IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 11, 2021, accepted December 15, 2021, date of publication January 18, 2022, date of current version February 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3144322

Efficient Electronic Patient Information Hiding Scheme With Tamper Detection Function for Medical Images

CHIA-CHEN LIN[1], CHIN-CHEN CHANG[2], (Fellow, IEEE), WEI-JIUN KAO[2], AND JUI-FENG CHANG[2]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan
[2]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407802, Taiwan

Corresponding authors: Chia-Chen Lin (ally.cclin@ncut.edu.tw) and Chin-Chen Chang (alan3c@gmail.com)

This work was supported in part by the Project of Ministry of Science and Technology of Taiwan under Grant MOS 110-2410-H-167 -004.

ABSTRACT With the rapid development of the electronic medical information systems and Internet technology, the frequency of electronic medical records (EMRs) and electronic patient information (EPIs) transmitted over the Internet and shared among authorized medical parties is gradually increased. Reversible data hiding in medical images is an efficient technique to embed an EPI or EMR into a medical image, such as X-ray, MRI etc. In this paper, a reversible data hiding scheme by using enhanced neighbor mean interpolation (ENMI) technology is proposed to expand each 2×2 block in the original image to a 3×3 block. The four original pixels remaining at the four corners in each expanded block. Five derived pixels by using the ENMI interpolation algorithm are used to carry 4-bits secret data and a 1-bit authentication code with our proposed efficient data hiding strategy. Experimental results confirm that the detection performance and the robustness of the hidden data with our proposed scheme are significantly higher than that of the Geeth & Geeth scheme and other six representative schemes.

INDEX TERMS Data hiding, interpolation, tamper detection, electronic patient information, medical images.

I. INTRODUCTION

In the last two decades, data hiding techniques have drawn more research attention because security is increasingly important as transmitted data grows exponentially. Moreover, data hiding techniques do not have several of the limitations found in cryptography. Modern cryptography is heavily based on mathematical theory and computer science practice. Among cryptographic algorithms [1], no matter whether they are symmetric cryptographic algorithms or asymmetric cryptographic algorithms, a plain message is always transmitted in a meaningless form, which may leave an adversary a clue. In contrast, data hiding techniques successfully use meaningful cover media, such as an image, audio, video and text, to carry the secret message [2]– [28]. This unique feature of data hiding techniques makes it hard for attackers to visually distinguish images carrying secrets from images that do not carry a secret, and thus the security of the hidden data is increased. Therefore, data hiding techniques have been explored with various approaches, and its application has finally expanded to the medical domain.

The representative data hiding method designed for images is based on the least significance bits, also called LSB [3]. Later, different mathematical techniques were applied to increase the hiding capacity. In addition to payload, reversibility is the other important criterion. Data hiding techniques can be divided into two categories according to reversibility: one is un-reversible data hiding (URDH) [2]–[4], [8]–[10], [13], [14], [15]–[17], [19], [22] and the other is reversible data hiding (RDH) [5]–[7], [11], [12], [7], [17], [18], [20], [21], [23]–[28]. The former is conventional data hiding, and distortions on the original images caused by data embedding always remain even after the hidden secrets have been extracted. In comparison, the latter guarantees that the original cover images can be completely restored once the hidden secrets have been extracted.

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Zunino.

For data hiding schemes belonging to the RDH branch, the applications cover military or medical domains, because cover images are not allowed to have any slight distortion in either military or medical applications. Representative RDH schemes include Tain's DE method [5], Nei's histogram shifting method [6], and many other RDH schemes based on various compression techniques, such as JPEG [24], VQ [7], BTC [21].

From the hidden data aspect, there are three purposes that can be achieved with data hiding schemes: 1) establishment of a confidential channel via the cover image so that a secret message can be sent to the recipient; 2) carrying a message with a transmitted image so that the hidden message can be extracted by a recipient to verify the integrity of the transmitted image; and 3) carrying a message with an image so that the image owner can use the carried message to claim ownership of an image. The first purpose can be implemented with conventional RDH. The second purpose can be done with image authentication, which is also called fragile watermarking. As for the third purpose, it can be done by robust watermarking. Once watermarking schemes have reversibility features; if the recovery information is embedded into the cover image along with the authentication code or ownership code, no matter whether the approach is fragile watermarking or robust watermarking, it will automatically inherit the recovery function. This means the tampered area can be restored with the assistance of hidden recovery information.

The first electronic medical record system was developed by the Regenstrief Institute in 1972. The idea of sharing electronic medical records (EMRs) was not widely adopted until 2000 when computers became more affordable and the Internet was becoming more accessible. Hospitals then started to develop electronic medical information systems to provide patients with better, safer and more efficient healthcare services. Because of this, the frequency of EMRs transmitted over the Internet and shared among authorized medical parties is gradually increased. As EMRs contains EPIs and patient medical records, such as X-rays and lab reports, it is crucial for EMRs to being efficiently and securely transmitted to authorized metical parties and without leaking EPIs and damaging patient privacy. As data hiding techniques can achieve the three objectives mentioned previously, researcher have explored whether data hiding in medical images is a feasible approach to protect EMRs and EPIs by providing confidentiality, integrity, authentication and even ownership functionalities.

In 2011, Chang *et al.* designed an RDH scheme with repetitive pixels for medical images [11]. In their scheme, they adopted Run Length Encoding (RLE), an efficient lossless compression algorithm, to losslessly embed information, such as EPI, into the LSBs of repetitive pixels of a medical image without using the binary location map. In 2013, Kumar *et al.* first designed an optimal hierarchical block division scheme based on four ways of block division [12]. Subsequently, a modified histogram shifting algorithm based on their optimal hierarchical block division scheme was proposed for medical images to achieve a high hiding capacity. Experimental results demonstrated that the average PSNR was more than 51 dB for 256×256-sized medical images by carrying secret data around 6,059 bits. In 2020, Gull *et al.* applied a dual data hiding strategy to increase hiding capacity while maintaining high image quality for medical images [22]. In their scheme, the secret message is preprocessed by using the Huffman encoding method. After Huffman coding is completed, a codebook is generated and can be used to generate the encoding indices for the embedding message, such as EPIs. The value of each generated index is then divided into two parts that can finally embedded into two meaningful images with an improved center folding strategy. Gull *et al.*'s scheme provides an average embedding rate of around 2.69 bpp while maintaining average image quality at 50 dB, but the average BER is only around 0.5. In the same year, Geeth & Geeth applied an interpolation technique to enlarge a medical image. Then the original pixels located at a 2×2 block of the original image is then left at four corners of the enlarged 3×3 block of the expanded image [10]. Finally, the pixel values generated by an interpolation algorithm are used to carry the secret message and authentication code. With the proposed scheme, they successfully decreased BER to 0.048 when the stego image is subjected to a salt and peppers noise attack for 0.1 while retaining a similar BER as before for the remaining attacks.

Although many RDH schemes with a tamper detection function have been proposed for medical images, it is crucial to further decrease the distortion under different attacks while maintaining high image quality. In this paper, a novel efficient electronic patient information hiding scheme is proposed to achieve the above objectives thru an interpolation-based data hiding strategy with tamper detection function.

The rest of this paper is organized as follows. Section 2 briefly reviews the enhanced neighbor mean interpolation (ENMI) which will be used in the proposed scheme. Section 3 explains our proposed interpolation-based reversible data hiding scheme. Section 4 demonstrates the experimental results, comparisons among our proposed scheme and other six existing schemes under different attacks. Finally, brief conclusions are given in Section 5.

II. RELATED WORK

Based on the experimental results demonstrated in [11], we found the estimated pixel values derived from the enhanced neighbor mean interpolation (ENMI) can provide a relatively high image quality in stego images. This section introduces the ENMI algorithm in Subsection 2.1. Next, some presentative data hiding schemes based on image interpolation are described in Subsection 2.2.

A. ENHANCED NEIGHBOR MEAN INTERPOLATION ALGORITHM

The ENMI algorithm is an improved version of the neighbor mean interpolation (NMI) algorithm, and was proposed in Chang *et al.*'s scheme [13]. Let assume there is a 2 × 2-sized block called B_I in an original image I. In the 2 × 2-sized block B_I, there are four pixels located at coordinates (0, 0), (0, 1), (1, 0), and (1, 1), and are denoted as $I(0, 0)$, $I(0, 1)$, $I(1, 0)$, and $I(1, 1)$, respectively. These four pixels are mapped to $I'(0,0)$, $I'(0,2)$, $I'(2,0)$ and I'(2,2) in the expanded block $B_{I'}$. The remaining five estimated pixels derived by ENMI are located at coordinates (0, 1), (1, 0), (1, 1), (1,2) and (2, 1) in the expanded block B'_I and are denoted as $I'(0, 1)$, $I'(1, 0)$, $I'(1, 1)$, $I'(1,2)$ and $I'(2, 1)$, respectively, as shown in Fig. 1(a). Basically, the five estimated pixels are derived by ENMI based on Equation (1) and then a 2×2-sized block B_I is expanded into a 3 × 3-sized block $B_{I'}$ as shown in Figure 1(a):

$$\begin{aligned} I'(0,1) &= (I(0,0) + I(0,1))/2, \\ I'(1,0) &= (I(0,0) + I(1,0))/2, \\ I'(1,1) &= (I(0,0) + I(0,1) + I(1,0) + I(1,1))/4, \\ I'(1,2) &= (I(0,1) + I(1,1))/2, \\ I'(2,1) &= (I(1,0) + I(1,1))/2. \end{aligned} \tag{1}$$

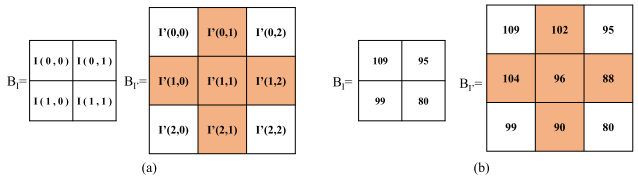

FIGURE 1. (a) ENMI mapping rules between the original 2 × 2-sized block B_I and its expanded 3 × 3-sized block $B_{I'}$, (b) example of ENMI.

B. RESENTATIVE DATA HIDING SCHEMES BASED ON IMAGE INTERPOLATION

In additional to the scheme [11] introduced in Section 1, there are many data hiding methods based on image interpolation algorithms [10], [15], [16], [18], [30], [32]. In 2009, Jung and Yoo first applied the image interpolation technique to hide secret data to a cover image [10]. To achieve their objective, Jung and Yoo designed a new image interpolation (NMI) algorithm, in which neighboring pixel values are used to calculate the mean, and then the calculated mean value is inserted into a pixel that has not been allocated yet. In Jung and Yoo's scheme, a cover image sized as H×W pixels is first downsized into a (H/2 + 1)×(W/2 + 1) sized image. Then, the downsized image is divided into non-overlapping 2 × 2 blocks. Note that the first pixel in each 2 × 2 block remains the same, while the remaining three pixels are derived by using NMI and the difference between each derived value and the first pixel is calculated to carry secret data. The payload for each derived value is related to its difference to the first pixel.

The same research team also combined another image interpolation algorithm called interpolation by neighboring pixels (INP) and LSB substitution to design a semi-reversible data hiding scheme in 2015 [15]. In 2017, Yang *et al.* adopted the same image interpolation algorithm as [15] did to generate the cover image, but applied least significant (LSB) substitution and optimal pixel adjustment process (OPAP) instead of simple addition to further increase the visual quality of the stego image [16]. In 2019, Mohammad *et al.* designed a new image interpolation algorithm to calculate the interpolated pixels based on remaining four unchanged corner pixels with different weights. For example, both I(0,0) and $I(0,1)$ contribute a one third impact and both I(0,1) and $I(1,1)$ only contribute a one and sixth impact to calculate the expanded pixel $I'(0, 1)$ as $I'(0, 1) = [(I(0, 0) + I(0, 1))/3 + (I(0, 1) + I(1, 1))/6]$ [18].

III. PROPOSED SCHEME

To create enough embedding space for medical images to carry EPIs and to ensure that the original medical images can be completely restored, our proposal is inspired by previous work introduced in Section 2 and also refers to various interpolation algorithms demonstrated in [13]. Finally, the ENMI algorithm proposed by Chang *et al.* [13] was selected to expand the original medical image I sized as $H \times W$ pixels into $((H/2) + 1) \times ((W/2) + 1) \times (3 \times 3)$-sized expanded image I' because ENMI successfully enhances 2dB than that of Nearest neighbor interpolation (NMI) based on the experimental results reported in [13].

To effectively detect the tamper regions, and embed EPIs into a medical image, the hidden data is divided into two categories: one is the authentication code (AC) and the other is the secret message (M), such as EPIs. It is noted the $AC = \{ac_1, ac_2, \ldots, ac_l\}$ and $M = \{m_1, m_2, \ldots, m_l\}$, where $l = ((H/2) + 1) \times ((W/2) + 1)$. The overall flowchart of our proposed scheme is depicted in Figure 2. Figure 2 shows how the secret message M will be embedded to a medical image with our proposed scheme and how the extracted authentication code will be used to judge whether or not the current block has been tampered. If it is un-tampered, the extracted message M will be recorded. After the verifying operation is completed, the restoration function is conducted. Finally, when all regions are judged as un-tampered, the original medical image I can be reconstructed and the secret message M can be obtained.

A. DATA EMBEDDING PHASE

Assume that a $H \times W$ medical image I is divided into $((H/2) + 1) \times ((W/2) + 1)$ non-overlapping blocks and each block is sized as 2 × 2 pixels. Once ENMI interpolation is completed, each block size is expanded as 3 × 3 pixels, and each expanded block is denoted as I'_B_l. As we mentioned in previous paragraph, the hidden data is divided into two categories: one is the authentication code (AC) and the other is the secret message (M), such as EPIs. It is noted the

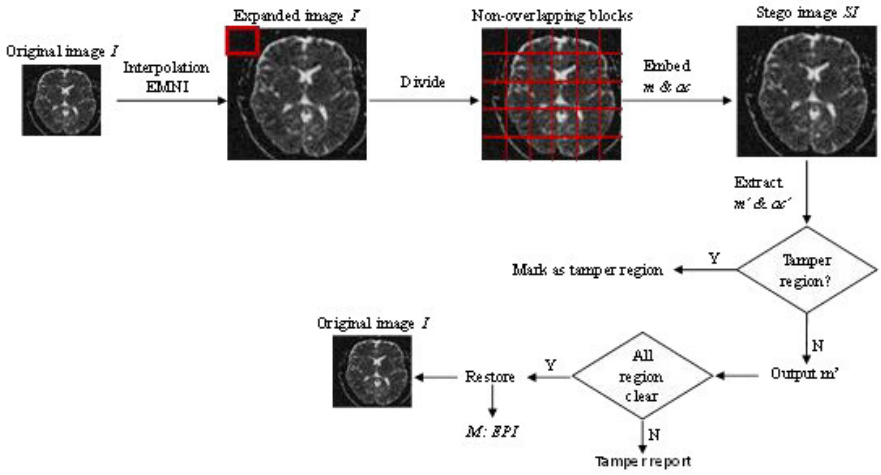

FIGURE 2. Flowchart of our proposed scheme.

$AC = \{ac_1, ac_2, \ldots, ac_l\}$ is pre-generated by using the secret key SK and a pre-determined random number generator, and $M = \{m_1, m_2, \ldots, m_l\}$, where $l = ((H/2)+1) \times ((W/2)+1)$.

The ENMI interpolation operation has been introduced in Subsection 2.1, thus, we skip the related descriptions here and focus on how to embed the authentication code and secret message into each non-overlapping block. The authentication code can be a binary stream and generated by a secret key SK with a pre-determined random number generator in advance. The detail embedding algorithm is as follow:

Input: Expanded image I, authentication code $AC = \{ac_1, ac_2, \ldots, ac_l\}$ and secret message $M = \{m_1, m_2, \ldots, m_l\}$, where $l = \left(H/2 + 1\right) \times \left(W/2 + 1\right)$

Output: Stego-image

Step 1: Determine threshold k, read a block I'_B_l, where $l = \left(H/2 + 1\right) \times \left(W/2 + 1\right)$, and calculate difference for pixels in the expanded block.

$$D(i,j) = I'(i,j) \, mod 2^k, \tag{2}$$

where $I'(i,j)$ denotes the pixel value located at (i,j) for the currently processing block and $D(i,j)$ indicates the difference derived from the current pixel $I'(i,j)$.

Step 2: For $I'(i, j-1), I'(i-1, j), I'(i+1, j)$ and $I'(i-1, j-1)$ located at the currently processing block I'_B_l as shown in Figure 3, modify the current pixel value based on the computed $D(i,j)$ according to the three modification rules, which are also demonstrated in Table 1, and hidden message m_l, where $m_l = \{0, 1\}$ and $l = ((H/2) + 1) \times ((W/2) + 1)$. If $D(i,j) \neq m_i$, find a new pixel value $newI'(i,j)$ which meet the following rule: $newI'(i,j) mod 2^k = m_i mod 2^k = m_i$.

The reference table shown in Table 1 presents a simple but important idea, that is, if the derived difference $D(i,j)$ is the same as that of hidden message m_l or authentication code ac_l,

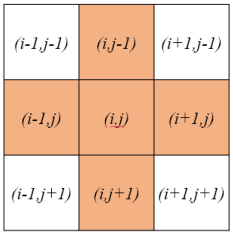

FIGURE 3. Relative positions for the current block I'_B_l.

TABLE 1. Reference table.

$D(i,j)$	0	1	2	3	4	5	6	7
m & ac	0	0	0	0	1	1	1	1
Status of $I'(i,j)$	C	UC	UC	C	C	UC	UC	C
secret-bit range	0				1			

it means the current pixel is located at the same secret-bit range as a hidden m_l or ac_l. Here, $D(i,j)$ is computed by Equation (2), m_l indicates the secret bit embedded into a given pixel and $m_l = \{0, 1\}$, and ac_l indicates the authentication bit embedded into a given pixel and $ac_l = \{0, 1\}$. The status of $I'(i,j)$ indicates whether the current $I'(i,j)$ should be changed or not according to three modification rules, which are as follows: where "C" means the current $I'(i,j)$ must be changed. In contrast, "UC" means the current $I'(i,j)$ can remain as it is:

Rule 1: The new pixel $newI'(i,j)$ must make sure its derived D is the same as either hidden message m_l or hidden authentication bit ac_l.

Rule 2: The difference between the new pixel *newI'(i,j)* and the original pixel *I'(i,j)* must be minimized.

Rule 3: The new pixel *newI'(i,j)* should not be located at the boundary area of its corresponding secret-bit rangee.

Here, we also define if $ac_l = 0$ or $m_l = 0$ then the secret-bit rangee is defined as "0," if $ac_l = 1$ or $m_l = 1$ then the secret-bit range is defined as "1." If the derived difference $D(i, j)$ located at secret-bit range "0" and $D(i, j)$ equals to "0" or "3," the current pixel *I'(i,j)* is located at the boundary area of secret-bit range "0". Following the same concept, if the derived difference $D(i, j)$ is located at secret-bit range "1" and $D(i, j)$ equals to "4" or "7", the current pixel *I'(i,j)* is also located at the boundary area of secret-bit range "1". If a pixel meets the first two rules, but it is located at the boundary area, then, a new pixel value should be found for the current pixel in order to comply with the three modification rules. This is because the hidden data is easily modified when the pixels are located in the boundary area once an attack occurs. Take pixel = 112 for example, if it is changed to 111, the derived D will be changed to "7" instead of "0," as the corresponding secret-bit range is changed and the hidden data is also changed. Only when the third rule is met, the robustness of the hidden data can be guaranteed.

Step 3: For pixel value *I'(i,j)* of expanded block I'_B_l as shown in Fig. 3, modify the current pixel value based on its derived D according to above three modification rules (Table 1), and hidden authentication code ac_l, where $ac_l = \{0, 1\}$ and $l = ((H/2) + 1) \times ((W/2) + 1)$.

Step 4: Output the proceeded stego block and then read the next block.

Step 5: Repeat Steps 2 to 4 until all blocks have been proceeded. Finally, a stego-image *SI* is generated and sent to the receiver.

To give a clear picture for our data embedding strategy, an example is demonstrated in this paragraph. Take $k = 3, I'(i, j) = 106$, hidden message $m_l = 0$ for example. Based on Equation (2), the corresponding $D(i, j) = 2 = 106\ mod 2^3 = 106\ mod 8$. The derived $D = 2$ located at secret-bit range "0" is the same as the hidden message $m_l = 0$. Therefore, $I'(i, j) = 106$ remains the same. As another example, for *I'(i,j)*=114, the hidden message $m_l = 1$. Based on Equation (1), the corresponding $D(i, j) = 2 = 114\ mod 2^3 = 114\ mod 8$. The derived $D = 2$ located at secret-bit range "0" is different from secret-bit range "1" of hidden message $m_l = 1$. Therefore, pixel *I'(i,j)* must be further changed based on the modification rules listed above. The possible candidates for the current $I'(i, j) = 114$ are listed in Table 2.

From Table 2, we can see the new pixel value for the current *I'(i,j)* is selected as 117 because its secret-bit range "1" is the same as hidden data m_l, the caused distortion is minimized and the new pixel value is not located at the boundary of secret-bit range "1". Once the data embedding is completed, the stego-image contains secret message and authentication code is then sent to the receiver.

TABLE 2. Possible candidates for the current $I'(i, j) = 114$.

D(i,j)	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
m & ac	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*1*
newI'(i,j)	*112*	*113*	*114*	*115*	*116*	*117*	*118*	*119*
secret-bit range	*0*				*1*			

B. DATA EXTRACTION AND VERIFICATION PHASE

Upon receipt, the receivers of the stego-image can extract the hidden authentication code to determine which blocks have been tampered. If a block has not been tampered, its extracted secret message will be outputted. If the integrity of the stego-image is verified, the extracted hidden data is contacted as the original message M. Details of data extraction and verification algorithms are shown below:

Input: Stego-image *SI*, secret key *SK*, random number generator and threshold k

Output: Secret message $M = \{m_1, m_2, \ldots, m_l\}$ and restored original image I

Step 1: Generate $AC = \{ac_1, ac_2, \ldots, ac_l\}$, where $l = ((H/2) + 1) \times ((W/2) + 1)$, by using secret key *SK* and the pre-determined random number generator.

Step 2: Read a block and an ac_l from AC, and compute the derived D' for pixel *SI(i,j)* of block SI_B_l, where $l = ((H/2) + 1) \times ((W/2) + 1)$ using Equation (1).

Step 3: Compare the derived D' and the corresponding ac_l, where $ac_l = \{0, 1\}$ and $l = ((H/2) + 1) \times ((W/2) + 1)$.

If they are equal, it means the current block has not been tampered. Otherwise, mark the block as "tampered," and go to Step 5.

Step 4. For pixels $SI'(i, j - 1), SI'(i - 1, j), SI'(i + 1, j)$ and $SI'(i - 1, j - 1)$, compute the derived D' as the hidden message m_i, where $m_l = \{0, 1\}$ and $l = ((H/2) + 1) \times ((W/2) + 1)$. Output the extracted hidden message m_i.

Step 5. Output error message as "E" to indicate that the current block has been tampered.

Step 6. Repeat Steps 2 to 5 until all blocks of stego-image *SI* have been proceeded.

Step 7. Contact all outputted messages. If the stego-image has not been tampered, all the extracted hidden message m_i will be contacted as secret message M. Otherwise, the combination of the extracted secret data and error message will be presented to the receiver.

With our data extraction and verification phase, the integrity of the stego-image is verified, and the hidden message can be successfully extracted to generate secret message M. Follow the example demonstrated in data embedding phase, when receiver detects the pixel value as 117 in the stego image, s/he can conclude $D(i, j) = 5 = 117 mod 2^3$

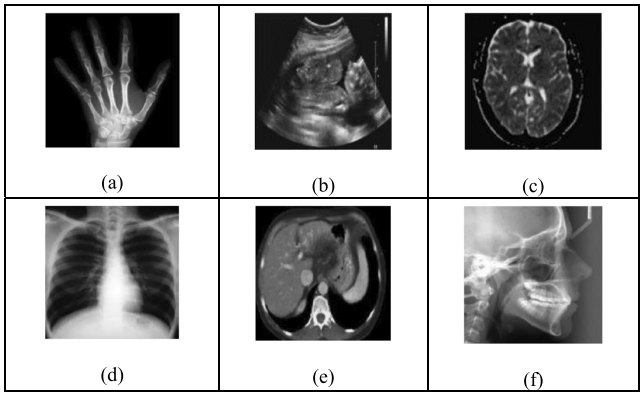

FIGURE 4. Six test images: (a) Palm X-ray image. (b) Ultra scan foetus image. (c) Brain MRI scan image. (d) Thorax X-ray image. (e) Brain CT scan image. (f) Cephalogram dental image.

and judges the hidden data is 1 based on judgement rules demonstrated in Table 2.

C. RESTORATION PHASE

Note that the restoration phase is triggered when stego-image *SI* is determined as an un-tampered image. In other words, the integrity of stego-image *SI* has been confirmed. A detailed description of our restoration algorithm is presented as below:

Input: Stego-image *SI*

Output: Original image *I*

Step 1: Read a block.

Step 2: Select pixel values $SI(i-1, j-1)$, $SI\ (i+1, j-1)$, $SI\ (i-1, j+1)$ and $SI(i+1, j+1)$ of block SI_B_l, where $l = ((H/2)+1) \times ((W/2)+1)$ to serve as the values for $I(i-1, j-1)$, $I(i1, j-1)$, $I(i-1, j)$ and $I(i, j)$ for block B_l of the original image I, where $l = ((H/2)+1) \times ((W/2)+1)$

Step 3: Repeat Steps 1 and 2 until all blocks of stego-image *SI* are completed.

If the stego-image has not been tampered during data transmission, the original image I can be completely restored after the hidden data is extracted with our proposed restoration algorithm. Even the stego-image has been tampered, the original image I still can be restored as long as the four corner pixels in stego-blocks have not been tampered.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed scheme on the image quality of a stego image, hiding capacity, and to observe the robustness under various attacks, we implemented our scheme with Matlab. Experiments were conducted on a personal computer of an Intel i7-4790 CPU, 8GB memory and Windows 10 Home basic 64-bits operating system. Six 512×512-sized medical gray-scale images were served as test images: (a) Palm X-ray image, (b) Ultra scan foetus image, (c) Brain MRI scan image, (d) Thorax X-ray image, (e) Brain CT scan image and (f) Cephalogram dental image as shown in Figure 4.

A. PERFORMANCE OF OUR PROPOSED SCHEME

The criteria used to evaluate the performance of the proposed scheme on image quality are the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM index). *PSNR* is defined as follows to evaluate the quality of the stego images.

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right), \tag{3}$$

where mean square error (MSE) is as follows:

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(I(i,j) - SI(i,j))^2, \tag{4}$$

where W stands for width and H stands for height of the images I and SI, respectively. $I(i,j)$ and $SI(i,j)$ are the pixel values of the original image and the stego image, respectively. As shown in Equation (3), the smaller the *MSE*, the larger the *PSNR*, and vice versa. In general, when a stego-image's *PSNR* is larger than 30 dB, the human vision system has difficulty in distinguishing the stego-image from the original image. The *SSIM* index, which is an index to evaluate the similarity between the original image and the stego image, is shown in Equation (5)

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{5}$$

where, μ_x and μ_y denotes the average pixel values of images x and y, respectively; and σ_x, σy, σxy denoted the standard deviation and cross-correlation for the images x and y, r espectively. As for c_1 and c_2, they are constants set internally by the *SSIM* function of Matlab.

In addition to the image quality of stego image, there are three other different kinds of criteria used. One is hiding capacity, and the general unit used to present the hiding capacity of a data hiding scheme is bit or bit per pixel (BPP). Another is the robustness of the hiding strategy under different attacks, such as bit error rate (BER) and normalized cross correlation (NCC). The other is security of the hidden data, such as information entropy $H\ (I)$, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). In general, there is always a trade-off between image quality and hiding capacity. Moreover, once the size of the hidden data becomes larger, the challenge of protecting the hidden data is also higher.

The definitions of BER, NCC, information entropy, NPCR and UACI are listed as follows:

$$BER = \frac{Extract\ wrong\ data}{Total\ embed\ data}, \tag{6}$$

$$NCC = \frac{(Total\ embed\ data - Extract\ wrong\ data)}{Total\ embed\ data}, \tag{7}$$

$$H\ (\mathbb{I}) = -\sum_{\iota=0}^{255}\Theta(\rho_\iota)\log 2(\Theta(\rho_\iota)), \tag{8}$$

where $\mathbb{I}$ denotes an input grayscale image with 256 grey-levels, $0 \leq \rho_\iota \leq 255$, and $\Theta(\rho_\iota)$ denotes the

TABLE 3. Image quality of six test images with $k = 3$.

Test images	*PSNR* (dB)	*SSIM*
Palm X-ray	48.09	0.981
Ultra scan foetus	46.54	0.985
Brain MRI scan	46.89	0.987
Thorax X-ray	46.65	0.983
Brain CT scan	46.81	0.984
Cephalogram dental	46.66	0.984
Average	46.94	0.984

TABLE 4. Image quality of six test images with $k = 4$.

Test images	*PSNR* (dB)	*SSIM*
Palm X-ray	41.35	0.897
Ultra scan foetus	40.33	0.946
Brain MRI scan	38.82	0.922
Thorax X-ray	40.11	0.941
Brain CT scan	40.18	0.932
Cephalogram dental	40.05	0.982
Average	40.14	0.937

probability of ρ_t.

$$UACI = \frac{1}{H \times W} \times \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{|I(i,j) - SI(i,j)|}{255} \times 100\%$$

$$NPCR = \frac{1}{H \times W} \times \sum_{i=1}^{H} \sum_{j=1}^{W} Diff(i,j) \times 100\%, \quad (9)$$

$$Diff(i,j) = \begin{cases} 0, & if\ I(i,j) = SI(i,j) \\ 1, & otherwise \end{cases} \quad (10)$$

where W stands for width and H stands for height of the images I and SI, respectively. $I(i,j)$ and $SI(i,j)$ are the pixel values of the original image and the stego image, respectively.

From Table 3, we can see when threshold k is set as 3, the average image quality of six test images is up to 46.94 dB and the *SSIM* index between the original image and stego image is around 0.984. Comparing the results listed in Tables 3 and 4, we can also find the original image and stego image are relatively similar to each other when $k = 3$ is compared with $k = 4$.

Since *PSNR*s and *SSIM*s with $k = 3$ are higher than those with $k = 4$, Fig. 5 demonstrates the results of our proposed scheme with $k = 3$. From a visual quality perspective, comparing the original images shown in Fig. 5-1(a) to 5-6(a) and stego images shown in Fig. 5-1(b) to 5-6(6), we can find the difference between the original image and stego image is barely recognizable. This confirms that our proposed scheme not only maintains a similar structure between the stego image and original image, but that the visual quality offered by our proposed scheme is quite high.

To further demonstrate the performance of our proposed scheme on hiding capacity, image quality and security of the hidden data, results are presented in Tables 5 and 6 with two different test image sets. The former is a test set with

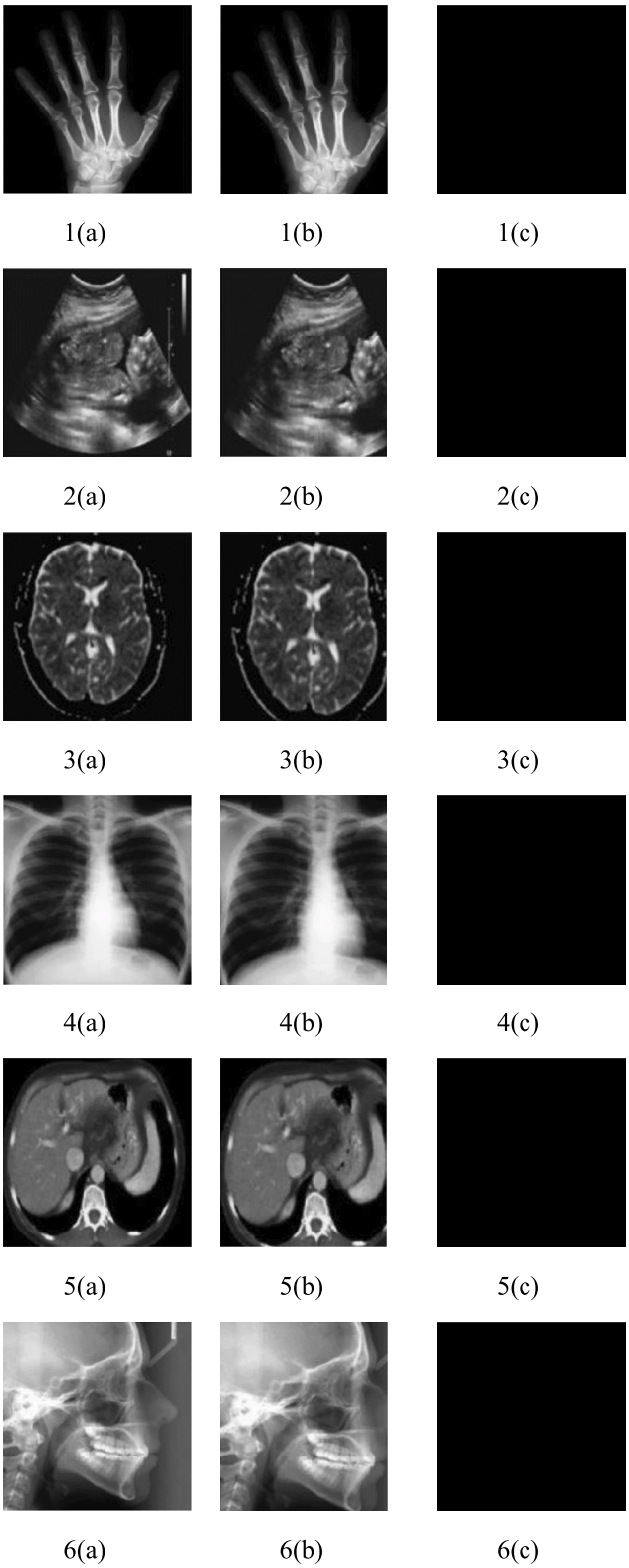

FIGURE 5. 1(a)-6(a) Original images. 1(b)-6(b) Stego images. 1(c)-6(c) Difference between original images and recovered images. ($k = 3$).

TABLE 5. Performance of our proposed scheme on hiding capacity, image quality security of the hidden data with $k = 3$.

Images	Capacity (bits)	PSNR (dB)	bpp	SSIM	Entropy	NPCR	UCAI
Palm X-Ray	327680	48.09	0.55	0.981	5.1757	45.29%	0.29%
Ultra Scan Fetus	327680	46.54	0.55	0.985	5.7307	46.22%	0.34%
Brain MRI SCAN	327680	46.89	0.55	0.987	5.5839	46.63%	0.33%
Thorax X-Ray	327680	46.65	0.55	0.983	6.5285	44.83%	0.33%
Brain CT scan	327680	46.81	0.55	0.984	6.1146	45.22%	0.33%
Cephalogram dental	327680	46.66	0.55	0.984	6.5389	45.04%	0.33%

TABLE 6. Performance of our proposed scheme on hiding capacity, image quality security of the hidden data with $k = 3$.

Images	Capacity (bits)	PSNR (dB)	bpp	SSIM	Entropy	NPCR	UCAI
Airplane	327680	45.52	0.55	0.983	6.3644	42.47%	0.32%
Baboon	327680	45.43	0.55	0.993	6.9619	42.68%	0.32%
Boat	327680	45.47	0.55	0.983	6.6375	42.66%	0.32%
Lena	327680	45.41	0.55	0.983	7.0809	42.67%	0.32%
Peppers	327680	45.46	0.55	0.984	7.2307	42.56%	0.32%

six medical images as shown in Fig. 2 and the latter is the other test set contains five general test images, such as "Airplane," "Baboon," "Boat," "Lena," and "Peppers." Since each block is sized as 2×2 pixels in the original image, and it has been expanded to a 3×3-sized block in the expanded image. Four pixels in each expanded block are used to carry 4-bits secret data and the pixel located at the central part in the 3×3-sized block is used to conceal a 1-bit authentication code. Therefore, a stego-image hides 327,680 bits in total. The hiding capacity of our proposed scheme is fixed (0.55 bpp) and is not affected by different thresholds ks. For the medical image set, the average *PSNR* is around 46.94 dB. As for the general image set, the average *PSNR* is around 45.458 dB.

According to the information entropy defined in Equation (7), if the information entropy is closer to 8, it means the current image has higher randomness. We can observe from Tables 5 and 6 that the information entropy of the stego images ranged from 5 to 7. As for *NPCR* and *UACI*, the former is used to determine the rate at which the pixels changed for a stego image caused by payload embedding, and it has a maximum theoretical value of 1. The latter is used to indicate the average intensity of the change of pixel values, and it has a theoretical value of 0.3346. In general, the smaller the *NPCR* and *UACI* are, the more minor the changes caused on the pixel during data embedding. The results of *NPCR*s and *UACI*s presented in Tables 5 and 6 confirm that our proposed data hiding strategy provides acceptable protection for the hidden data. Moreover, the more secure the hidden data is with our proposed scheme, the more diverse the pixel distribution of the cover image is.

TABLE 7. The detection performance and robustness of the hidden EPIs when under salt and peppers noise attack with 0.1 noise density and $k = 3$.

Name of images	Tampered blocks count	EPIs data	
		BER %	*NCC*
Palm X-ray	15891	19.38%	0.8062
Ultra scan Fetus	15791	19.42%	0.8058
Brain MRI scan	15736	19.38%	0.8062
Thorax X-ray	15986	19.47%	0.8053
Brain CT scan	15870	19.47%	0.8053
Cephalogram dental	15867	19.56%	0.8044
Average	15856	19.44%	0.8055

B. EXPERIMENTAL ANALYSIS IN MEDICAL IMAGES FOR VARIOUS ATTACKS

The proposed scheme can detect tampered blocks. If the integrity of the received stego image is verified, and the stego image is determined as un-tampered, the hidden secrets can be extracted. Moreover, the original image can be completely restored. Since a stego image may be attacked during data transmission, how many tampered blocks can be identified and the robustness of the hidden data under different attacks are two crucial criteria for a data hiding scheme. In this section, several attacks such as salt and peppers noise, filtering, JPEG compression, sharpening, etc., will be used to evaluate the performance of our proposed scheme on standing attacks. The results demonstrated in Subsection 4.1 show the image quality of the stego image with $k = 3$ is slightly higher than those with $k = 4$. Therefore, in the following experiments we only present test results with $k = 3$.

1) SALT AND PEPPERS NOISE

Table 7 presents the *NCC*s and *BER*s when a stego image is exposed to salt and peppers noise with a noise density 0.1, and Fig. 6 shows stego images attacked by salt and peppers noise and detection results with our proposed scheme. Table 7 shows an average of 15,856 blocks as tampered, *BER* as 19.44% and *NCC* as 0.8055 for the extracted hidden data when $k = 3$. Accordingly, more than 80% of the hidden EPIs can be extracted even under the salt and peppers noise attack.

2) ADDITIVE WHITE GAUSSIAN NOISE (AWGN)

Here, an AWGN attack is simulated as mean = 0 and variance = 0.02. When stego images are under an AWGN attack. The detection performance and robustness of the hidden data provided by our proposed scheme with threshold $k = 3$ are demonstrated in Table 8 and Fig. 7. An average of 34,596 blocks are identified as tampered, the *BER* of the hidden EPIs is 40.64%, and *NCC* is 0.5935, which means around 60% of hidden EPIs can be correctly extracted even when 52.8% blocks have been tampered.

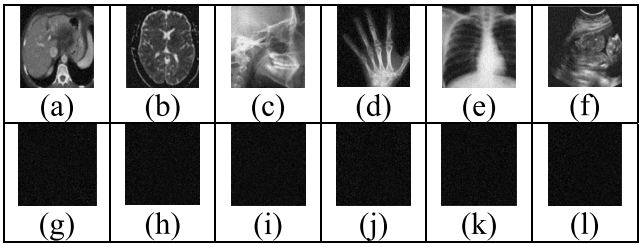

FIGURE 6. (a)-(f) attacked stego images. (g)-(l) tampered areas for a salt and peppers attack with $k = 3$.

TABLE 8. The detection performance and robustness of the hidden EPIs when under an AWG attack with $k = 3$.

Name of images	Tampered blocks count	EPIs data	
		BER %	*NCC*
Palm X-ray	30957	35.42%	0.6457
Ultra scan Fetus	34498	40.65%	0.5935
Brain MRI scan	34701	38.71%	0.6129
Thorax X-ray	36300	43.48%	0.5652
Brain CT scan	34897	40.82%	0.5918
Cephalogram dental	36227	44.79%	0.5521
Average	34596	40.64%	0.5935

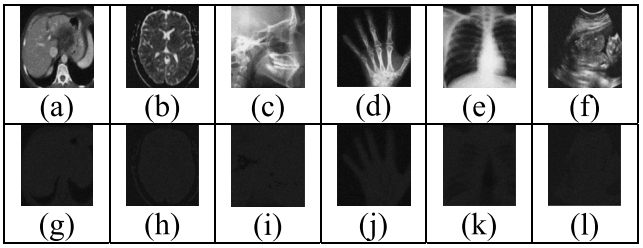

FIGURE 7. (a)-(f) attacked stego images. (g)-(l) tampered areas for an AWG attack with $k = 3$.

3) JPEG COMPRESSION

JPEG compression is quite frequently adopted to reduce bandwidth requirements during data transmission. From Table 9 and Fig. 8, we can see that an average of 41,765 blocks are identified as tampered, the *BER* of the hidden EPIs is 33.77%, and *NCC* is 0.6622, which means even more than around 64% of the blocks have been identified as tampered, there are still 33.77% of hidden EPIs can be correctly extracted.

4) FILTERING ATTACKS

In general, filtering is used to remove noise that exists in the images and increase the image quality of the image.

TABLE 9. The detection performance and robustness of the hidden EPIs when under a JPEG attack with QF=90 and $k = 3$.

Name of images	Tampered blocks count	EPIs data	
		BER %	*NCC*
Palm X-ray	48567	29.15%	0.7085
Ultra scan Fetus	38546	34.20%	0.6580
Brain MRI scan	42368	34.56%	0.6544
Thorax X-ray	39927	34.84%	0.6516
Brain CT scan	41225	34.64%	0.6536
Cephalogram dental	39960	35.26%	0.6474
Average	41765	33.77%	0.6622

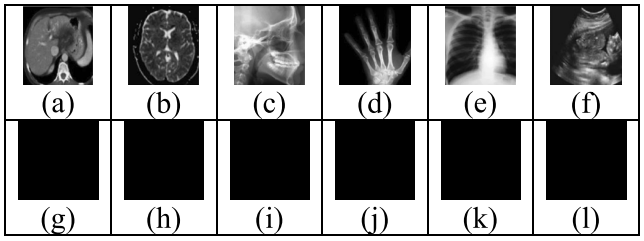

FIGURE 8. (a)-(f) attacked stego images. (g)-(l) tampered areas for a JPEG attack with GQ=90 and $k = 3$.

TABLE 10. The detection performance and robustness of the hidden EPIs when under filtering attack with kernel sized 3 × 3 and $k = 3$.

Name of images	Tampered blocks count	EPR data	
		BER %	*NCC*
Palm X-ray	28267	26.28%	0.7372
Ultra scan Fetus	32630	30.09%	0.6991
Brain MRI scan	38452	30.31%	0.6969
Thorax X-ray	36418	33.48%	0.6652
Brain CT scan	34605	31.23%	0.6877
Cephalogram dental	36069	32.49%	0.6751
Average	34406	30.64%	0.6935

However, filtering also can be treated as an attack just like a JPEG attack. A kernel window sized as 3×3 is used in our simulation. Fig. 10 and Table 9 demonstrate the detection performance and robustness of the hidden data provided by our proposed scheme with a threshold $k = 3$ when the stego image are under a filtering attack. An average of 34,406 blocks are identified as tampered, and the *BER* of the hidden EPIs is 30.64%, and *NCC* is 0.6935.

TABLE 11. The detection performance and robustness of the hidden EPIs when under a median filtering attack with a kernel windows sized 3 × 3 and $k = 3$.

Name of images	Tampered blocks count	EPR data	
		BER %	*NCC*
Palm X-ray	28267	26.28%	0.7372
Ultra scan Fetus	32630	30.09%	0.6991
Brain MRI scan	38452	30.31%	0.6969
Thorax X-ray	36418	33.48%	0.6652
Brain CT scan	34605	31.23%	0.6877
Cephalogram dental	36069	32.49%	0.6751
Average	34406	30.64%	0.6935

TABLE 12. The detection performance and robustness of the hidden EPIs when under a Weiner filtering attack with $k = 3$.

Name of images	Tampered blocks count	EPR data	
		BER %	*NCC*
Palm X-ray	29107	31.55%	0.6845
Ultra scan Fetus	34039	36.95%	0.6305
Brain MRI scan	40387	38.12%	0.6188
Thorax X-ray	37272	38.10%	0.6190
Brain CT scan	35367	37.98%	0.6202
Cephalogram dental	36928	38.32%	0.6168
Average	35516	36.83%	0.6316

TABLE 13. The detection performance and robustness of the hidden EPIs when under a sharpening attack with $k = 3$.

Name of image	Tampered blocks count	EPR data	
		BER %	*NCC*
Palm X-ray	30506	20.41%	0.7959
Ultra scan Fetus	36012	20.43%	0.7957
Brain MRI scan	42812	21.52%	0.7848
Thorax X-ray	39402	19.35%	0.8065
Brain CT scan	37433	20.52%	0.7948
Cephalogram dental	39286	20.18%	0.7982
Average	37575	20.40%	0.7959

5) MEDIAN FILTERING

Among various filterings, median filtering is a kind of non-linear filtering. The kernel window of 3×3 is the same as the filtering attack mentioned in the previous subsection, where

TABLE 14. Comparisons among seven representative data hiding schemes and our proposed scheme.

Images	Methods	Capacity	*PSNR*	*bpp*	*SSIM*
Palm X-ray	Proposed method(k=3)	327680	48.09	0.55	0.981
	Proposed method(k=4)	327680	41.35	0.55	0.897
	Geeth & Geeth [23]	393216	42.3256	1.5	0.981
	Shabir et al. [25]	91600	49.6855	0.3494	0.992
	Luo et al. [26]	36000	48.9464	0.1373	0.999
	Tian [5]	12200	41.1986	0.0465	0.991
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.999
Ultra scan fetus	Proposed method(k=3)	327680	46.54	0.55	0.985
	Proposed method(k=4)	327680	40.33	0.55	0.946
	Geeth & Geeth [23]	393216	42.312	1.5	0.980
	Shabir et al. [25]	91600	49.6873	0.3494	0.993
	Luo et al. [26]	36000	48.9464	0.1373	0.999
	Tian [5]	12200	41.1986	0.0465	0.990
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.999
Brain MRI scan	Proposed method(k=3)	327680	46.89	0.55	0.987
	Proposed method(k=4)	327680	38.82	0.55	0.922
	Geeth & Geeth [23]	393216	42.398	1.5	0.982
	Shabir et al. [25]	91600	49.6231	0.3494	0.993
	Luo et al. [26]	36000	48.9464	0.1373	0.999
	Tian [5]	12200	41.1986	0.0465	0.991
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.999
Thorax X-ray	Proposed method(k=3)	327680	46.65	0.55	0.983
	Proposed method(k=4)	327680	40.11	0.55	0.941
	Geeth & Geeth [23]	393216	42.3145	1.5	0.981
	Shabir et al. [25]	91600	49.6762	0.3494	0.992
	Luo et al. [26]	36000	48.9464	0.1373	0.999
	Tian [5]	12200	41.1986	0.0465	0.991
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.999
Brain CT scan	Proposed method(k=3)	327680	46.81	0.55	0.984
	Proposed method(k=4)	327680	40.18	0.55	0.932
	Geeth & Geeth [23]	393216	42.3732	1.5	0.981
	Shabir et al. [25]	91600	49.6822	0.3494	0.993
	Luo et al. [26]	36000	48.9464	0.1373	0.999

TABLE 14. *(Continued.)* **Comparisons among seven representative data hiding schemes and our proposed scheme.**

Cephalogram dental	Tian [5]	12200	41.4986	0.0465	0.990
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.998
	Proposed method(k=3)	327680	46.66	0.55	0.984
	Proposed method(k=4)	327680	40.05	0.55	0.942
	Geeth & Geeth [23]	393216	42.3429	1.5	0.982
	Shabir et al. [25]	91600	49.6882	0.3494	0.995
	Luo et al. [26]	36000	48.9464	0.1373	0.999
	Tian [5]	12200	41.1986	0.0465	0.990
	Xuan [27]	14600	48.1438	0.0556	0.998
	Lee et al. [28]	10880	48.4209	0.0412	0.998

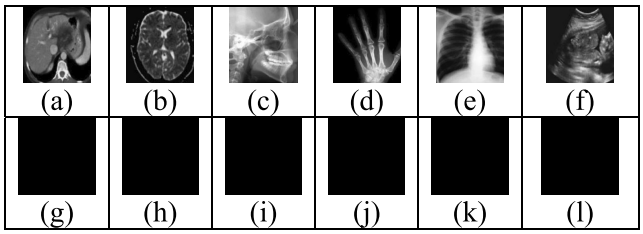

FIGURE 9. (a)-(f) marked attacked images. (g)-(l) tampered areas for a filtering attack with k=3.

the detection performance and robustness of the hidden data provided by our proposed scheme with a threshold $k = 3$ are demonstrated in Table 11 and Fig. 10. An average of 34,596 blocks are identified as tampered, the *BER* of the hidden EPIs is 30.64%, and NCC is 0.6935, which means around 70% of hidden EPIs can be correctly extracted when 52.8% of blocks have been tampered.

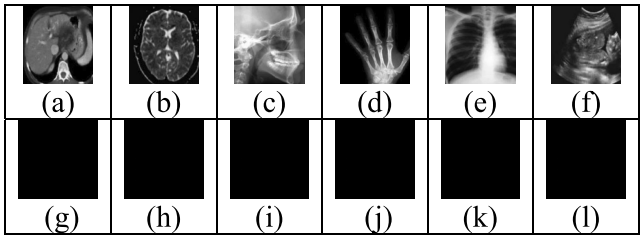

FIGURE 10. (a)-(f) marked attacked images. (g)-(l) tampered areas for a median filtering with $k = 3$.

6) WEINER'S FILTERING

Weiner's filtering is mainly used to remove blur in an image processing application. It can also be treated as an attack. When stego images are under Weiner's filtering attack, the detection performance and robustness of the hidden data provided by our proposed scheme with threshold $k = 3$ are demonstrated in Table 12 and Fig. 11. An average of 35,516 blocks are identified as tampered, the *BER* of the hidden EPIs is 36.83%, and *NCC* is 0.6316, which means there are around 63% of hidden EPIs that still can be correctly extracted when more than half of a stego image has been tampered.

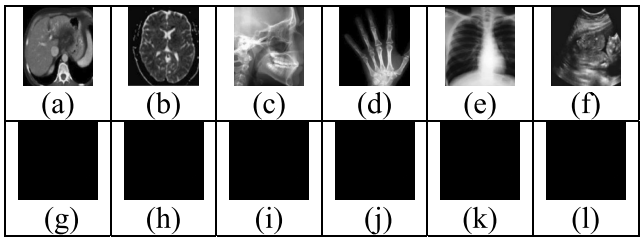

FIGURE 11. (a)-(f) marked attacked images. (g)-(l) tampered areas for a Weiner filtering attack with $k = 3$.

7) SHARPENING ATTACK

When stego images are under a sharpening attack, the detection performance and robustness of the hidden data provided by our proposed scheme with a threshold $k = 3$ are demonstrated in Table 12 and Fig. 11. An average of 35,516 blocks are identified as tampered, the *BER* of the hidden EPIs is 36.83%, and *NCC* is 0.6316, which means there are around 63% of hidden EPIs still can be correctly extracted when more than half of a stego image has been tampered.

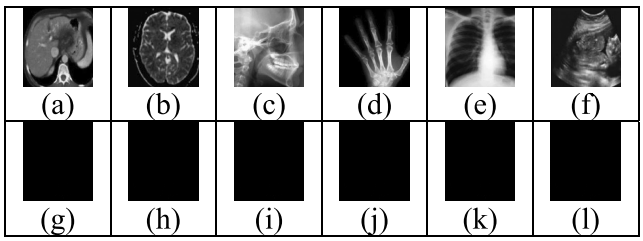

FIGURE 12. (a)-(f) marked attacked images. (g)-(l) tampered areas for a sharpening attack with $k = 3$.

TABLE 15. Comparisons of the Geeth & Geeth scheme and our proposed scheme on the detection performance and robustness of the hidden data: for test image "Thorax X-ray".

Attacks	Methods	Tampered blocks count	EPIs data	
			BER %	*NCC*
Salt and peppers	Proposed scheme(k=3)	15986 (24%)	19.47%	0.8053
	Proposed scheme(k=4)	15892 (24%)	19.68%	0.8032
	Geeth & Geeth [23]	11429 (17%)	4.89%	0.9521
AWGN	Proposed scheme(k=3)	36300 (55%)	43.48%	0.5652
	Proposed scheme(k=4)	36321 (55%)	43.49%	0.5651
	Geeth & Geeth [23]	15335 (23%)	50.12%	0.4998
median filtering	Proposed scheme(k=3)	36418 (56%)	33.48%	0.6652
	Proposed scheme(k=4)	35009 (53%)	37.17%	0.6283
	Geeth & Geeth [23]	14612 (22%)	41.11%	0.5977
Weiner filter	Proposed scheme(k=3)	37272 (57%)	38.10%	0.6190
	Proposed scheme(k=4)	39193 (60%)	31.72%	0.6828
	Geeth & Geeth [23]	15061 (23%)	47.67%	0.5286
sharpening	Proposed scheme(k=3)	39402 (60%)	19.35%	0.8065
	Proposed scheme(k=4)	38778 (59%)	21.77%	0.7823
	Geeth & Geeth [23]	15271 (23%)	49.23%	0.5015
JPEG(QF=90)	Proposed scheme(k=3)	39927 (61%)	34.84%	0.6516
	Proposed scheme(k=4)	40147 (61%)	25.41%	0.7459
	Geeth & Geeth [23]	14539 (22%)	43.19%	0.5743

C. COMPARISONS WITH OTHER SCHEMES

To further demonstrate the performance of our proposed scheme on image quality, hiding capacity, detection performance and robustness of the hidden data. We selected six representative data hiding schemes [5], [23], [25]–[28] for medical images to compare with our proposed scheme. Comparisons are listed in Table 14.

Table 14 Comparisons among seven representative data hiding schemes and our proposed scheme

Table 14 shows that our proposed scheme can conceal 327,680 bits into a stego image sized 768 × 768 pixels expanded by ENMI. Since a 2×2 block in the original image is expanded into a 3×3 block in a stego image, four secret bits and 1-bit authentication code are embedded into a 3×3 block. Therefore, our proposed scheme offers $0.5(= 5/9)$ bpp on average while the hiding capacity is up to 327,680 bits into the stego image in total. The hiding capacity offered by our proposed scheme is around 49 dB on average, which is lower than the Geeth & Geeth scheme [23] but significantly higher than the remaining five schemes. When $k = 3$, the *SSIM* of our proposed scheme remains 0.98.

Since the Geeth & Geeth scheme [23] outperforms our proposed scheme in terms of hiding capacity, we first select two medical images: "Thorax X-ray" and "Brain MRI scan"

TABLE 16. Comparisons of the Geeth & Geeth scheme and our proposed scheme on the detection performance and robustness of the hidden data: for test image "Brain MRI scan".

Attacks	Methods	Tampered blocks count	EPIs data	
			BER %	*NCC*
Salt &Peppers	Proposed scheme(k=3)	15736 (24%)	19.38%	0.8062
	Proposed scheme(k=4)	15784 (24%)	19.37%	0.8063
	Geeth & Geeth [23]	11489 (18%)	4.95%	0.9520
AWGN	Proposed scheme(k=3)	34701 (53%)	38.71%	0.6129
	Proposed scheme(k=4)	34550 (53%)	38.28%	0.6172
	Geeth & Geeth [23]	15297 (23%)	49.86%	0.5022
median filtering	Proposed scheme(k=3)	38452 (59%)	30.31%	0.6969
	Proposed scheme(k=4)	38213 (58%)	44.19%	0.5581
	Geeth & Geeth [23]	14804 (23%)	41.72%	0.5980
Weiner filter	Proposed scheme(k=3)	40387 (62%)	38.12%	0.6188
	Proposed scheme(k=4)	37651 (57%)	43.59%	0.5641
	Geeth & Geeth [23]	15271 (23%)	49.45%	0.5057
sharpening	Proposed scheme(k=3)	42812 (65%)	21.52%	0.7848
	Proposed scheme(k=4)	41597 (63%)	22.65%	0.7735
	Geeth & Geeth [23]	15324 (23%)	50.01%	0.4977
JPEG(QF=90)	Proposed scheme(k=3)	42466 (65%)	37.56%	0.6244
	Proposed scheme(k=4)	38918 (59%)	36.76%	0.6324
	Geeth & Geeth [23]	15027 (23%)	46.87%	0.5340

to further examine the detection performance and robustness of the hidden data. Tables 15 and 16 show in "salt and peppers attack," the fragile level of our proposed scheme is higher than the Geeth & Geeth scheme. Additionally, for the rest attacks, our proposed scheme always provides fewer BERs of the hidden data than those offered by the Geeth & Geeth scheme. For various filtering attacks, such as median filtering and so on, our average BER is around 11% higher than that of the Geeth & Geeth scheme when k = 3.

Geeth & Geeth scheme and ours is 21.93% and 52.80%, respectively. The detection performance of our proposed scheme is about two times that of the Geeth & Geeth scheme. Moreover, when half of the blocks have been tampered, the *BER* of the extracted EPIs is 29.87%, 39.36% for our proposed scheme with $k = 3$ and the Geeth & Geeth scheme, respectively. Thus, our proposed scheme provides the hidden data with higher protection and strong robustness, such that the *BER* offered by our scheme is significantly lower than the Geeth & Geeth scheme. The lower the *BER* is, the higher the similarity between the extracted secret data and the original secret data. In other words, both the detection performance and protection mechanism of our proposed scheme based on range-based instead of specific value for hidden authentication code or secret message are confirmed.

TABLE 17. Summary comparisons of the Geeth & Geeth scheme and our proposed scheme on the detection performance and robustness of the hidden data: "all medical images".

Attacks	Methods	Tampered blocks count	EPIs data	
			BER %	*NCC*
Salt & Peppers	Proposed scheme(k=3)	15829 (24%)	19.42%	0.80583
	Proposed scheme(k=4)	15851 (24%)	19.47%	0.80527
	Geeth & Geeth [23]	11421 (17%)	4.91%	0.95205
AWGN	Proposed scheme(k=3)	35201 (54%)	40.62%	0.59377
	Proposed scheme(k=4)	35127 (54%)	40.35%	0.5965
	Geeth & Geeth [23]	15325 (23%)	49.98%	0.50122
median filtering	Proposed scheme(k=3)	36804 (56%)	31.33%	0.6867
	Proposed scheme(k=4)	37587 (57%)	40.15%	0.59847
	Geeth & Geeth (2020)	14745 (22%)	41.40%	0.59795
Weiner filter	Proposed scheme(k=3)	38291 (58%)	37.92%	0.62082
	Proposed scheme(k=4)	37940 (58%)	37.60%	0.62398
	Geeth & Geeth [23]	15201 (23%)	48.53%	0.51625
sharpening	Proposed scheme(k=3)	40542 (62%)	20.62%	0.79385
	Proposed scheme(k=4)	39696 (61%)	22.21%	0.7779
	Geeth & Geeth [23]	15307 (23%)	49.65%	0.50017
JPEG(QF=90)	Proposed scheme(k=3)	40967 (63%)	36.09%	0.63907
	Proposed scheme(k=4)	39221 (60%)	31.04%	0.68965
	Geeth & Geeth [23]	14843 (23%)	45.22%	0.55212

V. CONCLUSION

In this proposed scheme, we first expanded each block sized 2×2 pixels to a 3×3-sized block, with the four original pixels remaining at the four corners in each expanded block. For five pixels derived by the ENMI interpolation algorithm, a simple data hiding strategy is designed with pixel concentration in each secret-bit range. In other words, pixels are mapped to one of mutually exclusive secret-bit ranges and all pixels carrying secret data are concentrated in each secret-bit range, so that the possibility of pixels changing to the other secret-bit range is significantly reduced even when an attack occurs.

Our proposed scheme provides data hiding strategy with the low computation complexity, and the experimental results show the average *BER*s of our proposed scheme as $k = 3$ and 4, which is 31.5% and 29.8%, respectively. These values are lower than that of the Geeth & Geeth scheme (39.3%). The number of tampered blocks accumulated by our proposed scheme with $k = 3$ and 4, are 34217 and 34223, respectively, which are higher than that of the Geeth & Geeth scheme (14374). In other words, the experimental results confirm that the detection performance and the robustness of the hidden data with our proposed scheme is significantly higher than that of the Geeth & Geeth scheme. With our proposed scheme, doctors can receive sufficient information (ex. medical images and EPIs) for medical treatment at the same time when receiving required medical image(s) in a single time. Moreover, they can judgement whether the received medical image(s) have been tampered or not. This will be quite helpful for telemedicine. In the future, we will further explore methods to improve on our proposed scheme in terms of a salt and pepper attack and then increase the robustness of the hidden data while increasing the hiding capacity.

REFERENCES

[1] W. Stallings, *Cryptography and Network Security Principles and Practices*, 7th ed. London, U.K.: Pearson, 2016.
[2] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. ICIP*, vol. 2. Austin, TX, USA: Austin Convention Center, 1994, pp. 86–90.
[3] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, Mar. 2004.
[4] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 313–336, 1996.
[5] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 889–896, Aug. 2003.

[6] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[7] C.-C. Chang, T.-S. Nguyen, and C.-C. Lin, "Reversible data hiding scheme for VQ indices based on absolute difference trees," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 7, pp. 2572–2589, Jul. 2014.

[8] R.-Z. Wang and Y.-D. Tsai, "An image-hiding method with high hiding capacity based on best-block matching and k-means clustering," *Pattern Recognit.*, vol. 40, no. 2, pp. 398–409, Feb. 2007.

[9] R. Rodriguez-Colin, F.-U. Claudia, and G. D. J. Trinidad-Blas, "Data hiding scheme for medical images," in *Proc. 17th Int. Conf. Electron., Commun. Comput. (CONIELECOMP)*, Feb. 2007, doi: 10.1109/CONIELECOMP.2007.14.

[10] K.-H. Jung and K.-Y. Yoo, "Data hiding method using image interpolation," *Comput. Standards Interfaces*, vol. 31, no. 2, pp. 465–470, Feb. 2009.

[11] Z. Chang and J. Xu, "Reversible run length data embedding for medical images," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw. (ICCSN)*, May 2011, pp. 260–263.

[12] C. V. Kumar, V. Natarajan, and D. Bhogadi, "High capacity reversible data hiding based on histogram shifting for medical images," in *Proc. Int. Conf. Commun. Signal Process.*, Melmaruvathur, India, Apr. 2013, pp. 730–733.

[13] Y.-T. Chang, C.-T. Huang, C.-F. Lee, and S.-J. Wang, "Image interpolating based data hiding in conjunction with pixel-shifting of histogram," *J. Supercomput.*, vol. 66, no. 2, pp. 1093–1110, Nov. 2013.

[14] C.-C. Wang, Y.-F. Chang, C.-C. Chang, J.-K. Jan, and C.-C. Lin, "A high capacity data hiding scheme for binary images based on block patterns," *J. Syst. Softw.*, vol. 93, pp. 152–162, Jul. 2014.

[15] K.-H. Jung and K.-Y. Yoo, "Steganographic method based on interpolation and LSB substitution of digital images," *Multimedia Tools Appl.*, vol. 74, no. 6, pp. 2143–2155, 2015.

[16] C.-N. Yang, S.-C. Hsu, and C. Kim, "Improving stego image quality in image interpolation based data hiding," *Comput. Standards Interfaces*, vol. 50, pp. 209–215, Feb. 2017.

[17] C.-C. Chang, C.-T. Li, and Y.-Q. Shi, "Privacy-aware reversible watermarking in cloud computing environments," *IEEE Access*, vol. 6, pp. 70720–70733, 2018.

[18] A. A. Mohammad, A. Al-Haj, and M. Farfoura, "An improved capacity data hiding technique based on image interpolation," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7181–7205, Mar. 2019.

[19] C.-C. Chang, J.-Y. Chen, Y.-H. Chen, and Y. Liu, "A reversible data hiding method for SMVQ indices based on improved locally adaptive coding," *Int. J. Netw. Secur.*, vol. 22, no. 4, pp. 575–583, 2020.

[20] Y.-Q. Chen, W.-J. Sun, L.-Y. Li, C.-C. Chang, and X. Wang, "An efficient general data hiding scheme based on image interpolation," *J. Inf. Secur. Appl.*, vol. 54, pp. 102584–102607, Oct. 2020.

[21] C.-C. Lin, J. Lin, and C.-C. Chang, "Reversible data hiding for AMBTC compressed images based on matrix and Hamming coding," *Electronics*, vol. 10, no. 3, p. 281, 2021, doi: 10.3390/electronics10030281.

[22] S. Gull, S. A. Parah, and K. Muhammad, "Reversible data hiding exploiting Huffman encoding with dual images for IoMT based healthcare," *Comput. Commun.*, vol. 163, pp. 134–149, Nov. 2020.

[23] R. Geetha and S. Geetha, "Efficient high capacity technique to embed EPR information and to detect tampering in medical images," *J. Med. Eng. Technol.*, vol. 44, no. 2, pp. 55–68, Feb. 2020.

[24] S. Sheidani, A. Mahmoudi-Aznaveh, and Z. Eslami, "CPA-secure privacy-preserving reversible data hiding for JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3647–3661, 2021, doi: 10.1109/TIFS.2021.3080497.

[25] A. P. Shabir, A. S. Javaid, and G. M. Bhat, "Data hiding in ISB planes: A high capacity blind stenographic technique," in *Proc. Int. Conf. INCOSET*, Tiruchirappalli, India, 2012, pp. 192–197.

[26] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.

[27] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *Electron. Lett.*, vol. 38, no. 25, pp. 1646–1648, Dec. 2002.

[28] C.-F. Lee and Y.-L. Huang, "An efficient image interpolation increasing payload in reversible data hiding," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6712–6719, Jun. 2012.

CHIA-CHEN LIN received the Ph.D. degree in information management from the National Chiao Tung University, in 1998. Since 2018, she has been the School Counselor with Providence University. She is currently a Professor with the Department of Computer Science and Information Management, Providence University. Her research interests include image and signal processing, information hiding, mobile agent, and electronic commerce. Since 2018, she has been a fellow of IET. From 2009 to 2012, she served as the Vice Chairman for the Tainan Chapter IEEE Signal Processing Society. She also serves as an Associate Editor and an Editor for several representative EI and SCIE journals.

CHIN-CHEN CHANG (Fellow, IEEE) received the Bachelor of Science degree in applied mathematics and the Master of Science degree in computer and decision sciences from the National Tsing Hua University, and the Ph.D. degree in computer engineering from the National Chiao Tung University. From 1989 to 2005, he worked with the National Chung Cheng University. His current title is the Chair Professor with the Department of Information Engineering and Computer Science, Feng Chia University, since February 2005. Prior to joining Feng Chia University, he was an Associate Professor with the National Chiao Tung University, a Professor with the National Chung Hsing University, and a Chair Professor with the National Chung Cheng University. He had also been a Visiting Researcher and a Visiting Scientist with Tokyo University and Kyoto University, Japan. During his service with the National Chung Cheng University, he served as the Chairman of the Institute of Computer Science and Information Engineering, the Dean of the College of Engineering, a Provost and then an Acting President of Chung Cheng University, and the Director of Advisory Office in Ministry of Education, Taiwan. On numerous occasions, he was invited to serve as a Visiting Professor, a Chair Professor, an Honorary Professor, an Honorary Director, an Honorary Chairman, a Distinguished Alumnus, a Distinguished Researcher, and a Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures. He is currently a fellow of IEE, U.K. He has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. Since his early years of career development, he consecutively won the Outstanding Talent in Information Sciences of the R. O. C., the AceR Dragon Award of the Ten Most Outstanding Talents, the Outstanding Scholar Award of the R. O. C., the Outstanding Engineering Professor Award of the R. O. C., the Distinguished Research Awards of National Science Council of the R. O. C., and the Top Fifteen Scholars in Systems and Software Engineering of the *Journal of Systems and Software*.

WEI-JIUN KAO received the master's degree in information engineering and computer science from Feng Chia University, in 2022. His research interests include image and signal processing and information hiding.

JUI-FENG CHANG received the master's degree in information engineering and computer science from Feng Chia University, in 2022. His research interests include image and signal processing and information hiding.