Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Reversible Authentication Watermarking Based on Improved 2D Histogram and Adaptive Difference Expansion

Zhengwei Zhang[*], **Xiu Li, Hao Yue, Fenfen Li**

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, Jiangsu, 223003, China
E-mail: zzw49010650@sina.com

Received: 6 July 2025; **Revised:** 11 August 2025; **Accepted:** 13 August 2025

Abstract: To address the limitations of low authentication accuracy and ineffective protection for complex-texture images/regions in existing reversible schemes, an improved algorithm based on two-Dimensional (2D) histogram and difference expansion is proposed. The core innovation involves classifying image pixels into texture and smooth categories using local complexity analysis. Distinct embedding strategies are then applied: texture pixels utilize adaptive prediction difference expansion for authentication watermark embedding, while smooth pixels utilizes channel shifting within the improved 2D histogram for authentication watermark embedding. Furthermore, a hierarchical embedding strategy enhances both authentication effectiveness and visual quality per sub-block. Experimental results demonstrate that the method achieves an average tamper detection rate of 92.78% (using 16×16 blocks) under complex attacks while maintaining PSNR above 46 dB. Compared to state-of-the-art methods, it significantly improves tamper localization for complex-texture content without compromising reversibility or visual fidelity.

Keywords: image authentication, tamper detection, difference expansion, two-Dimensional (2D) histogram, reversible image watermarking

MSC: 94A08, 94A15

1. Introduction

With the rapid development of the Internet and multimedia technology, digital images, audio, and video works have been all over every corner of People's daily life. However, the rapid development of information technology also enables criminals to easily tamper with, attack, and forge images. Ensuring the authenticity and integrity of images during storage and transmission has thus become an urgent problem to be solved [1, 2]. For instance, the illegal tampering of image content in fields such as military, remote sensing, and medical can have a significant impact on society. Therefore, protecting the authenticity, integrity, and reliability of image content in the massive multimedia data environment is of great significance. Based on this, the reversible image watermarking authentication algorithm [3] is proposed, which can detect the authenticity and integrity of the image content according to the authentication watermarking tampering, and judge whether the image is maliciously tampered with.

Various studies have been proposed for image authentication algorithms. Lo et al. [4] proposed an image authentication algorithm based on the shift of predictive difference histograms [5]. The algorithm first blocks the image,

DOI: https://doi.org/10.37256/cm.6520257735

embeds the authentication information into the original image by using the information hiding method of histogram shift, and then obtains the authentication image. The proposed scheme achieves good detection accuracy while maintaining the embedded image quality and good detection accuracy. Wang et al. [6] constructed two-dimensional histograms of the original image in the checkerboard structure, which selects the embedded channel from the preset parameters and determines the embedded channel peak point position, which will shift the embedded channel. Then the authentication information is embedded in the image block combined with the histogram information embedding method. In the process of the tampering detection, the layered tampering detection method is adopted to effectively improve the accuracy of the tampering detection. Lee et al. [7] used double watermarking for tampering detection and recovery. The algorithm beds both types of watermarks into the original image, so once one of the watermarks is destroyed, it can also recover the altered image according to the other. Yin et al. [8] further proposed that first, the Hilbert curve [9] was used to scan the image, then divide the resulting pixels into N subblocks, and finally use the improved pixel value sorting (Improved Pixel Value Ordering (IPVO)) [10] method to embed the N bit authentication code into the N subblocks. Since the conventional attack does not change the prediction error histogram corresponding to the subblock, it does not affect the extraction of authentication information, which may cause the tampering to not be effectively detected. Li et al. [11] proposed a reversible watermark algorithm based on wavelet transform to realize image content authentication. By the wavelet transform of image blocks, reduce the impact of tampering on the whole, block as the watermark carrier. The algorithm uses different keys to enhance security and randomness. Moreover, the accuracy of tampering detection is improved by multilevel detection and screening. This method effectively combines the watermark technology and the wavelet transform and is suitable for the application scenarios with high security requirements. Nguyen et al. [12] proposed the first independent certification method to solve the problem of non-independent certification scheme. This method performs two discrete wavelet transforms (Discrete Wavelet Transform (DWT)) [13] for each 8×8-sized subblock and beds 3-bit authentication code in the low-frequency subband using the prediction error extension technique. Because the authentication code is embedded in the DWT coefficient, there is usually no non-embedded block, but this method still has the problems of too large authentication unit and weak tamper localization ability. Hong et al. [14] combined the Least Significant Bit (LSB) [15] replacement method with the IPVO method to realize the authentication of non-embedded sub-blocks, but some pixels in the block can not be effectively verified, and there are some security risks.

Based on the above studies, this paper presents a reversible authentication scheme based on an improved 2D histogram and adaptive difference extension. Through the local complexity pixels are divided into two categories, for the texture of complex pixels using the adaptive difference extension method, according to the different texture features adaptive embedding strategy, make the watermark effectively embedded under various complex textures, effectively solving the existing algorithm cannot have complex texture image or image of the texture complex area realize effective protection problem. At the same time, difference extension can also help resist common attacks, such as compression or cropping, to improve watermark robustness; for low local contrast of smooth pixels, using eight neighborhood prediction and median prediction to generate 2D histograms to greatly increase the embedding capacity. In this paper, the four layers can enhance image processing and tamper attack resistance independently by embedding four layers, so that even if one layer is damaged, the other three layers can still maintain effectiveness and pixel correlation. Hierarchical embedding can also reduce the impact of the watermark on the original image, so as to effectively avoid the problems of the image visual quality reduction caused by embedding authentication information, and maintain a high visual quality.

2. Knowledge background

This paper investigates a reversible image authentication algorithm, which primarily employs the improved 2D histogram and adaptive difference expansion to embed the authentication watermark, thereby enabling accurate detection of image tampering and identification of the tampered locations. The fundamental knowledge involved in this algorithm includes 2D histogram, difference extension, local complexity calculation, and pixel classification.

2.1 *2D histogram*

The reversible data hiding based on 2D prediction error was proposed by Wang et al. [16] in 2013. The algorithm utilizes "four-neighborhood prediction" and "left-right prediction" to construct a 2D histogram [17]. The horizontal and vertical coordinates represent different prediction error values, and their 2D histogram is divided into many channels, and each channel corresponds to a one-dimensional histogram, as shown in Figure 1. Compared with the traditional histogram, this method can choose multiple channels to embed watermark, so the embedding capacity can be greatly improved. In the 2D histogram, it is divided into different channels. Its parameters determine the number of embedding channels, the capacity of embedded watermark information, and the quality of watermarked images. Specifically, the parameter c_b determines the number of embedding channels, the capacity of embedded watermark information, and the quality of watermarked images. If $c_b = 3$, the embeddable channels are c = -3, -2, -1, 0, 1, 2, 3, which means the channel selection is symmetric. In this paper, $c_b = 1$ is selected, so the embedding channels are c = 1, 0, -1, which indicates that the embeddable channels are symmetric.

However, when constructing the 2D histogram, since the embedding prediction error method adopted by Wang et al. uses four-neighborhood prediction and left-right prediction, it fails to make better use of the correlation between adjacent pixels, thereby reducing the embedding capacity of the low channels in the 2D histogram.

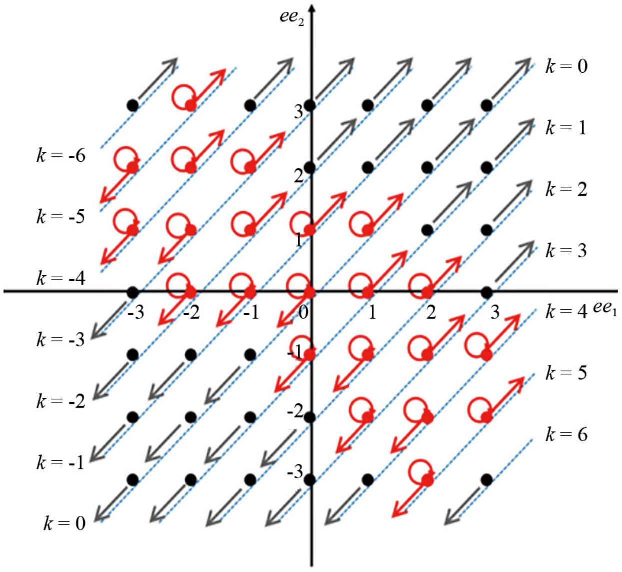


Figure 1. Two-dimensional histogram

Based on the above problems, this algorithm is improved. Then the original image is first divided into non-overlapping 6 × 6 sub-blocks, and then the pixels in the subblocks are divided into four sets as shown in Figure 2, namely "×" pixel set, "+" pixel set, "•" pixel set, and "○" pixel set. Let the eight neighborhood pixels adjacent to the predicted pixels be $\{x_1,\ x_2,\ \cdots\cdots,\ x_8\}$, and order them from small to large, that is, assuming the small to large rank as $\{x_{\sigma 1} \leq x_{\sigma 2} \leq \cdots\cdots x_{\sigma 8}\}$, and the subscript σ_i $(0 \leq i \leq n)$ is the initial position of the prediction error in the initial sequence $\{x_1,\ x_2,\ \cdots\cdots,\ x_8\}$. Then Formula (1) is used to obtain the prediction value P_1, the formula is not affected by the maximum value, minimum value, and part of the data changes, which can better describe the central trend of this group of data and achieve a better prediction effect. Another predicted value also adopts the more advantageous eight-neighborhood prediction, and P_2 is obtained by using the predicted value of Formula (2). By improving the way of constructing a two-dimensional histogram, the four-neighborhood prediction and left and right prediction are changed to median prediction and eight-neighborhood prediction, so as to make better use of the correlation of adjacent pixels and improve the embedding capacity of low channels in the 2D histogram.

$$P_1 = \left\lfloor \frac{x_{\sigma_4} + x_{\sigma_5}}{2} \right\rfloor \tag{1}$$

$$P_2 = \left\lfloor \frac{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8}{8} \right\rfloor . \tag{2}$$

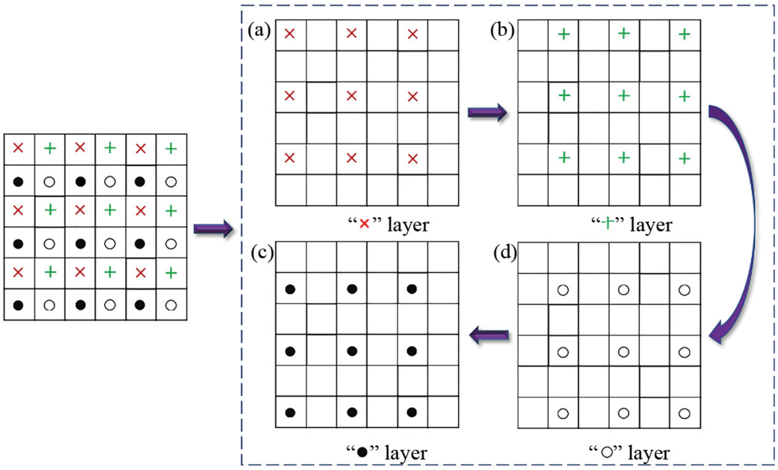


Figure 2. Hierarchical embedding strategy

2.2 *Difference extension*

Zhong et al. [18] proposed an efficient image reversible authentication method in 2023. The algorithm is divided into differential blocks and translation blocks based on the embeddable capacity of each subblock. The difference block embedding authentication information adopts the difference extension based on the prediction error, that is, the prediction error is first ranked, then chooses the extension direction according to the adaptation of the pixel gray value, and finally use the difference between the prediction errors for expansion. Compared with the traditional reversible image authentication algorithm, the proposed method uses the adaptive difference extension of prediction error to have higher embedding capacity and can realize the effective protection of complex texture images, and has higher ability of tamper detection and positioning. However, the algorithm does not fully take into account the positive and negative properties of each group of generated prediction error values when choosing the expansion direction, which is easy to cause residual stacking in the difference expansion process, leading to the failure to extract watermark and reducing the tampering detection rate.

Therefore, this paper uses two different methods to generate two groups of prediction error values, and obtains the best prediction error groups according to the comparison, so as to avoid the residual stacking in the difference expansion process and improve the tampering detection rate. First, the two groups of prediction error values were ranked from small to large, and the Formula (3) was used to determine the best prediction error group. Then, based on the complexity of positive and negative in the best prediction error group, it is proposed to embed the best prediction error group conditionally. In the first two cases, it expands upward and downward according to the positive and negative properties of the prediction error value, and in the third case, the secondary prediction error value is further calculated, so as to further improve the tampering detection rate. The specific steps are detailed in Section 3. In this paper, the best prediction error group adaptive difference extension method is adopted to embed authentication watermark, which fully considers the positive and negative nature of each group of prediction error value, avoids the generation of residual stacking, and improves the tampering detection rate.

$$\begin{cases} NL_1 = e^1_{\sigma(7)} - e^1_{\sigma(2)} \\ NL_2 = e^2_{\sigma(7)} - e^2_{\sigma(2)}. \end{cases} \tag{3}$$

2.3 *Local complexity calculation and pixel classification*

To effectively utilize the characteristics of pixels, the local complexity is used to classify pixels into texture pixels and smooth pixels, thus improving the embedding capacity. The pixel distribution is shown in Figure 3, and the local complexity $C(x)$ of the pixel x can be calculated by Formula (4). Take the "×" layer, its local complexity is calculated based on the eight neighborhood $\{x_1, x_2, \cdots\cdots, x_8\}$ of its neighboring pixels. Compared with other adjacent four neighbors to find local complexity, eight neighbors can make full use of the correlation of pixels and have more accurate results.

$$C(x) = \sqrt{\sum_{i=1}^{8} (\partial_i - \bar{X})^2 / 8} \tag{4}$$

∂_i is the difference value between adjacent pixels, $\partial_1 = |a-b|$, $\partial_2 = |b-c|$, $\partial_3 = |c-d|$, $\partial_4 = |d-e|$, $\partial_5 = |e-f|$, $\partial_6 = |f-g|$, $\partial_7 = |g-h|$, $\partial_8 = |h-a|$; $\bar{X} = (\partial_1 + \partial_2 + \partial_3 + \partial_4 + \partial_5 + \partial_6 + \partial_7 + \partial_8)/8$.

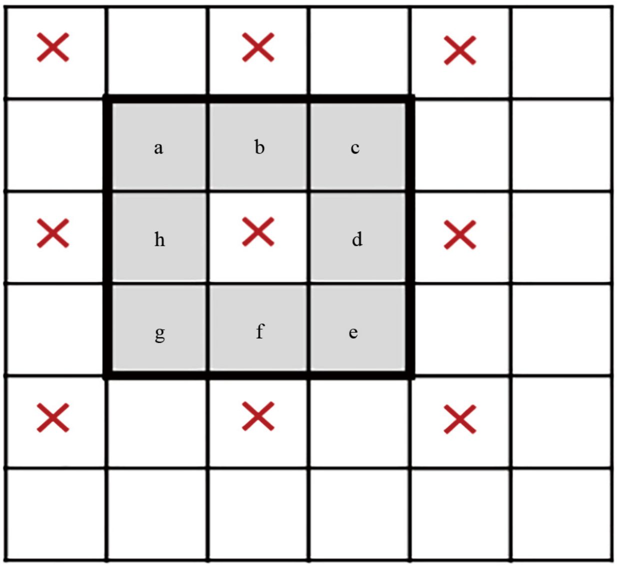


Figure 3. Schematic diagram of the local complexity calculation

Pixels are divided into smooth pixel subsets based on their pixel local complexity:

$$S_k = \{s_1, s_2, s_3, s_4\}, \ 1 \le k \le Ns. \tag{5}$$

Texture pixel subset:

$$C_t = \{c_1,\ c_2,\ c_3,\ c_4,\ c_5,\ c_6,\ c_7,\ c_8\}, 1 \le t \le Nc. \tag{6}$$

N_s and N_c are the total number of smooth and texture pixel subsets, respectively. The smooth pixel subset takes 4 pixels as a subset, such as Formula (5); the texture pixel subset takes 8 pixels as a subset, such as Formula (6), increases the number of elements of the texture pixel subset, which is mainly due to the prediction error of texture pixels is usually large. When sorting the prediction error of these elements, increasing the number of elements can usually reduce the high distortion of the embedded authentication information. Since this paper is embedded in the subplane, the change of one plane pixel does not affect the other plane, which guarantees the reversibility of the algorithm. Therefore, the complexity values remain unchanged during the reverse watermark extraction.

3. Image authentication algorithm

The process of authentication watermark embedding in this paper is shown in Figure 4. In the authentication of watermark embedded process, the size of $M \times N$ original images is first divided into no overlapping 6×6 subblocks, and then the pixels in the subblocks are divided into four sets, namely "$\times$" pixel set, "$+$" pixel set, "$\bullet$" pixel set and "$\circ$" pixel set. According to their local complexity, the layer pixels are divided into smooth pixels and texture pixels. Taking the "$\times$" layer as an example, the subset of texture pixels embeds the authentication information by the prediction difference adaptive difference extension, the smooth pixel subset uses the channel translation embedding authentication information in the 2D histogram, and the other "$+$" layer, "$\bullet$" layer and "$\circ$" layer also use similar methods to embed the authentication information.

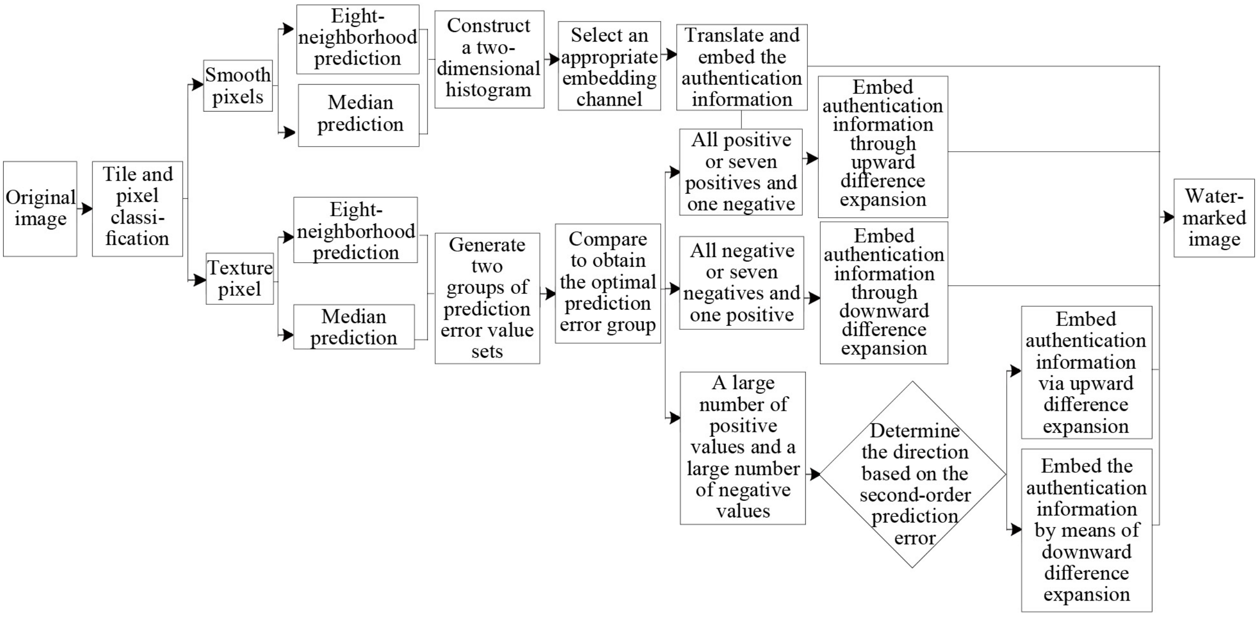

Original image
Tile and pixel classi-fication
Smooth pixels
Texture pixel
Eight-neighborhood prediction
Median prediction
Construct a two-dimensional histogram
Select an appropriate embedding channel
Translate and embed the authentication information
Eight-neighborhood prediction
Median prediction
Generate two groups of prediction error value sets
Compare to obtain the optimal prediction error group
All positive or seven positives and one negative
All negative or seven negatives and one positive
A large number of positive values and a large number of negative values
Embed authentication information through upward difference expansion
Embed authentication information through downward difference expansion
Determine the direction based on the second-order prediction error
Embed authentication information via upward difference expansion
Embed the authentication information by means of downward difference expansion
Water-marked image

Figure 4. Embedding process of the authentication code

3.1 *Texture pixel authentication code is embedded*

In order to minimize distortion and overflow risks caused by differential expansion when embedding watermark into texture pixels as much as possible, a method is adopted that calculates two groups of prediction error values, sorts them according to the error values, and selects the best group of prediction errors. Furthermore, since the predicted error values have a positive and negative nature, it is necessary to process the optimal predicted error group before embedding authentication information to reduce distortion.

The specific approach is shown as follows.

Step 1: Take the subset of "×" layer texture as an example, first calculate the two pixel prediction values P_2, P_1 by using Formulas (1) and (2) in Section 2.1.

Step 2: Calculate the two predicted values P_1, P_2, and then subtract them from the pixel value $I_{\text{ij}}((i,\ j)$ is the pixel position) respectively, as shown in the Formula $e_1 = I_{ij} - P_1,\ e_2 = I_{ij} - P_2$. Taking eight pixel prediction error values as a group, we get two sets of prediction error groups $E_1 = \left\{e_1^1,\ e_2^1,\ \cdots\cdots,\ e^1{}_8\right\}$ and $E_2 = \left\{e_1^2,\ e_2^2,\ \cdots\cdots,\ e^2{}_8\right\}$, and then rank the two sets of prediction error values from small to large, and get $\left\{e_{\sigma(1)}^1,\ e_{\sigma(2)}^1,\ \cdots\cdots,\ e_{\sigma(8)}^1\right\}$ and $\left\{e_{\sigma(1)}^2,\ e_{\sigma(2)}^2,\ \cdots\cdots,\ e_{\sigma(8)}^2\right\}$.

Step 3: Two groups of prediction error groups E_1, E_2 are obtained from the previous step, and the best prediction group is determined according to the Formula (3) in Section 2.2. If $NL_1 \le NL_2$, E_1 is selected as the optimal set of embedding prediction error values EE. If $NL_1 > NL_2$, E_2 is selected as the optimal set of embedding prediction error values EE, and choosing $e_{\sigma(7)}^1,\ e_{\sigma(2)}^1,\ e_{\sigma(7)}^2,\ e_{\sigma(2)}^2$ in its formula can effectively avoid the deviation caused by too large maximum and too small minimum. Selection of "Optimal Prediction Error Group", as shown in Figure 5 below.

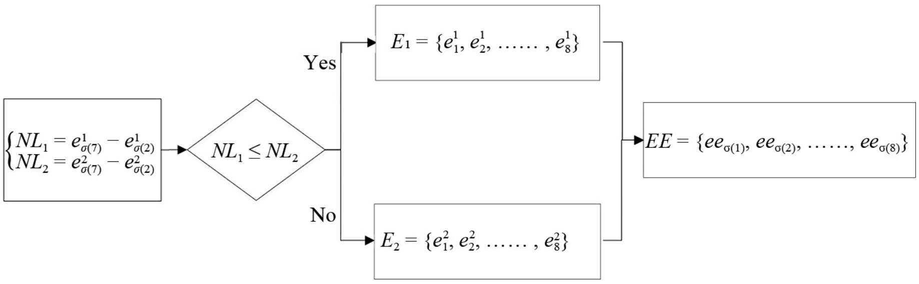

$\begin{cases} NL_1 = e_{\sigma(7)}^1 - e_{\sigma(2)}^1 \\ NL_2 = e_{\sigma(7)}^2 - e_{\sigma(2)}^2 \end{cases}$
$NL_1 \le NL_2$
Yes
No
$E_1 = \{e_1^1, e_2^1, \ldots\ldots, e_8^1\}$
$E_2 = \{e_1^2, e_2^2, \ldots\ldots, e_8^2\}$
$EE = \{ee_{\sigma(1)}, ee_{\sigma(2)}, \ldots\ldots, ee_{\sigma(8)}\}$

Figure 5. Selection of "optimal prediction error group"

Step 4: Generate the authentication codes b_1 by using the hash function *Hash* (*block*, *row*, *col*, *key*). Where the *block* represents the relevant information of the pixels contained in the current block, *row*, and *col* represent the rows and columns where the current block is located, respectively, and the *key* is the key previously agreed upon by the sender and the receiver.

Step 5: However, since the generated optimal prediction error group values $EE = \left\{ee_{\sigma_{(1)}},\ ee_{\sigma_{(2)}}\ \cdots\cdots,\ ee_{\sigma_{(8)}}\right\}$ have both positive and negative values, to avoid sign changes during the difference expansion process, the optimal prediction error value group is embedded in a case-by-case manner to reduce distortion.

The positive and negative conditions of the best prediction error group value can be divided into three scenarios: the first scenario is all positive or 7 positive and 1 negative, as shown in Formula (7); the second scenario is all negative or 7 negative and 1 positive, as shown in Formula (8); the third scenario is multiple positives and negatives, as shown in Formula (9). Where P represents the case of set $S \ge 0$ and N represents the case of set $S < 0$.

$$S=\left\{x \mid x \in\left\{ee_{\sigma_{(1)}},\, ee_{\sigma_{(2)}},\, \cdots\cdots,\, ee_{\sigma_{(8)}}\right\},\, x \geq 0\right\} \text{ or}$$

$$S=\left\{x \mid x \in\left\{ee_{\sigma_{(1)}},\, ee_{\sigma_{(2)}},\, \cdots\cdots,\, ee_{\sigma_{(7)}}\right\},\, x \geq 0\right\} \cup\left\{x \mid x \in\left\{ee_{\sigma_{(8)}}\right\},\, x<0\right\} \tag{7}$$

$$S=\left\{x \mid x \in\left\{ee_{\sigma_{(1)}},\, ee_{\sigma_{(2)}},\, \cdots\cdots,\, ee_{\sigma_{(8)}}\right\},\, x \leq 0\right\} \text{ or}$$

$$S=\left\{x \mid x \in\left\{ee_{\sigma(1)},\, ee_{\sigma(2)},\, \cdots\cdots,\, ee_{\sigma(7)}\right\},\, x \leq 0\right\} \cup\left\{x \mid x \in\left\{ee_{\sigma(8)}\right\},\, x>0\right\} \tag{8}$$

$$S=\left\{x \mid x \in ee_{\sigma(1)},\, ee_{\sigma(2)},\, \cdots\cdots,\, ee_{\sigma(8)}\right\} \tag{9}$$

where, $P=\{x \mid x \in S,\ x \geq 0\}$, $N=\{x \mid x \in S,\ x<0\}$ is the indicator function.

Step 6: The finally obtained group of optimal prediction error values is embedded the watermark according to different scenarios.

In the first case, when the values in the optimal prediction error group are as shown in Formula (7), that is, all positive or 7 positive and 1 negative, where most of the values in the optimal prediction error group are positive. To avoid the sign of the prediction error value changing after the difference expansion, the upward expansion is chosen, that is, the corresponding modified pixel value is increased, thus ensuring that the same sign can be subtracted and ensuring that the maximum value remains the maximum after the upward expansion and the sign does not change, as shown in Formula (10).

$$\begin{cases} h=e_{\sigma(n)}-e_{\sigma(n-1)} \\ H=2h+\mathrm{b}_1 \\ e'_{\sigma_{(n)}}=e_{\sigma_{(n-1)}}+H \end{cases} \tag{10}$$

where h represents the difference value of $e_{\sigma(n)}$, $e_{\sigma(n-1)}$, H is the extended value of h, $b_1 \in \{0,\ 1\}$ represents the 1-bit authentication code to be embedded, and $e'_{\sigma(n)}$ is the extended value of $e_{\sigma(n)}$.

In the second case, when the value in the optimal prediction error group is shown in Formula (8), that is, all negative or 7 negative 1 positive, most of the values in the optimal prediction error group are negative. In order to avoid the positive and negative changes of the prediction error value after the difference extension, the choice is to expand downwards, that is, the corresponding modified pixel value is reduced, thus ensuring that the same sign can be subtracted and ensuring that the minimum value remains the minimum after the downward expansion and the sign does not change, as shown in Formula (11).

$$\begin{cases} h=e_{\sigma(2)}-e_{\sigma(1)} \\ H=2h+\mathrm{b}_1 \\ e'_{\sigma(1)}=e_{\sigma(2)}-H. \end{cases} \tag{11}$$

In the third case, when the value in the optimal prediction error value group is shown in Formula (9), that is, more negative and more positive, that is, the positive and negative situation in the optimal prediction error group is more complex, the quadratic prediction error is used. At this point, let $EE = \left\{ee_{\sigma(1)},\ ee_{\sigma(2)}\cdots\cdots,\ ee_{\sigma(n)}\right\}$, $n = 8$, which indicates the optimal prediction error value after the judgment.

By applying Formulas (12) and (13), in the first case, if $ee_{down} \leq ee_{up}$, meaning the difference between the maximum and the second-largest values in the optimal prediction error is greater than the difference between the second-smallest and the minimum values, it indicates that expanding the optimal group upwards can better ensure that the sign of the prediction error value is not altered, as shown in (14). In the second case, if $ee_{down} > ee_{up}$, meaning the difference between the maximum and the second-largest values in the optimal prediction error is smaller than the difference between the second-smallest and the minimum values, the expansion should be downwards, as shown in Formula (15).

$$ee_{\text{down}} = ee_{\sigma(2)} - ee_{\sigma(1)} \tag{12}$$

$$ee_{up} = ee_{\sigma(n)} - ee_{\sigma(n-1)} \tag{13}$$

$$\begin{cases} h = ee_{up} - ee_{down} \\ H = 2h + \mathrm{b}_1 \\ ee'_{\sigma(n)} = ee_{down} + H + ee_{\sigma(n-1)} \end{cases} \tag{14}$$

$$\begin{cases} h = ee_{down} - ee_{up} \\ H = 2h + \mathrm{b}_1 \\ ee'_{\sigma(1)} = ee_{\sigma(2)} - ee_{up} - H. \end{cases} \tag{15}$$

Step 7: Calculate the pixel value after obtaining the embedded authentication information according to the formula $P_i' = \bar{P}_i + e'_{\sigma(n)}\left(\text{ or ee}'_{\sigma(n)}\right)$.

Step 8: For the other three layers, repeat Step 2 to Step 7 to obtain the watermarked image.

3.2 *The embedding of the smooth pixel authentication code*

In the process of smooth pixel embedded watermark, two sets of prediction errors are generated by using eight neighborhood prediction and median prediction, and then two-dimensional histogram translation embedded authentication information is constructed. The specific steps are as follows.

Step 1: Take the smooth pixel marked as "×" as an example. The Formulas (1) and (2) in Section 2.1 are used to calculate P_1 and P_2, and then subtract them from the pixel value $I_{\text{ij}}((i,\ j)$ is the pixel position) respectively, as shown in Formulas (16) and (17), to calculate $e_1,\ e_2$.

$$e_1 = I_{ij} - P_1 \tag{16}$$

$$e_2 = I_{ij} - P_2. \tag{17}$$

Step 2: Build a 2D histogram $H(e_1,\ e_2)$ according to the prediction error values obtained $e_1,\ e_2$ by the two different methods.

Step 3: In the 2D histogram, it is divided into different channels by $c = e_1 - e_2$. As described in Section 2.1, the parameter c_b determines the number of embedded channels, the capacity of the embedded watermark information and the quality of the watermarked image. If $c_b = 3$, the embedding channels $c = -3,\ -2,\ -1,\ 0,\ 1,\ 2,\ 3$, i.e., the channel selection is symmetric. In this paper, select $c_b = 1$, that is, the embedded channel is $c = 1,\ 0,\ -1$.

Step 4: Generate the authentication code b_2 by using the hash function $Hash$ ($block$, row, col, key).

Step 5: The authentication code information is embedded into the peak point of each channel, requiring to find the two peak points with the highest frequency on each channel. Suppose that their corresponding abscissa are $p1$ and $p2$, and $p1$ values are less than $p2$ values. Then for channel c, the corresponding two peak point coordinates can be expressed as $(p_1,\ p_1 - c)$ and $(p_2,\ p_2 - c)$.

Step 6: Each embeddable channel c is translated, and the specific embedding mode is shown in Formula (18). That is, by translating one unit to the lower left, both e_1 and e_2 are reduced by 1, and the corresponding host image pixel values of the difference value of $(e_1,\ e_2)$ are also reduced by 1. Similarly, by translating one unit to the upper right, that is, both e_1 and e_2 are added by 1, and the corresponding host image pixel values of the difference value of $(e_1,\ e_2)$ are also added by 1. The new coordinates after translation on the final channel can be expressed as $(e'_1,\ e'_2)$, and the pixel values in the original image can be expressed as $x'(i,\ j)$.

$$e_2 = e_1 - c \begin{cases} \text{if } e_1 < p_1, \text{ by translating one unit to the lower left} \\ \\ \text{if } e_1 > p_1, \text{ by translating one unit to the upper right.} \end{cases} \tag{18}$$

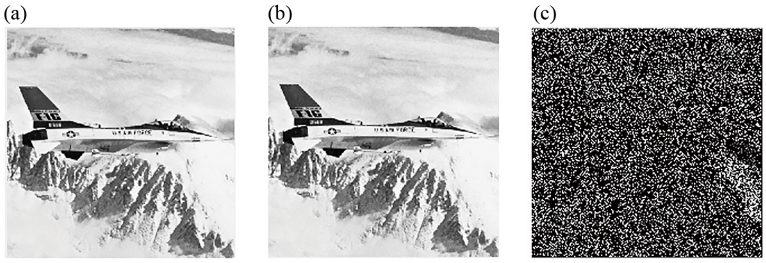
Step 7: The original image is divided into 6×6 sizes, and the watermark is embedded into each embedded channel in each subblock. The specific embedding process is as follows: If the corresponding difference $(e_1,\ e_2)$ of the pixel $x(i,\ j)$ in the subblock meets $e_2 = e_1 - c$ and $e_1 = p_1$, then a watermark information is embedded in the pixel position, that is, the pixel value is $x'(i,\ j) = x(i,\ j) - b_2$ (Here, $x'(i,\ j)$ represents the pixel values of the original image after translation, and b_2 represents a single watermark information, which has a value of 0 or 1); if the corresponding difference $(e_1,\ e_2)$ of the pixel $x(i,\ j)$ in the subblock meets $e_2 = e_1 - c$ and $e_2 = p_2$, then a watermark information is embedded in the pixel position, that is, the pixel value is $x'(i,\ j) = x(i,\ j) + b_2$.

The above process can be represented by Formula (19), traversing all subblocks until all the layers of "×" are fully embedded.

$$e_2 = e_1 - c \begin{cases} \text{if } e_1 = p_1,\ x'(i,\ j) = x(i,\ j) - b_2 \\ \\ \text{if } e_1 = p_2,\ x'(i,\ j) = x(i,\ j) + b_2. \end{cases} \tag{19}$$

Step 8: For the other three layers, repeat Step 2 to Step 7 to obtain the watermarked image.

Finally, it is combined with the watermarked image generated by the texture pixels to obtain the final watermarked image, as shown in Figure 6.



Original image Watermarked image Embedded watermark

Figure 6. Results diagram

4. Tampering detection process

4.1 *Extraction of texture pixel authentication information*

Depending on the priority of the layer, the authentication information is extracted in the opposite order of the embedding. In other words, if the embedding starts with the "×" layer and the "○" layer ends, then the extraction starts with the "○" layer and ends with the "×" layer. The authenticator extracts the authentication information and authenticates according to the inverse operation of the embedding method. Take the "○" layer as an example, which is shown below.

Step 1: After the authenticator receives the watermarked image, divide it into four layers, namely "×" layer, "+" layer, "●" layer and "○" layer. The authentication information is extracted for each layer separately. Taking the "○" layer as an example, the pixels are first classified, and the pixels are divided into texture pixels and smooth pixels based on the local complexity calculation in Section 2.3.

Step 2: Adaptively expand the texture pixels using Formulas (1), (2) from Section 2.1 to calculate P_1 and P_2. Grouping eight pixels together, two groups of prediction errors are obtained and sorted to yield $\left\{e^1_{\sigma(1)},\ e^1_{\sigma(2)},\ \cdots\cdots,\ e^1_{\sigma(8)}\right\}$ and $\left\{e^2_{\sigma(1)},\ e^2_{\sigma(2)},\ \cdots\cdots, e^2_{\sigma(8)}\right\}$.

Step 3: Judge the best prediction group by using Formula (3) in Section 2.2, and then divide it into three situations according to the positive and negative situations of the best prediction error group. In order to avoid the change caused by difference extension, the inverse operation of difference extension is used to restore the original value for each scenario separately. For upward scaling, restoration is performed according to Formula (20), and for downward scaling, restoration is performed according to Formula (21). Then, based on Formula (22), the authentication information b'_1 to be extracted is obtained.

$$\begin{cases} H = e'_{\sigma_n} - e'_{n-1} \\ h = \left\lfloor \dfrac{H}{2} \right\rfloor \\ e_{\sigma_n} = e'_{\sigma_{n-1}} + h \end{cases} \tag{20}$$

$$\begin{cases} H = e'_{\sigma_2} - e'_{\sigma_1} \\ h = \left\lfloor \dfrac{H}{2} \right\rfloor \\ e_{\sigma_1} = e'_{\sigma_2} - h \end{cases} \tag{21}$$

$$b'_1 = H - 2h. \tag{22}$$

Step 4: The remaining three layers should be extracted in the reverse order of the authentication information embedding process, starting with the "•" layer, followed by the "+" layer, and finally using the "×" layer.

4.2 *Smooth pixel authentication code extraction*

Step 1: First, construct a two-dimensional histogram $H(e_1,\ e_2)$ using $e_1,\ e_2$, divide different channels by $c = e_1 - e_2$ and select the embedding channel $c = 1,\ 0$ and -1.

Step 2: Find the peak point with the highest frequency in the selected channel, set to $(p_1,\ p_1 - c)$ and $(p_2,\ p_2 - c)$.

Step 3: In each subblock, for each embeddable channel, if the corresponding difference $(e_1,\ e_2)$ of the pixel $x(i,\ j)$ in the subblock meets $e_2 = e_1 - c$ and $e_1 = p_1$, then authentication information can be extracted, that is, $b'_2 = x(i,\ j) - x'(x,\ j)$; if the corresponding difference $(e_1,\ e_2)$ of the pixel $x(i,\ j)$ meets $e_2 = e_1 - c$ and $e_1 = p_2$, then the watermark information can be extracted using $b'_2 = x'(i,\ j) - x(x,\ j)$.

Step 4: Finally, the remaining three layers should be extracted and perform tampering detection in the reverse order of the authentication information embedding process, starting with the "•" layer, followed by the "+" layer, and finally using the "×" layer.

4.3 *Authentication process*

The extraction of texture pixel authentication information and smooth pixel authentication information has been described before. The authentication and tampering and positioning process of the reversible authentication algorithm will be introduced below.

Step 1: First, the watermarked image is divided into smooth pixels and texture pixels according to the local complexity.

Step 2: Then, extract the texture pixel authentication information b'_1 according to the method described in Section 3.1, and extract the smooth pixel authentication code according to the method described in Section 3.2 to obtain the authentication information b'_2.

Step 3: Then reuse the hash function Hash (*block*, *row*, *col*, *key*) according to the watermarked image received by the authentication party (where the *block* represents the relevant information of the pixels contained in the current block, *row*, and *col* represent the row and column where the current block are respectively, and the *key* is the agreed-upon key between the sender and the receiver). Generate texture pixel authentication code b_1 and smooth pixel authentication code b_2 accordingly.

Step 4: Finally, the regenerated texture pixel authentication code b_1 is compared with the extracted texture pixel authentication information b'_1, and the regenerated smooth pixel authentication code b_2 is compared with the extracted smooth pixel authentication information b'_2. If the pairs are equal, it means that the watermarked image is complete and not tampered with. If one of the two is unequal, it indicates that the watermarked image is tampered with, and the tampered area can be located through the unequal watermark pixels.

5. Analysis of the experimental results

To accurately evaluate the performance of the proposed algorithm, six 8-bit standard grayscale images with varying texture features and a size of 512 × 512 were selected as test images, namely *Cameraman*, *Pepper*, *Plane*, *Splash*, *Tank*, and *House*, as shown in Figure 7. The present algorithm simulation experiments are based on the windows11, using MATLAB 2016a. The PSNR value and correct detection rate were used as performance evaluation metrics. Compare the algorithm in this paper with the algorithms in references [18, 19], fully and objectively reflecting the visual quality of the watermarked images produced by the algorithm in this paper. At the same time, the tampering detection accuracy of this algorithm after simple and complex attacks is compared with [20–22], algorithm, which objectively measures the specific performance of the proposed algorithm. Additionally, this method can also be extended to color images through "channel separation processing" -the red, green, and blue channels of an RGB color image are disassembled into independent single-channel images, and the existing processing flow for grayscale images is executed for each channel separately.

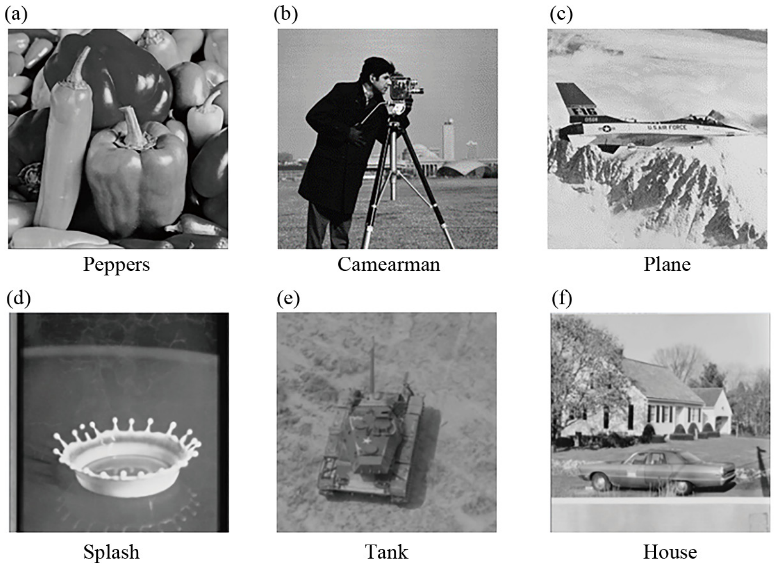


Figure 7. Experimental diagram

5.1 *Comparison of the watermarked images quality*

This experiment evaluates the visual quality preservation capability of the watermarking scheme by comparing PSNR values of watermarked images generated by different methods.

Table 1 compares the PSNR values of watermarked images from the proposed method against those from Zhong et al. [18] and Wang et al. [19], focusing on visual quality degradation introduced solely by watermark embedding under no-attack conditions. The results show that the PSNR of the proposed method is lower than that of Wang Hong et al. but higher than that of Zhong et al. Therefore, in order to ensure effective authentication, this method adopts the difference extension embedding method in the texture pixels, which leads to a large modification of the original image. However, literature [19] using IPVO embedding produced small modifications. Although the visual quality of the method in this paper decreases, generally the PSNR value greater than 40 dB is very close to the original image, while the PSNR value of the watermarked image obtained by this method is all greater than 46 dB, which has good image visual quality. Therefore, the algorithm in this paper has good authentication performance, but also can achieve good image visual quality.

Table 1. Comparison of Zhong et al.'s algorithm and PSNR of this algorithm

Image name	Zhong et al. [18]	Wang et al. [19]	This algorithm
Peppers	41.39	50.02	46.39
Cameraman	44.35	50.97	46.41
Plane	42.40	50.65	46.41
Splash	46.08	51.31	46.40
House	41.24	49.96	46.40
Tank	42.32	49.86	46.35

This experiment assesses the visual quality of the watermarked images under common tampering attacks.

Table 2. PSNR (dB) of attacked watermarked images

Image name	Peppers	Cameraman	Plane	Splash	House	Tank
Not under attack	46.39	46.41	46.41	46.40	46.40	46.35
Clipboard attack	37.22	24.04	34.79	33.76	31.58	36.88
Random alteration	36.84	36.88	36.53	31.53	37.82	35.27
Collage attack	34.33	27.48	33.27	32.32	30.56	34.63

Table 2 shows the PSNR values of the watermarked images after being subjected to tampering attacks. This experiment aims to assess the algorithm's robustness and its impact on the attacked image's structural integrity. It is evident that most experimental images, after being attacked, can still maintain PSNR values above 30 dB, thanks to the adaptive difference expansion employed in this paper. This algorithm helps to resist most common attacks, thereby enhancing the robustness, demonstrating the superiority of the proposed algorithm.

5.2 *Comparison of the correct detection rate*

5.2.1 *Comparison of the correct detection rate of simple attacks*

The simulation experiment makes simple attacks such as text addition, copy and paste, and content deletion, and calculates the correct detection rate. As can be seen from Table 3, the algorithm in literature [20] is slightly better than this method chosen in terms of the correct detection rate, because the algorithm uses the two-layer detection scheme and greatly reduces the leakage rate. However, the algorithm in literature [21] uses the principle of Logistic mapping to construct or embed the vulnerability watermark, and the correct detection rate is also high. But overall, the detection rate of this algorithm can be no less than 98%. This shows that the algorithm has good tampering detection performance against simple attacks.

Table 3. Correct detection rate (%) under simple attacks

Simple attack	Text addition	Copy and paste	Content deleted
This method	99.21	99.12	98.54
Suet al. [20]	99.55	99.79	99.81
Sahu et al. [21]	99.59	99.83	100

5.2.2 *Comparison of the correct detection rate of complex attacks*

To further measure the performance advantages of this algorithm, various complex tampering attacks were also conducted in this experiment and their correct detection rates were calculated. Moreover, to more intuitively compare the authentication performance of the proposed method with other methods and the impact of block size on authentication performance, the correct detection rates under different block sizes were also compared. Figure 8 for 4×4 blocks as the tamper detection unit, the watermarked image without attack, clip attack, random tampering attack (tampering proportion is 1%, randomly select 1% of the blocks, and modify the pixels to any value of not greater than 255 and not less than 0) and the *Plane* watermarked image and tamper detection mark after the collage attack.

Table 4 shows the correct detection rates for forgery detection using 4×4 blocks, 8×8 blocks, and 16×16 blocks compared with the experimental results of the algorithms proposed by Wang et al., and Yao et al. [22], after the watermarked images have been subjected to copy-move attacks, random tampering attacks, and collage attacks. According to the data in Table 4, the correct detection rate of the present method is significantly higher than that of the other two methods. The reason is that this method adopts the adaptive difference extension method for the texture pixels in the image, which can adjust the embedding strategy according to different texture features, so that the watermark is effectively embedded under various complex textures, thus improving the tampering detection rate. The other two methods do not fully utilize texture pixel to embed watermark and embed a smaller amount of watermark. As can also be seen from the data in Table 4, increasing the block size of the tamper detection unit can increase the initial correct detection rate. This phenomenon mainly stems from the larger block condition, each subblock has a larger capacity and can embed more authentication information. At the same time, the number of low-capacity subblocks (such as capacity 0 or 1) will also be reduced, thus increasing the detection rate to some extent. However, as the size of the subblock decreases, the corresponding tampering positioning effect becomes unsatisfactory.

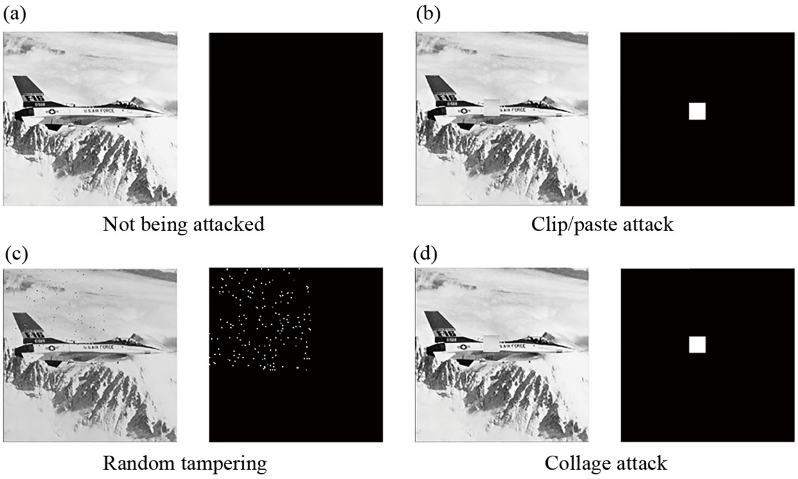
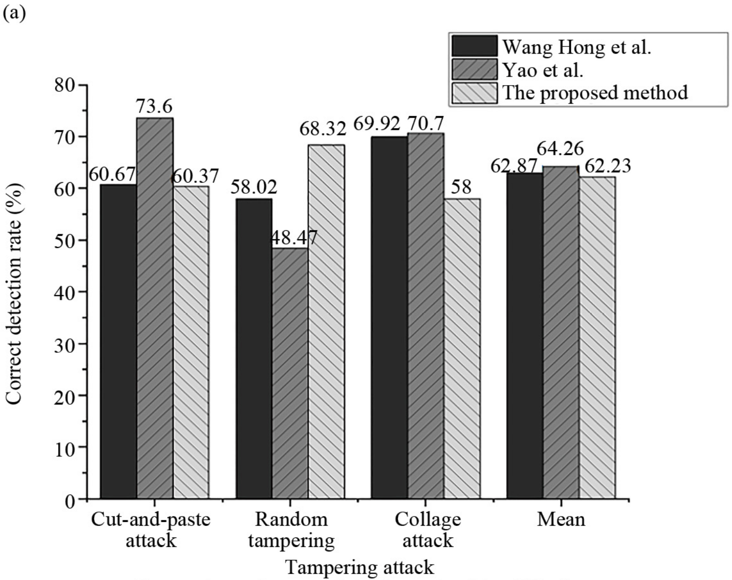


Figure 8. Tamper detection rate under different tampering rates

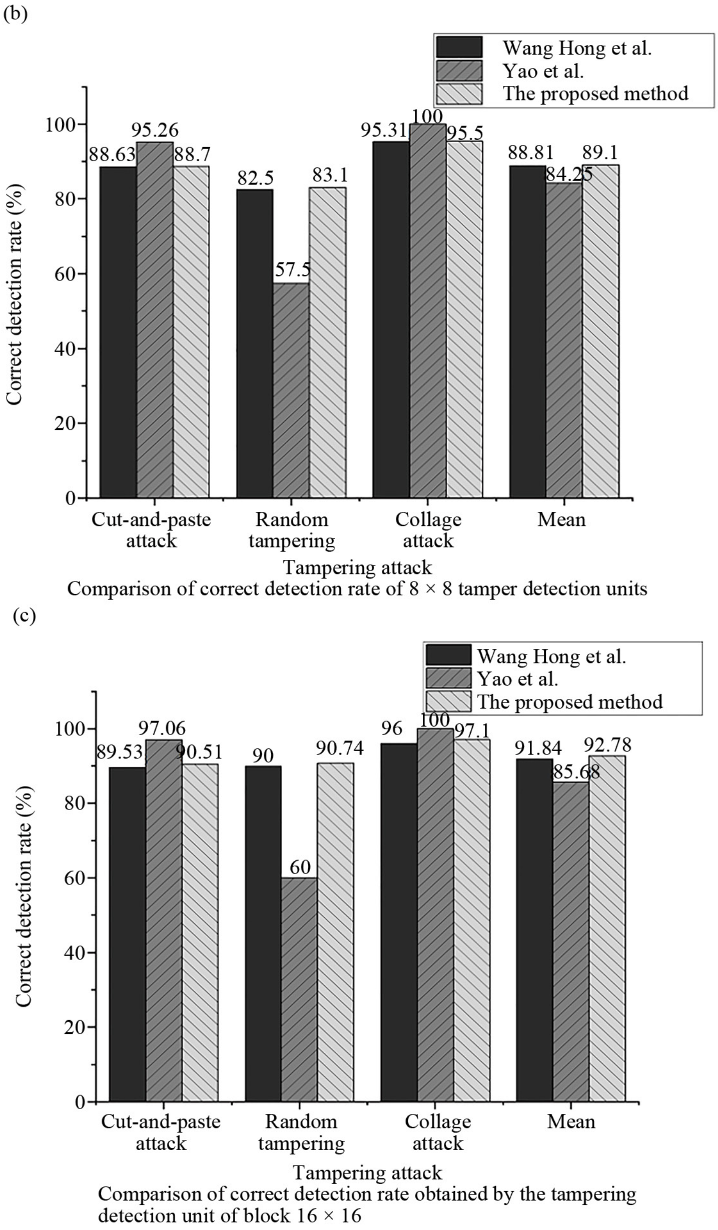
Table 4. Correct detection rate (%) under simple attacks

	Wang et al. [19]			Yao et al. [19]			This method		
Tampering attack	4×4	8×8	16×16	4×4	8×8	16×16	4×4	8×8	16×16
Clipboard attack	60.67	88.63	89.53	73.60	95.26	97.06	60.37	88.70	90.51
Collage attack	69.92	95.31	96.00	70.70	100.0	100.0	68.32	95.50	97.10
Random alteration	58.02	82.50	90.00	48.47	57.50	60.00	58.00	83.10	90.74
Average value	62.87	88.81	91.84	64.26	84.25	85.68	62.23	89.10	92.78

To more objectively demonstrate the superiority of this algorithm, as shown in Figure 9, using 4 × 4 blocks, 8 × 8 blocks, and 16 × 16 blocks as tampering detection units, we obtain the correct detection rate comparison with the algorithms of Wang Hong et al. and Yao et al. under copy-move attacks, random attacks, and collage attacks, respectively. To further demonstrate the advantages of our algorithm, we calculated the average values of three algorithms when subjected to different attacks using 4 × 4 blocks, 8 × 8 blocks, and 16 × 16 blocks as tampering detection units. As shown in Figure 9a, when using a 4 × 4 block as the tampering detection unit, the correct detection rate of the proposed algorithm for the watermarked image under collage attack is 2.31% higher than that of Wang Hong et al.'s algorithm, and the average value is 0.56% higher than that of Wang Hong et al.'s algorithm; The correct detection rate of this algorithm in the face of random tampering attacks is significantly better than Yao et al.'s algorithm, reaching as high as 19.85%. As shown in Figure 9b and c, when using 8 × 8 blocks and 16 × 16 blocks as tampering detection units, it can be visually seen that the average correct detection rate of this algorithm against three types of attacks is higher than that of the algorithms proposed by Wang Hong et al. and Yao et al., and the correct detection rate of this algorithm against random tampering attacks is also 25.6% and 30.74% higher than that of Yao et al.'s algorithm, respectively. As shown in Figure 9, the algorithm's correct detection rate for whether all 16 × 16 sub-blocks have been tampered with can basically reach over 90%. At the same block size, the algorithm in this paper is higher than that of Wang Hong et al. when performing collage attacks. The reason is that this paper not only embeds authentication information into smooth pixels, but also into texture pixels, increasing the correct rate of tampering detection. As can be seen from Figure 9, the bar chart indicates that the average correct detection rate of the algorithm proposed in this paper is higher than that of the algorithm by Wang Hong et al., regardless of whether the tampering detection unit is a 4 × 4 block, an 8 × 8 block, or a 16 × 16 block. It is also easy to see that the proposed algorithm is superior to the algorithm by Yao et al. in terms of correct detection rate for random tampering, and the stability of the proposed algorithm is better than that of Yao et al.'s algorithm.



Comparison of correct detection rate of 4 × 4 blocks



Comparison of correct detection rate of 8 × 8 tamper detection units

Comparison of correct detection rate obtained by the tampering detection unit of block 16 × 16

Figure 9. Correct detection rate comparison

6. Conclusion

This paper proposes an efficient reversible image authentication method based on the improved 2D histogram and adaptive difference expansion. Compared with existing reversible authentication methods, the proposed method fully exploits the characteristics of image pixels to embed authentication watermark information, effectively protecting images with complex textures or complex texture areas within images. In terms of robustness, the anti-interference characteristics of the two-dimensional histogram enable the method to have stronger tolerance to common attacks. In terms of embedding capacity, adaptive parameter adjustment significantly improves the effective embedding space of unit pixel blocks. In terms of computational efficiency, the time consumption in the preprocessing stage is reduced because complex feature screening processes are avoided. Experimental results indicate that the proposed method can effectively protect images with complex textures and partially complex texture regions within images. Although it has certain limitations in processing extremely low-texture images, overall, compared with existing reversible authentication methods, it demonstrates superior tamper detection and localization capabilities. Based on this, future work will focus on two aspects: first, optimizing the processing strategy for extremely low-texture images to enhance the method's adaptability; second, exploring a multi-channel collaborative embedding mechanism to improve the efficiency of extending to color images. Additionally, targeted analyses of the experimental results have been supplemented, including the reasons for the differences in detection rates under varying texture complexities and the specific impact of the hierarchical strategy on the balance between PSNR and detection performance.

Acknowledgments

This work is supported by National Statistical Science Research Project (2018LY12). At the same time this work is also supported by Huaian natural science research program (HAB202363).

Conflict of interest

The authors declare no competing financial interest that could have appeared to influence the work reported in this paper.

References

[1] Ouyang J, Huang J, Wen X, Shao ZH. A semi-fragile watermarking tamper localization method based on QDFT and multi-view fusion. *Multimedia Tools and Applications*. 2023; 82(10): 15113-15141. Available from: https://doi.org/10.1007/s11042-022-13938-1.

[2] Kim C, Yang CN. Self-embedding fragile watermarking scheme to detect image tampering using AMBTC and OPAP approaches. *Applied Sciences*. 2021; 11(3): 1146. Available from: https://doi.org/10.3390/app11031146.

[3] Ouyang JL, Huang JT, Wen XZ. A semi-fragile reversible watermarking method based on qdft and tamper ranking. *Multimedia Tools and Applications*. 2024; 83(14): 41555-41578. Available from: https://doi.org/10.1007/s11042-023-16963-w.

[4] Lo CC, Hu YC. A novel reversible image authentication scheme for digital images. *Signal Processing*. 2014; 98: 174-185. Available from: https://doi.org/10.1016/j.sigpro.2013.11.028.

[5] Thodi DM, Rodriguez JJ. Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*. 2007; 16(3): 721-730. Available from: https://doi.org/10.1109/TIP.2006.891046.

[6] Wang B, Mao Q, Su DQ. Image authentication algorithm based on two-dimensional histogram shifting. *Journal of Computer Applications*. 2015; 35(10): 2963-2968.

[7] Lee TY, Lin SF. Dual watermark for image tamper detection and recovery. *Pattern Recognition*. 2008; 41(11): 3497-3506. Available from: https://doi.org/10.1016/j.patcog.2008.05.003.

[8] Yin ZX, Niu XJ, Zhou ZL, Tang J, Luo B. Improved reversible image authentication scheme. *Cognitive Computation*. 2016; 8(5): 890-899. Available from: https://doi.org/10.1007/s12559-016-9408-6.

[9] Wang SS, Xu XH. Hilbert the algorithm for generating the curve scanning matrix and its MATLAB program code. *Chinese Journal of Image and Graphics*. 2006; (1): 119-122.

[10] Peng F, Li XL, Yang B. Improved PVO-based reversible data hiding. *Digital Signal Processing*. 2014; 25: 255-265. Available from: https://doi.org/10.1016/j.dsp.2013.11.002.

[11] Li D, Dai XL, Gui J, Liu JY, Jin X. A reversible watermarking for image content authentication based on wavelet transform. *Signal, Image and Video Processing*. 2024; 18(3): 2799-2809. Available from: https://doi.org/10.1007/s11760-023-02950-z.

[12] Nguyen TS, Chang CC, Yang XQ. A reversible image authentication scheme based on fragile watermarking in discrete wavelet transform domain. *AEU-International Journal of Electronics and Communications*. 2016; 70(8): 1055-1061. Available from: https://doi.org/10.1016/j.aeue.2016.05.003.

[13] Vaidya S, Chandra P. Robust digital color image watermarking based on compressive sensing and DWT. *Multimedia Tools and Applications*. 2024; 83(2): 3357-3371. Available from: https://doi.org/10.1007/s11042-023-15349-2.

[14] Hong W, Chen MJ, Chen TS. An efficient reversible image authentication method using improved PVO and LSB substitution techniques. *Signal Processing: Image Communication*. 2017; 58: 111-122. Available from: https://doi.org/10.1016/j.image.2017.07.001.

[15] Sinhal R, Sharma S, Ansarl IA, Bajaj V. Multipurpose medical image watermarking for effective security solutions. *Multimedia Tools and Applications*. 2022; 81(10): 14045-14063. Available from: https://doi.org/10.1007/s11042-022-12082-0.

[16] Wang SY, Li CY, Kuo WC. Reversible data hiding based on two-dimensional prediction errors. *IET Image Processing*. 2013; 7(9): 805-816.

[17] Ghaemi A, Danyali H, Kazemi K. Reversible data hiding in encryption domain based on two dimensional histogram shifting and secure encryption system. *Multimedia Tools and Applications*. 2022; 81: 33731-33757. Available from: https://doi.org/10.1007/s11042-022-12493-z.

[18] Zhong YY, Huang FJ. An efficient image reversible authentication method. *Journal of Software*. 2023; 34(12): 5848-5861.

[19] Wang H, Huang FJ. Attack and improvement of authentication scheme based on reversible information hiding technology. *Journal of Information Security*. 2022; 7(1): 56-65.

[20] Su GD, Chang CC, Chen CC. A hybrid-Sudoku based fragile watermarking scheme for image tampering detection. *Multimedia Tools and Applications*. 2021; 80: 12881-12903. Available from: https://doi.org/10.1007/s11042-020-10451-1.

[21] Sahu AK. A logistic map based blind and fragile watermarking for tamper detection and localization in images. *Journal of Ambient Intelligence and Humanized Computing*. 2022; 13(8): 3869-3881. Available from: https://doi.org/10.1007/s12652-021-03365-9.

[22] Yao H, Wei H, Tang Z. A real-time reversible image authentication method using uniform embedding strategy. *Journal of Real-Time Image Processing*. 2020; 17: 41-54. Available from: https://doi.org/10.1007/s11554-019-00904-8.