Multimedia Tools and Applications (2024) 83:4993–5030
https://doi.org/10.1007/s11042-023-15319-8

A prediction error based reversible data hiding scheme in encrypted image using block marking and cover image pre-processing

Shaiju Panchikkil[1] · **V. M. Manikandan**[1]

Received: 5 September 2022 / Revised: 3 January 2023 / Accepted: 6 April 2023/
Published online: 30 May 2023

Abstract
A drastic change in communication is happening with digitization. Technological advancements will escalate its pace further. The human health care systems have improved with technology, remodeling the traditional way of treatments. There has been a peak increase in the rate of telehealth and e-health care services during the coronavirus disease 2019 (COVID-19) pandemic. These implications make reversible data hiding (RDH) a hot topic in research, especially for medical image transmission. Recovering the transmitted medical image (MI) at the receiver side is challenging, as an incorrect MI can lead to the wrong diagnosis. Hence, in this paper, we propose a MSB prediction error-based RDH scheme in an encrypted image with high embedding capacity, which recovers the original image with a peak signal-to-noise ratio (PSNR) of ∞ dB and structural similarity index (SSIM) value of 1. We scan the MI from the first pixel on the top left corner using the snake scan approach in dual modes: i) performing a rightward direction scan and ii) performing a downward direction scan to identify the best optimal embedding rate for an image. Banking upon the prediction error strategy, multiple MSBs are utilized for embedding the encrypted PHR data. The experimental studies on test images project a high embedding rate with more than 3 bpp for 16-bit high-quality DICOM images and more than 1 bpp for most natural images. The outcomes are much more promising compared to other similar state-of-the-art RDH methods.

Keywords Medical image transmission · Snake scan · Prediction error strategy · High quality DICOM image · Reversible data hiding

✉ V. M. Manikandan
manikandan.v@srmap.edu.in

Shaiju Panchikkil
shaiju_panchikkil@srmap.edu.in

[1] SRM University-AP, Andhra Pradesh, India

1 Introduction

In the olden days, people had to travel distances to get the services of a specialist physician. It was difficult for the people to get a better diagnosis and hence the best treatment. The growth of communication technologies has made a positive impact on the conventional medical system. The transformation of the conventional health system to telehealthcare and e-health care system was inevitable. Designing and implementing algorithms for medical e-health care systems is in focus these days [38]. After all, better life follows better health.

Digitization has been a key factor that has also accelerated the growth of e-health care systems, demanding sharing of patient's personal information, medical history reports, diagnosis data, etc. This helps the patient to get the best possible diagnosis and treatment from an expert, who is at a remote location without any delay. But, unauthorized access to patient's personal details is illegal, which makes data security one of the most important challenges to be addressed. This has led the research community to focus on data-hiding techniques in highly sensible applications like medical image transmission.

In literature, there are encryption-based techniques that can secure the cover image, such as [48] integrating different scanning approaches with an El-Gamal encryption algorithm to encrypt the image and finally a chaotic system to scramble the pixels. Another example of secure transmission of the images is given in [47], which intends to shield the privacy and confidentiality of images sent over any communication medium. The digital data, whether it is a cover image, can also be secured by adding additional supporting content, such as a watermark. For example, [20] introduced an interpolation-based reversible watermarking technique for secure management of the Digital Imaging and Communications in Medicine (DICOM) images. An error matrix is computed by taking the difference between the original and interpolated images. The histogram of which gives 4 parameters: LN, LM, RM, and RN. Additional data gets embedded in the LSBs of LM and RM pixels. The highlight of the scheme is in using an adaptive linear minimum mean square error estimation-based quartered interpolation algorithm, that gave better PSNR and embedding results.

Unlike securing the cover medium, steganography is a method to secure the information through a cover medium, in a visually imperceptible way. Recently, researchers have also brought in techniques to secure the information by distributing the payload in multiple cover mediums, such as images [24] or by utilizing the RGB channels [25]. Image steganography based on generative adversarial network (GAN) that can generate perceptually indistinguishable stego images is also exploited [39].

Reversible data hiding (RDH) is a data hiding technique that is used extensively as a solution to hiding PHR into MI. It is very important to hide the PHR within the MI, in an imperceptible manner. The advantage of the RDH over any other data-hiding technique is in regenerating the MI at the receiver end while recovering the embedded PHR. Also, it is essential for the receiver to generate the same MI without any quality deterioration to make the correct diagnosis. Researchers have also been improvising RDH schemes for protecting privacy or patient content authentication, tamper localization, contrast enhancement, etc.

Most of the RDH algorithms have been designed based on different approaches. In compression-based schemes [22, 26], the cover image is compressed using different compression techniques that maximize utilization of the redundancy in the cover image and would ensure a lossless recovery. Another approach is histogram shifting-based [12, 21, 49], where the intensity bins are shifted based on the peak intensity value to hide the additional data in the empty bin. Here the peak intensity value decides the embedding capacity of the RDH scheme. Difference expansion is another technique [11, 32], where the secret

information embedding depends on the difference in the pixel intensities of neighboring pixels. Prediction error-based is a different approach [16, 31], where a prediction method will be employed to predict the pixels. Hence, the embedding capacity depends on the correct prediction of maximum pixels. The last category is interpolation-based [46]. Here the cover image is interpolated to generate an interpolated image. The interpolated pixels error is accounted for to embed the additional information through specific techniques like histogram shifting.

All the approaches need to find room for embedding confidential private information like the PHR data, either before encrypting the cover image or after the cover image encryption. Two schemes proposed in [36] vacated rooms before encrypting the cover image. Both rely on MSB prediction. In the first scheme, the prediction errors are rectified by modifying the pixels, which reduced the error difference, resulting in a much similar cover image after recovery. The second scheme avoided the blocks with prediction errors without embedding the private information. Thus, the first scheme could not recover the exact cover image in the worst-case scenario, while the second is fully recoverable. The second scheme divided the cover image into non-overlapping blocks of 1×8 pixels and used flags to indicate the presence of prediction error blocks. In another scheme [1], the cover image is encrypted using additive modulo 256, and the secret data is embedded by preserving the mean of pixels in each column. The mean of pixels in a column is stored in the topmost location of that column. Hence, the preserved pixel is unaltered while encrypting the image, and a single bit of additional data gets embedded in each pixel present in the column, except the topmost pixel. On the receiver side, after decryption, the mean of pixels in a column is compared with the top pixel value to extract the correct information. The original cover image gets recovered after extracting the hidden information.

The RDH scheme in [33] used Arnold transform scrambling technique to hide the private data. The image is processed as blocks of size $M \times M$ and is transformed into different matrices using the Arnold transform algorithm. The transformation depends on the transformation matrix employed. These different matrices generated are the means of embedded information. On the receiver side, each block is transformed to produce the maximum possible variations of the block using the Arnold transform. These blocks get decrypted. A convolutional neural network model identifies the correct block. Based on the recovered block, the embedded information gets extracted. Here the authors have utilized the cyclic property of Arnold transform to hide the private data. To reduce the overhead of transferring the machine learning model with the receiver in [33], the recovery of blocks of the cover image is facilitated through the correlation strength of the pixels in [34]. Whereas the scheme in [7] trained a linear regression-based predictor for RDH. A prediction error map is computed, and the data hider embeds the secret data along with the auxiliary information in the encrypted image. The pixel value predictor predicts the correct pixel from its original neighboring pixels. Hence, the auxiliary information includes a pixel value predictor and the prediction error map, which helps the receiver to recover the cover image. The scheme achieves an embedding greater than 0.5 bpp with good visual quality. A deep neural network (DNN) assisted RDH is proposed in [17]. The pixels employed for embedding the additional data get labeled through the trained DNN. All the pixels under the same label will make a single histogram. Likewise, based on multiple labels, multiple histograms are generated. Meanwhile, the prediction-error histogram of pixels under the same label gets computed by considering the 4 nearest neighbors across each pixel. As in general prediction-error expansion methods, prediction-error is modified to embed the secret data. Here, the algorithm mainly needs to compute the effective payload, embedding distortion, and search for

optimal expansion bins on the multiple histograms, which is a computational overhead. But they build two memos matrix in advance and preserve them to reduce repeated calculations. The performance of this RDH scheme is better in terms of minimum distortion embedding.

Interpolation-based schemes are also exploited for RDH. As in [28], implemented a directional pixel value ordering-based RDH using interpolation. Here, the cover image of size $P \times P$ pixels is interpolated into $(2P - 1) \times (2P - 1)$. A parameter k gets added and subtracted to the maximum and minimum pixels of the block, considering row, column, and diagonal directions. The process is repeated each time by decrementing k by 1 until the value of k becomes zero. The k introduced is to maintain the pixel's rank in the block. Data is embedded by first considering the pixels in the horizontal direction, then vertical direction pixels, and finally in the diagonal direction. Such an embedding, utilizing the overlapped pixels, resulted in better embedding in a PVO-based data hiding scheme with good visual quality. The interpolation-based schemes can give a good embedding capacity as new pixels get introduced in addition to the original pixels. An enhanced neighbor mean interpolation and modified neighbor mean interpolation technique gave a finer interpolated image in [15]. Additional data were embedded by accounting for the difference between the highest cover pixel and the interpolated pixel in each block of 3×3 pixels interpolated matrix, which facilitated generating the highest payload. Interpolation-based schemes are widely explored as they can expand the payload considerably. The scheme in [29] discusses an interpolation-based RDH with a high payload and computationally simplified technique. Initially, the given image $2N \times 2N$ is scaled down to $(N+1) \times (N+1)$ and then interpolated, by scanning each 2×2 overlapping block as 3×3 sized blocks. The original pixels of the downsized image are retained in the interpolated image. Considering the interpolated image as 2×2 non-overlapping blocks, each top left corner is an original pixel from the downsized image, and the rest are blank. These blank cells are filled using predefined equations using the original 2×2 pixels of the downsized image block. Now, the difference between the top corner pixel of a 2×2 non-overlapping block from the interpolated image and each of the rest 3 pixels defines the embedding capacity. The embedding in each interpolated pixel is limited by a value from 0 to 8. The sender sets this value, and whatever is the minimum between this value and the difference will be the embedded capacity. Hence, limiting the distortion. The employed interpolation technique along with the data hiding algorithm reduced the distortion in the marked image, resulting in better visual quality.

Similar to the MSB prediction-based RDH, [44] used prediction and embedded the secret information at the MSBs of pixels via a matrix encoding technique. The prediction error map based on orthogonal projection over the original cover image is also embedded to recover the correct cover image at the receiver side. The scheme achieved an embedding capacity of 0.412 bpp on average without bit errors. Whereas, [6] considered sparse representation at the patch level for hiding the additional information. Here the residual errors generated while performing sparse coding are embedded along with the private data within the encrypted image. This helps in recovering the original image completely. It is also an RDH scheme wherein the rooms for embedding the information are vacated before encryption. Unlike any other scheme, an RDH in the frequency domain is discussed in [45]. In this scheme, the wavelet coefficients are first computed via integer wavelet transform on the cover image. The embedding of secret data is carried out on the encrypted frequency coefficients using the histogram shifting technique.

As observed in the literature, redundancy in an image is a major factor exploited by most schemes to embed private information. Techniques that are incorporated and the approach make them different from one another. Working on RDH in an encrypted image

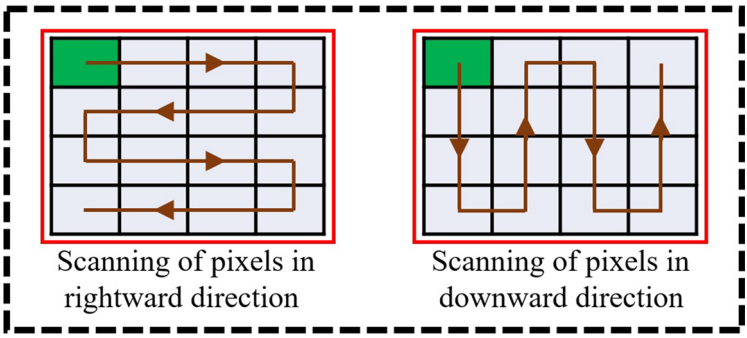

Fig. 1 Snake scan framework: scanning of pixels in the image

is challenging, as no two adjacent pixels are correlated in the encrypted domain. This paper discusses a high-capacity RDH scheme in an encrypted domain, focused on 16-bit high-quality DICOM images. To maximize the utilization of redundancy present in an image, we have adopted the snake scan framework for scanning the pixels in the medical image. Figure 1 shows the scanning framework.

All images are not the same, and the structural properties of each image impact the amount of information that can be stored. Few schemes may embed more data in images of a particular nature. This work is an initiative to exploit the correlation better. We pre-process the cover image using the two-way snake scan framework by varying the block size. Deciding upon the block size and the scan approach will maximize the usage of the available redundancy. Hence, multiple MSBs of pixels in the blocks, without prediction error, are used for hiding the encrypted patient health record (PHR) data. Prediction error marker information is self-embedded with the encrypted private PHR data before transmission. This marker information helps the receiver to extract the hidden PHR data and recover the patient medical image (MI) without any quality degradation. As any changes in the recovered MI can lead to a wrong diagnosis, we have prioritized recovering the correct original medical image at the receiver end. A general framework of the proposed scheme is given in Fig. 2. Experimental results show its potency over the existing other similar RDH schemes. Additionally, we have experimented with 8 bit natural images, and the results are discussed in the experimental section.

The rest of this article is structured as follows. The proposed work is discussed in detail in Section 2, wherein we explain the whole process of embedding the PHR data within the patient MI along with the recovery process by extracting all the hidden PHR data. An illustration of the proposed scheme is given in Figs. 6 and 8. Section 3 reveals the results obtained from the proposed work through various experiments carried over medical and natural images. A comparative study of the proposed work with other similar existing works is discussed in Section 4. Finally, we conclude the paper with a few future directions in Section 5.

2 Proposed scheme

The embedding rate and quality of the recovered image are both inevitable parts with medical images as a cover medium. The proposed work is motivated by the two RDH approaches proposed in [36]. One among the two schemes is fully reversible, and the maximum

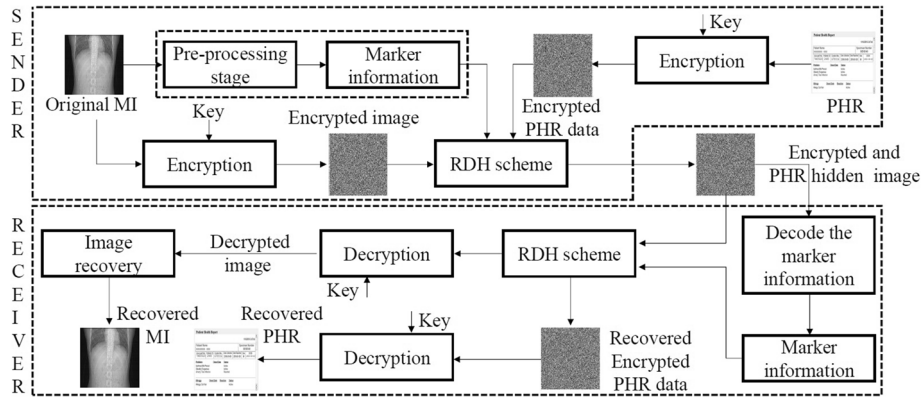

Fig. 2 General framework of the proposed work

achievable embedding rate is 1 bpp. They employed a single MSB plane replacement for embedding the additional data and developed an MSB prediction technique to recover the original image. Embedding of data is omitted in blocks with prediction errors by setting flags at the beginning and end of the marked block. Each flag is an 8 MSB bit, set to 1. They assumed that the chance of getting additional data with continuous eight 1 bits is rare. So, all the MSBs were embedded with additional data which was predictable. While previous and successive 8 MSBs were omitted from embedding when a non-embeddable block is found. The EPE scheme in [36] used the flag bit sequence for efficient recovery of the cover image. Dragoi and Coltuc [13] notified that a histogram analysis of runs of 1 can reveal the presence of flag bits and hence is vulnerable to threats. Hence, in the proposed work, we have avoided using flags but self-embedded the marker information for recovery. The experiments in [36] were carried out on natural images. The proposed scheme is applicable to medical image communication. Though there are medical images with 16, 12, and 8-bit pixel representations, researchers like in [2, 19] highlighted their study based on 8-bit pixel representation. Here, we investigate the MSB prediction technique by utilizing multiple MSB planes for embedding the PHR. We will discuss the results on high-quality DICOM medical images and 8-bit natural images. The highlights of the proposed RDH scheme are:

1 Scanning the image pixels follows the snake scan framework for better utilization of closely related pixels.
2 The pre-processing stage determines two parameters:
   - Whether to follow the rightward or downward snake scan framework for achieving the maximum embedding rate.
   - The block size that helps to embed the maximum information with negligible error.
3 The pre-processing stage helps the data hider to have an overall idea of the payload before the actual embedding process.
4 The requirement of flags is eliminated by having a predetermined space for embedding the information.
5 The receiver can extract all the embedded PHR information without any key, but the data decryption key is a must to read the actual PHR content.
6 The receiver can losslessly recover the cover image with the help of image decryption key after data extraction.

7 Experimental study is carried out on high quality 16-bit DICOM images and 8-bit natural images .

A data flow framework at the sender and the receiver side is shown in Figs. 3 and 4 respectively.

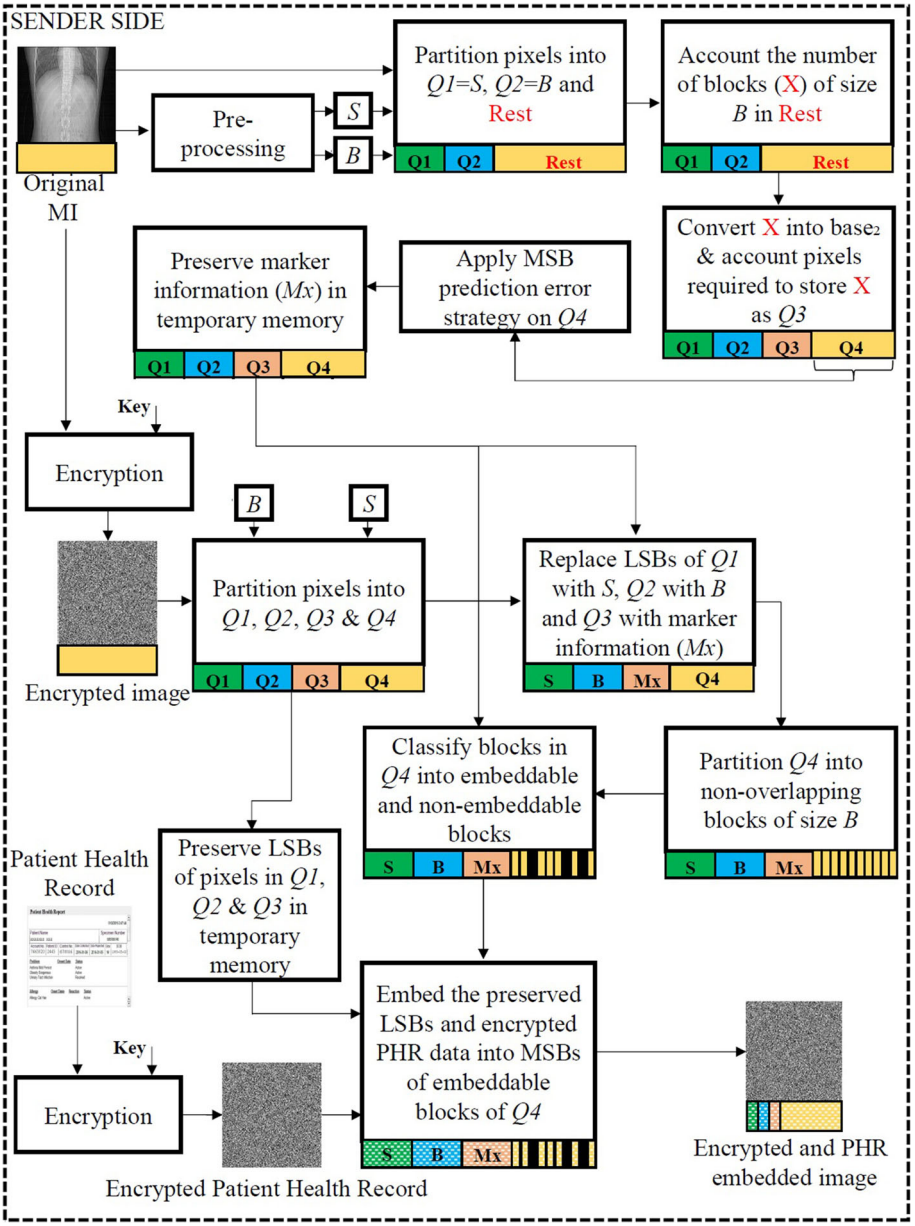

Fig. 3 Sender side data flow framework

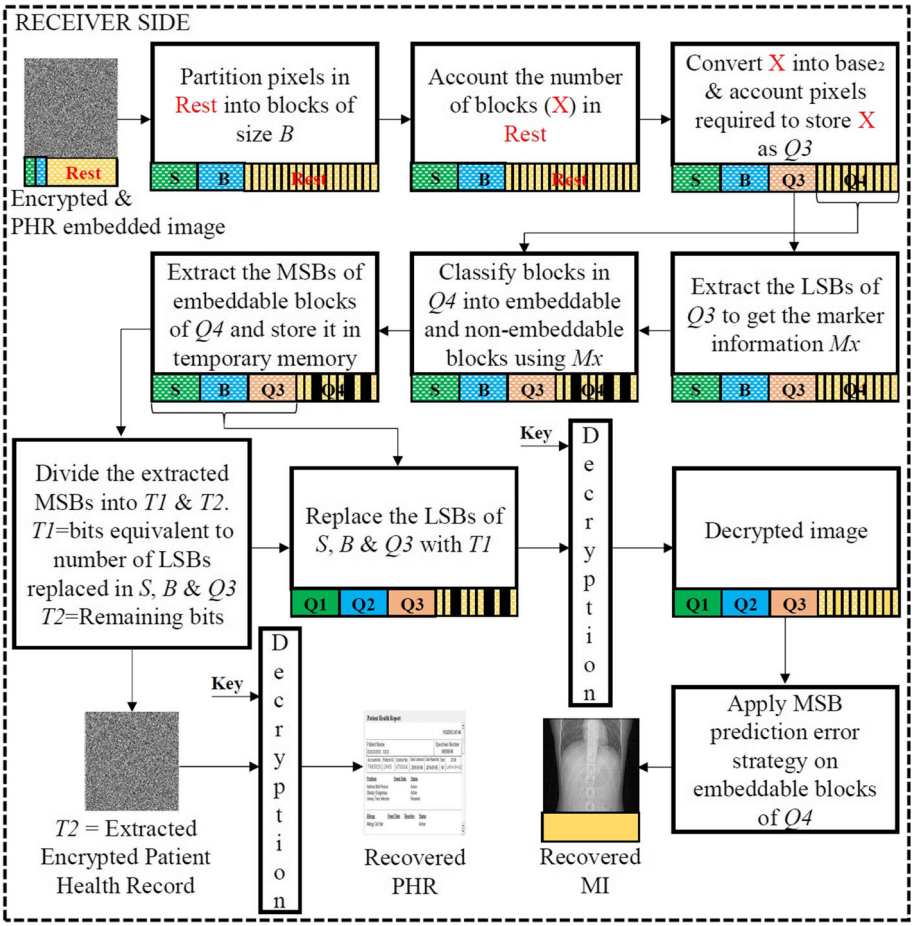

Fig. 4 Receiver side data flow framework

2.1 Pre-processing stage

The pre-processing stage identifies the type of snake scan approach to be employed while scanning the image and also identifies the appropriate block size that can give the maximum embedding rate. The original image "I" is chosen of size $R \times C$ where both R and C are "512" and each pixel $I_{i,j} \in [0, 255]$ on 8-bit image with $1 \leq i, j \leq 512$ and $I_{i,j} \in [0, 65535]$ in case of high quality 16-bit DICOM medical image. The output of pre-processing stage indicates the block size and the scan direction that should be followed while processing the image I. The total pixels of the image I is classified into 4 parts: $Q1$, $Q2$, $Q3$ and $Q4$ (refer Fig. 5).

- $Q1$ is the first pixel $I_{1,1}$.
- $Q2$ consists of next two contiguous pixel locations starting from the second pixel of the image I. Here, the second pixel will be either $I_{1,2}$ or $I_{2,1}$ based on the type of scanning approach employed.

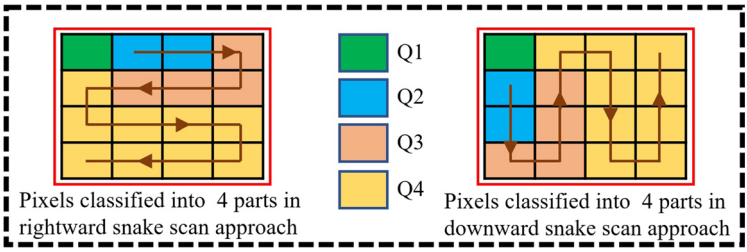

Fig. 5 Pixel partitioning

- $Q3$ consists of a fixed contiguous pixel locations starting from the fourth pixel of the image I.
- $Q4$ consists of all the remaining pixel locations after $(Q3+Q2+Q1)^{th}$ location [Please note that $Q1$ holds only one pixel].

$$\text{TP} = (R \times C) = Q1 + Q2 + Q3 + Q4, \tag{1}$$

where TP is the total pixels in the image I, that equals to 262144 for a 512×512 pixel image. "$Q1$" is used to preserve the scanning information, "$Q2$" for block size, and "$Q3$" for the marker information of each block present in "$Q4$". "$Q4$" has two portions: the first portion preserves the original LSBs of $Q1$, $Q2$, and $Q3$ that get modified while preserving the scanning information, block size, and marker information. The second portion embeds the PHR information.

Initially, the image "I" of size 512×512 pixels is scanned as non-overlapping blocks of size $1 \times W$ pixels, excluding the first 3 pixels. Therefore,

$$\text{Total available pixels} = (R \times C) - 3 \tag{2}$$

$$T1 = \left\lfloor \frac{\text{Total available pixels}}{W} \right\rfloor \tag{3}$$

where $T1$ is the total available blocks of size $1 \times W$ pixels. Therefore, the maximum marker information that can be generated is $T1$ bits. Here the marker information implies whether the pixels in the block are recoverable or not. Let A be the number of bit planes used to hide the private data. As per the proposed RDH scheme,

$$A = \begin{cases} 2 \text{ if the image is an 8-bit natural image} \\ \\ 4 \text{ if the image is a 16-bit DICOM image} \end{cases} \tag{4}$$

Please note that this is a pre–processing stage and we make few assumptions to define the actual values for the image processing block size and scanning pattern. In general, for $T1$ blocks, we will have $T1$ marker bits and to preserve $T1$ marker bits, we need $T1$ pixels. But, the proposed scheme embeds A bits in a single pixel. Hence, in order to preserve $T1$ marker bits, we need X pixels only.

$$X = \left\lceil \frac{T1}{A} \right\rceil. \tag{5}$$

Accordingly, $Q3$ consists of X locations to assure that $T1$ marker information is preserved as a combination of A bits. Now we divide the rest of the pixels that exist after the

first $X+3$ pixels of the input image MI I into non-overlapping blocks of size $1 \times W$ pixels, given by $T2$. It should be noted that the actual embedding rate will be calculated based on the embedding capability of blocks in $T2$.

$$T2 = \left\lfloor \frac{\text{Total available pixels-}X}{W} \right\rfloor, \tag{6}$$

where $T2$ is the total available blocks of size $1 \times W$ pixels after the first $X+3$ pixels. Please note that,

$$T2 \leq T1 \tag{7}$$

If B_i is the current block scanned with $1 \leq i \leq T2$, then the marker B_i^m is computed based on the prediction error strategy (discussed in Section 2.2.1):

$$B_i^m = \begin{cases} 0 \text{ if block } B_i \text{ is embeddable} \\ \\ 1 \text{ if block } B_i \text{ is not embeddable} \end{cases} \tag{8}$$

The above mentioned processes leads to the computation of the marker information. In practice we have adopted scanning of the image I via the snake scan approach that is implemented in two ways (refer to Fig. 5):

- scan from the fourth pixel on the first row towards the rightwards direction.
- scan from the fourth pixel on the first column towards the downwards direction.

M_R is the marker information obtained following block scanning via the rightward snake scan approach and M_D via following downward snake scan approach. In general, M_X is generated, where $X \in (R, D)$. It should be noted that M_R and M_D may or may not be the same for a single image.

$$M_X = [B_1^m \; B_2^m \; ... B_{T2}^m]. \tag{9}$$

Having got the marker information in both directions, we compute the total embeddable blocks S_X in both directions to decide the actual scanning direction on the image with the maximum embedding rate, where $X \in (R, D)$.

$$S_X = \{Y \mid Y = T2 - \sum_{i=1}^{T2} B_i^m\} \text{ where } X \in (R, D) \tag{10}$$

Note that $T2$ denotes the total blocks of size $1 \times W$ processed on pixels existing after the first $X+3$ pixels of the input image and $\sum_{i=1}^{T2} B_i^m$ accounts for the non–embeddable blocks in $T2$.

$$S = \begin{cases} 0 \text{ if } S_R > S_D : \text{Rightward snake scan} \\ \\ 1 \text{ otherwise : Downward snake scan} \end{cases} \tag{11}$$

S indicates the type of snake scan approach with maximum embedding rate, which will be utilized by the sender for further processing of the original MI I while embedding the PHR.

The pseudo-code for pre–processing the input MI is given in Algorithm 1. The output of the algorithm will be a multidimensional array $RESULT$. $RESULT$ will have entries of embedding rate in dual snake scan directions against each executed block size. We can identify the maximum embedding rate on the image MI by analyzing the array $RESULT$. The

corresponding snake scan direction S and the block size W against the maximum embedding rate get recorded for the actual processing of the MI.

2.2 Generating marker information

In most of the high-quality 16-bit DICOM medical images, we have observed that only 12 bits are utilized, for representing the image. This has been the key to using MSB substitution and MSB prediction RDH techniques for high-quality medical images. The proposed scheme is an RDH scheme in the encrypted image that embeds 2 bits of information in natural images and embeds as high as 4 bits of data from the PHR in a medical image. The technique utilizes an MSB prediction error approach for hiding the information, which in turn helps in the recovery process. In a $1 \times W$ pixel block, the first pixel in each embeddable block is unaltered, to facilitate the correct prediction of other pixels in the block. Therefore, $(W-1) \times A$ bits are embedded within a single block. If we can predict all the $(W-1)$ pixels correctly with the aid of the first pixel, then 0 will be the marker information of that block, else 1. This marker information is responsible for the embedding of PHR and the correct recovery of the original MI. Generally, two adjacent pixels are closely related, and the difference in their intensities is small. This is the strategy used for prediction error calculation.

2.2.1 Strategy of prediction

- Take P_i as the pixels in a block where $1 \leq i \leq W$.
- P_i is considered as the reference pixel for prediction.
- P_{i+1} is closely related to P_i and hence P_{i+1} is predicted using P_i.
- Let F indicate whether the image is an 8-bit or 16-bit image. Employing A bit MSB replacement in P_{i+1} yields 2^A combinations. i.e., for example, an MSB replacement of A bits with all set to 1 is evaluated as:

$$V_Z = \left(P_{i+1} + \sum_{i=F-(A-1)}^{F} 2^{(i-1)} \right) \bmod \left(\left(\sum_{i=0}^{(F-1)} 2^i \right) + 1 \right) \tag{12}$$

  Where V_Z indicate one combination of multiple MSB replacement and $1 \leq Z \leq 2^A$. The (12) implies that the difference between the original pixel value P_{i+1} and V_Z is $\sum_{i=F-(A-1)}^{F} 2^{(i-1)}$. i.e., in case of a 16-bit medical image the difference between P_{i+1} and V_{16} will be 61440. Also, it should be noted that, in the summation $\sum_{i=F-(A-1)}^{F} 2^{(i-1)}$), each term is valid only when the bit at position i is 1. So, different combinations of A MSB bits generates versions of P_{i+1}. It is obvious to note that one version is the original pixel P_{i+1} and there is much difference between this value and its other versions.
- While scanning pixels in a block, the previously scanned pixel acts as the reference for predicting the current pixel. In the current scenario, P_i is the reference for predicting P_{i+1}.
- The absolute value of the difference between the reference pixel and versions of the current pixel is calculated as:

$$D_Z = |P_i - V_Z| \tag{13}$$

- The difference between P_i and original value of P_{i+1} is D_1 , which is compared with differences calculated from other versions. Since, P_i and P_{i+1} are closely related, the

Input: The original image I of size $R \times C$ pixels, Block size array BS, Count of MSB bit planes A

Output: A multidimensional array $RESULT$, which contains the maximum achievable embedding rate in dual snake scan directions against each block size of BS

let $pixels = R \times C$;
let $RESULT$ be a multi-dimensional array of integers of size 15, initialized with zero's.
let BS be the block sizes ranging from D to $D + 14$, where the initial value of D can not be zero.
while $(BS \leq D + 14)$ **do**
    $CBS = BS$;
    $Total\_pixels = (R \times C) - 3$;
    $N = Total\_pixels/CBS$; /* Compute blocks of size $1 \times CBS$ */
    $Total\_available\_pixels = Total\_pixels - (N/A)$;
    $start = \left\lceil \frac{N}{A} \right\rceil + 3 + 1$; /* $start$ is the pixel location from where we generate the marker information for each block of size $1 \times CBS$ pixels */
    $kk1 = 0$;
    $i = start$;
    while $(i \leq pixels - CBS + 1)$ **do**
        Scan each block of I of size $1 \times CBS$ pixels from left to the right snake scan direction.
        Compute the marker information of each block BLK_i using the prediction error strategy.
        $kk1 = kk1 + 1$;
        Save the marker information in a binary array $B1[kk1]$.
        $i = i + CBS$;
    $kk2 = 0$;
    $i = start$;
    while $(i \leq pixels - CBS + 1)$ **do**
        Scan each block of I of size $1 \times CBS$ pixels from the top to bottom snake scan direction.
        Compute the marker information of each block BLK_i using the prediction error strategy.
        $kk2 = kk2 + 1$;
        Save the marker information in a binary array $B2[kk2]$.
        $i = i + CBS$;
    $B1\_embeddable\_blocks$=$kk1$-sum of 1's in $B1$;
    $B2\_embeddable\_blocks$=$kk2$-sum of 1's in $B2$;
    $B1\_embeddable\_bits$=$B1\_embeddable\_blocks \times (CBS - 1) \times A$;
    $B2\_embeddable\_bits$=$B2\_embeddable\_blocks \times (CBS - 1) \times A$;
    /* As we need to preserve the scan information, block size, and marker information in the LSBs of the first $3 + \left\lceil \frac{N}{A} \right\rceil$ pixels, the actual values of these LSBs will be added to the PHR data to support the recovery. */
    $Bitspace\_of\_LeftRight\_scan = B1\_embeddable\_bits - 3 \times A - N$;
    $Bitspace\_of\_TopDown\_scan = B2\_embeddable\_bits - 3 \times A - N$;
    $EmbeddingRate\_of\_LeftRight\_scan = Bitspace\_of\_LeftRight\_scan/pixels$;
    $EmbeddingRate\_of\_TopDown\_scan = Bitspace\_of\_TopDown\_scan/pixels$;
    Add the block size CBS and embedding rate results to the array $RESULT$.
    $BS = BS + 1$;
Output the array $RESULT$ which contains the achievable embedding rate in both the snake scan directions against each block size.

Algorithm 1 Pre-processing of input images.

Input: A block BLK of size $1 \times W$ and Count of MSB bit planes A.
Output: A bit "0" if the block is embeddable, else "1".
/* Please note that the 1st pixel in a block is a reference pixel. */
let $i = 1$;
while $(i \leq W)$ **do**
  /* We check whether each adjacent pixel on the right is predictable from the pixel on its left. ie., since the proposed scheme tries to embed A bits of information in each pixel, we alter the A MSBs of the adjacent pixel and see if the actual pixel is predictable from the pixel on its left. */
  Scan the block BLK, from left and access the two adjacent pixels at a time.
  let the left pixel be $Pred$ and the right pixel be X.
  Compute different values of X by changing all combinations of A MSBs. Let the different values of X be $X_1, X_2,.., X_n$. /* Here X_1 indicates the original value of X and $n = 2^A$. */
  If the difference between $pred$ and X_1 is lesser than the difference between $pred$ and other values of X, then the pixel X_1 is predictable through $pred$. Hence we make the $mark$ bit 0.
  /* The above process is repeated for until all the adjacent pixels are encountered for prediction. If all the adjacent pixels are predictable from its left pixel, the value 0 will be sent as the $mark$ bit, else 1. */
  $i = i + 1$;
Output the marker information $mark$.

Algorithm 2 Generating MARKER information of a block.

other differences will be quite large making the pixel P_{i+1} predictable through P_i. The same process is repeated for the rest of the pixels, and if all the $W-1$ pixels of the block are predictable via the previously recovered pixel, then the whole block is recoverable. This information is stored as marker information of the block.

2.3 Embedding PHR in MI

As mentioned in the pre-processing stage, the embedding of the PHR in MI is done in $Q4$. Figure 6 shows a small illustration of the embedding process. For consistency, the sender preserves A LSB bits of the first $Q3 + Q2 + 1$ pixels. Marker information M_X of each non-overlapping block (block considered after the first $Q3 + Q2 + 1$ pixels) is calculated, following the scanning approach "S" that is an outcome of the pre-processing stage. The block size to be considered for processing pixels in $Q4$ is another outcome of the pre-processing stage. Combining the scanning approach S and the suggested block size makes the RDH scheme achieve its maximum embedding capacity. The MI I is encrypted by XORing with a pseudo-random matrix generated using an image encryption key, resulting in an encrypted image I'. The pseudo-random matrix used is similar in size to that of I and

Input: The original image I of size $R \times C$ pixels, block size W, Count of MSB bit planes A, Block scan information S, Key $K1$ and Key $K2$

Output: An encrypted image with hidden PHR data

/* Please note that due to space limitation a few of the variables such as *pixels*, *start*, *Bitspace* have been taken from Algorithm 1. The values for the variables are computed as mentioned in Algorithm 1. */

let $kk1 = 0; i = start$;

while $(i \leq pixels - W + 1)$ **do**

  Fetch each block of size $1 \times W$, one after another from I.

  Compute the MARKER information of each block I_i using the prediction error strategy. /* refer Algorithm 2 */

  $kk1 = kk1 + 1$;

  Save the marker information in a binary array $B1[kk1]$.

  $i = i + W$;

Compute the $Bitspace$ as mentioned in Algorithm 1 for the block scan S.

Generate a binary stream of the PHR data of size $Bitspace$.

Encrypt the image I using the key $K1$ to generate E.

Encrypt the PHR data using the key $K2$.

Extract the A LSBs from each pixel of I starting from location [1 to $(start - 1)$] and save it in an array $EXIST$.

Append the encrypted PHR data to $EXIST$. /* Now the total size of $EXIST$ will be $Bitspace + A \times (start - 1)$. */

let $MARKER = B1$;

Replace the A LSBs of the 1st pixel in E with S.

Replace the A LSBs of the 2nd and 3rd pixels in E with the block size W. /* i.e., the A LSBs from the 2nd pixel and A LSBs from the 3rd pixel should cumulatively give the block size W. */

let $i = 4$; /* Now we scan the pixels from the 4th location of E */

while $(i \leq (kk1/A) + 1)$ **do**

  Replace the A LSBs of each pixel in E with bits from the $MARKER$ array.

  $i = i + 1$;

let $i = start$;

while $(i \leq pixels - W + 1)$ **do**

  /* From here, we access the pixels as blocks of size $1 \times W$ pixels from the location *start* of E. Please note that if the block is embeddable, then the 1st pixel in the block is a reference pixel that is not modified during the embedding process. */

  Replace the A MSBs of each embeddable pixel of the block with bits from the array $EXIST$.

  $i = i + W$;

Output the encrypted and PHR-embedded image E.

Algorithm 3 Embedding the PHR in MI.

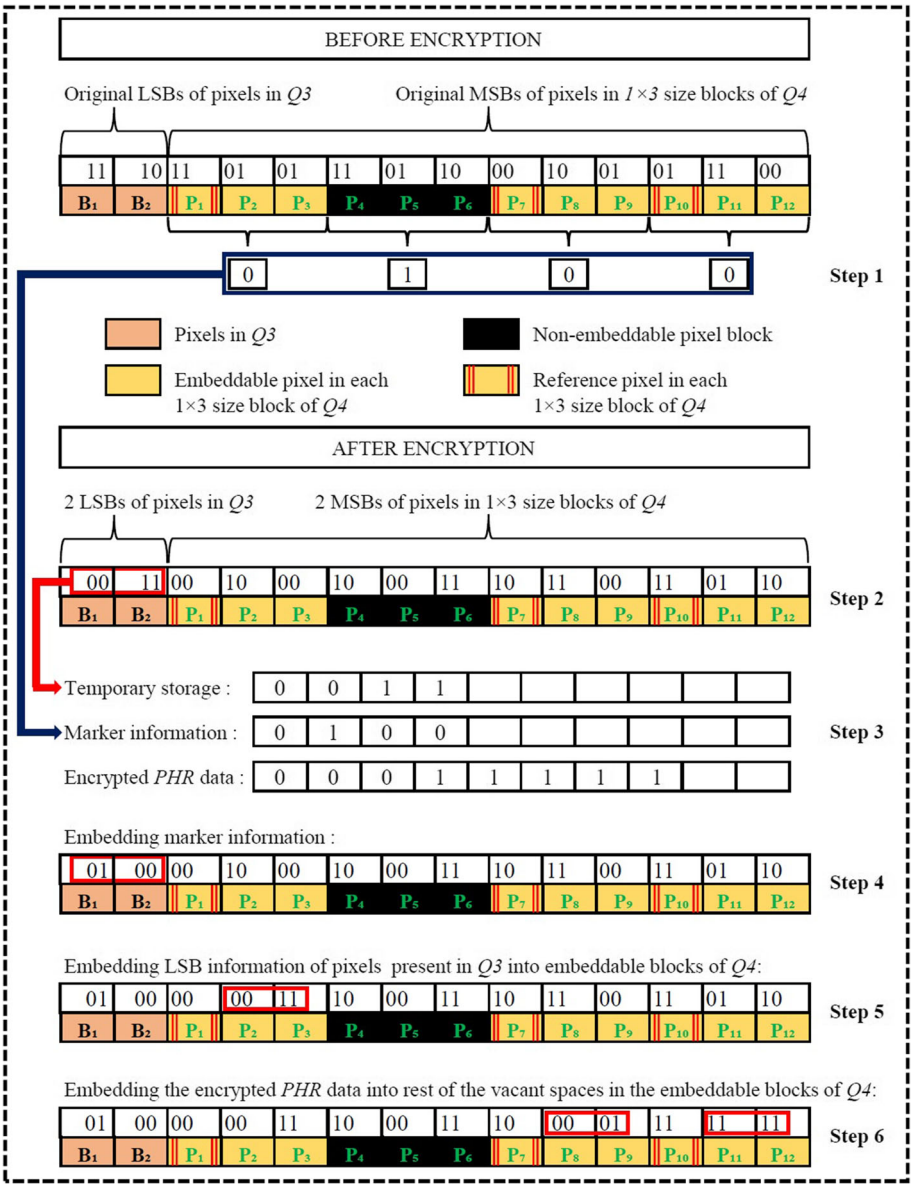

Fig. 6 Hiding PHR data within the encrypted MI

whose values lie in $[0, \sum_{i=0}^{F-1} 2^i]$. Here, F indicates whether the image is an 8-bit or 16-bit image.

S and block size are preserved in the first and the following two pixels via the LSB substitution technique. We also use the LSB substitution for storing the marker information M_X of blocks of $Q4$ in $Q3$. The original PHR data is encrypted using a data encryption key before embedding. Since we are replacing the LSBs of $Q1$, $Q2$, and $Q3$, the original LSBs are extracted before replacement. These extracted original LSBs and the encrypted

PHR data form the whole set of information embedded in the embeddable blocks of $Q4$. Please note that an embeddable block is a block with marker information 0. The PHR is encrypted to secure the data. Hence, flags are not required in the proposed method, as the marker information is preserved in $Q3$. The embedded marker information is used by the receiver, for extracting the hidden data and finally restoring the original pixels.

The pseudo-code for embedding the PHR information in the MI is given in Algorithm 3. The output of Algorithm 1 is a multi-dimensional array $RESULT$. The block size and scanning information for getting a maximum embedding rate on the image I is recorded in $RESULT$. This serves as an input to Algorithm 2, apart from the original image I and the Count of MSB bit planes A. Algorithm 2 details generation of MARKER bit for a block.

2.3.1 Illustration on embedding the encrypted *PHR*

Figure 6 gives an overview of the steps involved in embedding the encrypted PHR information with MI as the cover medium. In the pre-processing stage, the cover image gets scanned from the 4^{th} pixel in dual ways, as depicted in Fig. 5. The block sizes with corresponding embedding rates will also get determined during the pre-processing stage. Once the block size gets fixed, we can classify the pixels from 4^{th} pixel of the MI into $Q3$ and $Q4$. $Q3$ contains the number of pixels required to store the marker information generated on $Q4$. Then the encrypted PHR data, along with the auxiliary information gets embedded in $Q4$.

Referring to Fig. 6, the cover image shown is a one-dimensional matrix of 14 pixels scanned via the snake scan approach. For better understanding, we have shown only the 2 parts of the scanned image: $Q3$ and $Q4$. In converse, if we can understand how LSBs of $Q3$ and encrypted PHR data get embedded in $Q4$, it is just an extension to embed $Q2$ and $Q1$ along with $Q3$. Considering the block size as 1×3 pixels, we have $\lfloor 14/3 \rfloor = 4$ blocks in total. 4 blocks can have a maximum of 4 marker information. Let the RDH scheme embeds 2 bpp. Hence, $Q3$ must reserve 2 pixels for embedding 4 marker bits, and the remaining 12 pixels out of 14 are classified under $Q4$. As we embed the marker information via LSB substitution of pixels in $Q3$, the original LSBs of $Q3$ should get preserved. Hence, we add the original LSBs of $Q3$ with encrypted PHR data to make the whole set of information, that needs to be embedded in $Q4$. The embedding process in $Q4$ follows (refer to Fig. 6):

1 Divide the available 12 pixels of $Q4$ into non-overlapping blocks of 1×3 pixels. This partitions the pixels into 4 blocks.

   Using the MSB prediction error strategy (refer to Section 2.2.1), generate the marker information of each block. Marker information of 0 indicates that the pixels in the block are recoverable, hence embeddable. If there is any prediction error, the marker information of the block is set to 1. This marker information generated gets stored in a temporary location for later utilization.
2 The encrypted MI, and the 2 LSBs of each pixel in $Q3$ are stored in another temporary location, as these LSBs will get replaced with the 4 marker bits generated in step 1.
3 Out of the 4 blocks of $Q4$, let 3 blocks be embeddable. Also, the first pixel in each block acts as a reference pixel, which implies that the rest 2 pixels are only available for embedding. As we are embedding 2 bits of information per pixel, we have $3 \times (2 \times 2) = 12$ vacant spaces in $Q4$ for the purpose of embedding data. The first 4 vacant spaces will be used for preserving the 2 LSBs of each pixel of $Q3$. The remaining spaces can be utilized for embedding the encrypted PHR data. Hence, 8 bits of encrypted PHR data gets preserved in a temporary location.

4 The 2 LSBs of each pixel in $Q3$ get replaced with the marker information. (Please note that in certain cases, there may be an extra vacant space in $Q3$ even after embedding the whole marker information. In such cases, the additional bit will be embedded with marker information 1 to ensure correct recovery.)
5 The 2 LSBs of each pixel in $Q3$ that were preserved in step 2, are embedded in the first 4 vacant spaces of $Q4$ via the MSB replacement method. i.e., in the current scenario, the 4 LSBs of pixels of $Q3$ are embedded into MSBs of $P2$ and $P3$ of $Q4$.
6 We hide the encrypted PHR data generated during step 3 in the remaining vacant spaces. i.e., the 8 bits of the encrypted PHR data is embedded by replacing the 2 MSBs of embeddable pixels of block 3 and 4 of $Q4$.

2.4 Extracting the hidden PHR from the encrypted and marked MI I''

The receiver is aware that the proposed scheme uses A MSB planes for embedding. Hence, A bpp gets replaced in each pixel. The LSB of the first pixel of the received Encrypted and Marked MI I'' get extracted into S'. If S' is 0, then the scanning of blocks will follow the "rightward snake scan approach" else, the scanning of blocks will follow the "downward snake scan approach". Decoding the LSBs of the 2^{nd} and 3^{rd} pixel gives the block size W information. The receiver partitions the rest of the pixels into $Q3$ and $Q4$ in the same manner as mentioned in the pre-processing stage. A LSB bits of each of the pixels in $Q3$ are decoded to get the marker information M'_X, where $X \in (R, D)$. Based on the marker information M'_X, all the recoverable blocks of size $1 \times W$ accounted in $Q4$ get scanned, and the A MSBs of all pixels are extracted, excluding the first pixel of the block (the first pixel of the recoverable block is the reference pixel). The first $A \times (Q3 + Q2 + 1)$ MSBs extracted from $Q4$ are used to restore the original ($Q1$, $Q2$, and $Q3$) pixels of I'' by replacing the A LSBs of each pixel in $Q1$, $Q2$, and $Q3$. The rest of the bits extracted from recoverable non-overlapping blocks are the recovered hidden PHR, which is in encrypted form. As the proposed RDH scheme is separable, the receiver can extract all the hidden encrypted PHR information directly but can decode the original information using the data decryption key.

2.5 Restoration of the original MI

Please note that during the extraction of embedded encrypted PHR, the pixels in $Q1$, $Q2$, and $Q3$ locations get restored with the original encrypted pixel values. Therefore, the pixels in $Q1$, $Q2$, and $Q3$ will be the original MI pixels after decryption. Also, the actual pixels in non-embeddable blocks get restored. The only pixels that need to be recovered are the $W - 1$ pixels present in embeddable blocks, excluding the reference pixel. These pixels get recovered using the MSB prediction strategy based on the previously scanned pixel (refer to Fig. 7 and Section 2.2.1). A small illustration of the recovery procedure is given in Fig. 8.

2.5.1 Illustration of the recovery process

The steps shown in Fig. 8 are a continuation of the explanation given in Fig. 6. It is known that the first pixel of the received image gives the scanning information. Also, 2^{nd} and 3^{rd} pixel detail the block size used to process the image. Based on the block size and the specified scanning approach, we will find the number of non-overlapping blocks of size $1 \times W$ pixels from the total pixels available after the 3^{rd} pixel of the received encrypted and marked MI I''. In the example shown in Fig. 8, we have 14 pixels in total to be scanned (note that the first 3 pixels. i.e., $Q1$ and $Q2$ are not shown in the example).

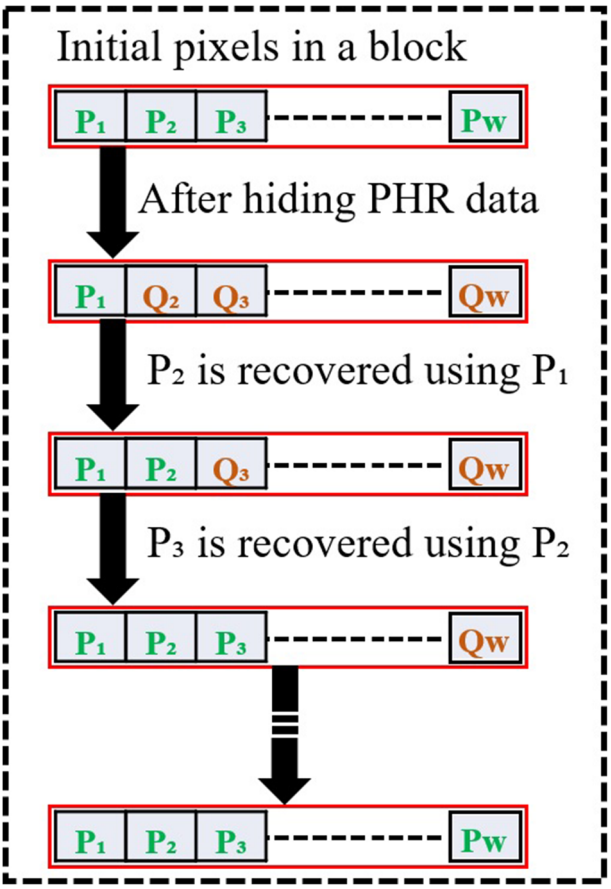

Fig. 7 Recovery of *W-1* pixels in a block

1 The receiver decodes the block size from $Q2$ and processes the image as 1×3 pixel blocks. i.e., the receiver scans the 14 pixels and understands that the pixels can be partitioned into $\lfloor 14/3 \rfloor = 4$ blocks of 1×3 pixels. Hence, there can be a maximum of 4 marker bit information. Also, the receiver knows that 2 bpp is the information stored in each pixel. Thus, the receiver classifies the total 14 pixels into $Q3$ and $Q4$, with $Q3$ being the first 2 pixels expected to hold the 4 marker bits.
2 Decode the 2 LSBs of each pixel in $Q3$ by following the scanning approach decoded via $Q1$, storing it in a temporary location. This gives the marker information.
3 Divide the pixels in $Q4$ into non-overlapping blocks of 1×3 pixels. Also, classify the blocks into embeddable and non-embeddable blocks based on the marker information.
4 Extract the 2 MSBs of each pixel present in the embeddable blocks of $Q4$, excluding the reference pixel. These MSBs get preserved in a temporary location.
5 Since we have decoded 4 LSBs as marker bits from $Q3$, the first 4 bits extracted from $Q4$ in step 4 will replace the existing 4 LSBs of $Q3$.
6 The remaining MSBs extracted during step 4 are the extracted hidden encrypted PHR data.

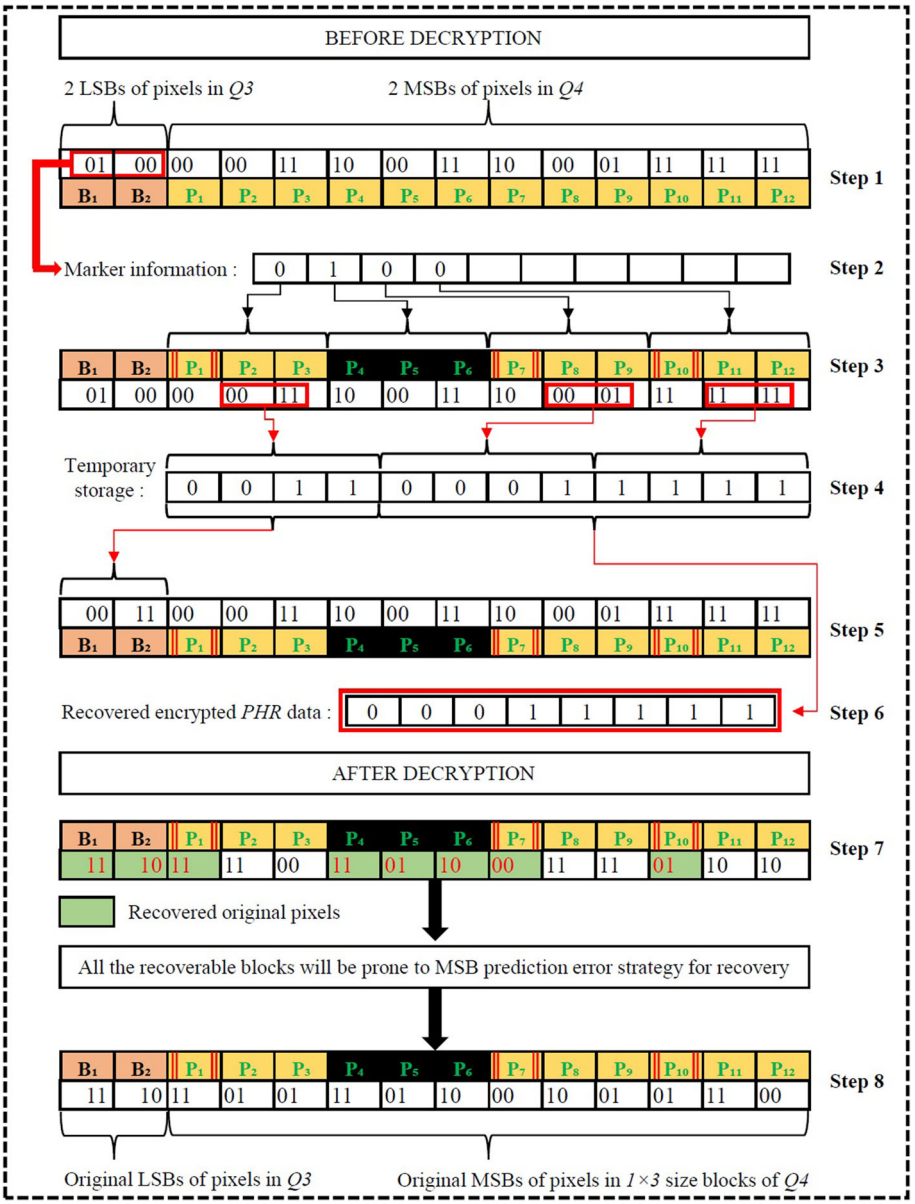

Fig. 8 Recovering of embedded PHR data and original MI

7 The next step is to recover the original cover image - MI. As we can observe, the pixels in $Q3$ have been restored to their actual encrypted value in step 5. All the pixels in non-embeddable blocks and the reference pixels of embeddable blocks are also in their original encrypted state. Hence, after decrypting MI, we restore the mentioned pixels to their original values.

Input: The encrypted and PHR-embedded image E of size $R \times C$ pixels, Count of MSB bit planes A, Key $K1$ and Key $K2$
Output: The recovered image E and extracted PHR information $DPHR$

Extract the A LSBs from the 1st pixel of E. /* Gives scanning direction. Further processing will be carried out by following this scanning direction. */
Extract the A LSBs from each 2nd and 3rd pixel of E. /* They cumulatively form the block size W. */
let $pixels = R \times C$;
let $Total\_pixels = pixels - 3$;
$N = Total\_pixels/W$; /* Compute blocks of size $1 \times W$ */
let $start = \left\lceil \frac{N}{A} \right\rceil + 3 + 1$;
$blocks\_in\_PHR\_area = (pixels - (3 + (N/A)))/W$;
$pixels\_representing\_PHR\_blocks = blocks\_in\_PHR\_area/A$;
let $kk1 = pixels\_representing\_PHR\_blocks + 1$;
let $i = 4$;
while $(i \leq kk1 + 3)$ **do**
    Scan the pixels from 4th location of E one after another.
    Extract the A LSBs from each pixel and place it in the array $MARKER$
    $i = i + 1$;
let $i = start$;
while $(i \leq pixels - W + 1)$ **do**
    Scan the blocks of pixels of size $1 \times W$ one by one.
    Extract the A MSBs from each pixel in the embeddable block using the $MARKER$ array. /* $MARKER$ array denotes whether each block is embeddable or not. If a block is embeddable, then MSBs are extracted from all the pixels excluding the reference pixel. */
    Store the extracted MSBs into an array $EXIST$. /* Note that the array $EXIST$ includes the original LSBs of encrypted pixels in $Q1$, $Q2$, $Q3$ of the image and the PHR data as well. */
    $i = i + W$;
/* Now we split the $EXIST$ into two parts: the original encrypted LSBs and the PHR data. */
Save the bits from location 1 to $(start - 1) \times A)$ of $EXIST$ in an array $DLSB$ and rest of the bits in the array $DPHR$.
let $i = 1$;
while $(i \leq (start - 1))$ **do**
    Scan each pixel one after another of E.
    Replace the A LSBs of each pixel with bits from $DLSB$.
    $i = i + 1$;
Decrypt the image E using the Key $K1$.
Decrypt the data $DPHR$ using the Key $K2$.
let $i = start$;
while $(i \leq pixels - W + 1)$ **do**
    Scan each block of pixels BLK of size $1 \times W$ of E one after another.
    Access the $MARKER$ information for the block.
    if the mark bit from $MARKER$ is 0, then recover the block BLK using Algorithm 5.
    /* Please note that if the mark bit of a block from $MARKER$ is 1, then the block is non-embeddable, and hence they got recovered earlier, after the decryption stage itself. */
    $i = i + W$;
Output the recovered image E and the extracted PHR data $DPHR$.

Algorithm 4 Recovery of the MI and the hidden PHR information.

Input: A block BLK of size $1 \times W$ and Count of MSB bit planes A.
Output: A block BLK of size $1 \times W$ with recovered pixels.
`/* Please note that the 1st pixel in a block is a reference pixel, and hence it is the original unmodified pixel, which need not be recovered. */`
let $i = 1$;
while $(i \leq W)$ **do**
  Scan the pixels in the block BLK from left and access two adjacent pixels at a time.
  let the left pixel be $Pred$ and the right pixel be X.
  Compute different values of X by making different combinations of A MSBs in X. Let the different values of X be X_1, X_2,.., X_n. `/* Here` X_1 `indicates the original value of` X `and` $n = 2^A$. `*/`
  Compute the difference between the $Pred$ and different values of X.
  Replace the pixel X in the block BLK with the new value of X, that gives minimum difference with the $Pred$.
  `/* The above process gets repeated until all the adjacent pixels get encountered for recovery. */`
  $i = i + 1$;
Output the recovered block BLK.

Algorithm 5 Recovering the pixels in a block.

8 The only pixels that need to be recovered are the pixels present in the embeddable blocks, except their reference pixel. The pixels in embeddable blocks are hence predicted using our MSB prediction error strategy. Thus, the whole cover image MI is restored in its original form.

The pseudo-code for restoring the original MI and the PHR information is given in Algorithm 4, which requires recovering pixels of embeddable blocks through the MSB prediction strategy. The pixels recovery through MSB prediction strategy is given in Algorithm 5.

3 Results of experiments

This section discloses the experimental results carried out to understand the efficacy of the proposed work. Simulation of the work is done with Matlab R2019a. The system configuration includes a 64-bit operating system, installed RAM of 8.00 GB, and Intel Core i5-9300H processor @ 2.40GHz. All the images used in the experiments were initially converted into a standard size of 512×512 pixels before further processing.

3.1 Data-set description

The experimental study of the proposed RDH scheme in encrypted images is conducted on 8-bit natural images and 16-bit DICOM images. The high-quality DICOM image database used for the experimental purpose is downloaded from the DICOM library [10] and the CT images [3] from the cancer imaging archive [9]. The natural image data

set used for the experimental purpose is taken from the USC-SIPI image database [40]. The table that describes different images present in the USC-SIPI image set is given in Table 1.

The DICOM medical images include 100 CT images from the cancer imaging archive and 497 images from [10], which consists of different modalities. Table 2 shows the different modalities of DICOM images considered in our experiment. The original images whose results are highlighted explicitly in this experimental study are shown in Fig. 9.

3.2 Influence of snake scan approach

We have observed that the way pixels of an image get scanned impacts the embedding capacity. There is a drastic change in the pixel intensities when we go in one direction, which may be the opposite in the other case. Our MSB prediction error strategy works on neighboring pixels correlated to each other, which adds to the embedding capacity. To utilize the closely related pixels to the maximum extent possible, we have devised the scanning of pixels in an image to follow the snake scan framework (refer to Fig. 1). Table 3 gives an idea of variation in the embedding rate while processing the image with different block sizes and with 2 ways of snake scan framework. It is observable that the *boat* image has attained its maximum embedding capacity of 1.3744 bpp when the image follows a downward snake scan approach with a processing block size of 1×13 pixels.

A few factors that led to the selection of a snake scan framework for the proposed model are:

- The proposed scheme standardizes images into 512 pixels before processing. The images are processed as blocks of size $1 \times W$ pixels. But, we should note that the size "W" with which each natural image is processed differs. Hence, the block size varies, and it may not be possible to divide the 512 pixels in a row exactly. Moreover, the initial pixels of the image have been used to preserve the details of scan direction and block size.
- Now, as we can't divide the 512 pixels in a row exactly, with W pixels, there is a chance that a few of the pixels at the end of the row may fall into a new block of size $1 \times W$ pixels. Hence, we should include a few more pixels from the immediate next row to complete the block size requirement. i.e., suppose we have scanned the first row, and at the end, let the last few pixels fall within a new block. To complete the block size with W pixels, a few more pixels from the immediate second row are needed. Say some pixels from the beginning of the second row are chosen to fill the block. The probability that these pixels are correlated to the last pixels present at the end of the first row is less. But if we choose pixels from the end of the second row, then the probability that they are correlated to the other pixels from the first row is very high. This certainly helps our prediction strategy.

Table 1 USC-SIPI data-set features

Sl. No.	Category / Features	Number of images
1	Aerials	38
2	Sequences	69
3	Textures	64
4	Miscellaneous	39
Total images:		210

Table 2 Modalities of DICOM images

Sl. No.	Modalities / Features	Number of images
1	CT (computed tomography)	461
2	MRIs (magnetic resonance imaging)	135
3	OT (others)	1
Total images:		597

3.3 Basis for the selection of different block sizes

The following points act as the basis for the selection of different block sizes in the experimental setup:

- It is possible to use a smaller block size for embedding the secret data. But as we go for smaller block sizes, for example, in the case of a 1×2 block, only half of the pixels in each block are used for embedding, and the other half is used as reference pixels for prediction. Hence, with smaller block sizes, we can't utilize the maximum number of pixels present in the image for embedding the data.
- The other possibility is to use higher block sizes. In the case of higher block sizes, only one pixel in each block gets reserved as a reference pixel, and the rest of the pixels can be utilized for embedding the data. Hence, It is always better to go with higher block sizes for getting the best embedding rate. But we must guarantee that all the pixels in the block are predictable using the reference pixel. Say, if any one of the pixels is not predictable, then the whole block is not embeddable, and the block remains unutilized. This impacts the embedding rate to a greater extent. If all the pixels in an image are highly correlated, then high block size beats the rest. But this won't be the case practically.

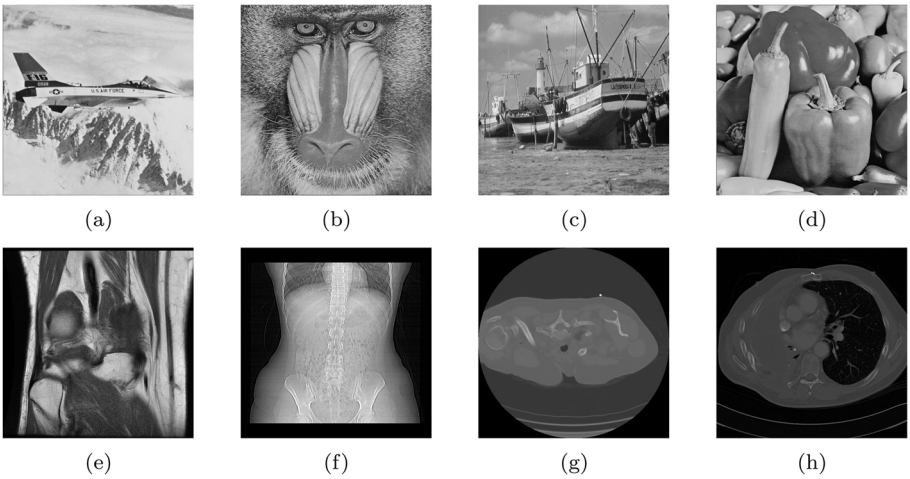

Fig. 9 Natural images: (a) Airplane, (b) Baboon, (c) Boat, (d) Peppers. DICOM images: (e) Medical Image-1, (f) Medical Image-2, (g) Medical Image-3, (h) Medical Image-4

Table 3 Computing optimal embedding rate on *boat* image

Sl. No.	Block size	Snake scan (rightward direction)	Snake scan (downward direction)
1	1×5	1.0268	1.1246
2	1×6	1.1033	1.2161
3	1×7	1.1471	1.2743
4	1×8	1.1758	1.3160
5	1×9	1.1907	1.3393
6	1×10	1.1999	1.3564
7	1×11	1.2035	1.3648
8	1×12	1.2051	1.3689
9	1×13	1.1997	1.3744
10	1×14	1.1931	1.3721
11	1×15	1.1870	1.3699
12	1×16	1.1839	1.3671
13	1×17	1.1724	1.3614
14	1×18	1.1596	1.3524
15	1×19	1.1523	1.3463

Hence, the experimental study is carried out over the block sizes (1×5, 1×6, ..., 1×19) to investigate the best trade-off between the prediction error rate and the embedding rate.

3.4 Embedding potential

The embedding potential of a scheme indicates how much information can be loaded into a cover image. It is measured in terms of bits per pixel (bpp). Figure 10 shows how the embedding rate of each image varies from an initial lower value to a higher value and then starts decreasing. The explicit results of experimenting on 4 natural images and 4 high-quality DICOM images are detailed in Tables 4 and 5. We have observed that the maximum embedding rate of a natural image lies within a block size 1×5 to 1×19. Specifically, the *Airplane* image attained its peak embedding rate at a block size of 1×16 pixels, *Baboon* image at 1×7 pixels, *Boat* image at 1×13 pixels, and *Peppers* image at 1×16 pixels. In 16-bit high-quality DICOM medical images, the block size is chosen as 511 empirically.

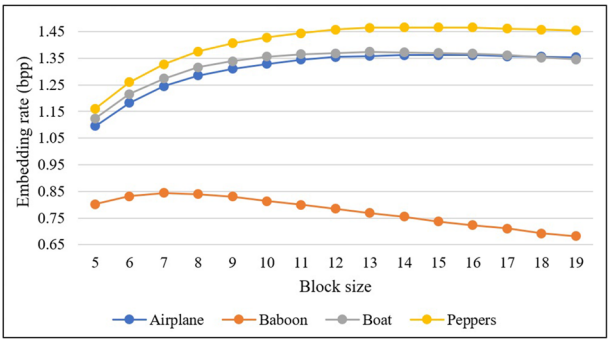

Fig. 10 Variation in embedding rate of different images

Table 4 Capacity of embedding and Bit Error Rate on Natural images

Name of the image	Block size	Embedded bits	Embedding rate (bpp)	BER
Airplane	1×16	357112	1.362274	0
Baboon	1×7	221338	0.844337	0
Boat	1×13	360278	1.374352	0
Peppers	1×16	384352	1.466187	0

The highest achieved embedding rate on a DICOM image is 3.9824 bpp. Based on the way pixels of an image are scanned, the $Airplane$ and $Baboon$ gained their peak embedding rate, when they followed the rightward snake scan framework, while the $Boat$ and $Peppers$ gained their maximum capacity through the downward snake scan framework. Similarly, out of the 4 medical images $Image-1$ and $Image-4$ acquired their maximum embedding capacity by scanning the pixels via the rightward snake scan framework, whereas the other two gained their maximum capacity via the downward snake scan framework.

We have experimented with all 497 DICOM images from the DICOM library [10] and 100 images from the cancer imaging archive. The graph in Fig. 11 shows an outline of the payload on the 497 images. The lowest embedding found is 3.3987 bpp, and the highest is 3.9824 bpp. We could achieve a 3.9762 bpp embedding rate on average, which is much better while handling medical images. Figure 12 shows the payload characteristic on the 100 CT medical images from the cancer imaging archive with an average embedding rate of 3.9784 bpp.

The $USC-SIPI$ image database is a collection of 210 natural images with 4 groups: $Arials$, $Sequences$, $Textures$ and $Misc$. The proposed scheme embeds 2 bpp in an 8-bit natural image. The graph in Fig. 13 shows the variation of embedding rate achieved on different groups of the 210 images. The rate of embedding was low in highly textured images, and 1.3496 bpp is the average embedding rate achieved.

3.5 Analysis on the amount of bit errors

Regeneration of the secret information embedded in a cover medium is pivotal in any RDH scheme. Hence, methods to improve the payload capacity should not affect the bit error rate (BER). The proposed RDH scheme is very effective in handling high-quality DICOM medical images as the cover medium. Experimenting on the mentioned high-quality medical image databases totaling 597 images, we could embed a maximum of 3.9824 bpp and a minimum of 3.3987 bpp PHR data without any bit errors at the receiver side. Refer to Tables 4 and 5 for the BER incurred at the receiver side. The value 0 implies that the proposed scheme works well on all images extracting entire hidden information.

Table 5 Capacity of embedding and Bit error rate on DICOM images

Sl. No.	Name of the image	Embedded bits	Embedding rate (bpp)	BER
1	Medical Image-1	962360	3.671112	0
2	Medical Image-2	1043960	3.982391	0
3	Medical Image-3	1039880	3.966827	0
4	Medical Image-4	1041920	3.974609	0

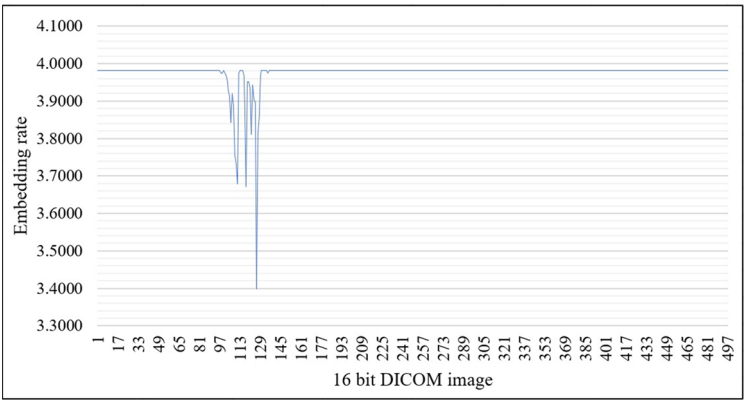

Fig. 11 Embedding rate calculated on 497 sample DICOM images

3.6 Quality assessment of the recovered image

We have assessed the quality of the recovered images with reference to the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). PSNR analyzes the quality of the recovered image, prioritizing the noise factor. It is defined as the ratio of maximum possible value MAX_VAL of the pixels in an image to the mean square error (MSE), mathematically expressed as:

$$10 \log_{10} \left(\frac{MAX\_VAL^2}{MSE} \right) \text{ dB} \tag{14}$$

MSE calculates the mean square of differences between the corresponding pixel values of two analogous images. When the MSE tends to 0, PSNR will tend to ∞. SSIM estimates the quality of the recovered image based on structural dissimilarity. The loss in structural information is quantified, which relies on components like luminance, contrast, etc. The

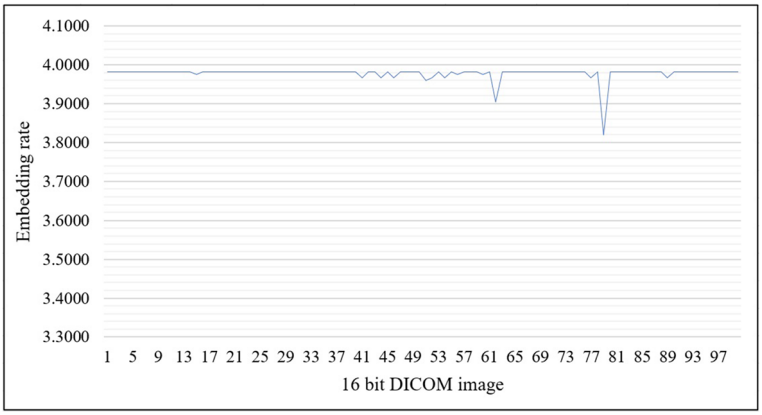

Fig. 12 Embedding rate on 100 CT images from cancer imaging archive

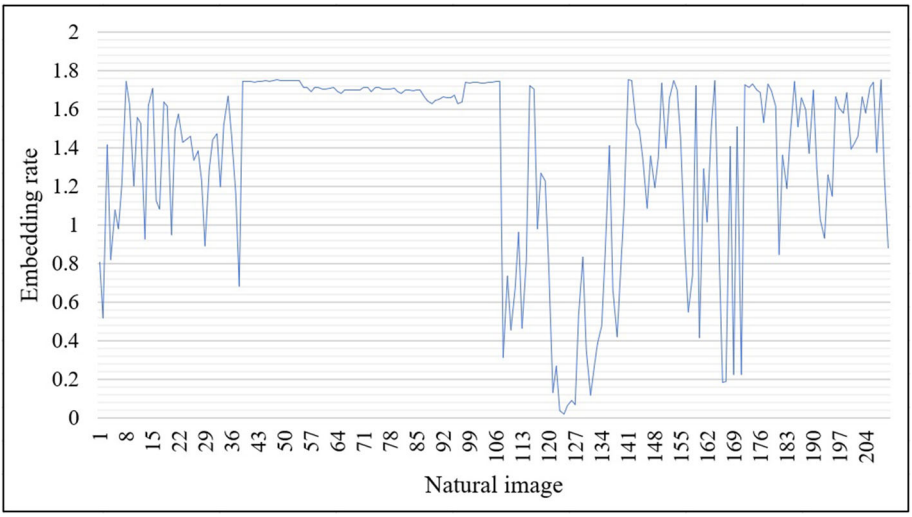

Fig. 13 Embedding rate calculated on 210 sample natural images

value of SSIM will be 1 when there is no loss in the structural information between the original and the recovered image. Table 6 shows the PSNR and SSIM outcomes resulting from experimenting with the 8 images.

3.7 Analyzing randomness property after embedding PHR data

The level of disorderliness of pixels in an image is a measure of security. Extremely randomized neighboring pixels, hide the essential details from external perception. This is an essential property of an encrypted image. The proposed work is an RDH in an encrypted image. Hence, the pixel values of the encrypted image get modified during the process of hiding the PHR data. An encrypted image retains the property of randomness. Hence, the proposed scheme should also manage to maintain the randomness for securing the image features or characteristics. Entropy is a quantity that measures the randomness of pixels in an image. Table 7 reveals the entropy details acquired from the 8 images. It conveys that the entropy of the encrypted image and the entropy calculated after hiding the PHR data

Table 6 Image quality estimates

Sl. No.	Name of the image	PSNR (dB)	SSIM
1	Airplane	∞	1
2	Baboon	∞	1
3	Boat	∞	1
4	Peppers	∞	1
5	Medical Image-1	∞	1
6	Medical Image-2	∞	1
7	Medical Image-3	∞	1
8	Medical Image-4	∞	1

Table 7 Entropy estimation

Name of the image	Entropy of the original image	Entropy after encrypting the image	Entropy after hiding HR data	Entropy of the recovered image	Correlation coefficient C ($I1, I2$)
Airplane	6.70246	7.99925	7.99862	6.70246	1
Baboon	7.35834	7.99936	7.99831	7.35834	1
Boat	7.19137	7.99928	7.99821	7.19137	1
Peppers	7.59365	7.99917	7.99849	7.59365	1
Medical Image-1	12.00740	15.80749	15.80707	12.00740	1
Medical Image-2	6.40019	15.8084	15.80835	6.40019	1
Medical Image-3	7.66034	15.80827	15.80789	7.66034	1
Medical Image-4	9.39531	15.80982	15.80783	9.39531	1

are nearly the same. Hence, the proposed scheme manages to maintain a high entropy level even after embedding the PHR data.

3.8 Analyzing correlation coefficients on the PHR data

The strength of the correlation between two variables can be assessed by measuring the correlation coefficients. In other words, the correlation coefficient is a measure of the linear relationship between the variables. The formula for calculating the correlation coefficient is:

$$C(I1, I2) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{I1_i - M_{I1}}{S_{I1}} \right) \left(\frac{I2_i - M_{I2}}{S_{I2}} \right). \tag{15}$$

Here, $I1$ and $I2$ are the two variables, N indicates the number of observations, $M_{I1}\&M_{I2}$ indicates the mean value and $S_{I1}\&S_{I2}$ indicates the standard deviation. Alternatively, the correlation coefficient is the ratio of the covariance of the variables to the product of their standard deviation. A correlation coefficient value of 1 indicates a strong positive correlation, a value of 0 indicates no correlation, and a value of -1 implies that the two variables are having a strong negative correlation.

We have computed the correlation between the embedded PHR data and the extracted PHR data. We have observed that the information embedded and the information extracted are strongly correlated. The values of correlation coefficients of the information over the different images are given in Table 7. The similarity between the original cover image, the encrypted image, and the marked image is assessed through entropy and histogram analysis.

3.9 Analysis on distribution of pixel intensities

An inspection of the distribution of pixel intensities has a huge effect on letting out the image details. If we could observe the distribution of pixel intensities on an encrypted image, they always follow an unwavering and steady pattern. Hence, the gray level distribution needs to be uniform to secure the characteristics of an image. We have analyzed the same by finding the histogram of images after encryption and after hiding the encrypted PHR data. As shown in Fig. 14, the proposed method retains a homogeneous distribution of pixel

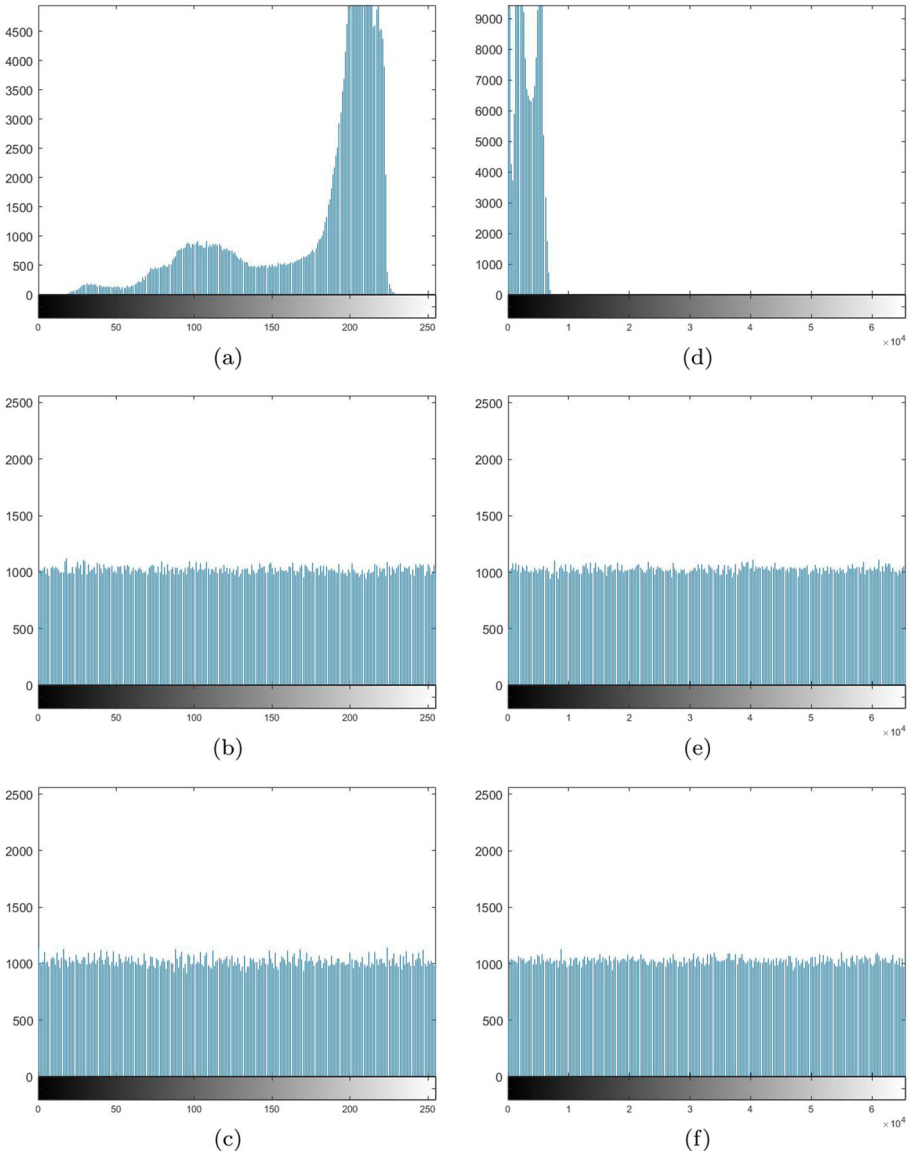

Fig. 14 Histogram results on $Airplane$ image (a, b & c) and $Image-1$ DICOM image (d, e & f) [Figure (a), (d): Original image histogram, Figure (b), (e): histogram after encrypting the image, Figure (c), (f): histogram after hiding the additional information]

intensities, despite modifying the encrypted pixel values during the course of embedding the additional information.

3.10 Experimental time analysis

This section briefs the overall experimental time analysis. The execution time taken for embedding the additional data and the recovery process, which includes recovery of the

Table 8 Average execution time on natural images

Image category	Total images	Embedding time (in seconds)	Recovery time (in seconds)
Aerials	38	1.6478	1.5382
Textures	64	1.1438	1.4298
Sequences	69	1.7908	1.7819
Misc	39	1.4963	1.7704

cover image and extraction of the hidden information is computed on all images. The average embedding time and the average recovery time on images under different categories of the $USC-SIPI$ image data set are given in Table 8. The average time taken for embedding the secret data over the 210 images of $USC-SIPI$ image data set is 1.5131 seconds, and that of recovery is 1.6284 seconds.

The experimental time computed on 20 randomly selected medical images is shown in Fig. 15. As can be observed from the graph, the time consumed for embedding the PHR data is below 1 second, averaging 0.0943 seconds on testing over the 597 medical images, and the average recovery time taken is 4.9371 seconds. Please note that the computation time taken in the recovery process is higher, and this is a cause of the recovery mechanism that we have employed. The recovery process involves identifying the partitions $Q3$, $Q4$, recovering the pixels in $Q3$ through multi LSBs replacement, and predicting the pixels in $Q4$ through multi MSBs prediction error technique (Refer Section 2). However, in practical applications, the impact of this increased computation time won't be so serious. Also, it is noteworthy that a higher embedding rate is achieved as a result.

3.11 Theoretical time analysis

In this section, we discuss the theoretical time complexity analysis of the proposed scheme. This includes the analysis of the embedding process and the analysis of data extraction and image recovery.

3.11.1 Embedding time analysis

The embedding process in the proposed scheme includes two stages: pre-processing stage and the data embedding stage. In the pre-processing stage, we identify the suitable block

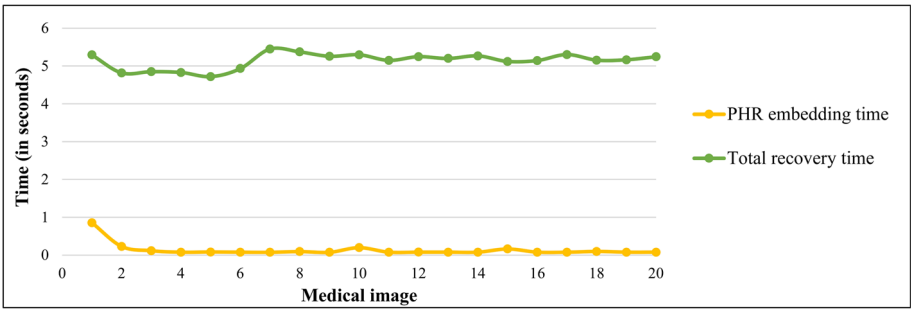

Fig. 15 Execution time computed on 20 medical images

size for achieving the maximum embedding rate. During this phase, the cover image is processed in dual snake scan directions (horizontal snake scan and vertical snake scan) by considering various block sizes. The computation in one direction involves calculating the marker bit length, splitting the image into 4 parts $\{Q1, Q2, Q3, Q4\}$ based on the computed marker bit length, and finally estimation of the embedding capacity from $Q4$ after reserving length for preserving the LSBs of $Q1$, $Q2$, and $Q3$. Since the given image of size $R \times C$ pixels is processed by considering non-overlapping blocks of fixed side-length $1 \times B$ pixels at a time, the theoretical time complexity to compute the embedding rate is $O(N)$, where $N = R.C$.

In the data embedding stage, in addition to the steps followed in the pre-processing stage, we preserve the LSBs of pixels of $Q1$, $Q2$, $Q3$, and appending the additional data to it, will take a constant amount of time. Encrypting the image and replacing the LSBs of $Q1$, $Q2$, and $Q3$ with scan direction, block size, and marker bits will account for a constant time. Finally, we embed the preserved LSBs and additional data into the MSBs of pixels of $Q4$ using the marker bits, which again takes a constant time. Hence, the overall theoretical time complexity of the proposed reversible data hiding process is $O(N)$, where $N = R.C$.

3.11.2 Recovery time analysis

The recovery process of the proposed scheme involves the extraction of the hidden information and the recovery of the cover image. Initially, we extract the scan direction and the block size from the initial pixels of $Q1$ and $Q2$ respectively. The rest of the pixels are divided into non-overlapping blocks based on the block size information extracted from $Q2$. Further, we calculate the size of $Q3$, and the LSBs are extracted, which are the marker information required for further processing. All these processes will take a constant time. The embedded information (PHR and LSBs for cover image restoration) will be extracted from the pixels of $Q4$ by processing it block-wise by considering the marker information. The extracted LSB information will be used to replace the LSBs of $Q1$, $Q2$, and $Q3$. This process will have a time complexity of $O(N)$.

Further, the image should be decrypted, and the original image pixels should be restored through the MSB prediction error strategy. Since we need to process all the pixels of the image, and the processing will take constant time, the overall time complexity of the data extraction and image recovery process is $O(N)$ where N is the total number of pixels in the image ($N = R.C$, when the image size is $R \times C$ pixels).

3.12 Memory consumption

The memory consumed while executing the proposed scheme on $Airplane$, $Baboon$, $Boat$, and $Peppers$ images are shown in Fig. 16. It should be noted that the simulation of the proposed scheme is done in MATLAB. The details of the system configuration are given at the beginning of the Results of Experiments section. Figure 16 shows the memory consumed during the embedding and the recovery process separately.

4 Comparison with other RDH schemes

We have improved the scheme proposed in [36], where they have achieved the highest embedding capacity of 1 bpp, which couldn't guarantee full reversibility. They have also discussed a modified approach, where flags are used to make it fully reversible, wherein

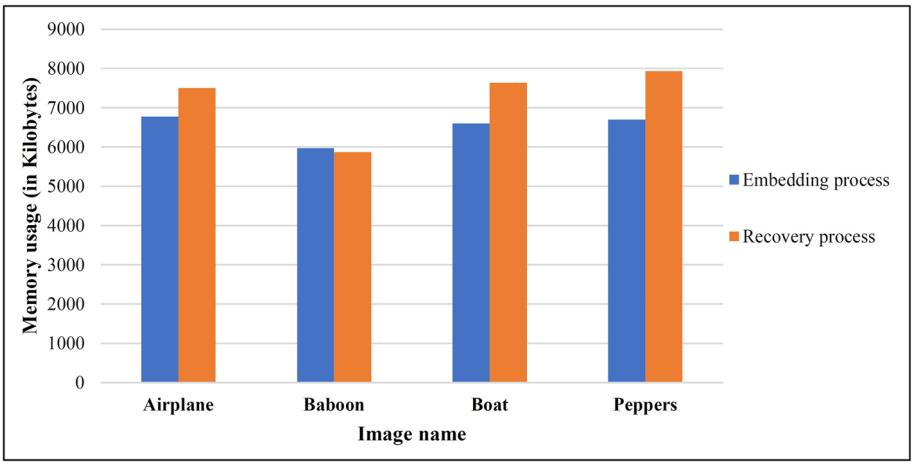

Fig. 16 Memory used by the proposed scheme

the embedding rate came down a little. Our scheme tries to enhance the scheme which is perfectly reversible, with a maximum embedding capacity of 1.7538 bpp and an average embedding capacity of 1.3496 bpp on natural images. In their approach, a single MSB replacement technique is adopted for embedding secret data, and the pixels in an image are processed as a fixed block size of 1×8 pixels to make maximum information embedded. In our RDH scheme on the natural image, dual MSBs from predictable blocks are utilized to embed the additional information. Also, we have introduced a new scanning framework called snake scan in two directions, which along with optimal block size selection (not restricted to 1×8 pixel blocks) has elevated the payload capacity. We have used a few images from the natural image set to compare the embedding capacity of our scheme with other similar schemes. Table 9 shows that our scheme performs much better when compared to most of the recent RDH schemes in encrypted images.

A comparison of the methodology used by various similar RDH schemes and the proposed scheme is shown in Table 10. All the schemes that have been compared, are the schemes where the extraction of the embedded additional information is separated from the image recovery except [33] and [35]. The bit error generated during the extraction is 0 in all the schemes. There are two approaches disclosed in [36]. One approach is the CPE approach, where they adopted correcting the prediction error generated by modifying the pixels such that the regeneration will lead to a pixel value that is much closer to the original pixel in the worst case. Hence, there will be errors while recovering the cover image. Whereas the EPE approach overcomes the prediction error in each block by implementing flags giving the scheme the capability to completely recover without any loss. Similarly, the proposed scheme and others are completely reversible without any recovery loss. There are schemes that do not reserve room prior to the embedding of the additional information like [33, 35, 37, 43, 44], while others, unlike the proposed scheme reserves room. The visual traits of the cover image after direct decryption is good in [43] and [36]. This is not the case with others, unlike [44], the proposed scheme generates an error map that is self-embedded after encryption to support recovery.

Figure 17 shows the intermediate outcomes resulting from experimenting with the DICOM images. A comparison using the medical image as the cover medium between the

Table 9 Results of embedding rate on different images with reference to various schemes

Scheme	Airplane	Baboon	Boat	Peppers
Method used in [1]	0.004	0.004	0.004	0.004
Method used in [23]	0.008	0.008	0.008	0.008
Method used in [45]	0.040	0.040	0.040	0.040
Method used in [35]	0.031	0.031	0.031	0.031
Method used in [33]	0.055	0.055	0.055	0.055
Method used in [43]	0.265	0.047	0.145	0.213
Method used in [30]	0.279	0.034	0.104	0.158
Method used in [44]	0.426	0.365	-	0.425
Method used in [27]	0.483	0.483	0.483	0.483
Method used in [14]	0.764	0.124	0.560	0.434
Method used in [36]	0.962	0.838	0.962	> 0.920
Method used in [18]	0.992	0.884	0.985	0.990
Method used in [41]	2.023	-	1.286	1.511
Proposed scheme	**1.362**	**0.844**	**1.374**	**1.466**

proposed scheme with [4, 8, 36, 42, 50] and [5] is given in the Table 11. The k in [5] indicates embedding data on a base k numeral framework. As we can observe, the proposed method can embed more PHR information than any other method. This is because of the fact that most of the pixels in the medical images are black, and only 12 bits store valid information in a 16-bit DICOM image. Hence, we exploited these facts quite well to establish a good trade-off between the embedding rate and the quality of the recovered MI, giving it a competitive edge over the other related RDH schemes.

5 Conclusion

Medical image-based RDH schemes from the research community mostly implement and discuss their results concerning 8-bit medical images, though the widely used medical

Table 10 Comparison based on the type of methodology employed

Methodology used	Separable scheme	Bit error during recovery	Error during image recovery	Room reserving
Method in [37]	YES	NO	NO	NO
Method in [35]	NO	NO	NO	NO
Method in [6]	YES	NO	NO	YES
Method in [50]	YES	NO	NO	YES
Method in [43]	YES	NO	NO	NO
Method in [36]-CPE	YES	NO	YES	YES
Method in [36]-EPE	YES	NO	NO	YES
Method in [33]	NO	NO	NO	NO
Method in [44]	YES	NO	NO	NO
Proposed scheme	YES	NO	NO	YES

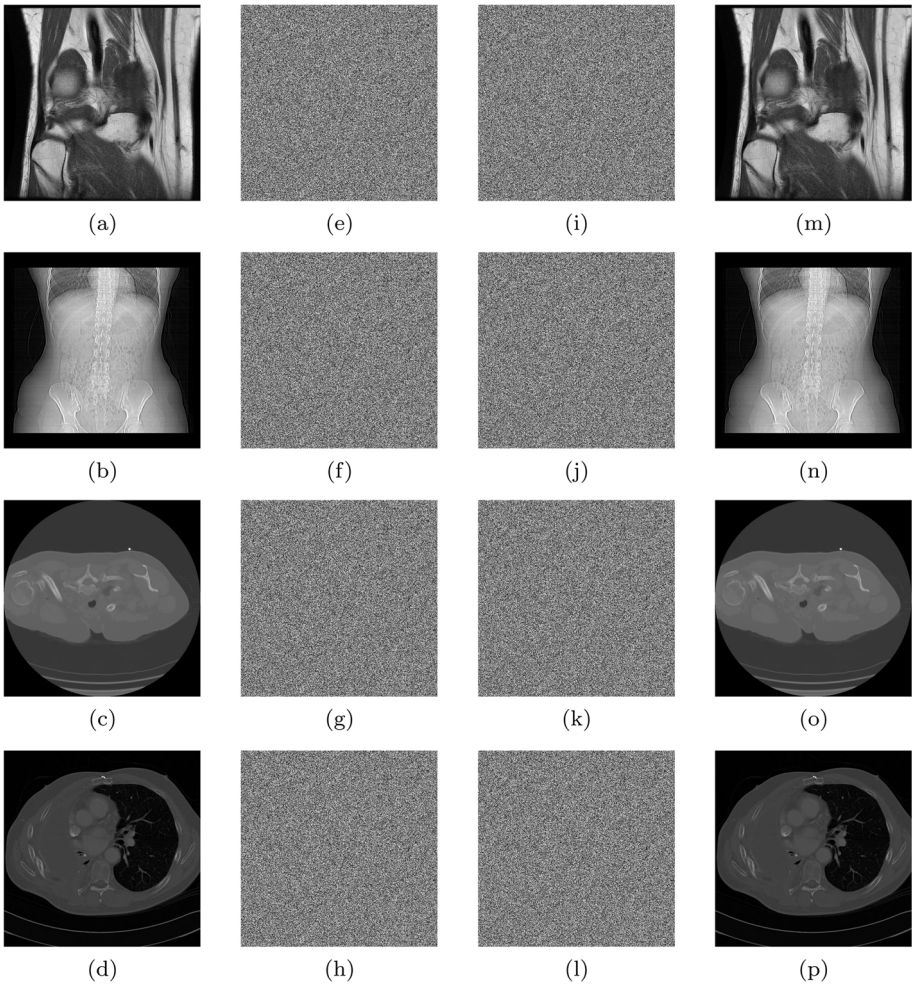

Fig. 17 Sample results on medical images. [Figure (a), (b), (c), (d): Original images, Figure (e), (f), (g), (h): corresponding encrypted images, Figure (i), (j), (k), (l): encrypted and data embedded images, Figure (m), (n), (o), (p): corresponding recovered images]

images are of high quality in nature. The proposed scheme is fully compatible with the 16-bit DICOM images. We have discussed the RDH scheme on natural images as well. The average payload capacity on high-quality DICOM images is 3.9766 bpp, and that of natural images is 1.3496 bpp. The proposed scheme uses marker information that is self-embedded to support recovery. Though the proposed RDH scheme has an overhead of preserving the marker information, the scheme better utilizes most of the MSBs of each pixel to gain a high payload. The scheme is fully reversible and error-free with $BER = 0$, $PSNR = \infty$ dB, and $SSIM = 1$. Also, the proposed scheme brings in the pixel scanning framework that has a good impact on achieving the best trade-off between the embedding capacity and the quality of the recovered image or between the embedding capacity and the prediction error. The work can be categorized under separable RDH schemes where the receiver can decode the

Table 11 Comparison with different schemes on medical images

Scheme	Image	Parameters		
		EI	ER PSNR	
Chen et al. [8]	Medical Image-1	1,31,072	0.50	∞
	Medical Image-2	1,31,072	0.50	∞
	Medical Image-3	1,31,072	0.50	∞
	Medical Image-4	1,31,072	0.50	∞
Wu et al. [42]	Medical Image-1	2,62,144	1	∞
	Medical Image-2	2,62,144	1	∞
	Medical Image-3	2,62,144	1	∞
	Medical Image-4	2,62,144	1	∞
Zheng et al. [50]	Medical Image-1	2,62,144	1	∞
	Medical Image-2	2,62,144	1	∞
	Medical Image-3	2,62,144	1	∞
	Medical Image-4	2,62,144	1	∞
Puteaux and Puech [36]	Medical Image-1	2,62,144	1	∞
	Medical Image-2	2,62,144	1	∞
	Medical Image-3	2,62,144	1	∞
	Medical Image-4	2,62,144	1	∞
Anushiadevi et al. [4]	Medical Image-1	2,62,144	1	∞
	Medical Image-2	2,62,144	1	∞
	Medical Image-3	2,62,144	1	∞
	Medical Image-4	2,62,144	1	∞
Bhardwaj [5] k=3	Medical Image-1	7,86,432	3	∞
	Medical Image-2	7,86,432	3	∞
	Medical Image-3	7,86,432	3	∞
	Medical Image-4	7,86,432	3	∞
Proposed scheme	**Medical Image-1**	**9,62,360**	**3.67**	∞
	Medical Image-2	**10,43,960**	**3.98**	∞
	Medical Image-3	**10,39,880**	**3.97**	∞
	Medical Image-4	**10,41,920**	**3.97**	∞

embedded PHR data without knowing the image decryption key but requires a data decryption key. The receiver requires the image decryption key to recover the original cover MI. One future direction is to consider an improved MSB prediction mechanism that can give a competitive edge in dealing with the changes in the textured surfaces of the images. In the current manuscript, we have considered the communication channel to be noiseless and error-free. Practically, this won't be the case. Hence, another future direction is to make the scheme robust over the noisy channel. Also, it would be interesting to see different other scanning approaches adapting to the requirements of the RDH as a future scope.

Author Contributions Mr. Shaiju Panchikkil contributed to the design, and implementation of the algorithm and preparation of the manuscript.

Dr. V. M. Manikandan contributed to the critical evaluation and proofreading of the manuscript.

Data Availability The datasets generated during and/or analysed during the current study are available in the following repositories:

1. USC-SIPI: https://sipi.usc.edu/database/
2. DICOM Library: https://www.dicomlibrary.com/
3. CT images from cancer imaging archive: https://www.kaggle.com/datasets/kmader/siim-medical-images?select=dicom_dir/

Declarations

Informed consent and permission The experiments for the proposed work have been carried out on standard image datasets available publicly.
Competing interests The authors declare that they have no conflict of interest.

References

1. Agrawal S, Kumar M (2017) Mean value based reversible data hiding in encrypted images. Optik 130:922–934
2. Al-Haj A, Abdel-Nabi H (2021) An efficient watermarking algorithm for medical images. Multimed Tools Appl 80(17):26021–26047
3. Albertina B, Watson M, Holback C, Jarosz R, Kirk S, Lee Y, Lemmerman J (2016) Radiology data from the cancer genome atlas lung adenocarcinoma [tcga-luad] collection. The Cancer Imaging Archive
4. Anushiadevi R, Praveenkumar P, Rayappan JBB, Amirtharajan R (2021) Uncover the cover to recover the hidden secret-a separable reversible data hiding framework. Multimed Tools Appl 80(13):19695–19714
5. Bhardwaj R (2021) An enhanced reversible patient data hiding algorithm for e-healthcare. Biomed Signal Process Control 64:102–276
6. Cao X, Du L, Wei X, Meng D, Guo X (2015) High capacity reversible data hiding in encrypted images by patch-level sparse representation. IEEE Trans Cybern 46(5):1132–1143
7. Chen K, Chang CC (2019) Error-free separable reversible data hiding in encrypted images using linear regression and prediction error map. Multimed Tools Appl 78(22):31441–31465
8. Chen YC, Shiu CW, Horng G (2014) Encrypted signal-based reversible data hiding with public key cryptosystem. J Vis Commun Image Rep 25(5):1164–1170
9. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M et al (2013) The cancer imaging archive (tcia): maintaining and operating a public information repository. J Digit Imaging 26(6):1045–1057
10. DICOM (2022) Accessed 01 Apr 2022 image database, URL: https://www.dicomlibrary.com/
11. Dragoi IC, Coltuc D (2014) Local-prediction-based difference expansion reversible watermarking. IEEE Trans Image Process 23(4):1779–1790
12. Dragoi IC, Coltuc D (2016) Adaptive pairing reversible watermarking. IEEE Trans Image Process 25(5):2420–2422
13. Dragoi IC, Coltuc D (2020) On the security of reversible data hiding in encrypted images by msb prediction. IEEE Trans Inf Forensics Secur 16:187–189
14. Ge H, Chen Y, Qian Z, Wang J (2018) A high capacity multi-level approach for reversible data hiding in encrypted images. IEEE Trans Circuits Syst Video Technol 29(8):2285–2295
15. Hassan FS, Gutub A (2021) Efficient image reversible data hiding technique based on interpolation optimization. Arab J Sci Eng 46(9):8441–8456
16. He W, Xiong G, Weng S, Cai Z, Wang Y (2018) Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion. Inf Sci 467:784–799
17. Hou J, Ou B, Tian H, Qin Z (2021) Reversible data hiding based on multiple histograms modification and deep neural networks. Signal Process: Image Commun 92:116–118
18. Huang D, Wang J (2020) High-capacity reversible data hiding in encrypted image based on specific encryption process. Signal Process: Image Commun 80:115–632
19. Kaw JA, Loan NA, Parah SA, Muhammad K, Sheikh JA, Bhat GM (2019) A reversible and secure patient information hiding system for iot driven e-health. Int J Inf Manag 45:262–275
20. Khosravi MR, Yazdi M (2018) A lossless data hiding scheme for medical images using a hybrid solution based on ibrw error histogram computation and quartered interpolation with greedy weights. Neural Comput Appl 30(7):2017–2028

21. Kim S, Qu X, Sachnev V, Kim HJ (2018) Skewed histogram shifting for reversible data hiding using a pair of extreme predictions. IEEE Trans Circuits Syst Video Technol 29(11):3236–3246
22. Kumar R, Chand S (2016) A reversible high capacity data hiding scheme using pixel value adjusting feature. Multimed Tools Appl 75(1):241–259
23. Li M, Li Y (2017) Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding. Signal Process 130:190–196
24. Liao X, Yin J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features. IEEE Transactions on Dependable and Secure Computing
25. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. IEEE Trans Circuits Syst Video Technol 30(3):685–696
26. Malik A, Singh S, Kumar R (2018) Recovery based high capacity reversible data hiding scheme using even-odd embedding. Multimed Tools Appl 77(12):15803–15827
27. Malik A, Wang H, Chen T, Yang T, Khan AN, Wu H, Chen Y, Hu Y (2019) Reversible data hiding in homomorphically encrypted image using interpolation technique. J Inf Secur Appl 48:102–374
28. Meikap S, Jana B (2018) Directional pvo for reversible data hiding scheme with image interpolation. Multimed Tools Appl 77(23):31281–31311
29. Mohammad AA, Al-Haj A, Farfoura M (2019) An improved capacity data hiding technique based on image interpolation. Multimed Tools Appl 78(6):7181–7205
30. Nguyen TS, Chang CC, Chang WC (2016) High capacity reversible data hiding scheme for encrypted images. Signal Process: Image Commun 44:84–91
31. Ou B, Li X, Wang J (2016) High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. J Vis Commun Image Represent 39:12–23
32. Ou B, Li X, Zhao Y, Ni R, Shi YQ (2013) Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans Image Process 22(12):5010–5021
33. Panchikkil S, Manikandan V, Zhang YD (2022) A convolutional neural network model based reversible data hiding scheme in encrypted images with block-wise arnold transform. Optik 250:168–137
34. Panchikkil S, Manikandan V, Zhang YD (2022) An efficient spatial transformation-based entropy retained reversible data hiding scheme in encrypted images. Optik 261:169–211
35. Panchikkil S, Manikandan V, Zhang YD (2022) A pseudo-random pixel mapping with weighted mesh graph approach for reversible data hiding in encrypted image. Multimedia Tools and Applications pp 1–29
36. Puteaux P, Puech W (2018) An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images. IEEE Trans Inf Forensics Secur 13(7):1670–1681
37. Qian Z, Zhang X (2015) Reversible data hiding in encrypted images with distributed source encoding. IEEE Trans Circuits Syst Video Technol 26(4):636–646
38. Shimizu K (1999) Telemedicine by mobile communication. IEEE Eng Med Biol Mag 18(4):32–44
39. Tan J, Liao X, Liu J, Cao Y, Jiang H (2021) Channel attention image steganography with generative adversarial networks. IEEE Trans Netw Sci Eng 9(2):888–903
40. USC (2022) accessed 01 Apr 2022 image database, URL: http://sipi.usc.edu/database/
41. Wang Y, He W (2021) High capacity reversible data hiding in encrypted image based on adaptive msb prediction. IEEE Transactions on Multimedia
42. Wu X, Chen B, Weng J (2016) Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer. J Vis Commun Image Represent 41:58–64
43. Xiao D, Xiang Y, Zheng H, Wang Y (2017) Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism. J Vis Commun Image Represent 45:1–10
44. Xie XZ, Chang CC, Chen K (2020) A high-embedding efficiency rdh in encrypted image combining msb prediction and matrix encoding for non-volatile memory-based cloud service. IEEE Access 8:52028–52040
45. Xiong L, Xu Z, Shi YQ (2018) An integer wavelet transform based scheme for reversible data hiding in encrypted images. Multidim Syst Sign Process 29(3):1191–1202
46. Xu D, Wang R (2016) Separable and error-free reversible data hiding in encrypted images. Signal Process 123:9–21
47. Yousif SF, Abboud AJ, Alhumaima RS (2022) A new image encryption based on bit replacing, chaos and dna coding techniques. Multimed Tools Appl pp 1–41
48. Yousif SF, Abboud AJ, Radhi HY (2020) Robust image encryption with scanning technology, the el-gamal algorithm and chaos theory. IEEE Access 8:155184–155209
49. Yu J, Zhu C, Zhang J, Huang Q, Tao D (2019) Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. IEEE Trans Neural Netw Learn Syst 31(2):661–674
50. Zheng S, Li D, Hu D, Ye D, Wang L, Wang J (2016) Lossless data hiding algorithm for encrypted images with high capacity. Multimed Tools Appl 75(21):13765–13778