme, Table 3 also displays the results of several performance metrics, including information entropy, number of pixel changing rate (NPCR), unified average changed intensity (UACI), and mean absolute error (MAE), for different values of k. The calculation of information entropy is depicted in Equation (10). The entropy of an image is a statistical form of a feature that represents the characteristics of the image's grayscale distribution. If the image pixel distribution is highly random, the information entropy value is closer to 8. In terms of entropy performance, the entropy of the proposed scheme at different k values is similar to the original entropy, indicating that the proposed scheme does not cause significant damage to the image pixel distribution in the process of information embedding. NPCR helps count the number of different pixels, while UACI serves to count the average change in pixel between two images. The calculation methods for NPCR and UACI are given in Eqs. (11) and (13), respectively. As shown in Table 3, the proposed scheme uses only gray pixels for embedding secret data, so the NPCR does not exceed 50%, but the UACI is very low for different test images at different k values, and the highest UACI is only 2.8957%, which indicates that the proposed scheme produces a small average pixel variation when embedding secret data. MAE is an indicator measuring image quality; its calculation method is shown in (14). The smaller the MAE value, the better the image quality. From Table 3, we can find that the MAE of the proposed scheme is small for different k values, with a maximum of 7.3841. This indicates that the stego image produced by the proposed scheme is very similar to the original image.

$$\text{entropy} = \sum_{i=0}^{255} P(i) \log \frac{1}{P(i)}, \tag{10}$$

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \tag{11}$$

$$D(i,j) = \begin{cases} 1, & O(i,j) \neq S(i,j) \\ 0, & otherwise \end{cases}, \tag{12}$$

$$\text{UACI} = \frac{\sum \left| O(i,j) - S(i,j) \right|}{M \times N \times 255} \times 100\%, \tag{13}$$

$$\text{MAE} = \frac{1}{M \times N} \sum_{i=1}^{M \times N} \left| O(i,j) - S(i,j) \right|. \tag{14}$$

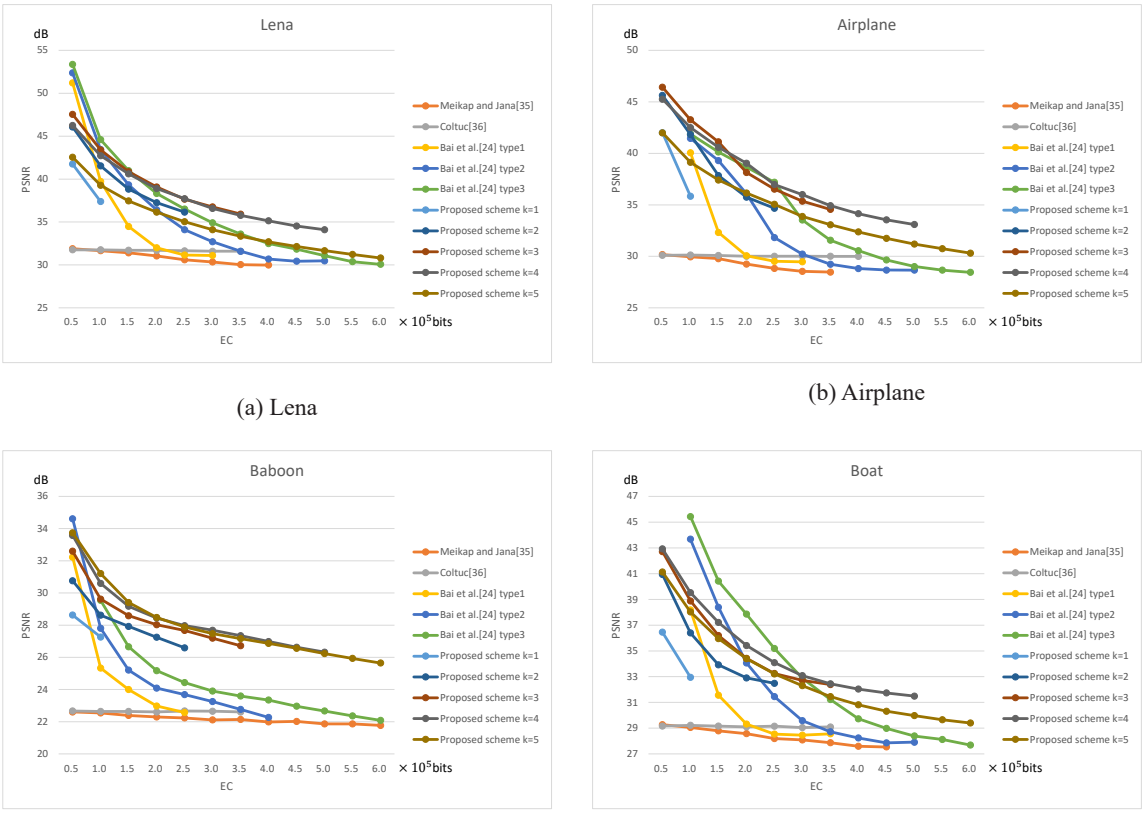
Here i represents the image pixel value ranging from 0 to 255, and $P(i)$ is the probability of image pixel value i. The width and height of the image are represented by M and N, respectively. $O(i, j)$ is the pixel in row i, column j of the original image. $S(i, j)$ is the pixel in row i, column j of the stego image.

Table 3. Additional performance of the proposed scheme at different k values

Image	Original entropy	k	Entropy	NPCR (%)	UACI (%)	MAE
Airplane	6.7059	k=1	6.7216	42.5232	0.7163	1.8267
		k=2	6.7268	43.3937	0.7451	1.9000
		k=3	6.7426	45.2965	0.8447	2.1539
		k=4	6.7869	47.1523	1.1227	2.8628
		k=5	6.9047	48.4787	1.7998	4.5895
Baboon	7.3579	k=1	7.2979	48.0118	2.3617	6.0222
		k=2	7.2984	48.0209	2.3689	6.0406
		k=3	7.3000	48.1346	2.4004	6.1210
		k=4	7.3067	48.3822	2.5082	6.3959
		k=5	7.3298	48.8056	2.8957	7.3841
Barbara	7.6321	k=1	7.6109	45.6791	1.5565	3.9692
		k=2	7.6124	45.9171	1.5738	4.0133
		k=3	7.6168	46.5527	1.6353	4.1700
		k=4	7.6307	47.7188	1.8391	4.6898
		k=5	7.6673	48.6755	2.3867	6.0860
Boat	7.1914	k=1	7.1816	46.3425	1.1132	2.8386
		k=2	7.1845	46.4890	1.1287	2.8783
		k=3	7.1899	46.8243	1.1812	3.0122
		k=4	7.2065	47.5079	1.3759	3.5086
		k=5	7.2471	47.8912	1.9250	4.9087
Couple	7.0581	k=1	7.3360	45.9244	1.0830	2.7616
		k=2	7.3514	46.0964	1.0977	2.7992
		k=3	7.3556	46.5797	1.1575	2.9517
		k=4	7.3609	47.2912	1.3525	3.4490
		k=5	7.3765	47.7936	1.9130	4.8782
Lena	7.4455	k=1	7.4368	44.3581	0.7204	1.8371
		k=2	7.4385	44.6827	0.7428	1.8942
		k=3	7.4433	45.6352	0.8257	2.1055
		k=4	7.4615	47.1966	1.0819	2.7590
		k=5	7.5068	48.4161	1.7436	4.4462
Peppers	7.5944	k=1	7.5894	46.0861	0.9413	2.4003
		k=2	7.5909	46.1746	0.9566	2.4393
		k=3	7.5957	46.4764	1.0153	2.5889
		k=4	7.6086	47.0299	1.2139	3.0954
		k=5	7.6333	46.9028	1.7812	4.5422

4.2 Execution Results and Analysis

Image quality and a scheme's hiding ability always have an inverse relationship, and image quality is necessarily lost when hiding ability increases. We can thus try to reduce the damage to the image with the same embedding capability. We compare the PSNR of the proposed scheme with that of the methods presented in [24, 35-36]. Figure 8 illustrates the PSNR values for four test images, clearly demonstrating the impact of varying embedding capacities. From the representation shown in Figure 8, it becomes evident that, when the embedding capacity (EC) is low, the PSNR of the proposed scheme falls behind that of Bai et al. [24]. But after the EC reaches 3×10^5 bits, it can be observed that the PSNR of the proposed scheme surpasses those of all other methods as the embedding capacity increases. As the embedding capacity increases, the superiority of the mapping table used in the embedding process of the proposed scheme begins to emerge. As the embedding capacity increases, that is, our k increases, a pixel changes at most one-half of the embedded data because we use a mapping table for embedding. Moreover, the PSNR of the proposed scheme decreases more slowly as the EC increases. Therefore, the proposed scheme proves to be more efficient in terms of cost when it comes to embedding a large quantity of confidential information.



(a) Lena

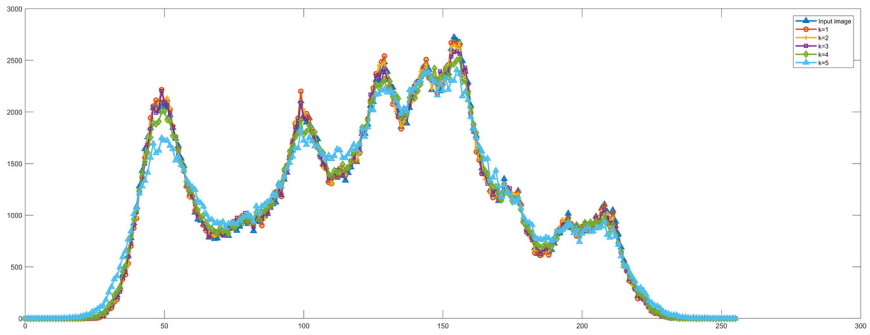
(b) Airplane

(c) Baboon

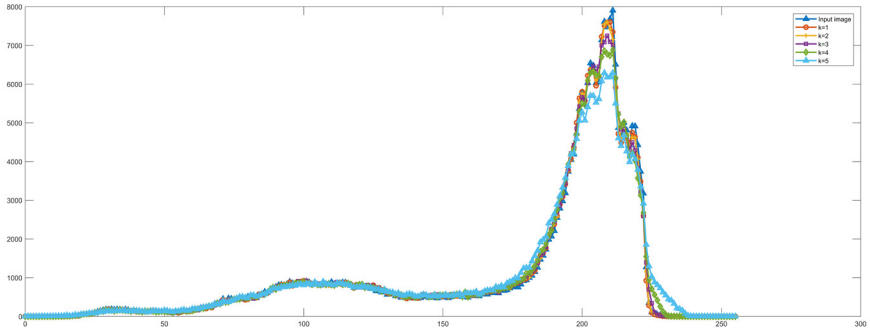
(d) Boat

Figure 8. Comparison of PSNR with different embedding capabilities

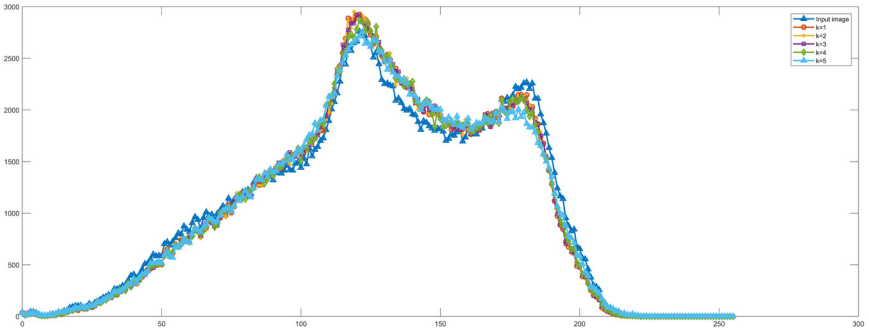
We also use histogram steganalysis to assess the security of the proposed algorithm. Histogram steganalysis is an analysis method used to detect the presence of steganographic information in an image. In histogram steganalysis, the main focus is on the pixel value distribution and statistical features of the image. To provide clear illustration, we conducted the experiments using four standard images: "Lena", "Airplane", "Baboon", and "Boat". Figure 9 shows the histograms of the input images and the stego images generated by the proposed scheme for different k values. The histogram of the stego image is observed to be nearly identical to that of the original image, indicating the successful implementation of the steganographic technique and the preservation of the image quality. Although the histogram of the stego image tends to become smooth when the value of k increases, the location and trend of the peaks do not change much. When the value of k is smaller, the histogram of the stego image is observed to be nearly identical to that of the original image, which implies that the proposed scheme is able to resist the histogram steganalysis method when the embedded secret data is small, and the embedded secret data has better steganography.



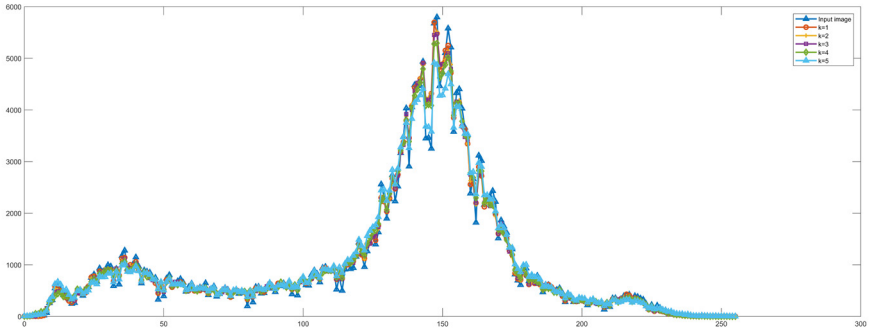
(a) Lena



(b) Airplane



(c) Baboon



(d) Boat

Figure 9. Comparison of histogram with different k values

In addition, we also compared our scheme with other methods [15, 17, 20, 24] in terms of EC, PSNR and SSIM. From the performance measures in Figure 8, we find that type 3 of [24] performs better than type 1 and type 2 in both embedding capability and PSNR direction; in Table 4, we therefore directly compare with against type 3. From Table 4, we can see that the PSNR and SSIM of the proposed scheme are higher than those of the other methods given a similar EC. Moreover, when k = 5, the proposed method achieves a higher EC than most of the other schemes, while when k = 5, the proposed scheme obtains a higher PSNR than even the low-hiding [15, 17]. These data fully illustrate the superiority of the proposed scheme in terms of high hiding capacity and PSNR.

Table 4. Comparison with different schemes on EC, PSNR and SSIM

Image		[15]	[17]	[20]	[24] type3	k=2	k=3	k=4	k=5
Lena	EC	325,167	356,174	248,982	585,225	262,145	393,217	524,289	655,361
	PSNR	26.82	30.15	34.86	29.92	35.96	35.45	33.91	30.46
	SSIM	0.9287	0.9451	0.9244	0.8383	0.9403	0.9252	0.8720	0.7223
Peppers	EC	317,769	358,143	265,249	585,225	262,139	393,118	521,709	634,751
	PSNR	25.99	28.13	33.33	28.49	33.96	33.64	32.56	29.90
	SSIM	0.9319	0.9468	0.9009	0.8079	0.9038	0.8903	0.8423	0.7098
Baboon	EC	309,319	635,591	447,173	585,225	262,083	393,052	524,033	654,811
	PSNR	20.09	21.29	24.63	22.01	26.41	26.37	26.15	25.36
	SSIM	0.7726	0.7983	0.7904	0.6419	0.8648	0.8600	0.8426	0.7831
Boat	EC	355,002	428,547	303,200	585,225	262,145	393,217	524,177	647,081
	PSNR	24.91	26.15	30.88	27.57	32.40	32.18	31.36	29.14
	SSIM	0.8334	0.8714	0.8579	0.8078	0.9025	0.8924	0.8540	0.7467
Couple	EC	314,805	441,800	284,262	585,225	262,031	392,839	522,269	646,126
	PSNR	24.20	24.86	28.50	26.60	32.53	32.29	31.47	29.25
	SSIM	0.8484	0.8806	0.8987	0.7632	0.9237	0.9140	0.8790	0.7747
Barbara	EC	494,284	473,225	333,301	585,225	262,145	393,217	524,289	655,331
	PSNR	23.41	22.92	26.79	23.99	28.22	28.12	27.79	26.67
	SSIM	0.8084	0.8244	0.8435	0.744	0.9033	0.8928	0.8553	0.7470
Airplane	EC	358,689	314,330	184,280	585,225	262,145	393,217	524,289	655,346
	PSNR	24.66	27.43	36.19	28.39	33.38	33.10	32.12	29.58
	SSIM	0.9226	0.9413	0.9521	0.8573	0.9567	0.9402	0.8816	0.7249

To further evaluate the performance of the proposed algorithm, we used two standard image databases, the Uncompressed Color Image Database (UCID) and the USC-SIPI Image Database (SIPI). In the experiments, note that all images were first converted to grayscale before testing the performance of the different methods. We compare our scheme with the methods proposed in [13, 15, 25-28]. It can be seen from Figure 10 and Figure 11 that the proposed scheme not only has a good average PSNR at low bpp but also has a higher average PSNR than other schemes at high bpp, both in the UCID dataset and in the SIPI dataset. This indicates that the proposed scheme is not cost-effective only for individual images, but also has good visual quality and embedding capacity for most images. In other words, the proposed scheme can help embed secret data in most images without serious visual corruption.

5 Conclusion

In this paper, we propose a novel data hiding scheme using a voting strategy for prediction; we first use a voting strategy to predict pixel values, then directly embed the data into the predicted pixels according to the designed mapping table. In the process of information extraction, this scheme also uses the voting strategy to predict the pixel value. Then, according to the difference between the predicted value and the hidden pixel, it finds the secret data represented by the operation in the mapping table and converts it into a k-bit binary number. After this, our secret data can be extracted. Experimental results show that the proposed scheme has a high payload capacity of up to 655,361 bits, which is much better than many existing schemes. Moreover, the proposed scheme has a significant PSNR advantage when embedding

large amounts of information. The method of using mapping tables for data embedding will be less destructive to pixels than other existing schemes when a large amount of information is embedded. Moreover, the histogram of the stego image produced by the proposed scheme is similar to the original image in terms of peaks and histogram trends, which means that the proposed scheme makes it difficult for secrert information to be detected by the histogram steganalysis method.

In the future, we could improve the prediction method and embedding rules based on the amount of data to be embedded, so that the scheme can have a relatively high PSNR even at low EC.

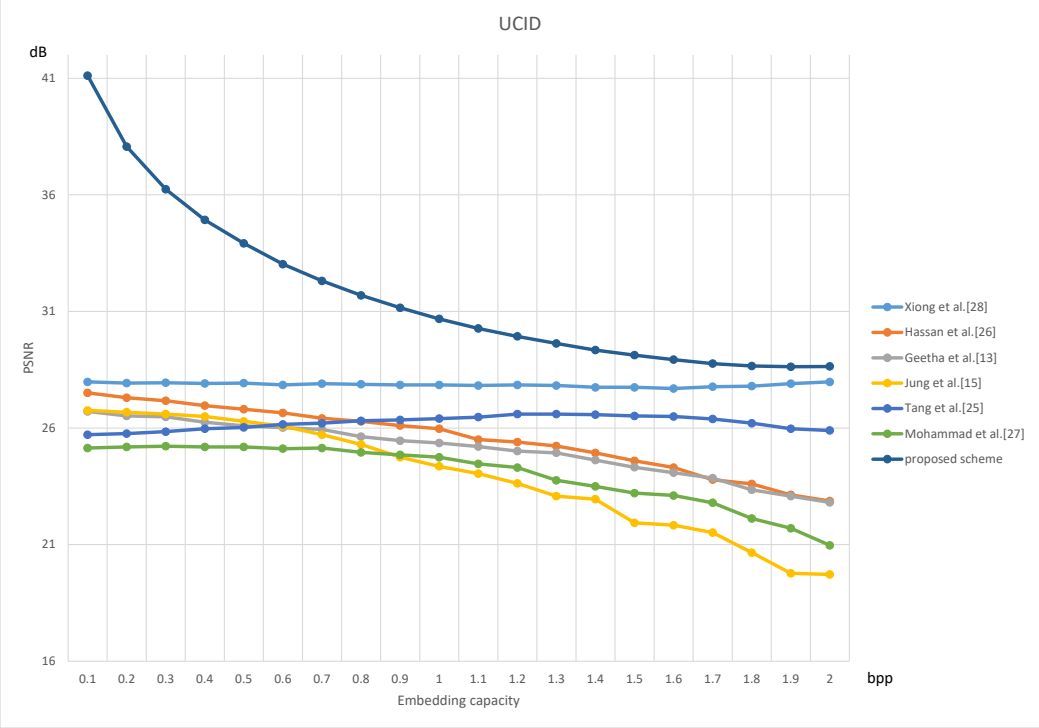


Figure 10. Comparison of PSNR under different embedding capabilities in the UCID dataset

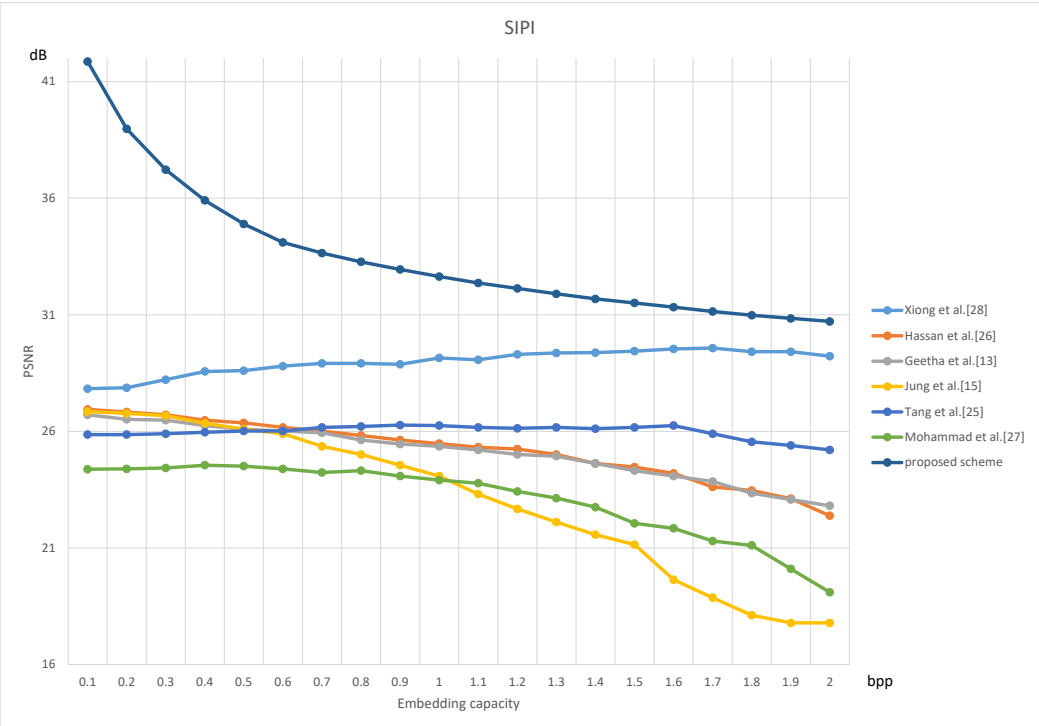


Figure 11. Comparison of PSNR under different embedding capabilities in the SIPI dataset

References

[1] G. Coatrieux, C. Le Guillou, J.-M. Cauvin, C. Roux, Reversible Watermarking for Knowledge Digest Embedding and Reliability Control in Medical Images, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 13, No. 2, pp. 158–165, March, 2009, doi: 10.1109/TITB.2008.2007199.

[2] Y.-C. Chen, C.-W. Shiu, Distributed Encrypted Image-Based Reversible Data Hiding, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 101-107, January, 2021.

[3] L. Almazaydeh, Secure RGB image steganography based on modified LSB substitution, *International Journal of Embedded Systems*, Vol. 12, No. 4, pp. 453–457, May, 2020, doi: 10.1504/IJES.2020.107644.

[4] S. Das, K. Muhammad, S. Bakshi, I. Mukherjee, P. K. Sa, A. K. Sangaiah, A. Bruno, Lip biometric template security framework using spatial steganography, *Pattern Recognition Letters*, Vol. 126, pp. 102–110, September, 2019, doi: 10.1016/j.patrec.2018.06.026.

[5] P. S. Huang, S.-K. Yang, A Novel Data Hiding Approach Using Addition and Multiplication on Groups of Three Pixels, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 521-531, May, 2021.

[6] Y. Hu, H.-K. Lee, J. Li, DE-Based Reversible Data Hiding With Improved Overflow Location Map, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 2, pp. 250–260, February, 2009, doi: 10.1109/TCSVT.2008.2009252.

[7] X. Li, B. Li, B. Yang, T. Zeng, General Framework to Histogram-Shifting-Based Reversible Data Hiding, *IEEE Transactions on Image Processing*, Vol. 22, No. 6, pp. 2181–2191, June, 2013, doi: 10.1109/TIP.2013.2246179.

[8] Y.-Y. Chen, C.-H. Hsia, H.-Y. Kao, Y.-A. Wang, Y.-C. Hu, An Image Authentication Method for Secure Internet-Based Communication in Human-Centric Computing, *Journal of Internet Technology*, Vol. 21, No. 7, pp. 1893-1903, December, 2020.

[9] M. Fan, S. Zhong, X. Xiong, Reversible Data Hiding Method for Interpolated Images Based on Modulo Operation and Prediction-Error Expansion, *IEEE Access*, Vol. 11, pp. 27290–27302, March, 2023, doi: 10.1109/ACCESS.2023.3258461.

[10] R. Wang, G. Wu, Q. Wang, L. Yuan, Z. Zhang, G. Miao, Reversible Data Hiding in Encrypted Images Using Median Edge Detector and Two's Complement, *Symmetry*, Vol. 13, No. 6, Article No. 921, June, 2021, doi: 10.3390/sym13060921.

[11] L. Liu, A. Wang, C.-C. Chang, Separable Reversible Data Hiding in Encrypted Images With High Capacity Based on Median-Edge Detector Prediction, *IEEE Access*, Vol. 8, pp. 29639–29647, February, 2020, doi: 10.1109/ACCESS.2020.2972736.

[12] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi, Reversible Watermarking Algorithm Using Sorting and Prediction, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pp. 989–999, July, 2009, doi: 10.1109/TCSVT.2009.2020257.

[13] R. Geetha, S. Geetha, Efficient Rhombus Mean Interpolation for Reversible Data Hiding, *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2018, pp. 1007–1012. doi: 10.1109/RTEICT42901.2018.9012133.

[14] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible Image Watermarking Using Interpolation Technique, *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 1, pp. 187–193, March, 2010, doi: 10.1109/TIFS.2009.2035975.

[15] K.-H. Jung, K.-Y. Yoo, Data hiding method using image interpolation, *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 465–470, February, 2009, doi: 10.1016/j.csi.2008.06.001.

[16] C.-T. Huang, C.-Y. Lin, C.-Y. Weng, Dynamic Information-Hiding Method with High Capacity Based on Image Interpolating and Bit Flipping, *Entropy*, Vol. 25, No. 5, Article No. 744, May, 2023, doi: 10.3390/e25050744.

[17] C.-F. Lee, Y.-L. Huang, An efficient image interpolation increasing payload in reversible data hiding, *Expert Systems with Applications*, Vol. 39, No. 8, pp. 6712–6719, June, 2012, doi: 10.1016/j.eswa.2011.12.019.

[18] Y.-T. Chang, C.-T. Huang, C.-F. Lee, S.-J. Wang, Image interpolating based data hiding in conjunction with pixel-shifting of histogram, *Journal of Supercomputing*, Vol. 66, No. 2, pp. 1093–1110, November, 2013, doi: 10.1007/s11227-013-1016-6.

[19] X. Zhang, Z. Sun, Z. Tang, C. Yu, X. Wang, High capacity data hiding based on interpolated image, *Multimedia Tools and Applications*, Vol. 76, No. 7, pp. 9195–9218, April, 2017, doi: 10.1007/s11042-016-3521-0.

[20] A. Malik, G. Sikka, H. K. Verma, Image interpolation based high capacity reversible data hiding scheme, *Multimedia Tools and Applications*, Vol. 76, No. 22, pp. 24107–24123, November, 2017, doi: 10.1007/s11042-016-4186-4.

[21] A. Malik, G. Sikka, H. K. Verma, A Reversible Data Hiding Scheme for Interpolated Images Based on Pixel Intensity Range, *Multimedia Tools and Applications*, Vol. 79, No. 25-26, pp. 18005–18031, July, 2020, doi: 10.1007/s11042-020-08691-2.

[22] P. Pal, B. Jana, J. Bhaumik, A secure reversible color image watermarking scheme based on LBP, lagrange interpolation polynomial and weighted matrix, *Multimedia Tools and Applications*, Vol. 80, No. 14, pp. 21651–21678, June, 2021, doi: 10.1007/s11042-021-10651-3.

[23] B. Jana, Reversible data hiding scheme using sub-sampled image exploiting Lagrange's interpolating polynomial, *Multimedia Tools and Applications*, Vol. 77, No. 7, pp. 8805–8821, April, 2018, doi: 10.1007/s11042-017-4775-x.

[24] X. Bai, Y. Chen, G. Duan, C. Feng, W. Zhang, A data hiding scheme based on the difference of image interpolation algorithms, *Journal of Information Security and Applications*, Vol. 65, Article No.

103068, March, 2022, doi: https://doi.org/10.1016/j.jisa.2021.103068.

[25] M. Tang, J. Hu, W. Song, S. Zeng, Reversible and adaptive image steganographic method, *AEU - International Journal of Electronics and Communications*, Vol. 69, No. 12, pp. 1745–1754, December, 2015, doi: 10.1016/j.aeue.2015.08.011.

[26] F. S. Hassan, A. Gutub, Efficient reversible data hiding multimedia technique based on smart image interpolation, *Multimedia Tools and Applications*, Vol. 79, No. 39-40, pp. 30087–30109, October, 2020, doi: 10.1007/s11042-020-09513-1.

[27] A. A. Mohammad, A. Al-Haj, M. Farfoura, An improved capacity data hiding technique based on image interpolation, *Multimedia Tools and Applications*, Vol. 78, No. 6, pp. 7181–7205, March, 2019, doi: 10.1007/s11042-018-6465-8.

[28] X. Xiong, L. Wang, Z. Li, C. Ye, Y. Chen, M. Fan, Y. Zhu, An adaptive high capacity reversible data hiding algorithm in interpolation domain, *Signal Procesing*, Vol. 194, Article No. 108458, May 2022, doi: 10.1016/j.sigpro.2022.108458.

[29] H.-X. Chi, J.-H. Horng, C.-C. Chang, Reversible Data Hiding Based on Pixel-Value-Ordering and Prediction-Error Triplet Expansion, *Mathematics*, Vol. 9, No. 14, Article No. 1703, July, 2021, doi: 10.3390/math9141703.

[30] W. He, K. Zhou, J. Cai, L. Wang, G. Xiong, Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion, *Journal of Visual Communication and Image Representation*, Vol. 49, pp. 351–360, November, 2017, doi: 10.1016/j.jvcir.2017.10.001.

[31] W. He, G. Xiong, S. Weng, Z. Cai, Y. Wang, Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion, *Information Sciences*, Vol. 467, pp. 784–799, October, 2018, doi: 10.1016/j.ins.2018.04.088.

[32] W. He, Z. Cai, Y. Wang, Flexible spatial location-based PVO predictor for high-fidelity reversible data hiding, *Information Sciences*, Vol. 520, pp. 431–444, May, 2020, doi: 10.1016/j.ins.2020.02.003.

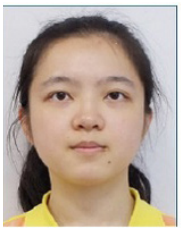
[33] F. Peng, X. Li, B. Yang, Improved PVO-based reversible data hiding, *Digital Signal Processing*, Vol. 25, pp. 255–265, February, 2014, doi: 10.1016/j.dsp.2013.11.002.

[34] T. Zhang, X. Li, W. Qi, Z. Guo, Location-Based PVO and Adaptive Pairwise Modification for Efficient Reversible Data Hiding, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 2306–2319, January, 2020, doi: 10.1109/TIFS.2019.2963766.

[35] S. Meikap, B. Jana, Directional PVO for reversible data hiding scheme with image interpolation, *Multimedia Tools and Applications*, Vol. 77, No. 23, pp. 31281–31311, December, 2018, doi: 10.1007/s11042-018-6203-2.

[36] D. Coltuc, Improved Embedding for Prediction-Based Reversible Watermarking, *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 3, pp. 873–882, September, 2011, doi: 10.1109/TIFS.2011.2145372.

Biographies



Hengxiao Chi received the B.E. degree in Computer Science and Technology from Southeast University, Nanjin, P. R. China, in 2019. She is currently pursuing the Ph.D degree in the Department of Information Engineering and Computer Science, Feng Chia University. Her current research interests include cryptography, information security, and information hiding.

Chin-Chen Chang received a B.S. degree in Applied Mathematics, an M.S. degree in Computer and Decision Sciences from National Tsing Hua University, and a Ph.D. degree in Computer Engineering from National Chiao Tung University. He is currently the Chair Professor of Feng Chia University. His current research interests include database design, cryptography, and data structures.

Chia-Chen Lin (also known as MIN-HUI LIN) received her Ph.D degree in information management from the National Chiao Tung University. She is currently a distinguished professor of the National Chin-Yi University of Technology and University-Affair Advisor of Providence University. Her research interests include information hiding, mobile agent, and electronic commerce.