Displays 90 (2025) 103105

Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

Reversible data hiding in encrypted DICOM images via pattern matching and dynamic Huffman coding☆

Yijie Lin [a], Jui-Chuan Liu [a,*], Ching-Chun Chang [b], Chin-Chen Chang [a,*]

[a] Department of Information Engineering and Computer Science, Feng Chia University, NO.100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, Republic of China
[b] Information and Communication Security Research Center, Feng Chia University, NO.100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, Republic of China

ARTICLE INFO

Keywords:
DICOM images
Encrypted images
Medical data
Reversible data hiding
Smart healthcare

ABSTRACT

With the rapid advancement of smart healthcare, multimedia technology and healthcare are becoming increasingly integrated, making the security of sensitive patient data more critical than ever. Reversible data hiding has emerged as a key technology in multimedia information security and is widely applied in smart healthcare. Digital Imaging and Communications in Medicine (DICOM) is a standard format for medical image transmission and serves as the basis for our proposed reversible data hiding scheme tailored for encrypted DICOM images. By employing pixel bit-slicing, pattern matching, and dynamic Huffman coding, we embed secret data into the encrypted images. Upon receiving the encrypted images, the recipients can either extract the embedded data or reconstruct the original medical images using access keys. Our proposed scheme is fully reversible and offers high encryption performance. Compared with state-of-the-art schemes, it provides a higher embedding capacity with the best cases reaching over 10 bits per pixel.

1. Introduction

As information technology advances rapidly, various industries are entering the digital era. In particular, the healthcare sector has undergone significant transformation, especially accelerated by the outbreak of COVID-19. Since then, smart healthcare has attracted significant attention worldwide. E-health and telemedicine have become important channels for doctors and patients to communicate without physical contact [1]. However, transmitting large amounts of data carries risks, as many criminals target confidential patient information. Therefore, protecting personal privacy and hospital-owned data is crucial. In response to these risks, a growing number of information security technologies [2–5] have been widely applied in the field of smart healthcare [6,7]. Among them, data hiding [8,9] is an effective multimedia information security technology. It can be applied to carriers including, but not limited to, text, audio, images, and video [10,11].

Data hiding techniques are typically divided into irreversible data hiding [12–15] and reversible data hiding [16–21]. Given the high standards for image quality in smart healthcare, irreversible data hiding is sometimes chosen when acceptable visual quality can still be maintained. However, reversible data hiding is generally preferred, as it fully preserves the original image [22]. For applications with stricter confidentiality requirements, encrypted images are commonly used. Over the past decade, reversible data hiding in encrypted images (RDHEI) [23–28] has developed into a mature field [29,30]. Digital Imaging and Communications in Medicine (DICOM) [31], a widely used standard for medical image storage and transmission. It has seen various data hiding schemes [32–34] applied to DICOM images. Nevertheless, to the best of our knowledge, at present, only three reversible data hiding schemes have been specifically developed for encrypted DICOM images [35–37]. These pioneering works have laid the groundwork for this field, establishing it as a novel and promising research area. In this context, we propose a reversible data hiding scheme for encrypted DICOM images using pattern matching and dynamic Huffman coding (PM-DHC). The proposed scheme begins with pixel bit-slicing and follows by pattern matching to determine the corresponding pattern for each pixel. The patterns of all pixels are then compressed using dynamic Huffman coding and further optimized based on specialized types. Subsequently, the confidential information is encrypted using a data hiding key, while the auxiliary information required for image

☆ This paper was recommended for publication by Prof Guangtao Zhai.

* Corresponding authors.
E-mail addresses: p1263670@o365.fcu.edu.tw (Y. Lin), p1200318@o365.fcu.edu.tw (J.-C. Liu), ccc@fcu.edu.tw (C.-C. Chang), ccc@o365.fcu.edu.tw (C.-C. Chang).

https://doi.org/10.1016/j.displa.2025.103105
Received 12 May 2025; Received in revised form 27 May 2025; Accepted 29 May 2025
Available online 29 May 2025

reconstruction is encrypted with an encryption key to produce the encrypted stream and thereby generate the encrypted image. For the receiver, if the data hiding key is available, the confidential information can be successfully extracted. Furthermore, if the encryption key is available, the original image can be accurately reconstructed. The main contributions of the proposed PM-DHC scheme are listed below:

- A reversible data hiding scheme in encrypted DICOM images is proposed, using pattern matching and dynamic Huffman coding.
- The proposed PM-DHC scheme excels in terms of encrypted visual quality, encryption efficiency, and security.
- Compared to state-of-the-art schemes, the proposed PM-DHC scheme offers a significant advantage in embedding capacity.

The remainder of this paper is organized as follows: Section II reviews previous related studies; Section III outlines the proposed PM-DHC scheme in detail; Section IV presents the experiments conducted and analyzes the corresponding results; finally, Section V concludes this paper.

2. Related works

This section provides an overview of the relevant related works. Subsection II.A introduces the Digital Imaging and Communications in Medicine standard; subsection II.B reviews state-of-the-art reversible data hiding schemes in encrypted DICOM images; and subsection II.C provides a brief overview of Huffman coding.

2.1. Digital Imaging and Communications in Medicine

Digital Imaging and Communications in Medicine (DICOM) [31] is an international standard for storing, transmitting, and managing medical images and related information. It is widely used for data exchange between CT, MRI, X-ray, and other medical equipment. DICOM supports multiple pixel depths, including the commonly used 16-bit format. It provides a very high dynamic range and detailed presentation capabilities to facilitate accurate diagnoses. The unified standard promotes seamless connectivity and efficient collaboration between medical equipment and information systems.

2.2. Reversible data hiding in encrypted DICOM images

With the widespread adoption of DICOM, recent state-of-the-art research has increasingly focused on reversible data hiding in encrypted DICOM images to safeguard confidential medical information. In 2021, Dzwonkowski and Rykaczewski [35] proposed an efficient coding scheme that utilized cyclic binary Hamming (7,4) and Golay (23,12) codes to embed additional data into the least significant bits of encrypted images. In 2022, Dzwonkowski and Czaplewski [36] introduced a technique that involved binary decomposition of input data and sorting of the resulting binary sequence. By leveraging the compression properties of run-length encoding, they improved embedding capacity by utilizing the most significant bit plane within each predefined data block. In 2024, Panchikkil et al. [37] proposed a method that divided images into non-overlapping blocks of equal size and adaptively embedded information related to actual pixel values into the encrypted blocks using a most significant bit prediction error approach. These advancements have laid a strong foundation for further research into reversible data hiding in encrypted DICOM images. Building upon these foundations, the proposed scheme first classifies each pixel into a pattern type using pattern matching, and then applies dynamic Huffman coding for compression, thereby achieving higher embedding capacity.

2.3. Huffman coding

Huffman coding [38] is a widely used lossless data compression algorithm that constructs an optimal binary tree based on the frequency of symbol occurrences. It assigns shorter codes to symbols with higher frequencies and longer codes to symbols with lower frequencies, thereby effectively reducing the overall size of the data. For example, consider the string “AAAABBCD”. The occurrence frequency for each symbol is as follows: A: 4, B: 2, C: 1, D: 1. If fixed-length coding were used, the symbols A, B, C, and D would be represented by 00, 01, 10, and 11, respectively, resulting in an encoding length of 16 bits: 0000000001011011. When using Huffman coding, the symbols are assigned the following codes: A: 0, B: 10, C: 110, D: 111. The resulting encoded string is 00001010110111, which is 14 bits long.

3. Proposed PM-DHC scheme

This section specifically introduces the PM-DHC scheme we proposed for reversible data hiding in encrypted DICOM images. Fig. 1 shows the flow of the PM-DHC scheme. The sender first performs pixel bit-slicing on the original DICOM image to obtain different patterns. Next, the symbols of the patterns are Huffman encoded according to their frequencies. With a generated data hiding key K_d and a generated encryption key K_e, different secret data embedding strategies are then applied to the different patterns to construct the encrypted image. After receiving the encrypted image, the receiver can extract the secret data using the data hiding key K_d and reconstruct the DICOM image using the encryption key K_e.

3.1. Pixel bit-slicing and pattern matching

The DICOM images are typically stored with 16 bits per pixel but use a 12-bit depth for grayscale representation. As a result, the 4 most significant bits (MSBs) of each pixel are set to 0. For this special data structure of a DICOM image, the 16 bits of each pixel are sliced into 4 parts. With each part represented by a 4-bit symbol, it forms a distinct 4-symbol pattern.

Table 1 summarizes 25 observed patterns, with one requiring further processing. In Table 1, the symbols in a pattern are defined as follows: *o* represents 0000; *s* represents the secret data embedded in the first stage; *a*, *b*, and *c* represent other distinct 4-bit symbols.

The pattern for Type 1 is *oooo*, meaning the pixel has 16 bits of 0, i.e., the pixel value is 0. This indicates that no information needs to be preserved; the preserved bit count is 0, and the number of vacated bits is 16. The patterns for Type 2– Type 4 are *oooa*, *ooao*, and *oaoo* respectively, meaning that *a* needs to be preserved. *a* consists of 4 bits, and the rest 12 bits can be vacated. The patterns for Type 5– Type 7 are *ooaa*, *oaoa*, and *oaao*, respectively. Although *a* appears twice, only one *a* needs to be preserved using 4 bits, and 12 bits can be vacated. The patterns for Type 8– Type 10 are *ooab*, *oaob*, and *oabo*. In these cases, both *a* and *b* need to be preserved by occupying 8 bits and leaving 8 vacated bits. For Type 1– Type 10, the first stage of embedding must be performed and the secret data is embedded into the first 4 most significant bits. The first *o* symbol is changed to the symbol *s*. Type 11– Type 20 are similar to Type 1– Type 10 except that an *o* is replaced by an *s*. The pattern for Type 21 is *saaa*, where *a* appears three times but only one *a* needs to be preserved. *a* is 4 bits, leaving 12 vacated bits. The patterns for Type 22– Type 24 are *saab*, *saba*, and *sbaa*. With *a* and *b* preserved a total of 8 bits, the number of vacated bits is 8. Finally, Type 25 corresponds to *sabc* and requires to preserve 12 bits. After the types of pixels are encoded using dynamic Huffman coding, Type 25 undergoes further processing.

3.2. Dynamic Huffman coding

For each pixel, additional auxiliary information is required to record its corresponding type. Since different types occur with varying frequencies, fixed-length coding is not the best choice. Therefore, Huffman coding is selected. Shorter codes are assigned to high-frequency types, while longer codes are assigned to low-frequency types. Given that the

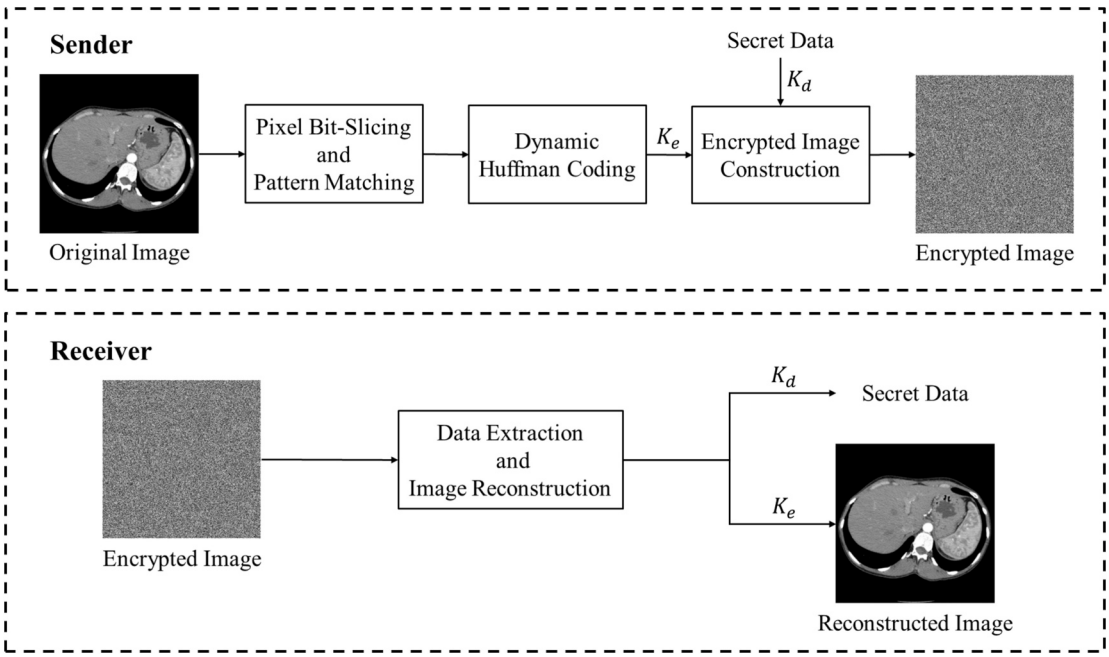

Fig. 1. Flow of the proposed PM-DHC scheme.

Table 1
Pattern Matching Guide.

Type	Pattern	Preserved Bits	Vacated Bits
1	*oooo*	0	16
2	*oooa*	4	12
3	*ooao*	4	12
4	*oaoo*	4	12
5	*ooaa*	4	12
6	*oaoa*	4	12
7	*oaao*	4	12
8	*ooab*	8	8
9	*oaob*	8	8
10	*oabo*	8	8
11	*ssss*	0	16
12	*sssa*	4	12
13	*ssas*	4	12
14	*sass*	4	12
15	*ssaa*	4	12
16	*sasa*	4	12
17	*saas*	4	12
18	*ssab*	8	8
19	*sasb*	8	8
20	*sabs*	8	8
21	*saaa*	4	12
22	*saab*	8	8
23	*saba*	8	8
24	*sbaa*	8	8
25	*sabc*	12	4

proposed scheme involves different vacated bits, dynamic Huffman coding is introduced. The coding adjusts according to the relationship between Huffman coding and vacated bits to achieve shorter codes by deleting or merging less frequent types.

As shown in Algorithm 1, the dynamic Huffman coding process is detailed in pseudocode using MATLAB functions. First, the pixel type table *PTT*, which records the types of all pixels, is input. Then, all unique types and their counts are extracted, followed by the calculation of their frequencies to build the Huffman tree. For each type, the current Huffman code *hc* and vacated bits *vb* are obtained. The pure vacated bits *pvb* is derived as $pvb = vb - hc$. After getting all *pvb* values, they are compared with the *pvb* of the last type, which is initialized as the original Type 25. If a *pvb* value is smaller than that of the last type, the current type is set as the new last type. After all types have been processed, the Huffman tree is rebuilt based on the updated types and frequencies, and this process is repeated until the *pvb* of the last type is the smallest one. Finally, the Huffman code is output based on the final Huffman tree, and the Huffman stream *HS* is produced.

Algorithm 1 Dynamic Huffman Coding

Input	Pixel type table *PTT*.
Output	Huffman codes *HC*.
Step 1	// Initialization
	types = unique(*PTT*)
	counts = histcounts(*PTT*, [types, max(types) + 1])
	FT = [types', counts']
Step 2	// Build Huffman tree iteratively
	do
	mergeCount = 0;
	prob = counts / sum(counts)
	huffmanTree = huffmandict(types, prob)
	// *vb* from pattern matching guide, *hc* from Huffman tree.
	$pvb_{num} = vb_{num} - hc_{num}$;
	for i = 1 **to** num − 1
	$pvb_i = vb_i - hc_i$;
	if $pvb_i < pvb_{num}$
	counts[num] = counts[num] + counts[i];
	counts[i] = 0;
	mergeCount = mergeCount + 1;
	end if
	end for
	types = types[counts > 0]
	counts = counts[counts > 0]
	while mergeCount > 0
Step 3	// Generate Huffman stream
	HS = huffmanenco(*PTT*, huffmanTree)
Step 4	Output Huffman stream *HS*.

For Type 25, 12 bits must be preserved. To mitigate the resulting redundancy, pixel prediction is employed to compress the data. First, the reference pixel *rp* must be selected. Since most medical devices that generate DICOM images scan horizontally, horizontal pixel values are used as references. As shown in Eq. (1), *i* and *j* represent the row and column indices of the current pixel *p*. For prediction, the pixel value to the left of the current pixel is used. However, if $j = 1$, meaning the current pixel is in the first column and there is no pixel to its left, the pixel directly above is selected as the reference. If the current pixel is

located at $i = 1, j = 1$, i.e., in the upper-left corner, it is excluded from the compression process.

The error e is then calculated as shown in Eq. (2), where the predicted pixel rp is subtracted from the current pixel p. Since the pixel values of adjacent pixels in the image are typically similar, most errors are either 0 or small values close to 0, resulting in a highly concentrated distribution. Leveraging this high-frequency characteristic, Huffman coding is employed to efficiently compress the errors and vacate more bits.

$$rp_{i,j} = \begin{cases} p_{i,j-1}, & if j \neq 1 \\ p_{i-1,j}, & if j = 1 and i \neq 1 \end{cases} \tag{1}$$

$$e_{i,j} = p_{i,j} - rp_{i,j} \tag{2}$$

3.3. Encrypted image construction

The stream cipher algorithm is used as the encryption method for the proposed PM-DHC; however, other effective encryption algorithms can be applied as well. In subsection III.A, all preserved and vacated bits are obtained, while in subsection III.B, each pixel is associated with a type determined by the pattern matching guide, encoded using dynamic Huffman coding, and the corresponding Huffman stream is generated. Additionally, Type 25 is further compressed to vacate more bits.

Our final encrypted stream consists of the lengths of the auxiliary information, the auxiliary information and the secret data. The auxiliary information includes preserved bits, the dynamic Huffman coding rule, the dynamic Huffman stream, the Huffman coding rule of Type 25, and the Huffman stream of Type 25. The lengths of these components are not encrypted and are stored at the beginning of the stream. First, the auxiliary information AI is encrypted by generating a pseudo-random stream R_{AI} that is of the same length as the auxiliary information, and the encrypted auxiliary information EAI is then calculated according to Eq. (3). The space remaining after deducting the auxiliary information and its length information is used to embed the secret data. Similarly, the secret data SD is encrypted with the pseudo-random stream R_{SD}, which is of the same length as the secret data, as shown in Eq. (4), to obtain ESD. The generated streams R_{AI} and R_{SD} are used as the encryption key K_e and the data-hiding key K_d respectively. Finally, the lengths of the auxiliary information, the encrypted auxiliary information EAI, and the encrypted secret data ESD are concatenated to form the encrypted stream. Every 16 bits of this stream are then treated as a pixel to construct the encrypted image.

$$EAI = AI \oplus R_{AI} \tag{3}$$

$$ESD = SD \oplus R_{SD} \tag{4}$$

3.4. Data extraction and image reconstruction

Upon receiving the encrypted image, the receiver first converts it into an encrypted stream. The receiver then reads the length information to identify the starting positions of the auxiliary information and the secret data. If a data hiding key K_d (i.e., R_{SD}) is present, the secret data SD can be extracted according to Eq. (5). If an encryption key K_e (i.e., R_{AI}) is available, the auxiliary information AI can be extracted according to Eq. (6). Next, using the length information, the auxiliary information is split into components: preserved bits, the dynamic Huffman coding rule, the dynamic Huffman stream, the Type 25 Huffman coding rule, and the Type 25 Huffman stream. Since the bit depth of the DICOM image is 12 bits, the first 4 MSBs of all pixels are initially set to 0. Then, based on the dynamic Huffman coding rule and the dynamic Huffman stream, the type of each pixel is determined. The preserved bits are subsequently filled into the corresponding positions of each pixel according to its pixel type. For Type 25, the reference pixel rp is calculated according to Eq. (1), and the error e is derived using the Type 25 Huffman coding rule and the Type 25 Huffman stream. The original pixel p is then calculated according to Eq. (7). After processing all pixels, the DICOM image can be reconstructed.

$$SD = ESD \oplus R_{SD} \tag{5}$$

$$AI = EAI \oplus R_{AI} \tag{6}$$

$$p_{i,j} = rp_{i,j} + e_{i,j} \tag{7}$$

4. Experimental results

In this section, experimental analyses are conducted to evaluate the performance of the proposed PM-DHC scheme and to compare it with state-of-the-art reversible data hiding schemes for encrypted DICOM images. The test images are sourced from the DICOM Library [39], which includes datasets such as CT, MR, OPT, and OT. Fig. 2 shows six 512×512 test images selected from a series within a CT dataset in the library. The experimental results demonstrate that the proposed scheme outperforms the state-of-the-art in terms of performance.

Fig. 3 illustrates the images at different stages and their corresponding histograms. Fig. 3 (a-c) depict the original image, encrypted image, and reconstructed image respectively, while Fig. 3 (d-f) show their corresponding histograms. It can be observed that due to the content characteristics of medical images, the background often appears predominantly black with low pixel values. As a result, the high frequencies in the histogram are concentrated in the low pixel value region. The visual appearance of the encrypted image resembles random noise, and its histogram is relatively evenly distributed, indicating good encryption performance. Since the proposed PM-DHC scheme is reversible, the original image can be reconstructed losslessly and the reconstructed image is identical to the original, along with its corresponding histogram.

Table 2 provides a comprehensive evaluation of the encrypted images generated by the proposed PM-DHC scheme. To assess visual quality, the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are utilized. The results indicate that the PSNR of the test image is approximately 4.87 dB, while the SSIM is around 0.0004. A lower PSNR and SSIM signify reduced similarity between the encrypted and original images, thereby demonstrating that the proposed scheme achieves a good visual encryption effect.

The encryption performance is further analyzed using the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). For a 16-bit DICOM image, the theoretical ideal NPCR value is 99.9985 %, and the ideal UACI value is 33.3338 %. However, given that this study targets medical images, it is important to note that many DICOM images produced by medical imaging devices often contain substantial black background regions. Consequently, the ideal UACI value approaches 50 % in the presence of extensive black backgrounds. Experimental results reveal that both NPCR and UACI are close to their respective ideal values, underscoring the efficacy of the proposed encryption scheme.

An entropy analysis was conducted to evaluate the randomness of the encrypted images. Since each pixel in a DICOM image is represented using 16 bits, the experimental results show that the average information entropy of the encrypted images is 15.81, which is close to the theoretical maximum of 16. This finding demonstrates that the grayscale distribution of the encrypted images approximates complete randomness, thereby ensuring high security.

To demonstrate the effectiveness and necessity of the proposed dynamic Huffman coding, the embedding capacity, measured in bits per pixel (bpp), is evaluated using different coding methods. As shown in Table 3, fixed-length coding, Huffman coding, and the proposed dynamic Huffman coding are compared. When fixed-length coding is employed, the average embedding capacity is 6.4757 bpp. With Huffman coding, effective compression is achieved and the average

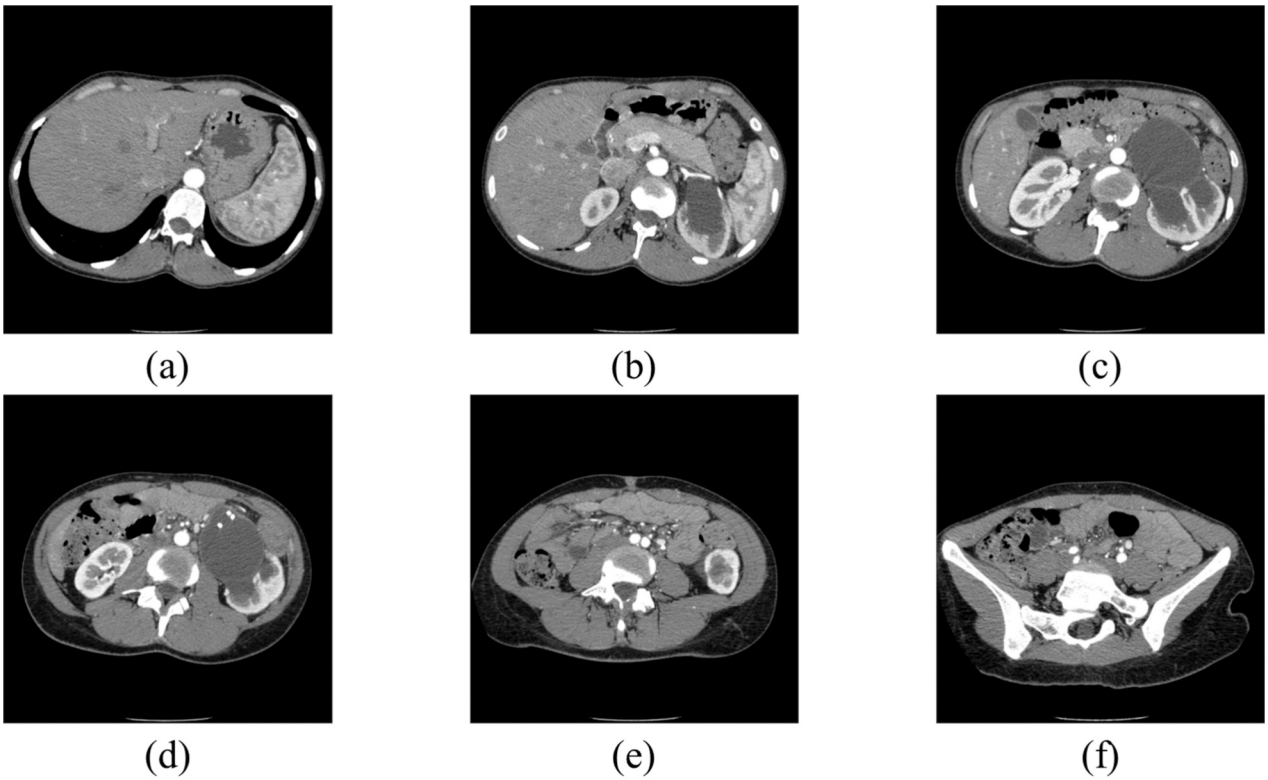

Fig. 2. Test images.

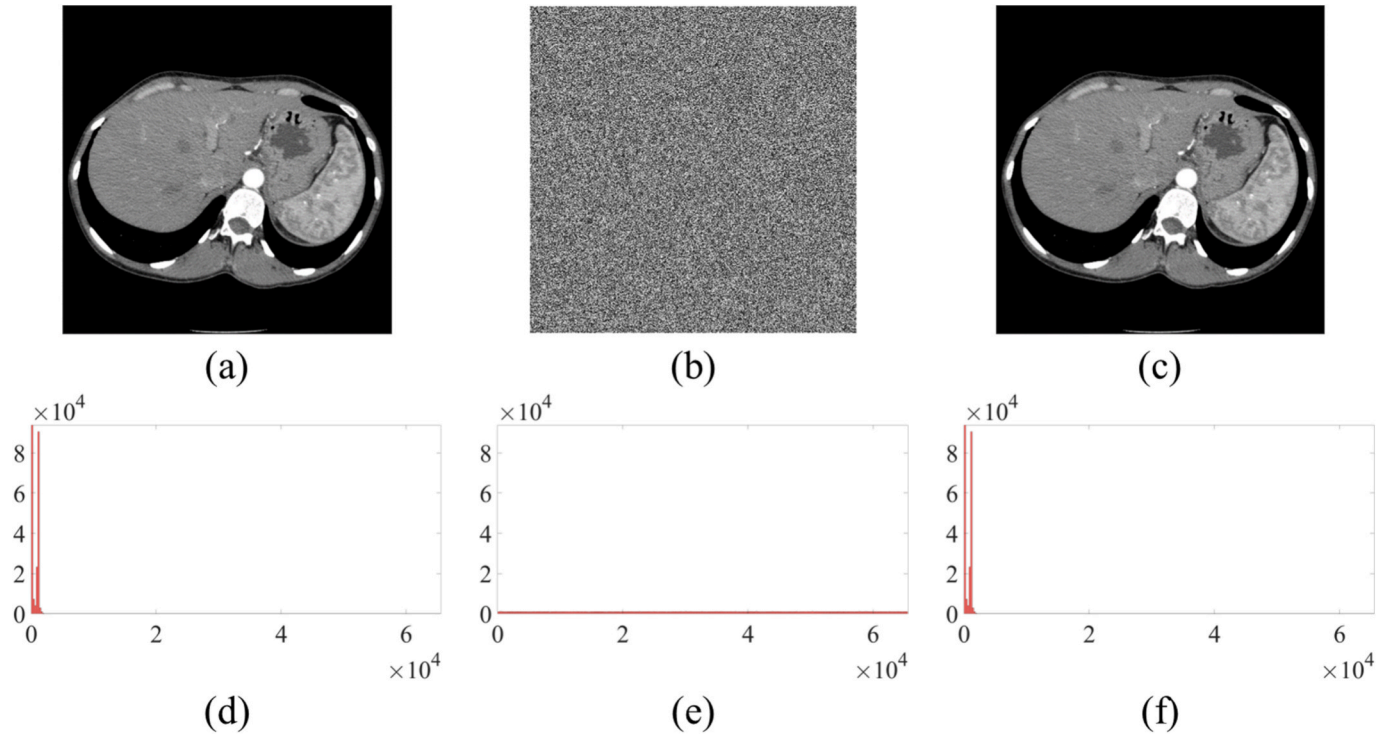

Fig. 3. Images and histograms at different stages. (a) Original image. (b) Encrypted image. (c) Reconstructed image. (d–f) Corresponding histograms for (a–c), respectively.

embedding capacity is increased to 8.4928 bpp. The dynamic Huffman coding proposed in this study achieves even greater efficiency, further improving the average embedding capacity to 8.8826 bpp. These results demonstrate that the dynamic Huffman coding proposed in this study offers significant advantages in compression performance, and its efficient compression capability directly enhances the embedding capacity.

The embedding capacity of the proposed scheme is further compared with those of other state-of-the-art reversible data hiding schemes in

Table 2
Visual quality, encryption performance, and security evaluation of encrypted images generated by the proposed PM-DHC scheme.

Metric	(a)	(b)	(c)	(d)	(e)	(f)
PSNR	4.8720	4.8752	4.8676	4.8499	4.8645	4.8712
SSIM	0.0004	0.0004	0.0003	0.0003	0.0003	0.0004
NPCR	99.9992	99.9973	99.9985	99.9989	99.9992	99.9996
UACI	49.2208	49.1937	49.2914	49.3944	49.3178	49.1361
Entropy	15.8086	15.8062	15.8078	15.8081	15.8082	15.8076

Table 3
Evaluation of embedding capacity in the proposed PM-DHC scheme using different coding methods.

Coding Method	(a)	(b)	(c)	(d)	(e)	(f)
Fixed-Length Coding	6.1184	6.2547	6.5222	6.6852	6.7693	6.5044
Huffman Coding	8.0696	8.1911	8.5338	8.7328	8.8520	8.5776
Dynamic Huffman Coding	8.0902	8.5229	8.8396	9.0221	10.1971	8.6239

encrypted DICOM images. As shown in Table 4, the average embedding capacity of Dzwonkowski and Rykaczewski's scheme [35] is 1.7143 bpp when using the Hamming (7,4) code and 1.9130 bpp when using the Golay (23,12) code. The average embedding capacity of Dzwonkowski and Czaplewski's scheme [36] is 7.8739 bpp, while Panchikkil et al.'s scheme [37] achieves an average embedding capacity of 5.8146 bpp. In comparison, the proposed scheme achieves an average embedding capacity of 8.8826 bpp, which is significantly higher than those of the other schemes. Even when compared with Dzwonkowski and Czaplewski's scheme, which offers the highest embedding capacity among the existing schemes, the proposed scheme improves the embedding capacity by approximately 1 bpp. These results demonstrate that the proposed scheme provides significant advantages in embedding capacity.

To avoid potential bias in the experimental results caused by a small sample size, a total of 300 test images were sampled from various datasets in the DICOM Library [39] to evaluate the embedding capacity performance and compare it with other state-of-the-art reversible data hiding schemes for encrypted DICOM images. Fig. 4 and Table 5 present the comparisons of embedding capacities. The results of Dzwonkowski and Rykaczewski's [35] scheme across different images are constant, with the Hamming (7,4) code and Golay (23,12) code achieving 1.7143 bpp and 1.9130 bpp, respectively. In contrast, the maximum value of Dzwonkowski and Czaplewski's [36] scheme is as high as 8.2248 bpp, the minimum is 5.0167 bpp, and the average is 7.8078 bpp, indicating that the performance of their method is unstable and highly depends on the smoothness of the image. The maximum value of Panchikkil et al.'s [37] scheme is 5.8594 bpp, the minimum is 5.5273 bpp, and the average is 5.8151 bpp. Although the performance is stable, the embedding capacity remains relatively lower. The proposed PM-DHC scheme outperforms existing state-of-the-art schemes and achieves a maximum

Table 4
Comparison of the embedding capacity among the proposed PM-DHC scheme and other reversible data hiding schemes in encrypted DICOM images.

Schemes	(a)	(b)	(c)	(d)	(e)	(f)
[35] with Hamming (7,4) code	1.7143	1.7143	1.7143	1.7143	1.7143	1.7143
[35] with Golay (23,12) code	1.9130	1.9130	1.9130	1.9130	1.9130	1.9130
[36]	7.5984	7.7782	8.0400	8.1915	8.1467	7.4884
[37]	5.7129	5.8350	5.8447	5.8447	5.8506	5.7998
Proposed	8.0902	8.5229	8.8396	9.0221	10.1971	8.6239

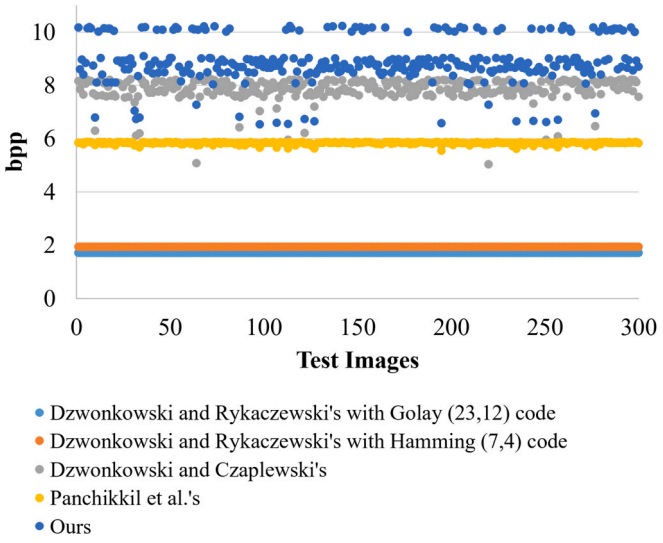

Fig. 4. Comparison of the embedding capacity among the proposed PM-DHC scheme and other reversible data hiding schemes in encrypted DICOM images using 300 test images.

Table 5
Maximum, minimum, and average embedding capacity comparison among the proposed PM-DHC scheme and other reversible data hiding schemes in encrypted DICOM images using 300 test images.

Schemes	Maximum	Minimum	Average
[35] with Hamming (7,4) code	1.7143	1.7143	1.7143
[35] with Golay (23,12) code	1.9130	1.9130	1.9130
[36]	8.2248	5.0167	7.8078
[37]	5.8594	5.5273	5.8151
Proposed	10.2174	6.5149	8.8899

value of 10.2174 bpp, which is approximately 2 bpp higher than the current best-performing Dzwonkowski and Czaplewski's [36] scheme. The minimum value reaches 6.5149 bpp, which is about 1 bpp higher than the best-performing Panchikkil et al.'s [37] scheme. In terms of average bpp, it exceeds the best-performing Dzwonkowski and Czaplewski's [36] scheme by more than 1 bpp. These experimental results clearly demonstrate that the proposed scheme offers a significant advantage in embedding capacity.

To further evaluate the efficiency of the proposed PM-DHC scheme, Table 6 summarizes the execution times of the scheme evaluated using 300 test images. These results demonstrate that the proposed scheme can complete image encryption and data embedding in about 2 s, confirming its efficiency.

5. Conclusions

This paper introduces a novel PM-DHC reversible data hiding scheme for encrypted DICOM images. By taking advantage of the multimedia characteristics of medical images, the scheme achieves high embedding capacity through pattern matching and dynamic Huffman coding. This approach enables the seamless embedding of confidential information, such as diagnostic reports and patient data, directly into encrypted medical images to reduce additional storage risks and prevent index

Table 6
Minimum, average, and maximum execution times of the proposed PM-DHC scheme using 300 test images.

Maximum	Minimum	Average
2.47	2.07	2.17

confusion. Experimental results show that the proposed scheme outperforms state-of-the-art schemes in terms of encrypted visual quality, encryption efficiency, and security, while also offering superior embedding capacity. These advantages make the scheme suitable for practical scenarios such as secure image transmission and integrated patient data storage in medical institutions, although deployment would require compatibility with clinical systems and data regulations. Future research will focus on further exploring the multimedia characteristics of medical images to enhance embedding capacity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data are contained within the article.

References

[1] Q. Wang, M. Su, M. Zhang, R. Li, Integrating digital technologies and public health to fight Covid-19 pandemic: key technologies, applications, challenges and outlook of digital healthcare, *Int. J. Environ. Res. Public Health* 18 (11) (2021) 6053.

[2] F. Chen, Y. Tang, C. Wang, J. Huang, C. Huang, D. Xie, T. Wang, C. Zhao, Medical cyber–physical systems: A solution to smart health and the state of the art, *IEEE Trans. Comput. Social Syst.* 9 (5) (2021) 1359–1386.

[3] Y. Sun, J. Liu, K. Yu, M. Alazab, K. Lin, PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare, *IEEE Trans. Ind. Inf.* 18 (3) (2021) 1981–1990.

[4] P. Sarosh, S.A. Parah, B.A. Malik, M. Hijji, K. Muhammad, Real-time medical data security solution for smart healthcare, *IEEE Trans. Ind. Inf.* 19 (7) (2022) 8137–8147.

[5] S. Datta, S. Namasudra, Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile edge computing, *IEEE Trans. Consum. Electron.* (2024).

[6] C.C. Chang, X. Wang, J.H. Horng, I. Echizen, Progressive transmission of medical images via a bank of generative adversarial networks, *J. Healthcare Eng.* 2021 (2021) 9917545.

[7] Y. Lin, C.C. Lin, C.C. Chang, C.C. Chang, An IoT-based electronic health protection mechanism with AMBTC compressed images, *IEEE Internet Things J.* (2024).

[8] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, *IBM Syst. J.* 35 (3.4) (1996) 313–336.

[9] M. Wu, B. Liu, Data hiding in binary image for authentication and annotation, *IEEE Trans. Multimedia* 6 (4) (2004) 528–538.

[10] C.C. Chang, Reversible linguistic steganography with Bayesian masked language modeling, *IEEE Trans. Comput. Social Syst.* 10 (2) (2022) 714–723.

[11] Y. Lin, J.C. Liu, C.C. Chang, C.C. Chang, A puzzle matrix oriented secret sharing scheme for dual images with reversibility, *Signal Process.* 236 (2025) 110056.

[12] A. Anand, A.K. Singh, A hybrid optimization-based medical data hiding scheme for industrial internet of things security, *IEEE Trans. Ind. Inf.* 19 (1) (2022) 1051–1058.

[13] A. Anand, A.K. Singh, H. Zhou, ViMDH: visible-imperceptible medical data hiding for internet of medical things, *IEEE Trans. Ind. Inf.* 19 (1) (2022) 849–856.

[14] Y. Lin, C.C. Lin, J.C. Liu, C.C. Chang, Verifiable (t, n) secret image sharing scheme based on slim turtle shell matrix, *J. Inf. Security Appl.* 80 (2024) 103679.

[15] C.C. Chang, I. Echizen, Steganography in game actions, *IEEE Access* (2025).

[16] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Trans. Circuits Syst. Video Technol.* 16 (3) (2006) 354–362.

[17] Y.Q. Shi, X. Li, X. Zhang, H.T. Wu, B. Ma, Reversible data hiding: advances in the past two decades, *IEEE Access* 4 (2016) 3210–3237.

[18] W. He, Z. Cai, Reversible data hiding based on dual pairwise prediction-error expansion, *IEEE Trans. Image Process.* 30 (2021) 5045–5055.

[19] L. Xiong, X. Han, C.N. Yang, Y.Q. Shi, Reversible data hiding in shared images with separate cover image reconstruction and secret extraction, *IEEE Trans. Cloud Comput.* (2024).

[20] Y. Lin, J.C. Liu, C.C. Chang, C.C. Chang, An innovative recompression scheme for VQ index tables, *Future Internet* 16 (8) (2024) 297.

[21] Y. Lin, J.C. Liu, C.C. Chang, C.C. Chang, Lossless recompression of vector quantization index table for texture images based on adaptive huffman coding through multi-type processing, *Symmetry* 16 (11) (2024) 1419.

[22] C.C. Chang, Bayesian neural networks for reversible steganography, *IEEE Access* 10 (2022) 36327–36334.

[23] X. Zhang, Reversible data hiding in encrypted image, *IEEE Signal Process Lett.* 18 (4) (2011) 255–258.

[24] X. Zhang, Separable reversible data hiding in encrypted image, *IEEE Trans. Inf. Forensics Secur.* 7 (2) (2011) 826–832.

[25] Y. Wang, W. He, High capacity reversible data hiding in encrypted image based on adaptive MSB prediction, *IEEE Trans. Multimedia* 24 (2021) 1288–1298.

[26] Z. Hua, Y. Wang, S. Yi, Y. Zhou, X. Jia, Reversible data hiding in encrypted images using cipher-feedback secret sharing, *IEEE Trans. Circuits Syst. Video Technol.* 32 (8) (2022) 4968–4982.

[27] Y. Wang, G. Xiong, W. He, High-capacity reversible data hiding in encrypted images based on pixel-value-ordering and histogram shifting, *Expert Syst. Appl.* 211 (2023) 118600.

[28] X. Jiang, Y. Xie, Y. Zhang, Y. Ye, F. Xu, L. Li, Y. Su, Z. Chen, Reversible data hiding in encrypted images using reservoir computing based data fusion strategy, *IEEE Trans. Circuits Syst. Video Technol.* (2024).

[29] C.C. Chang, C.T. Li, Y.Q. Shi, Privacy-aware reversible watermarking in cloud computing environments, *IEEE Access* 6 (2018) 70720–70733.

[30] C.C. Chang, C.T. Li, K. Chen, Privacy-preserving reversible information hiding based on arithmetic of quadratic residues, *IEEE Access* 7 (2019) 54117–54132.

[31] O.S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM): a Practical Introduction and Survival Guide*, Vol. 10, Springer, Heidelberg, 2012.

[32] A. Elhadad, A. Ghareeb, S. Abbas, A blind and high-capacity data hiding of DICOM medical images based on fuzzification concepts, *Alex. Eng. J.* 60 (2) (2021) 2471–2482.

[33] M.A. Ahmad, M. Elloumi, A.H. Samak, A.M. Al-Sharafi, A. Alqazzaz, M.A. Kaid, C. Iliopoulos, Hiding patients' medical reports using an enhanced wavelet steganography algorithm in DICOM images, *Alex. Eng. J.* 61 (12) (2022) 10577–10592.

[34] N. Rajesh Kumar, R. Bala Krishnan, G. Manikandan, V. Subramaniyaswamy, K. Kotecha, Reversible data hiding scheme using deep learning and visual cryptography for medical image communication, *J. Electron. Imaging* 31 (6) (2022) 063028.

[35] M. Dzwonkowski, R. Rykaczewski, Reversible data hiding in encrypted DICOM Images using cyclic binary golay (23, 12) code, *IEEE Access* 9 (2021) 60503–60515.

[36] M. Dzwonkowski, B. Czaplewski, Reversible data hiding in encrypted DICOM images using sorted binary sequences of pixels, *Signal Process.* 199 (2022) 108621.

[37] S. Panchikkil, V.M. Manikandan, P. Pratim Roy, S. Wang, Y. Zhang, An adaptive block-wise prediction error-based (AdaBPE) reversible data hiding in encrypted images for medical image transmission, *CAAI Trans. Intell. Technol.* (2024).

[38] D.A. Huffman, A method for the construction of minimum-redundancy codes, *Proc. IRE* 40 (9) (1952) 1098–1101.

[39] DICOM (2022) Accessed 21 November 2023 image database, URL: https://www.dicomlibrary.com/.