Multimedia Tools and Applications (2023) 82:38613–38637
https://doi.org/10.1007/s11042-023-14717-2

Reversible robust fragile multi-watermarking scheme for color images

Shaohua Duan[1] · Yuhan Qian[1] · Junjie Liu[1] · Hanwen Wang[1] · Xiaoyi Zhou[1]

Received: 14 August 2021 / Revised: 14 December 2021 / Accepted: 4 February 2023 /
Published online: 23 March 2023

Abstract

This research introduced a reversible multi-watermarking scheme for color images with robustness and fragility. The robust watermarking can be used for copyright protection while the fragile watermarking is used to decide whether the image is tampered. The scheme divides the color image into R, G and B layers. The first two layers are embedded with a robust watermark via integer wavelet transform (IWT) and differential histogram shift. Then the embedded layers are used to generate a hash sequence as a fragile watermark. Layer B is used to embed fragile watermark. In order to enhance the watermark invisibility, the prediction error extension (PEE) algorithm is optimized by prediction error length mapping (PELM). To improve the extraction accuracy for tamper detection, a mapping correction scheme is proposed. The performance of the proposed scheme is evaluated using Kodak and USC-ISUI data set. Experimental results show that the proposed scheme has balanced imperceptibility and robustness, and also achieved reversibility and high detection rate. Specifically, the peak signal-to-noise ratio(PSNR) which is used to verify the imperceptibility of the watermarked image is 43.234db. The average normalized correlation coefficient (NC) of the extracted watermark is greater than 0.91, even it suffers from common attacks such as JPEG, noise attack and filter attack. And the accuracy of tamper detection is higher than 90% under malicious attacks.

Keywords Reversible multi-watermarking · Integer wavelet transform · Differential histogram shift · Prediction error length map · Tamper detection map correction

✉ Xiaoyi Zhou
xy.zhou.xy@gmail.com

Shaohua Duan
smartjack10101@gmail.com

Yuhan Qian
qianyuhan077@gmail.com

Junjie Liu
liujunjie10055@gmail.com

Hanwen Wang
asd781263509@gmail.com

[1] Hainan University, Hainan, China

1 Introduction

The development of Internet technology enables digital multimedia to be spread in real-time via Internet. In the digital world, even non-professional users can easily process digital images with tools such as Photoshop, Assetizr and Picasa. Due to the various methods of image destruction, it is often difficult to protect copyrights. Digital watermarking has been recognized as a promising approach for ensuring the authenticity and integrity of images. It is an art of hiding secret information (the watermark) within digital data, such that the embedded watermark can be identified or extracted to confirm the validity of the data [17]. The secret information may be text, image, audio, or video type of contents [19].

1.1 Related work

Watermarking technology can be divided into robust watermarking and fragile watermarking. Robust watermarking is mainly for copyright insurance. Fragile watermarking is sensitive to a broad variety of bends [17]. However, most watermarking schemes embed single watermark with single purpose, which have great limitations in practical applications. Generally, in many applications such as copyright protection, data integrity, tamper detection, fingerprint recognition, and broadcast monitoring, a multi-watermarking scheme is needed.

Some scholars have proposed to embed two robust watermarks at the same time to achieve better copyright protection. Singh et al. [22] presents lifting wavelet transform (LWT) and discrete cosine transform (DCT) based robust watermarking approach for telehealth applications, which hides the signature watermark and patient report into the host medical image. Kumar et al. [12] presents a dual watermarking technique using discrete wavelet transform (DWT), singular value decomposition (SVD), and SPIHT (set partitioning in hierarchical trees). It embeds two encoded watermarks into the transformed host image. Su et al. [23] proposed a new blind watermarking algorithm. By modifying the direct current (DC) coefficient in DCT, the watermark is embedded four times to achieve better robustness. Zhang and Darwish et al. [3, 31] respectively combined with particle swarm optimization (PSO) and adaptive genetic algorithm to balance both watermarking robustness and imperceptibility.

The above schemes embed two watermarks only for robustness. In order to obtain more functions in a scheme, many scholars have paid attention to multi-watermarking with both robustness and fragility. Hurrah et al. [8] embeds a robust watermark using an efficient inter-block coefficient differencing algorithm. The authentication of the content has been ensured by embedding a fragile watermark in the spatial domain. Kamili et al. [9] applies DCT for robust watermark embedding. Then it uses prediction error extension to embed the fragile watermark in the spatial domain. Ahmadi et al. [1] embeds a robust watermark into the blue channel of RGB color space based on DWT, HVS and SVD domains with a specialized PSO optimization to balance the trade-off between robustness and imperceptibility. A fragile watermark is embedded into all channels of RGB color space. Duan et al. [4] embeds a robust watermark into the two layers of R and G based on transformations such as NSST and DWT, and then it embeds the hash value calculated by the two layers of R and G as a fragile watermark into the B layer.

Although multi-watermarking can simultaneously meet the requirements of copyright protection and malicious tampering of multimedia content, the above algorithms will lead to irreversible distortion of the image. Therefore, reversible watermarking has attracted scholars' attention in recent years. The advantage of reversible watermarking is that if there is no attack, the watermarked image can be completely restored to the original one. These types

of watermarks can be used in fields with higher requirements for images, such as medicine and military. Tian et al. [25] first proposed the concept of Difference Expansion (DE) in 2003, DE is used for watermark embedding, a new reversible data embedding method is proposed, but the embedding capacity of the embedding scheme is not large. In order to increase the embedding capacity, Lee et al. [13] uses pixel correlation to change the scan mode and adjust the block size to increase the maximum embedding capacity of the original block from 3 bits to 8 bits. Wang et al. [27] introduced a reversible data hiding algorithm based on two-way differential expansion, which expands the difference between two adjacent pixels in two directions, and embeds information in the left pixel. Girdhar et al. [6] proposed a reversible watermarking algorithm to embed information by moving the difference between the vertices in the 3D mesh model. On the basis of DE, Li et al. [14] put forward a pixel value ordering (PVO) method. The maximum and minimum values of each block are predicted by other pixels of the block according to their pixel value orders. Zhou et al. [35] and Fan et al. [5] respectively proposed a novel PVO with changeable step size (CPVO) algorithm and an improved PVO (IPVO) algorithm to improve the image quality.

In recent years, reversible watermarking algorithms based on interpolation are introduced to scale the image and embed the watermark into the sampling points. Kaw et al. [10] proposed a new large-capacity and reversible watermarking algorithm based on pixel repetition, which uses the best pixel repetition rate to securely embed the watermark into the image. On this basis, Parah et al. [18] proposed a new reversible watermarking scheme based on the pixel repetition method (PRM) and modular algorithm, which can provide high embedding capacity while maintaining good imperceptibility. Hassan et al. [7] chose the existing enhanced neighbor mean interpolation (ENMI) and modified neighbor mean interpolation (MNMI) technology to scale-up the original image before embedding the secret data. This method achieves good image quality under high embedding capacity.

The above schemes all implement the reversibility of watermarking, but their schemes are fragile to common attacks. To solve this problem, reversible robust watermarking scheme is satisfied with the demand. Wang et al. [29] proposed a new spatially-based robust steganography algorithm based on the significant bit difference expansion (SBDE) method, which increases the embedding capacity, but the distortion caused by invalid embedding is relatively high. Kumar et al. [11] improved the algorithm of Wang et al. [29] and proposed a new robust reversible data hiding (RRDH) scheme based on double-layer embedding. The redundancy of HSB plane elements is effectively used, and the distortion caused by invalid embedding is reduced. To further improve robustness, [20] and [28] apply wavelet transforms and histogram-based methods to embed reversible robust watermarks. Roy et al. [20] present a robust reversible image watermarking scheme based on DCT and histogram shifting. By modifying AC coefficient, a binary watermark is embedded in each transform block and the location map is embedded into the cover image by histogram shift technology. Wang et al. [28] applied a high pass filter to each block to generate a histogram which is a Laplacian-like distribution. The watermark is embedded into the blocks by shifting the generated histogram. Experimental results show that the above schemes are robust to common attacks with reversibility.

In recent years, some adaptive algorithms [33, 34, 36] and deep learning algorithms [30] have been applied to digital watermarking. Adaptive schemes are put forward to determine the embedding position of the watermark. [16, 32] proposed adaptive watermarking schemes using integer wavelet transform. Zhang et al. [32] and Meng et al. [16] respectively use HVS and element relations to adaptively select the position of the embedded watermark to achieve better invisibility. Ansari et al. [2] Use artificial bee colony (ABC) to control the embedding strength of the watermark to achieve a balance of invisibility and robustness.

1.2 Contributions

In summary, although reversible robust watermark and fragile watermark have achieved fruitful results. A reversible robustness and fragility multi-watermark scheme for color images has not been proposed. The main research work of this paper is as follows:

(1) In this paper, a comprehensive reversible watermarking scheme of robustness and fragility for color images is proposed for the first time. This paper uses integer wavelet transform and differential histogram shift scheme to achieve reversible robust watermarking, and improve the existing PEE method to embed fragile watermarking. Both robust and fragile watermarks can be embedded into color images at the same time. After extracting the watermark, the watermarked Image can be restored, and the recovered Image is exactly the same as the original host Image.
(2) In this paper, integer wavelet transform and differential histogram shift method are combined to achieve a robust watermarking scheme. In order to make watermarked image more invisible, this paper proposes a pre-embedding scheme to select the optimal embedding position and generate the local map. The R and G layers of the color image use the same local map to reduce the local map storage capacity. See Section 3.1 for details.
(3) In this paper, the traditional PEE method is improved. A prediction error length map (PELM) was first proposed, instead of using local map to record the embedding position of watermark, we record the length map. For the position of too large prediction error (the prediction error length is too large), we will not embed watermarks to avoid large distortions. During the generation of tamper detection graph, a map correction scheme for tamper detection is proposed to improve the extraction accuracy. In this scheme, the blocks are divided into three classes, tampered blocks, suspected blocks and untampered blocks. After the initial tamper detection graph is generated, the proposed correction scheme is used to judge the suspicious block twice to improve the tamper detection accuracy. See Section 3.2 for details.

The rest of the paper is arranged as follows. Section 2 introduces background knowledge, integer wavelet transform, difference histogram shift, and we proposed prediction error length map(PELM). Section 3 describes the embedding and extraction process of the watermark in detail. Section 4 makes experimental results of the proposed method. Finally, a summary is made in Section 5.

2 Algorithm basis

2.1 Integer wavelet transform

IWT (Integer wavelet transform) was proposed and proved by Sweldens et al. [24] in 1996. It presents the lifting scheme, a simple construction of second-generation wavelets. Compared with DWT, IWT can improve computational efficiency and achieve lossless image reconstruction.

Figure 1 shows the decomposition and reconstruction process of lifting wavelet transform. Lifting wavelet transform is divided into three steps, splitting, predicting and updating. After lifting scheme, the input signal s_j can be decomposed into a low-frequency part s_{j-1} and a high-frequency part d_{j-1}. For the low-frequency subset s_{j-1}, perform

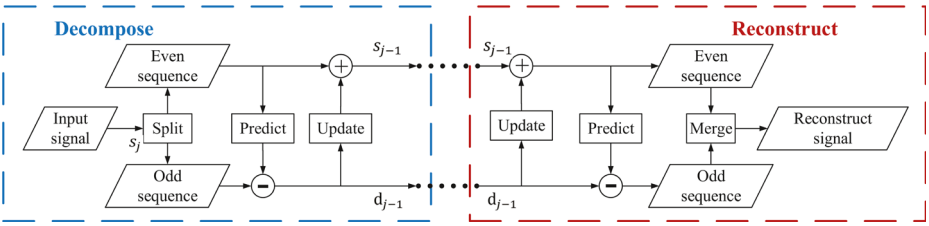

Fig. 1 Lifting wavelet transform process

the same splitting, predicting, and updating to further decomposed s_{j-1} into s_{j-2} and $d_{j-2}, \cdots$.In this way, after n times of decomposition, the wavelet of the original data s_j is represented as $\{s_{j-n}, d_{j-n}, d_{j-n+1}, \cdots, d_{j-1}\}$. s_{j-n} represents the low-frequency part, $\{d_{j-n}, d_{j-n+1}, \cdots, d_{j-1}\}$ is the series of the high-frequency part from low to high. The specific steps are as follows:

Split is to divide the original signal $s_j = \{s_{j,k}\}$ into two disjoint subsets. The length of each subset is half of the original signal. It is common to divide a sequence into even sequences $e_{j-1,k} = \{s_{j,2k}\}$ and odd sequences $o_{j-1,k} = \{s_{j,2k+1}\}$, that is, $Splits_j = (e_{j-1}, o_{j-1})$.

Predict takes advantage of the correlation between even and odd sequences. One sequence (usually an even sequence e_{j-1}) is used to predict the other sequence (usually an odd sequence o_{j-1}). The difference d_{j-1} between the actual value o_{j-1} and the predicted value $p(e_{j-1})$ reflects the degree of approximation between the two, which is called the wavelet coefficient. Generally speaking, the more relevant the data, the smaller the wavelet coefficient. If the prediction is reasonable, the dataset d_{j-1} contains much less information than the original subset o_{j-1}. The prediction process is as follows: $d_{j-1,k} = o_{j-1,k} - p_k(e_{j-1})$. Where p_k can take the e_j data itself, which is $p_k(e_{j-1,k}) = e_{j-1,k}$ or take the average value of the adjacent data, $p_k(e_{j-1}) = \frac{e_{j-1,k}+e_{j-1,k+1}}{2}$.

After the splitting step, the global features (such as the mean) of the subset may be inconsistent with the original data. To maintain these characteristics of the original data, an *Update* process is essential. The process is as follows: $s_{j-1} = e_{j-1} + U(d_{j-1})$. Where s_{j-1} is the low-frequency part of s_j. The update operator can also take different functions, such as $U_k(d_{j-1}) = \frac{d_{j-1,k}}{2}$, $U_k(d_{j-1}) = \frac{d_{j-1,k-1}+d_{j-1,k}}{4} + \frac{1}{2}$.

2.2 Difference histogram shift

Due to the correlation between adjacent audio information, Liang et al. [15] realized audio robust and reversible watermarking using high-order differential histogram shift. In the image, there is a correlation between pixels. After the color image is divided into three layers of R, G, and B, any layer also meets this characteristic. On this basis, this paper proposes a differential histogram shift robust watermarking scheme for an image. Divide the embedded area into N blocks ($N = \frac{H}{n} \times \frac{W}{n}$, H is the length of the host image, W is the width of the host image, and n is the size of each sub-block). That is, each block has $n \times n$ pixels, which is defined as $block^l$, where l is the block number ($l = 1, 2, 3 \cdots N$). In one block, half of pixels are weighted as 1, and the others are -1. Thus, each group will get a difference

value $D(l)$, the calculation formula is as Formula (1).

$$D(l) = \sum_{i=1}^{n} \sum_{j=1}^{n} (-1)^{i+j} \times block^{l}(i, j) \tag{1}$$

Use this method to calculate the difference of N blocks. Then generate a prediction error histogram, as shown on the left side of Fig. 2.

Suppose the offset of the histogram is B, $B = T + G$. T is the absolute value of the maximum prediction error, G is the embedding strength. If $B < T$, the watermark cannot be completely extracted. Shift the histogram to embed the watermark. If the watermark $w(i) = 1$, use Formula (2) for shifting, and if $w(i) = 0$, no operation is performed. The difference histogram after embedding is shown on the right in Fig. 2.

$$D(l)' = D(l) + \frac{D(l)}{|D(l)|} \times B \qquad if \quad w(i) = 1 \tag{2}$$

The change of the block difference value $D(l)$ is achieved by shifting the pixels in the l block. To ensure that the change of the pixel value is an integer, the offset of each pixel is set to $\beta(k) = \lfloor \frac{B+(k-1)}{M} \rfloor$, $M = n \times n$, $k = 1, 2, 3 \cdots M$, change the $block^{l}(i, j)$ as Formula (3).

$$block^{l}(i, j)' = block^{l}(i, j) + (-1)^{i+j} \times \frac{D(l)}{|D(l)|} \times \beta(k) \tag{3}$$

When extracting the watermark, only the $D(l)$ of the watermark image needs to be calculated. Use Formula (4) to extract the watermark, and the grouped image is shown on the right in Fig. 2.

$$w = \begin{cases} 0 & D(l) \in class2 \\ 1 & D(l) \in class1, 3 \end{cases} \tag{4}$$

Similarly, according to the range of $D(l)$, reversible recovery is performed. When restoring, use Formula (1) to calculate the $D(l)'$ of the watermark image, and use Formula (5) to restore.

$$D(l)^{r} = D(l)' - \frac{D(l)'}{|D(l)'|} \times B \qquad if \quad w(i) = 1 \tag{5}$$

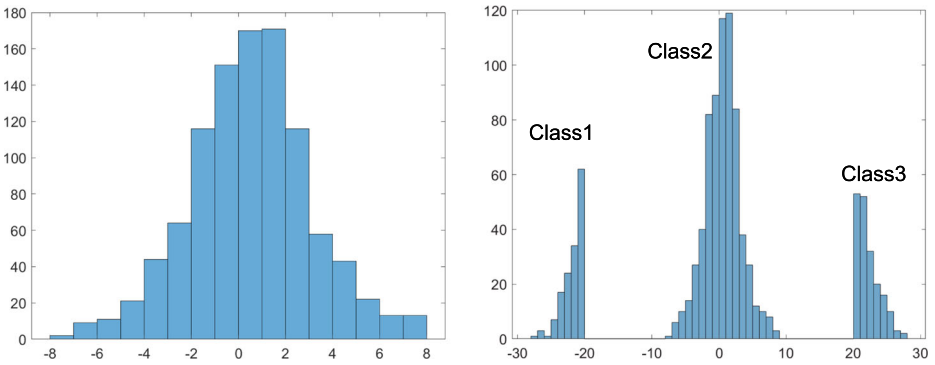

Fig. 2 Histogram of prediction error and embedded histogram

To realize the change of $D(l)$, the group l in the watermark image is restored using Formula (6).

$$block^l(i,j)^r = block^l(i,j)' - (-1)^{i+j} \times \frac{D(l)'}{|D(l)'|} \times \beta(k) \tag{6}$$

2.3 Prediction error length map

The prediction error length map (PELM) is proposed to replace the local map in the traditional PEE method. Figure 3 is an 8×8 block from the image, which is used as an example for illustration. The blue block pixels in the figure are used to predict the green block pixels, so that each 8×8 block has 18 prediction positions, which can be embedded in the way of PEE.

This paper use the method of diamond prediction [21] to predict the green block. The following is a detailed explanation of the embedding, extraction and recovery process of a pixel.

The prediction and embedding formula is shown in Formula (7-10).

First, use a 3×3 block size to predict the central pixel value C. Pc is the average of the four pixels above and below the center pixel, Pe is the prediction error, as in Formula (7).

$$Pe = C - Pc \tag{7}$$

w is the watermark sequence, and only the position where $w(i) = 1$ is embedded. Use Formula (8) to calculate the prediction error value after embedding the watermark Pe'.

$$Pe' = \begin{cases} Pe & w(i) = 0 \\ \frac{Pe}{|Pe|} \times (2 \times |Pe| + 1) & w(i) = 1 \end{cases} \tag{8}$$

99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
94	94	94	94	94	94	94	94
73	73	73	75	73	73	75	75
71	71	71	71	71	72	71	71
72	74	72	71	72	72	71	71
74	74	68	71	71	72	72	71
71	72	72	71	74	74	71	72

Fig. 3 Map of the predicted error of a block

Modify the center pixel, the modified center pixel value is C', as in Formula (9).

$$C' = Pc + Pe' \tag{9}$$

Calculate the prediction error value length l and store it.

$$l = \lceil log_2(Pe) \rceil \tag{10}$$

The watermarking extraction formula is shown in Formula (11-13). First calculate the predicted value of the pixel after the embedded watermark, C' is the central pixel value of the watermarked image, Pc' is the predicted central pixel value of the embedded watermark, and Pe' is the calculated prediction error value.

$$Pe' = C' - Pc' \tag{11}$$

Calculate the prediction error value length of the watermark map l'.

$$l' = \lceil log_2(Pe') \rceil \tag{12}$$

The watermark extraction method is as formula (13).

$$w(i) = \begin{cases} 0 & l = l' \\ 1 & l \neq l' \end{cases} \tag{13}$$

After the watermark is extracted, the center pixel is restored, and Pe^r is the prediction error after restoration.

$$Pe^r = \begin{cases} Pe' & w(i) = 0 \\ \frac{Pe}{|Pe|} \times \frac{|Pe|}{2} & w(i) = 1 \end{cases} \tag{14}$$

Obtain the restored center pixel value C^r according to Pe^r

$$C^r = Pc' + Pe^r \tag{15}$$

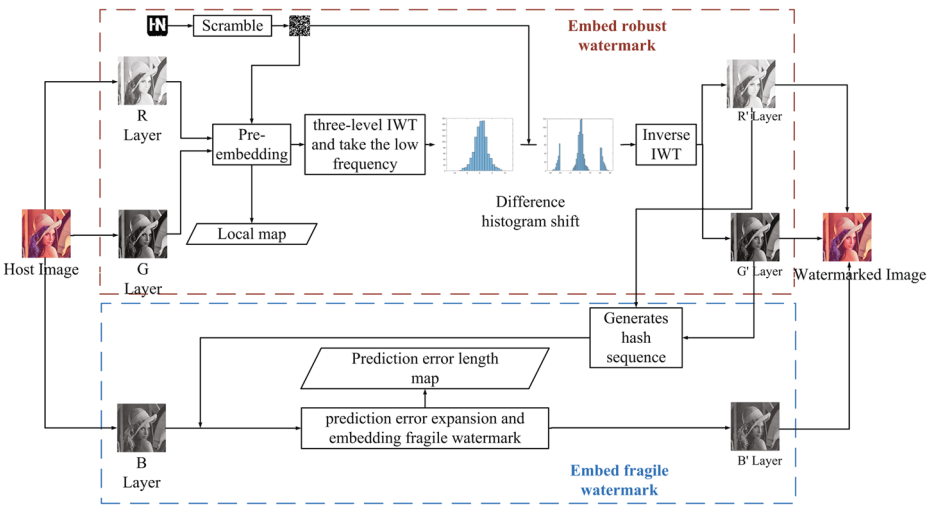

Fig. 4 The flow chart of watermark embedding

3 Proposed watermarking scheme

In this paper, the robust watermark is embedded into R layer and G layer by using integral wavelet transform and differential histogram shift. Then, the embedded R′ and G′ are used to generate the hash sequence. Hash sequence, as fragile watermark, is embedded into B layer by PEE. Figure 4 shows the flow chart of watermark embedding.

In the procedure of extracting the watermark, extract the fragile watermark first, the hash sequence H_1 is generated by the R′ and G′ layers. Then, the hash sequence H_2 in the B′ layer is extracted by the predictive error length map. The tamper detection map is generated by comparing H_1 and H_2. Secondly, extract the robust watermark use IWT and difference histogram shift. Finally, decide whether the suspected image is tampered according to tamper detection graph. If yes, the tamper detection graph will be displayed. Otherwise, the watermarked image will be recovered. Figure 5 shows the flow chart of watermark extraction procedure.

3.1 Copyright protection: robust watermarking scheme

3.1.1 Robust watermark embedding

Robust watermark embedding is divided into two stages, the first stage is the pre-embedding, and the second stage is watermark embedding. The pre-embedding stage is mainly used to determine the embedding position and generate the local map. In this paper, the R layer and the B layer of the host image are first divided into 16×16 non-overlapping blocks, each block is transformed by 3-IWT, and the 2×2 low frequency area is selected to calculate the difference value $D(l)$. This paper uses the method of sorting the difference value to determine the final embedding position. The smaller the difference value is, the better the invisibility of the watermarked image. Figure 6 is a heat map generated from the

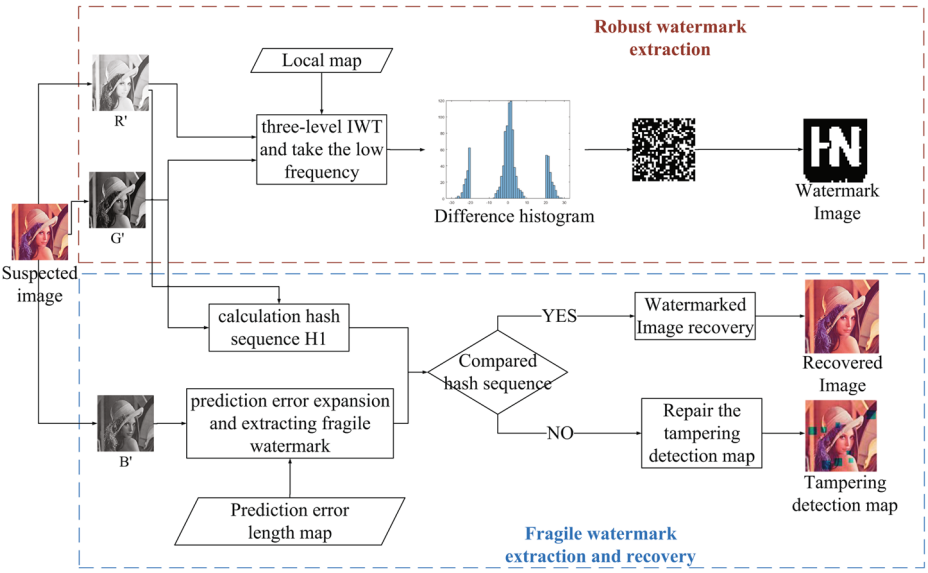

Fig. 5 The flow chart of watermark extraction

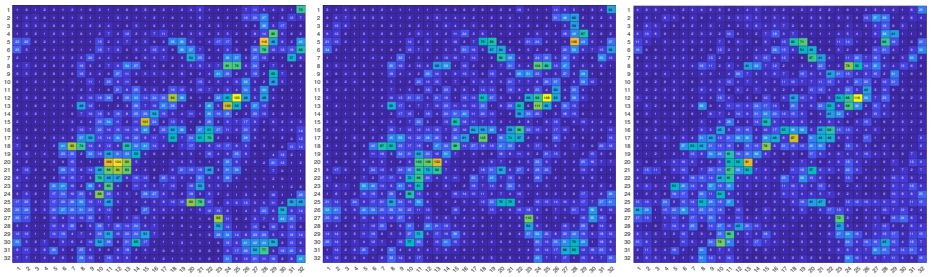

Fig. 6 Heat map of differential value of R, G and B layers

differential value calculated after the block transformation of the R layer, G layer and B layer of Lena. The closer the color in the heat map is to blue, the smaller the difference is, and the closer to green, the greater the difference is. In this paper, the R and the B layer use the same local map for the reason that the distribution of the heat maps of the three layers is similar, which can also be seen from Fig. 6. In the embedding stage, the local map generated in the pre-embedding stage is used to embed the watermark. The specific embedding process of the watermark is as follows. Steps 1-4 are the pre-embedding process, and 5-7 are the embedding process.

Step1. Perform Arnold scrambling on the watermark image and generate a binary watermark sequence w, the watermark sequence length is len_w.

Step2. Divide layer R and layer G into non-overlapping 16×16 blocks and calculate the difference using Formula (1). Use Formula (16) and (17) to get the sequence S_R and S_G. Add the corresponding positions of the two sequences to generate the sequence S, as in the Formula (18). Sort the sequences from small to large $S_{sort} = Sort(S)$.

$$S_R = \{D(l)^R(1), D(l)^R(2), \cdots, D(l)^R(N)\} \tag{16}$$

$$S_G = \{D(l)^G(1), D(l)^G(2), \cdots, D(l)^G(N)\} \tag{17}$$

$$S = S_R + S_G = \{D(l)^R(1) + D(l)^G(1), \cdots, D(l)^R(N) + D(l)^G(N)\} \tag{18}$$

Step3. According to the order of S sequence, we pre-embed. Since overflow may occur during embedding, pre-embedding is performed before the formal embedding. Use Formula (19) to calculate the number of blocks pre-embedded $N(N \geq \lceil \frac{len_w}{2} \rceil)$, len_w is the watermark sequence length, a is the linear increment coefficient.

$$N = a \times \lceil \frac{len_w}{2} \rceil \qquad a \geq 1 \tag{19}$$

Step4. Take the front N block in S_{sort}. Then take $T' = Max(D(l))$, T' is the predicted shift threshold. Set the robust watermark embedding strength as G. Calculate the predicted offset $B' = T' + G$.

Step5. Use the B' generated in step 4 to pre-embed. If all the watermark sequences can be embedded, then generate the local map (Fig. 7 shows the robust watermark local map of Lena) and let $B = B', T = T'$. Otherwise, increase the value of a to increase the number of pre-embedded blocks and return to step 3. Figure 8 shows the flow chart of generating local map.

Step6. According the local map generated in Step 5, perform three-level IWT on labeled small block of the R and G layer. Using the histogram shift method, as shown in Formula (2), embed the robust watermark in the low-frequency area.

Fig. 7 Robust watermark local map of Lena

Step7. Perform inverse IWT on each embedded block, and replace the original image block to generate a watermarked image (R′, G', B).

Step8. Store the local map, the predicted shift threshold T and the robust watermark embedding strength G.

3.1.2 Robust watermark extraction

Extraction process of the robust watermark is as follows:

Step1. Read the store T, G and the local map.

Step2. Divide an image into blocks. According to the local map, perform IWT transform on the watermarked block. Use Formula (1) to calculate the difference $D'(l)$.

Step3. This paper uses three watermark extraction schemes [15] to extract the watermark. The first two schemes use the control range method. The third scheme uses the K-means clustering algorithm, such as Formula (20–22).

$$w_1'(i) = \begin{cases} 0 & if \quad E' \in [-T - \frac{G}{2}, T + \frac{G}{2}] \\ 1 & otherwise \end{cases} \tag{20}$$

$$w_2'(i) = \begin{cases} 0 & if \quad E' \in [-T - \frac{G}{3}, T + \frac{G}{3}] \\ 1 & otherwise \end{cases} \tag{21}$$

$$w_2'(i) = \begin{cases} 0 & if \quad E' \in class2 \\ 1 & if \quad E' \in class1 \ or \ class3 \end{cases} \tag{22}$$

Step4. The watermark sequence is extracted by Formula (20–22). The watermark is extracted from Formula (23).

$$w'(i) = \begin{cases} w_1'(i) & if \quad w_1'(i) = w_2'(i), w_1'(i) = w_3'(i) \\ w_1'(i) & if \quad w_1'(i) = w_2'(i), w_1'(i) \neq w_3'(i) \\ w_2'(i) & if \quad w_1'(i) \neq w_2'(i), w_2'(i) = w_3'(i) \\ w_3'(i) & if \quad w_1'(i) \neq w_2'(i), w_1'(i) = w_3'(i) \end{cases} \tag{23}$$

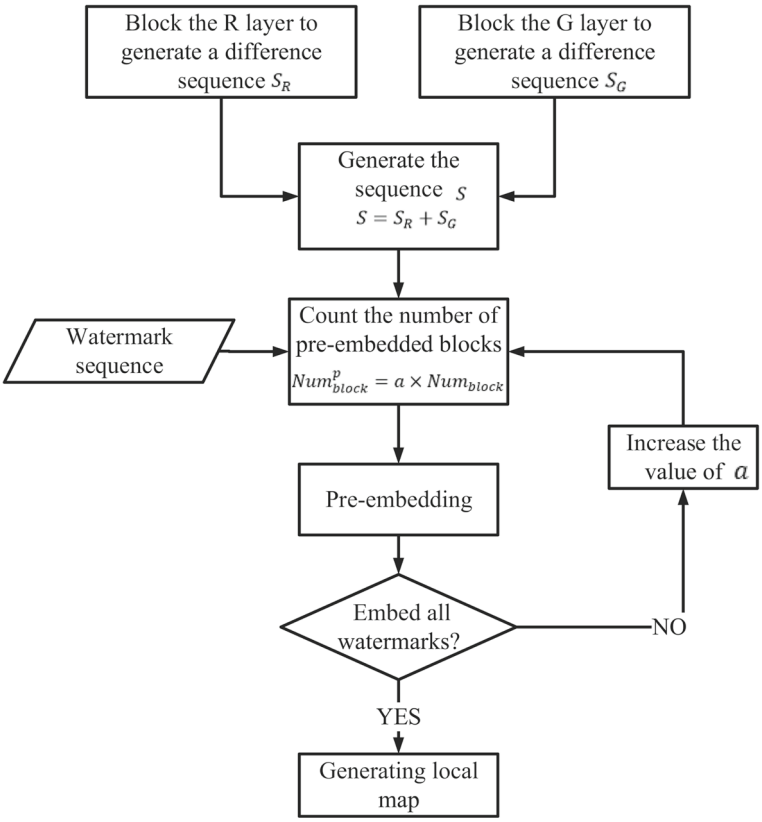

Fig. 8 The flow chart of generating local map

3.1.3 Robust watermark recovery

If the image has not been tampered with, restore the image with robust watermark. The recovery process is as follows:

Step1. Use the local map to get the watermarked blocks in the R and G layers.
Step2. The small blocks are transformed by IWT transformation to get the low-frequency areas. Then use Differential Histogram Shift to restore, as in Formula (5 and (6).
Step3. After all the embedded blocks are recovered, perform inverse IWT to generate the restored image.

3.2 Tamper detection: fragile watermark scheme

3.2.1 Fragile watermark embedding

When embedding the fragile watermark, the predictive error length map is presented. The watermark can be embedded only when the position of the watermark sequence is 1. Please refer to Section 2.3 for more details. The specific process of fragile watermark embedding is as follows:

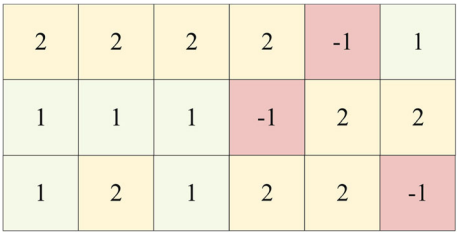

Fig. 9 Map of the predicted error length of a block

Step1. The R′, G′ of the embedded watermark image is subjected to 8 × 8 blocks. Each block uses SHA-256 to generates a hash value and intercepts the front 18 bits as a fragile watermark.

Step2. Layer B is divided into the same size blocks. Embed fragile watermarks in each block by Formula (8). Set the embedding length threshold μ, in the embedding process, if the length of the prediction error value is more than μ or the embedded pixels overflows, this position will not be embedded and the PELM is recorded as -1. Otherwise, the PELM records the length of the prediction error value, as shown in Formula (24). An 8 × 8 block can be embedded with 18-bit watermark, so 18 prediction error lengths need to be stored. The PELM of a small block is shown in Fig. 9.

$$Map_{len}(x, y) = \begin{cases} -1 & l > \mu \ or \ C' > 255 \ or \ C' < 0 \\ log_2(Pe) + 1 & l \leq \mu \ and \ 0 \leq C' \leq 255 \end{cases} \tag{24}$$

Step3. After all the small blocks are embedded with fragile watermarks, the length map Map_{len} is stored.

3.2.2 Fragile watermark extraction

This paper uses a predictive error length map to embed fragile watermarks, which may cause some locations not to be embedded ($Map_{len}(x, y) = -1$), and these locations will be skipped in the tamper detection. In this paper, fragile watermark extraction is composed of two stages. The first stage is the comparison of watermark hash sequences (step 1-3). The second stage is to restore the tamper detection map (step 4). The specific fragile watermark extraction process is as follows:

Step1. Divide the watermarked image into 8 × 8 non-overlapping blocks. Then use R′ and G′ layers to generate hash sequences for each block $H_1(i), i \in \{1, 2, ..., 4096\}$.

Step2. Read the stored Map_{len}. Use Formula (12) to calculate the length of prediction error l' for the B layer blocks. Then use Formula (13) to extract the watermark to generate $H_2(i), i \in \{1, 2, ..., 4096\}$.

Step3. The extracted watermark in Step 2 was compared with that generated by R′ and G′ layers, generate the initial tamper detection map Map_{tamper}, as shown in Formula (25).

$$Map_{tamper} = \begin{cases} 0 & H_1(i) = H_2(i) \\ 1 & H_1(i) \neq H_2(i) \end{cases} \tag{25}$$

Step4. Set $\varphi(0 \leq \varphi \leq 8 \times 8)$, if at least φ positions in an $n \times n$ block are not embedded, mark this block as a suspected block. The possible locations of the suspected blocks are divided into three classes. Class1: center block, Class2: corner block, Class3: edge block, as shown in Fig. 10.

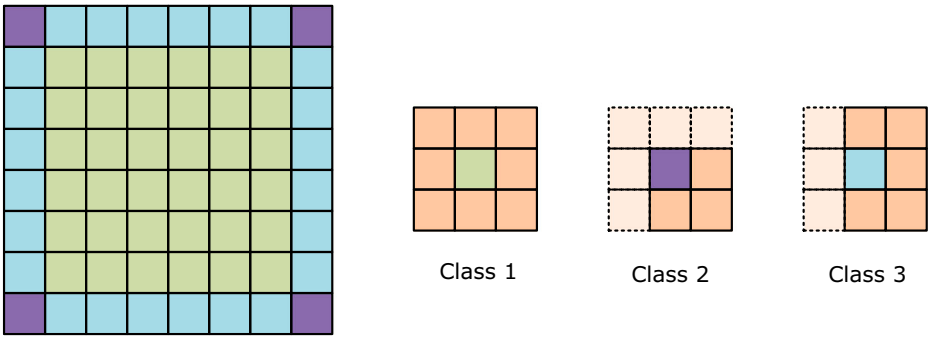

Fig. 10 Suspected block classification

Step5. Set a threshold θ to determine whether the suspected block, which is marked in Step 4, is tampered, . Different types of suspected blocks have different values of θ_k. $k \in \{1, 2, 3\}$ is the block class. In this paper, for the center block, $k = 1, \theta_k = 5$, for the corner block, $k = 2, \theta_k = 1.5$, for the edge block, $k = 3, \theta_k = 2.9$. Determine whether the suspected block is tempered by the neighboring 9 blocks. Use Formula (26) to update the position of the suspected block in Map_{tamper} and obtain the final tamper detection map.

$$\begin{cases} T_{map} = 0 & \theta_{set} - \alpha \times Num_{miss} \geq \beta \times Num_{tramper} + \gamma \times Num_{suspect} \\ T_{map} = 1 & \theta_{set} - \alpha \times Num_{miss} < \beta \times Num_{tramper} + \gamma \times Num_{suspect} \end{cases} \tag{26}$$

3.2.3 Fragile watermark recovery

If the image is not tampered, restore the image with fragile watermark. The recovery process is as follows:

Step1. Formula (14) is used to calculate the restored prediction error value, and Formula (15) is used to obtain the restored center pixel C^r to restore the modified center pixel.

Step2. Restore all the modified center pixels to get the restored image.

4 Experimental results

The proposed scheme was implemented by Matlab R2021a in the PC with Intel Core i5-4200H CPU@2.80GHz and 16GB RAM, of which the OS is 64-bit Windows 10 professional. This paper uses Kodak and USC-ISUI data set for testing, the color image having size $512 \times 512 \times 3$ and watermark image having size 32×32. The invisibility, reversibility, robustness and tamper detection of the proposed method has been evaluated in terms of Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index Measure (SSIM) for watermarked image, Normalization cross correlation (NC) for watermark image, Accuracy (ACC) for tamper detection diagram. Section 4.1 describe the watermark invisibility and reversibility. Sections 4.2 and 4.3 represent the watermark robustness test and fragile test, respectively. Section 4.4 describe the watermark capacity.

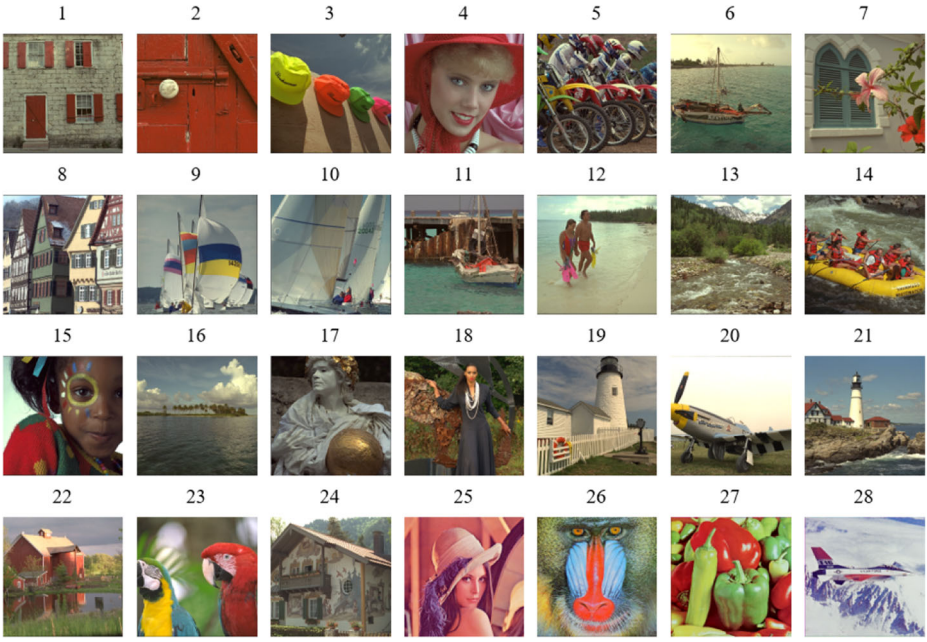

Fig. 11 28 images used in the experiment

4.1 Watermark invisibility and reversibility analysis

Figure 11 shows test images, and use the Formula (27) to balance invisibility and robustness. Change the target value by adjusting the parameters a_1, a_2, a_3, and p. By changing the embedding strength G, the PSNR value, SSIM value and NC value will change, and the target value changes accordingly. The closer the target value is to 1, the better the balance. According to the target value, different images choose the best embedding strength G.

$$target = a_1 \times \frac{PSNR}{p} + a_2 \times SSIM + a_3 \times NC \tag{27}$$

$$avgNC = \frac{\sum_{i=0}^{10} NC_i}{10} \tag{28}$$

Table 1 shows the experimental parameters and results, where T and G are the predicted shift threshold and the robust watermark embedding strength when embedding different images, respectively. PSNR1 and SSIM1 are the values of the watermark and host image after the robust watermark is embedded, PSNR2 and SSIM2 are the values after the robust and fragile watermark are embedded, PSNR3 and SSIM3 are the values of the restored image and the host image, $avgNC$ is the average of the watermark NC values after the watermarked image has been attacked by JPEG (QF = 40%), Histogram equalization, Gaussian noise (0.5%), Salt-pepper noise (0.5%), Speckle noise (1%), Average filter 3×3, Median filter 3×3, Motion filter 3×3, Gaussian LPF 3×3, and Crop 25%. The calculation formula of $avgNC$ is shown in Formula (28). NC_i is the NC value of the extracted watermark image and the original watermark image when carrying out different attacks. Table 1 shows the average PSNR2 is 43.234, and the SSIM2 values are all greater than 0.98, which indicates the good imperceptibility of the proposed scheme. The reversibility can be shown

Table 1 Experimental parameters and results

Picture	T	G	PSNR1	SSIM1	PSNR2	SSIM2	PSNR3	SSIM3	AvgNC
1	10	9	44.1589	0.9979	43.8594	0.9976	Inf	1	0.9547
2	5	11	45.6875	0.9997	45.087	0.9996	Inf	1	0.9488
3	5	11	45.6875	0.9952	45.1389	0.9945	Inf	1	0.9514
4	9	11	43.7493	0.9972	43.3243	0.9968	Inf	1	0.9613
5	99	–	–	–	–	–	–	–	–
6	11	11	42.8857	0.9955	42.6116	0.995	Inf	1	0.9742
7	8	12	43.7493	0.994	43.3627	0.9933	Inf	1	0.9734
8	73	–	–	–	–	–	–	–	–
9	4	13	45.1161	0.9888	44.5361	0.9869	Inf	1	0.9383
10	5	12	45.1161	0.9918	44.5267	0.9903	Inf	1	0.9482
11	8	12	43.7493	0.9895	43.3925	0.9886	Inf	1	0.9665
12	7	13	43.7493	0.9961	43.3442	0.9955	Inf	1	0.9736
13	31	8	37.9401	0.9897	37.8733	0.9895	Inf	1	0.9164
14	24	10	39.1253	0.9857	38.9961	0.9851	Inf	1	0.9384
15	15	10	41.7903	0.9934	41.5335	0.9927	Inf	1	0.9572
16	6	11	45.1161	0.9933	44.6332	0.9922	Inf	1	0.9513
17	10	9	44.1589	0.9899	43.7029	0.9884	Inf	1	0.9607
18	16	5	43.2961	0.9889	42.9471	0.9878	Inf	1	0.9141
19	9	8	45.1161	0.9918	44.6347	0.9906	Inf	1	0.933
20	51	–	–	–	–	–	–	–	–
21	10	8	44.6112	0.9958	44.1641	0.9951	Inf	1	0.9429
22	11	9	43.7493	0.9907	43.3562	0.9898	Inf	1	0.959
23	9	7	45.6875	0.9967	45.0703	0.9961	Inf	1	0.9324
24	14	7	43.2961	0.9906	42.9827	0.9897	Inf	1	0.9546
25	10	10	43.7493	0.9993	43.3584	0.9991	Inf	1	0.9579
26	24	5	40.5065	0.9969	40.406	0.9968	Inf	1	0.9162
27	16	5	43.2961	0.9991	42.9555	0.999	Inf	1	0.9179
28	8	8	45.6875	0.987	45.0608	0.9849	Inf	1	0.9113
avg	11.4	9.4	43.631	0.9934	43.234	0.9926	Inf	1	0.9461

by PSNR3 and SSIM3, which is all INF and all 1, respectively. This shows that the image is completely reversible, that is to say, the watermarked image can be completely restored to the original image. The original image, watermarked image and the restored image of seven random selected in the test image are shown in Fig. 12.

As can be seen from Table 1, the T values of Image 5, 8 and 20 are large, which means that the B value, the offset of the histogram, of these three images is also large, and is easy to cause pixel value overflow. Therefore, these three images cannot be used to embed watermarks and the PSNRs and SSIMs are not available. To explore the transformation of PSNR and NC with G. This paper uses a 32×32 watermark image and Lena, Baboon, Peppers, Airplane image to conduct experiments. Figure 13 shows the changes in PSNR and $avgNC$ values of different images as G increases by 50 from 1. It can be seen that $avgNC$ shows a logarithmic growth pattern and PSNR slowly declines. If G is approximately less than 10, $avgNC$ increases sharply with G, and then slowly after that. It means that if G is

Fig. 12 The original image, watermarked image and the restored image of seven image random selected in the test image: (O1-O7) original image, (W1-W7) watermarked image, (R1-R7) restored image

kept in the range of about 10, the balance between robustness and invisibility reaches the best.

4.2 Watermark robustness test

In practice, the watermarked image may experience various distortions, such as JPEG attack, noise attack, filter attack, etc. Table 2 shows the comparison between the proposed

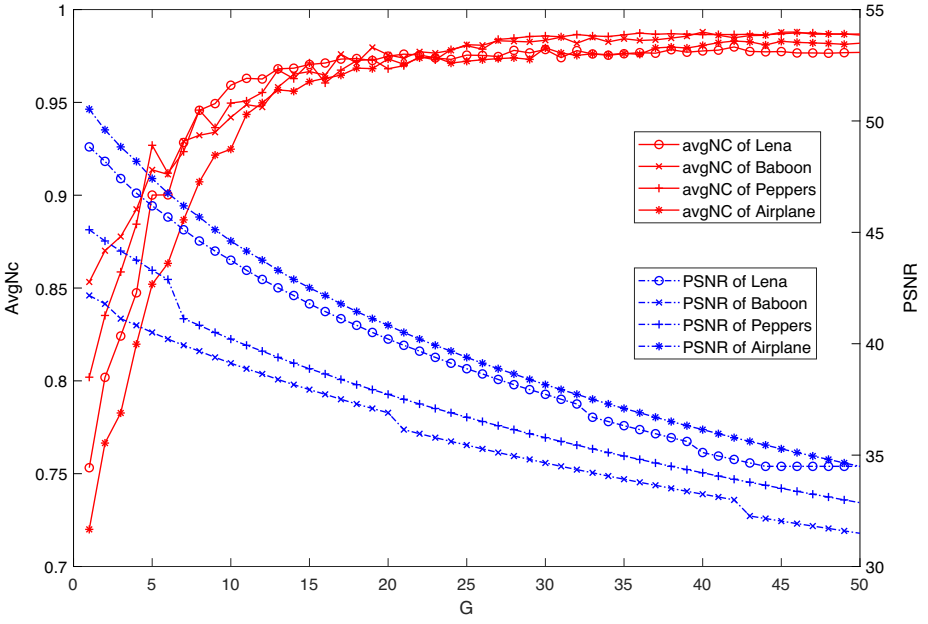

Fig. 13 The changes of T, PSNR, $avgNC$ value of different images as the G increases

Table 2 Comparison with [26] NC and BER value. Bold represents the highest value

	Proposed		IWT [26]		DWT [26]		CT [26]	
	NC	BER	NC	BER	NC	BER	NC	BER
No attack	**1**	**0**	1	0	0.999	0.006	0.999	0.004
GF[3 × 3]	**0.999**	**0.090**	0.984	0.799	0.965	1.744	0.992	0.419
GF[5 × 5]	**0.998**	**0.097**	0.949	2.550	0.930	3.498	0.964	1.789
GN(0.001)	**0.997**	**0.223**	0.993	0.370	0.961	1.939	0.994	0.290
S&P(0.01)	0.978	1.406	**0.990**	**0.497**	0.982	0.881	0.951	0.652
S&P(0.02)	0.932	4.432	**0.965**	**1.765**	0.938	2.336	0.816	3.469
S&P(0.05)	0.806	13.3	**0.821**	**7.928**	0.799	8.282	0.641	11.98
AF[3 × 3]	0.931	4.148	**0.982**	**0.879**	0.973	1.332	0.966	1.702
MF[3 × 3]	0.974	1.621	**0.988**	**0.586**	0.982	0.879	0.986	0.684
JPEG(90)	0.964	2.274	**0.964**	1.789	0.956	**2.222**	0.952	2.396
JPEG(80)	0.934	4.138	0.923	3.871	0.923	3.861	**0.949**	**2.564**
JPEG(70)	**0.901**	6.09	0.853	7.377	0.901	**4.974**	0.891	5.455
SN(0.001)	**0.999**	**0.039**	0.848	7.596	0.841	7.963	0.961	1.964

scheme and [26]. NC and BER in the table are the average of all test images in Fig. 11. For Gaussian filter, Gaussian noise, Speckle noise and high QF JPEG compression attack, the

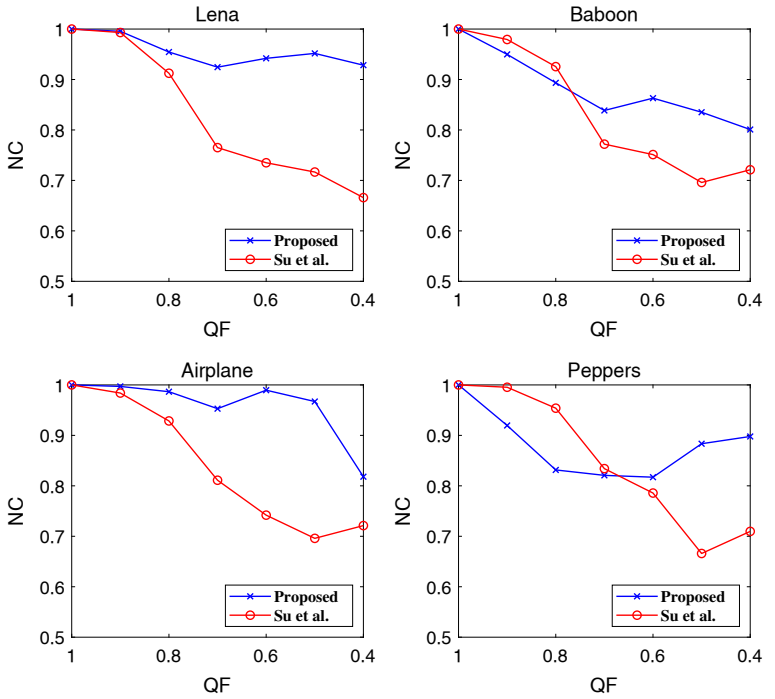

Fig. 14 The comparison with the Su et al.'s scheme [23] after the JPEG attack

Table 3 The NC values of noise attack test for Lena. Bold represents the highest value

	Proposed $T = 10, G = 10$	[9]	[4] $\Delta = 72, \lambda = 12$	[23] $\Delta = 40$
PSNR	43.75dB	41.70dB	43.39 dB	43.59dB
Attack	*NC*			
Image darken	**1**	0.984	0.999	0.621
Gaussian noise (0.1%)	**1**	0.985	0.95	0.999
Salt-pepper noise (1%)	**0.984**	0.92	0.757	0.901
Salt-pepper noise (2%)	**0.952**	–	0.603	0.746
Speckle noise (1%)	**0.97**	0.928	0.823	0.97
Average filter 3X3	**0.996**	0.939	0.988	0.974
Median filter 3X3	**1**	0.927	0.979	1
Gaussian LPF 3X3	**1**	0.938	1	1

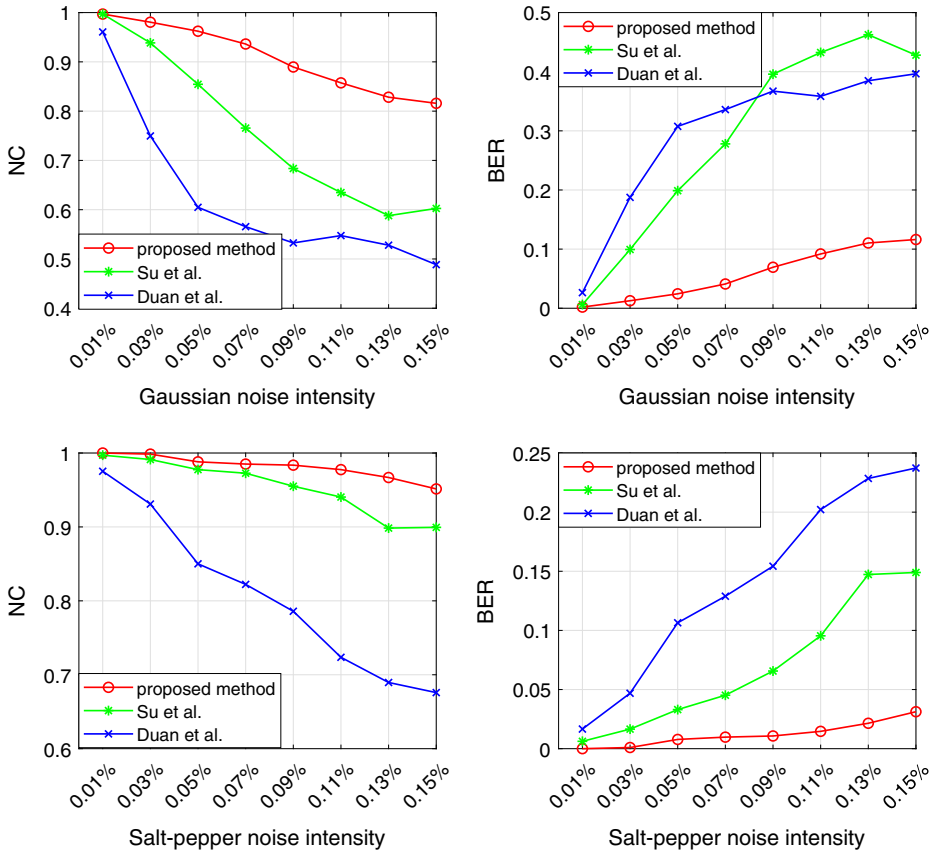

Fig. 15 The comparison with the Su et al.'s scheme [23] and Duan et al.'s method [4] after the Noise Attack

proposed scheme is better than all the schemes in [26]. For Salt-pepper noise, Average filter and Median filter, The proposed scheme is better than or close to the scheme in [26]. It proves that this scheme has better robustness.

Lossy compression technology is usually used to encode color images for efficient storage and communication. In this test, JPEG compression was used to attack watermark images. Figure 14 shows the comparison with the Su et al.'s scheme [23] after the JPEG attack. Figure 14 contains a comparison of four pictures of Lena, Baboon, Airplane, and Peppers. When $QF>0.5$, the NC values of four pictures are above 0.8. When $QF = 0.6$, $QF = 0.5$, $QF = 0.4$, the proposed scheme has a higher NC value than [23] and is more robust.

Noise attacks and filter attacks are classic attacks against digital watermarking. Watermarked images are easily and inevitably affected by noise during transmission. Moreover, filters of different sizes work on the watermarked image may also make the embedded watermark disappear. The test compares the proposed scheme with [4, 9, 23]. By adjusting

Table 4 Comparison of imperceptibility between PELM and PEE

Image	G	PELM		PEE	
		PSNR	SSIM	PSNR	SSIM
1	9	43.8594	0.9976	36.1916	0.987
2	11	45.087	0.9996	40.9857	0.9986
3	11	45.1389	0.9945	41.8918	0.9929
4	11	43.3243	0.9968	40.7986	0.9949
6	11	42.6116	0.995	37.1574	0.9877
7	12	43.3627	0.9933	40.6687	0.9915
9	13	44.5361	0.9869	40.8137	0.982
10	12	44.5267	0.9903	40.7111	0.9854
11	12	43.3925	0.9886	37.9594	0.9821
12	13	43.3442	0.9955	40.3149	0.9926
13	8	37.8733	0.9895	33.36	0.9791
14	10	38.9961	0.9851	36.2218	0.9786
15	10	41.5335	0.9927	38.8046	0.9887
16	11	44.6332	0.9922	39.8932	0.9846
17	9	43.7029	0.9884	39.725	0.9829
18	5	42.9471	0.9878	38.2945	0.9822
19	8	44.6347	0.9906	37.6839	0.9855
21	8	44.1641	0.9951	38.0721	0.9891
22	9	43.3562	0.9898	39.0899	0.9857
23	7	45.0703	0.9961	41.7806	0.9945
24	7	42.9827	0.9897	37.6653	0.9831
25	10	43.3584	0.9991	39.1369	0.9968
26	5	40.406	0.9968	32.6901	0.983
27	5	42.9555	0.999	39.2861	0.9972
28	8	45.0608	0.9849	41.6101	0.9808
avg	9.4	43.234	0.9926	38.8323	0.98746

the parameters of different algorithms, the imperceptibility of each algorithm is similar. In this case, the anti-noise attack and anti-filtering attack are tested. The comparison results are shown in Table 3. It can be seen that the NC value of the watermark extracted by the proposed scheme is better than [4, 9, 23] under Gaussian noise, Salt-pepper noise, Speckle noise, Average filter, Median filter, and Gaussian LPF attacks. The comparison with the schemes [4, 23] in terms of Gaussian noise and salt and pepper noise attacks is shown in Fig. 15. It can be seen from Fig. 15 that the NC values of the watermark extracted by the proposed scheme are all greater than [4, 23], the BER values are less than [4, 23]. This shows that the scheme proposed in this paper can effectively resist noise attacks and is better than [4, 23].

4.3 Watermark fragility test

Watermarked images may be subjected to malicious attacks In this paper, the PEE method is improved, and the PELM is proposed to reduce the impact on the invisibility of the watermark. Table 4 shows comparison of imperceptibility between PELM method and PEE. The PELM method proposed in this paper is superior to the PEE in both PSNR value and SSIM value.

Common malicious attacks mainly include random block attacks and object attacks. Random block attack is an attack on blocks of random size and location. Object attack adds or

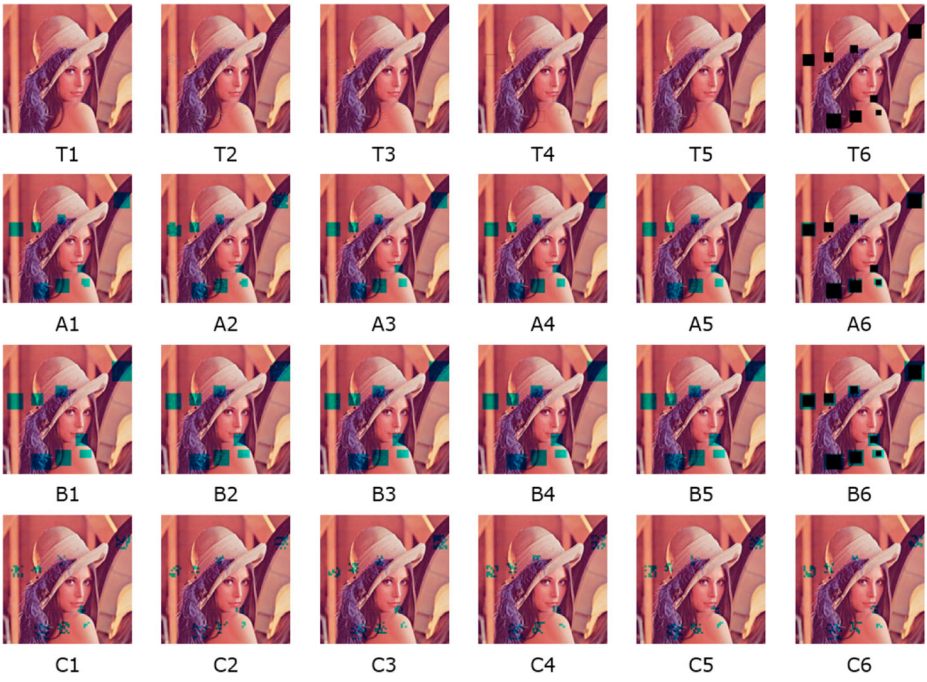

Fig. 16 Random block attack and tamper detection: (T1-T6) Images after random block attack: (T1) blurring, (T2) salt and pepper noise, (T3) Gaussian noise, (T4) average filter, (T5) sharpening, (T6) cropping. A1-A6 are tamper detection diagrams extracted from the scheme, B1-B6 are tamper detection diagrams extracted from Duan et al.'s scheme [4], C1-C6 are tamper detection diagrams extracted from Kamili et al.'s scheme [9]

deletes a specified area of an image. This paper proposes a tampering detection map correction program, and in the test, select $\alpha = 1, \beta = 0.4$ as the value in Formula (27) for experiment. Figure 16 shows Lena being attacked by random blocks. Figure 17 shows the results of object attacks. The comparison of tamper detection accuracy between the proposed scheme and the schemes of Duan et al. [4] and Kamili et al. [9] is shown in Fig. 18. When compared with [4, 9], except for the S&P random block attack, the ACC is slightly lower than other schemes, and the rest of the random block attack and object attack are better than the other two schemes.

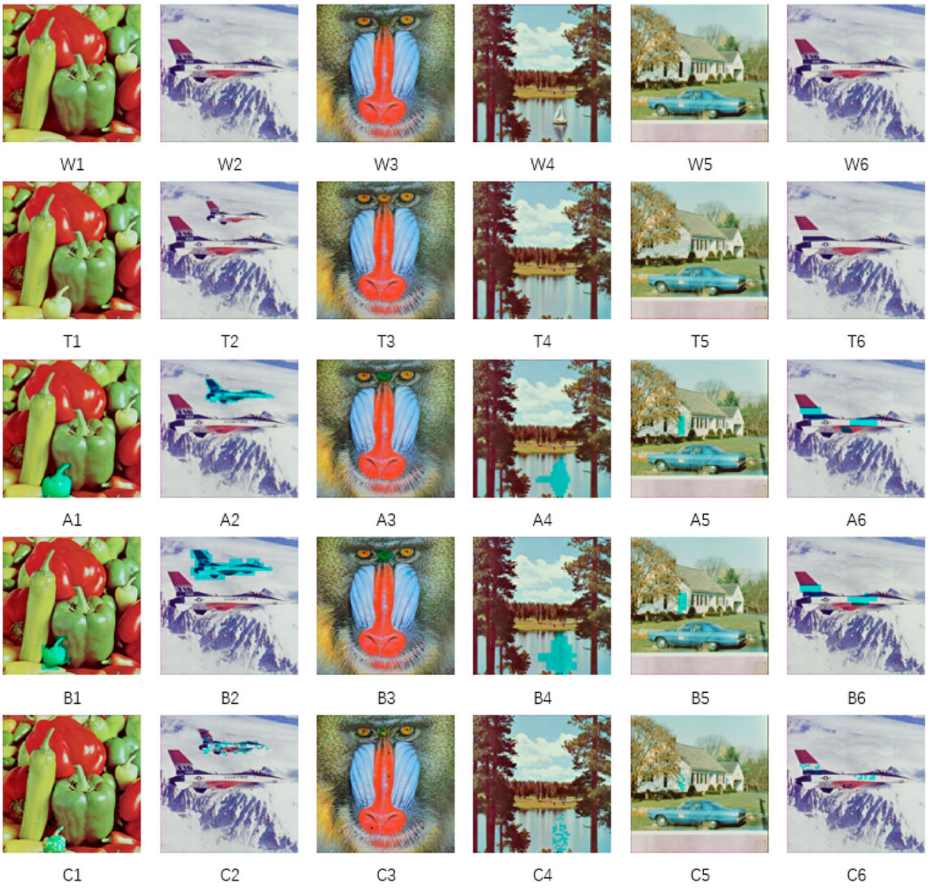

Fig. 17 Results of object attacks: (W1-W6) watermarked image (T1-T6) images after object attack: (T1) attacked image in "Peppers" by adding a new pepper, (T2) attacked image in "Airplane" by adding another airplane, (T3) attacked image in "Baboon" by adding an eye, (T4) attacked image in "Sail-boat" by removing the boat, (T5) attacked image in "House" by removing a window, and (T6) attacked image in "Airplane" by removing the "USA AIR FORCE", and (A1-A6) tamper detection diagrams extracted from the proposed scheme, (B1-B6) tamper detection diagrams extracted from Duan et al.'s scheme [11], (C1-C6) tamper detection diagrams extracted from Kamili et al.'s scheme [9]

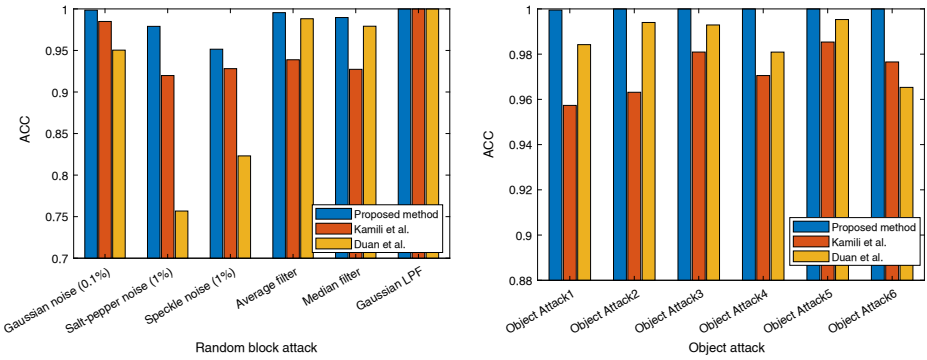

Fig. 18 The comparison with the Duan et al.'s scheme [4] and Kamili et al.'s scheme [9] after tamper detection

4.4 Watermark capacity

This paper analyzes the robust watermark capacity by embedding rate. The bits per pixel (bpp) is used to represent the pure payload (that is, the watermark capacity that can be embedded per pixel). The watermark in this paper is a 32×32 binary image, and the host image is a 512×512 color image, thus, the embedding capacity is $(32 \times 32)/(512 \times 512 \times 3) = 0.0013bpp$. The capacity of the method [4, 23] is the same as that of the proposed in this paper. The capacity of the method [9] is $(64 \times 64)/(512 \times 512 \times 3) = 0.0052bpp$. Obviously, the proposed method has the lowest capacity. This is because in order to achieve better robustness and invisibility, only one bit of watermark is embedded in a 16×16 image block. Therefore, in the follow-up work, we will consider how to embed more watermarks while maintaining robustness and invisibility.

5 Conclusion

This paper introduces a reversible multi-watermarking scheme for color images with robustness and fragility. The main work is as follows: 1) IWT and histogram shift are used to realize robust watermarking; 2) The pre-embedding method can select a more appropriate embedding location. In addition, the PEE algorithm is optimized by PELM to reduce the embedding capacity. These optimizations greatly improve the invisibility of the watermark; 3) a tamper detection map correction scheme is proposed to improve the accuracy of tamper detection.

The experimental results show that the average PSNR value is 43.234 dB, which has good imperceptibility. Meanwhile, it is robust to common attacks. Especially in noise attacks, it achieves the highest robustness compared to those proposed in literature [4, 9, 22]. When the watermarked Lena is attacked by 1% pepper-and-salt noise, the NC value is higher than 0.98. The accuracy of tamper detection reaches 99.9% under object attacks, which exceeds the schemes proposed by Kamili et al. [9] and Duan et al. [4].

However, there is still much room for improvement in our scheme. 1) After embedding the watermark, some pixels may overflow and cannot be further embedded. 2) The location map and PELM in this paper are lack of security considerations during storage and transmission. If they are attacked, the watermark cannot be extracted. Therefore, in the future

work, this problem will be solved by cryptography or hiding them into the image. 3) Pre-embedding consumes a lot of time. How to reduce the time overhead is also a work need to do in the future.

Acknowledgements The research was supported by Hainan Province Basic, Applied Basic Research Program (Natural Science Field) High-level Talent Project (Grant No. 2019RC044) and Hainan Province Key R&D plan project(No. ZDYF2022GXJS224).

Declarations

Conflict of Interests The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Ahmadi SBB, Zhang G, Rabbani M, Boukela L, Jelodar H (2021) An intelligent and blind dual color image watermarking for authentication and copyright protection. Appl Intell 51:1701–1732
2. Ansari IA, Pant M, Ahn CW (2017) Artificial bee colony optimized robust-reversible image watermarking. Multimed Tools Appl 76:18001–18025
3. Darwish SM, Hassan OF (2020) A new colour image copyright protection approach using evolution-based dual watermarking. Journal of Experimental & Theoretical Artificial Intelligence, pp 1–23
4. Duan S, Wang H, Liu Y, Huang L, Zhou X (2020) A novel comprehensive watermarking scheme for color images. Security and Communication Networks, 2020
5. Fan G, Pan Z, Gao E, Gao X, Zhang X (2021) Reversible data hiding method based on combining IPVO with bias-added rhombus predictor by multi-predictor mechanism. Signal Process 180:107888
6. Girdhar A, Kumar V (2019) A reversible and affine invariant 3D data hiding technique based on difference shifting and logistic map. Journal of Ambient Intelligence and Humanized Computing
7. Hassan FS, Gutub A (2021) Efficient image reversible data hiding technique based on interpolation optimization. Arab J Sci Eng, pp 1–16
8. Hurrah NN, Parah SA, Loan NA, Sheikh JA, Elhoseny M, Muhammad K (2019) Dual watermarking framework for privacy protection and content authentication of multimedia. Future Generation Comput Syst 94:654–673
9. Kamili A, Hurrah NN, Parah SA, Bhat GM, Muhammad K (2020) DWFCAT: Dual watermarking framework for industrial image authentication and tamper localization. IEEE Transactions on Industrial Informatics 17:5108–5117
10. 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
11. Kumar R, Jung K-H (2020) Robust reversible data hiding scheme based on two-layer embedding strategy. Inf Sci 512:96–107
12. Kumar C, Singh AK, Kumar P (2019) Dual watermarking: an approach for securing digital documents. Multimed Tools Appl, pp 1–16
13. Lee C-F, Shen JJ, Lai YH (2018) Data hiding using multi-pixel difference expansion. In: 2018 3Rd international conference on computer and communication systems (ICCCS). IEEE, pp 56–60
14. Li X, Li J, Li B, Yang B (2013) High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Process 93:198–205
15. Liang X, Xiang S (2020) Robust reversible audio watermarking based on high-order difference statistics. Signal Process 173: 107584
16. Meng L, Liu L, Tian G, Wang X (2021) An adaptive reversible watermarking in IWT domain. Multimed Tools Appl 80:711–735
17. Mohanarathinam A, Kamalraj S, Venkatesan GP, Ravi RV, Manikandababu C (2019) Digital watermarking techniques for image security: a review. Journal of Ambient Intelligence and Humanized Computing, pp 1–9
18. Parah SA, Sheikh JA, Akhoon JA, Loan NA (2020) Electronic Health Record hiding in Images for smart city applications: A computationally efficient and reversible information hiding technique for secure communication. Futur Gener Comput Syst 108:935–949
19. Qasim AF, Meziane F, Aspin R (2018) Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review. Comput Sci Rev 27:45–60

20. Roy S, Pal AK (2018) A robust reversible image watermarking scheme in DCT domain using Arnold scrambling and histogram modification. Int J Inf Comput Secur 10:216–236
21. Sachnev V, Kim HJ, Nam J, Suresh S, Shi YQ (2009) Reversible watermarking algorithm using sorting and prediction. IEEE Trans Circuits Syst Video Technol 19:989–999
22. Singh AK (2019) Robust and distortion control dual watermarking in LWT domain using DCT and error correction code for color medical image. Multimed Tools Appl 78:30523–30533
23. Su Q, Chen B (2018) Robust color image watermarking technique in the spatial domain. Soft Comput 22:91–106
24. Sweldens W (1998) The lifting scheme: a construction of second generation wavelets. SIAM J Math Anal 29:511–546
25. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circ Syst Vid Technol 13:890–896
26. Valandar MY, Barani MJ, Ayubi P (2020) A blind and robust color images watermarking method based on block transform and secured by modified 3-dimensional hénon map. Soft Comput 24:771–794
27. Wang W (2020) A reversible data hiding algorithm based on bidirectional difference expansion. Multimed Tools Appl 79:5965–5988
28. Wang H, Li X, Xiao M, Zhao Y (2021) A novel robust reversible watermarking method against JPEG compression. International conference on artificial intelligence and security, pp 312–322, Springer
29. Wang W, Ye J, Wang T, Wang W (2017) Reversible data hiding scheme based on significant-bit-difference expansion. IET Image Process 11:1002–1014
30. Yue Z, Ding S, Zhao L, Zhang Y, Cao Z, Tanveer M, Jolfaei A, Zheng X (2019) Privacy-preserving time series medical images analysis using a hybrid deep learning framework. ACM Trans Internet Technol 37:1–22
31. Zhang L, Wei D, Dual DCT-DWT-SVD (2019) Digital watermarking algorithm based on particle swarm optimization. Multimed Tools Appl 78:28003–28023
32. Zhang Z, Wu L, Xiao S, Gao S (2017) Adaptive reversible image watermarking algorithm based on IWT and level set. EURASIP J Adv Signal Process 2017:1–16
33. Zhang J, Zhou X, Yang J, Cao C, Ma J (2019) Adaptive robust blind watermarking scheme improved by entropy-based SVM and optimized quantum genetic algorithm. Math Probl Eng, 2019
34. Zhou X, Cao C, Ma J, Wang L (2018) Adaptive digital watermarking scheme based on support vector machines and optimized genetic algorithm. Math Probl Eng, 2018
35. Zhou K, Ding Y, Bi W (2021) High-capacity PVO-based reversible data hiding scheme using changeable step size. Multimed Tools Appl 80:1123–1141
36. Zhou X, Ma Y, Zhang Q, Mohammed MA, Damaševičius R (2021) A reversible watermarking system for medical color images: Balancing capacity, imperceptibility, and robustness. Electronics 10:1024