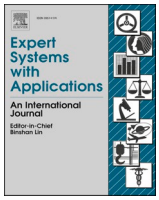
Expert Systems With Applications 226 (2023) 120228

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Tampering localization and self-recovery using block labeling and adaptive significance

Qiyuan Zhang [a], Xiaochen Yuan [a,*], Tong Liu [a], Chan-Tong Lam [a], Guoheng Huang [b,*], Di Lin [c], Ping Li [d,e]

[a] Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, China
[b] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[c] College of Intelligence and Computing, Tianjin University, Tianjin, China
[d] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[e] School of Design, The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
Block-level Partially Symmetric Mapping
Block Adaptive Significances (BAS)
Image Self-Recovery
Block-based labeling (BBL)
Pixel-based labeling (PBL)

ABSTRACT

This paper proposes a scheme for localization and restoration of image tampered regions using block labelling and adaptive significance. To generate the watermark information which includes authentication data and recovery data, we propose a block coordinate labelling method, which extracts the exact coordinate position information of each block, while the recovery data is composed of Block Adaptive Significances (BAS) and bitmaps, which are composed of high and low adaptive significance. To detect the tampered area more effectively, we propose a dual detection approach that combines the block-based labeling (BBL) and pixel-based labeling (PBL). We embed the authentication data into each pixel in the block sequentially and embed the position coordinate information of the block into the whole image in ascending order. The PBL approach can be used to rapidly complete tamper detection when the requirements for PBL are satisfied, whereas the BBL is used to increase the possibility of successfully detecting tampering if the conditions are not satisfied. Furthermore, we propose a block-level partially symmetric mapping and apply it to self-recovery bits in block units, thereby reducing the possibility of recovery bits being lost. The experimental results show that in our scheme, the average precision reaches 86.70%, which is 4% higher than the existing results, and the average F1score reaches 92.02%, which is 2% higher than the existing results.

1. Introduction

The advancement of network technology has promoted the development of the digital industry, and more and more important information is stored using digital images. At the same time, some lawbreakers may use image processing software to tamper with the image, which will have a major impact on the real content of the image. For example, destroying or tampering with medical photos may lead to misdiagnosis; maliciously tampering with news pictures will distort the facts; tampering with court evidence pictures will seriously affect the fairness of criminal trials. Due to modern computerized image processing techniques, an individual cannot instinctively detect its integrity and authenticity with the naked eye. Therefore, ensuring the authenticity and integrity of digital images has become one of the most important information security researches. Van et al. (Van Schyndel et al. 1994) proposed the concept of digital watermark for the first time, and discussed the method of adding invisible digital watermark on 8 bit gray scale images. Yeung et al. (Yeung & Mintzer, 1997) first proposed adding invisible watermarking to high-quality color and gray-scale images. This invisible watermarking has functions for determining whether an image has been tampered with. Fridrich et al. (Fridrich & Goljan, 1999) proposed the concept of self-correcting and implemented image self-recovery based on DCT transform and quantized coefficients for the first time. In 2008, Zhang et al. (Zhang and Wang, 2008) implemented a method for lossless restoring images with a large amount of embedded information. But this method has great limitations. First, in order to ensure lossless restoration of the image, the amount of information used for embedding is very large, resulting in a Peak-Signal-to-Noise-Ratio

* Corresponding authors.
E-mail addresses: qiyuan.zhang@mpu.edu.mo (Q. Zhang), xcyuan@mpu.edu.mo (X. Yuan), p2209360@mpu.edu.mo (T. Liu), ctlam@mpu.edu.mo (C.-T. Lam), kevinwong@gdut.edu.cn (G. Huang), di.lin@tju.edu.cn (D. Lin), p.li@polyu.edu.hk (P. Li).

https://doi.org/10.1016/j.eswa.2023.120228
Received 20 December 2022; Received in revised form 29 March 2023; Accepted 20 April 2023
Available online 25 April 2023

(PSNR) of only 28.7 dB for the watermarked image. Second, the premise of lossless image restoration is that the tampered area is not larger than 3.2% of the original image area. Under the current research, He et al. (He et al. 2011) envisaged using classical information encryption techniques to encrypt the transmitted information to solve the above difficulties. The legitimate user at the information receiving end uses the key to decode the encrypted data and receive the data. Moreover, the information in the image is retrieved and compared with the original content information to determine if the image content has been changed. Self-recovery based on watermarking is a hot research topic of many scholars in recent years, and it is also an important branch of digital watermarking technology. In addition to judging whether the image content has been tampered with, it can also accurately find and repair the tampered area (Lu and Liao, 2000, Puhan and Ho, 2007, Zhang and Wang, 2007, Chen and Wang, 2009, Lin et al. 2009, Chuang et al. 2011). At present, image self-recovery research based on fragile watermarking is primarily concerned with the problem of ensuring watermark invisibility and improving the performance of tampering detection and recovery.

It is well known that if we can precisely find the tampered area, the quality of the recovered image can be increased further. For accurately locating the tampered area, there are two standard watermark embedding methods for image authentication: pixel-based (Vali et al. 2018, Huang et al. 2019) and block-based (Lee & Lin, 2008) measures. In the pixel-based method, the authentication information was generated and embedded into each pixel, and it should be recalculated for comparison during the tamper detection procedure. While in the block-based method, the image was split into blocks, and the embedding and extraction of the watermark were based on each block independently. Although the pixel-based method could achieve more accurate results, it is limited by extensive calculations and weak robustness to resist attack. While in the block-based method, block dependencies can make the scheme more robust against various attacks, such as copy-move and collage. Taking both accuracy and security into account, we propose a method adding authentication and recovery information into each pixel of blocks. The authentication information here is scattered throughout each pixel. The unit is also a block when extracting the watermark, making it possible to identify tampering with even a single pixel in a block.

After reviewing lots of state-of-the-art techniques, four challenges in this topic are summarised as follows: (1) the tamper detection localization requires greater accuracy; (2) the computational time needs to be reduced; (3) the watermark is not imperceptible enough; (4) the quality of the recovered image needs to be enhanced. To address challenges (1) and (2), we propose a dual detection mechanism that strikes a balance between detection speed and accuracy. Our approach involves using Pixel-based Labeling (PBL) for rapid tamper detection and Block-based Labeling (BBL) to provide a more accurate detection result. By combining the PBL and BBL detection methods, a fast and accurate approach to detecting forgeries is achieved. Next, to address challenges (3), the proposed watermarking scheme is designed to achieve satisfactory functional performance with fewer watermarks embedded. As it is commonly understood, fewer watermarks lead to better imperceptibility of the scheme. To address challenges (4), we use Block Adaptive Significances (BAS) and bitmap as approximate information for image restoration, and use Block-level Partially Symmetric Mapping (BPSM) to increase the probability of image restoration. Overall, the contributions of the paper are summarized as follows:

1. We propose a dual approach for tampering detection that strikes a balance between detection speed and accuracy. Our first method, PBL tampering detection, enables rapid localization of tampered regions. During PBL, we extract authentication information and check whether it conforms to the ascending sequence. However, PBL is only effective if the coordinate information for the first block is correct. To supplement this, we propose a second method, BBL tampering detection, which locates tampered regions by comparing the extracted watermark with the calculated watermark of each block.
2. We propose to use the BAS and bitmap obtained from both the mean and standard deviation for each block's whole set of pixels as self-recovery information. In addition, in order to guarantee the security of the whole scheme, we propose to use the BPSM method for encoding the recovery information. Besides that, the BPSM also retrieves the lost recovery data from the untampered region, which further improves the quality of the recovered image.
3. We evaluate the proposed method using two databases, BOWS2 (Bas & Furon, 2007) dataset and USC-SIPI (Weber 2006) standard image dataset. From the two datasets, images of different textures are used, and forgeries of various tampering rates are included and simulated for experiments. Comparison with existing works shows the superiority of this method. Moreover, we also conduct experiments on robustness against two categories of attacks and experiments at different tampering rates on the images, the proposed method shows satisfying performance in both scenarios.

The remaining sections of this paper are as follows: in section 2, we review some recent self-recovery techniques based on watermarking. The method that we proposed is detailed in section 3. And section 4 performs the evaluation of proposed scheme through extensive experiments. In addition, this section also gives the experimental results against multiple attacks and compares our scheme with the existing works. Finally, a brief conclusion is summarized and the future works are discussed in section 5.

2. Related work

This section describes some of the existing watermarking methods. We explore two common watermark embedding techniques in subsection 2.1: frequency domain and spatial domain approach. Additionally, subsection 2.2 offers some new techniques for tampering detection and self-recovery.

2.1. Watermark embedding techniques

Usually, there are two kinds of embedding methods: one is to embed the generated watermark into the frequency domain, while the other is to embed it into the spatial domain. When the watermark is embedded in the frequency domain, it takes advantage of the human eye's visual features, where the coefficients are more significant at the textures of detail sub-bands, and the human eye is not easily able to identify changes in the image after embedding the watermark information. The frequency domain-based methods show the advantage of larger watermark capacity and better invisibility, and are usually used in the field of robust watermarking. While in the field of fragile watermarking, embedding watermarks into the spatial domain can be very sensitive to changes in image content. Therefore, in order to well locate the tampered area of the image, many authors choose to embed the watermark into spatial domain. Hu et al (Hu et al. 2013; Hu et al., 2013) showed a joint encoding and temper detection method by Absolute Moment Block Truncation Coding (AMBTC). The watermark information was embedded into the Least Significant Bit (LSB) plane by matching the parity of the subdivided bitmap. And also in Hu et al. (Hu et al. 2013; Hu et al., 2013), the length of the authentication data was created to be flexible to improve embedding efficiency. The PSNR of their embedded watermarked image was 34.48 dB. Hong et al. (Hong et al. 2020) proposed an AMBTC tamper detection system that maintained excellent detectability and high image fidelity. To create authentication codes, the proposed method switches sequentially in the bitmap of the AMBTC codes. The PSNR of their watermarked image was 52.45 dB, but their scheme cannot restore the tampered region. Bravo et al. (Bravo-Solorio et al. 2018) presented a receiver that used check

information to identify altered pixels before executing a restorable iterative mechanism to determine the watermarked pixels' original values. The scheme proposed by Bravo et al. was able to recover about 95% of the changed pixels in the image, but their proposed scheme can only be effective if the tampering rate is less than 26%. Besides, the PSNR of watermarked image was 37.9 dB. Roy et al. (Roy et al. 2012) introduced the hardware upgrade method of the digital watermark structure that can instantly add a watermark to compressed video streams, which is semi-fragile and invisible. The 3 LSB planes of the image were given as reference bits and check bits in this method, and the PSNR of the watermarked image was above 35 dB. In this work, we also choose to employ the LSB watermark embedding approach since it is less obvious and more attack-resistant.

2.2. Image tamper detection and self-recovery methods

Tampering detection schemes can be divided into two categories according to their purposes: one focuses on tampering detection under different attacks, and the other has the ability to reconstruct the tampered area within the image. Haghighi et al. (Haghighi et al. 2019) came up with a method for fragile blind quad watermarking which was using the wavelet transform and a genetic algorithm to find and recover the image. In this paper, two new approaches, Partner-block and Mirror-aside, were developed to improve the quality of the final result. Although the scheme proposed by Haghighi et al. could cover a larger tampered area, the PSNR of the restored image was 34.45 dB in the case of 50% tampering rate. Sarreshtedari et al. (Sarreshtedari & Akhaee, 2015) proposed that the reference bits used to restore the tampered area be produced using the Set Partitioning In Hierarchical Trees (SPIHT) technique. Meanwhile, it designed a channel code to protect the reference bits against tampering. It successfully recovered up to 30% of the tampering without leaving any observable distortion. However, because the method of Sarreshtedari (Sarreshtedari & Akhaee, 2015) generated the authentication data from Most Significant Bit (MSB) planes, it could not resist the attack of copy-move. And the PSNR of the recovered image was only 40.8 dB. Mahmood et al. (Mahmood et al. 2018) proposed an efficient technique to uncover regional duplication in digital images. This method created w $\times$ w overlapping blocks from the approximate sub-bands of the displacement-invariant stationary wavelet transform. To reveal regional duplication in digital photos, special features generated from overlapping patches are employed. The F1score of tampering detection with size of 4 $\times$ 4 blocks was 86%. A blind double watermarking technique for images that consists of a strong watermark for protecting copyright and a weaker watermark for locating tampered regions has been presented by Ahmadi et al. (Ahmadi et al. 2021). For the purpose of detecting image manipulation by modifying diagonal singular values, fragile watermarks were included in each channel of the RGB color space. In their scheme, the PSNR of the watermarked image was 52 dB. Chen et al. (Chen et al. 2009) introduced a block-based fragile watermark method that used Fuzzy C-means (FCM) clustering techniques to create relationships between image blocks for image authentication and tamper resistance. The results showed that this method could achieve better tampering detection results and higher watermarking image quality, the PSNR of watermarked image was 44 dB. Molina-Garcia et al. (Molina-Garcia et al. 2020) created a method for tampered region identification and self-recovery in color images that needed to be initially divided into blocks. The watermark was created for each i^{th} block and embedded in other blocks in accordance with the embedding order specified by the permutation procedure. The intended approach employs a bit adjustment stage after embedding the watermark produced by each block into 2-LSB, thus the invisibility of the watermarked image could be guaranteed. But experiments showed that in the case of 50% tampering, the average PSNR of the restored image was only 26 dB. Shehab et al (Shehab et al. 2018) came up with a watermarking method in medical applications to find and recover the tampered area. After the image was divided into 4 $\times$ 4 sized blocks, which were utilized to collect information for resisting vector quantization attack authentication using SVD. The experiments showed that the average PSNR of restored images was 38.96 dB. Tai et al. (Tai & Liao, 2018) used wavelet transform to embed the fragile watermark, including the data of authentication and recovery. To address the issue of tampering coincidence, they developed a two-level self-recovery technique using 3 $\times$ 3 block-neighborhood. But experiments showed that the average PSNR of restored images was only 37.1 dB. Adbelhakim et al. (Abdelhakim et al. 2019) created a recovery method based on unsupervised machine learning and fragile watermarking. They implemented DCT transform on the block, which constructed the authentication bits. For restoration, they used K-means clustering to calculate the recovery data. Nevertheless, if the content damage appeared on both the regular block and the mapped-block, it was unable to achieve the recovery. Besides, their tamper detection time was 160.47 s, and the restoration time was 168.12 s. Singh et al. (Singh et al. 2019) introduced a self-embedded watermarking technique based on quantization and DCT transformation, which used two-level coding to generate content recovery bits. Recovery can be achieved with high image quality when the tamper rate is 50%. Dadkhah et al. (Dadkhah et al. 2014) came up with a tamper detection and self-recovery system with SVD, which created two distinct detection keys of image blocks. To improve the accuracy of tampering localization, the image was segmented into a combination of 4 $\times$ 4 and 2 $\times$ 2 sized blocks. Under this method, the mean *PSNR* of watermarked image and the recovered image were 43.39 dB and 38.22 dB, respectively. A blind fragile watermarking approach has been presented by Sinhal et al. (Sinhal et al. 2020) in order to achieve tamper detection and self-recovery for color images. A key-based pseudorandom binary sequence was employed for tamper detection. The MSB recovery data of the various blocks are concatenated with 6 bits to create the watermark sequence. However, when the tampering rate is 50%, their restored image quality was only 27.02 dB. Chang et al. (Chang et al. 2020) introduced a technique for generating watermarks with fewer bits by using a bit-reduction based AMBTC approach. They further embedded these watermarks into the original image using a turtle shell based data hiding technique. To recover the original image, they employed an adaptive weight-based recovery algorithm. However, the restoration quality of their method was limited, as indicated by the relatively low PSNR value of the average restored image, which was only 34.65 dB. Lin et al. (Lin et al. 2023) proposed a novel method for image authentication and tamper detection using AMBTC compressed codes as authentication codes, followed by VQ compressed codes as credentials for information recovery. However, the experimental results only demonstrated the correct rate of image tampering at a lower tampering rate, and the average PSNR of their recovered images was only 39.28 dB.

In many prior works, the emphasis has been placed on image restoration methods, while ignoring the critical importance of image tampering detection. However, it is worth noting that the accuracy of tamper detection has a direct impact on the quality of image restoration. Our paper addresses this issue by proposing the use of two tamper detection methods, with the aim of improving the accuracy and efficiency of tamper detection. By utilizing these two techniques, our method achieves higher accurate results while also improving the efficiency of the tamper detection process. Our approach aims to strike a balance between accuracy and efficiency, as we recognize that both factors are critical for effective tampering detection. Furthermore, we employ the BAS and bitmap techniques to ensure the high quality of recovered images, on the premise of accurate tamper detection. Through the utilization of these methods, our paper provides a comprehensive and effective solution for image restoration that accounts for the importance of tamper detection.

3. Proposed method

Our proposed scheme could be divided into two parts: (1) watermark generation and embedding, which is the preprocessing of the image to

generate the watermarked image; (2) tampered area detection and self-recovery. During the procedure of watermark generation, we propose the Block Coordinate Label (BCL) method to generate the authentication data. And the recovery data consists of two parts, the first part is the BAS which is calculated from each block of the original image, and the second part is the bitmap generated from the calculation result of BAS. Then we propose the BPSM on the recovery data to improve the success rate of recovery. Finally, the watermarked image is generated by embedding the authentication and recovery information into the original image. Given an image that may be maliciously tampered with, firstly, we extract the watermark information from the divided blocks. Then by applying the BPSM on blocks accordingly, the recovery information of the modified area is thus obtained. Then by overlaying the recovered area onto the received image, we can obtain the recovered image. In the following subsections, the generation and embedding of watermark information will be detailed in 3.1. And the procedure of tampered area localization and self-recovery will be shown in 3.2.

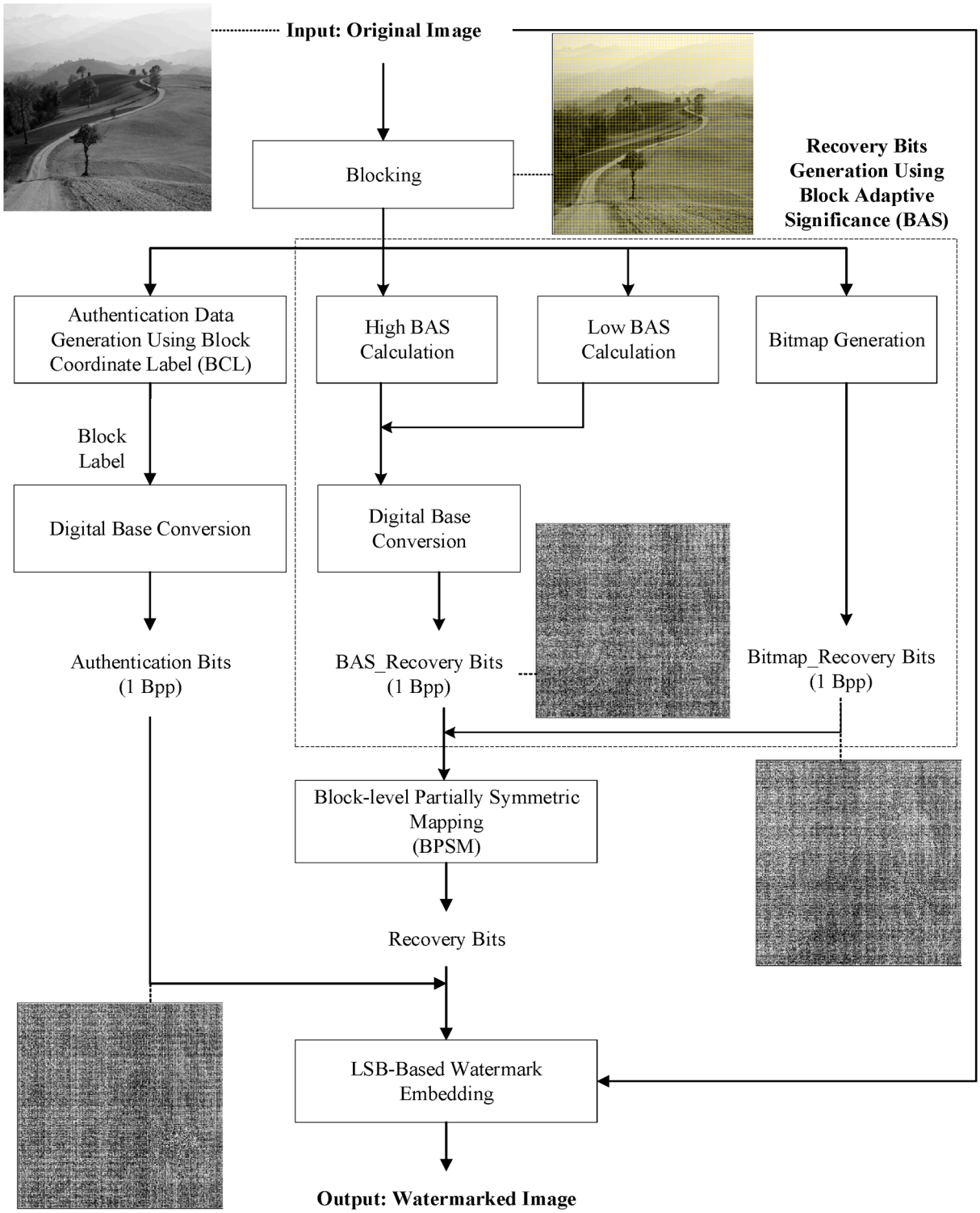


Fig. 1. The Generation Flowchart of Watermarked Image.

3.1. Watermark generation and embedding

In our proposed method, the watermark is composed of two components: authentication and recovery information, which are embedded independently into the spatial domain to ensure the effectiveness of tampered region localization and restoration. To achieve this, we first separate the host image into blocks, and the authentication information marks and records the coordinates of each block. The recovery information is then generated using the average and standard deviation of the pixels in the block, which includes high BAS, low BAS, and bitmap. To further enhance the effectiveness of our proposed method, we employ the BPSM technique to reorganize the generated recovery information. After binary processing, the recovery and authentication information is integrated into their respective blocks. This approach ensures that the authentication and recovery information is accurately embedded into the image, improving the robustness of our method against tampering attacks.

Fig. 1 shows the flowchart of watermark generation and embedding. Firstly, the $M \times N$ sized grayscale original image I_O is segmented into blocks I_T, which are of size $M_T \times N_T$ that do not overlap. During this process, each block has been successively coordinated. After that, we apply the BCL to every block $I_T(L_X, L_Y)$ to generate the authentication data A_{data}. The recovery data R_{data} consists of two parts. The first part is the BAS which is composed of low BAS ($L_A(L_X,L_Y)$) and high BAS ($H_A(L_X, L_Y)$), and is used for restoration of the corresponding pixel values; and the second part is the bitmap ($B_{map}(L_X, L_Y)$) which records the exact location of L_A and H_A in the corresponding block. Next, we perform binary conversion on the authentication data and recovery data of each block, thus to calculate the *Authentication Bits* A_{bits} and the *BAS_Recovery Bits* R_{BAS}. After that, we apply BPSM to R_{BAS} and the *Bitmap_Recovery Bits* R_{Bitmap} using (1), to calculate the storage location of block recovery feature. In this way, the corresponding storage location (HIB_x, HIB_y) of the recovery feature is calculated. Finally, the calculated A_{bits} and R_{bits} are embedded into the corresponding blocks, and the watermarked image I_w is therefore generated.

$$BPSM\rightarrow\begin{cases} HIB_x = \dfrac{M}{M_T} - L_X + 1 \\ HIB_y = \dfrac{N}{N_T} - L_Y + 1 \end{cases} \quad (1)$$
$$L_X \in \left[1,\ \frac{M}{M_T}\right],\ L_Y \in \left[1,\ \frac{N}{N_T}\right]$$

where L_X *and* L_Y are the block-based coordinate, HIB_x and HIB_y are the storage location of the corresponding recovery information.

Algorithm 1: Procedure of Generation and Embedding of the Watermark

Input: Original Image I_O of size $M \times N$;
Output: Watermarked Image I_w of size $M \times N$;

1 Divide I_O into non-overlapping blocks by using (2), and the size of each block is $M_T \times N_T$.

$$I_O = \cup_{i=1}^{\frac{M}{M_T}} \cup_{j=1}^{\frac{N}{N_T}} I_T(L_X,\ L_Y)$$
$$L_X \in \left[1,\ \frac{M}{M_T}\right],\ L_Y \in \left[1,\ \frac{N}{N_T}\right] \quad (2)$$

where (L_X, L_Y) is the block-based coordinate.

2 Apply the proposed BCL to generate the Authentication data A_{data} using (3).

$$A_{data} = BCL(I_T) \quad (3)$$

3 Calculate the recovery data R_{data} including low adaptive significance ($L_A(L_X, L_Y)$), high adaptive significance ($H_A(L_X, L_Y)$), and bitmap ($B_{map}(L_X, L_Y)$).

(*continued on next column*)

(*continued*)

Algorithm 1: Procedure of Generation and Embedding of the Watermark

$$R_{data}\begin{cases} L_A(L_X, L_Y) = x_{avg} - \sigma\sqrt{\dfrac{q}{m-q}} \\ H_A(L_X, L_Y) = x_{avg} + \sigma\sqrt{\dfrac{m-q}{q}} \\ B_{map}(L_X, L_Y) = \begin{cases} 1, everypixelin I_O(L_X, L_Y) > x_{avg} \\ 0, everypixelin I_O(L_X, L_Y) \le x_{avg} \end{cases} \end{cases} \quad (4)$$

Where x_{avg} is the average value of pixels in the block, σ denotes the standard deviation, m is the number of pixels in the block and q is the number of pixels whose value is greater than x_{avg}.

4 Perform binary conversion on Authentication data and Recovery data to get ***Authentication Bits*** A_{bits} and the ***BAS_Recovery_Bits*** R_{BAS}.

5 Apply proposed BPSM to R_{BAS} and R_{Bitmap} to generate the ***Recovery Bits*** R_{bits}.

$$R_{bits} = BPSM(R_{BAS},\ R_{Bitmap}) \quad (5)$$

6 Generate the watermarked image I_w by embedding the watermark, which consists of ***Authentication Bits*** and ***Recovery Bits*** into the LSB planes of each block.

$$I_w = \textbf{\textit{LSB\_Embedding}}(I_O,\ A_{bits},\ R_{bits}) \quad (6)$$

Fig. 2 shows the detail of embedding demonstration of authentication and recovery bits. As discussed above, to generate the watermark information, first, the blocks of size $M_T \times N_T$ that non-overlap are generated from the original image. The size of the block we use here is $M_T = N_T = 4$. The generated authentication data is transformed into authentication bits with a block size of 16 bits using the proposed BCL. The recovery bits are composed of high BAS, low BAS and bitmap, where high BAS and low BAS are both of 8 bits, and the bitmap is of 16 bits per block. In total, three bits per pixel are required to embed the watermark bits. In our work, we embed the watermark information via the LSB replacement. We embed the 16-bit authentication bits into the third-to-last bit of the pixels in each block, which is used to reduce the possibility of misjudgment. In this way, one bit is embedded into each pixel, thus if anyone pixel is tampered with, we can accurately get the coordinates of that tampered block. Similarly, the high BAS and low BAS are embedded into the penultimate bit of each pixel, and the bitmap is embedded into the last bit of each pixel.

3.2. Tamper detection and image self-recovery

Since any region of the received image could have been maliciously changed, the algorithm for localizing and recovering these tampered regions is proposed in this section to protect the image content, and the flowchart of tampering detection and self-recovery is given in Fig. 3. Our proposed approach would identify and locate the tampered region since that was obtained from the other party. In our method, the dual tamper detection mechanism is proposed, the BBL and the PBL, to achieve a balance between detection speed and accuracy. The extracted recovery bits including *BAS_Recovery_Bits* and *Bitmap_Recovery_Bits* are used for tampered regions recovery.

Given the received image I_R, first, we block I_R into non-overlapping blocks of size $M_T \times N_T$, then BPSM is applied and the watermark information is extracted, including the authentication bits and recovery bits respectively. In the tampering detection part, the authentication bits are separated from the watermark in block units, and then each block obtains 16 binary numbers. Next, the obtained binary numbers are converted into two decimal numbers indicated as block labels. To achieve a compromise between detection speed and accuracy, we propose a dual detection method based on PBL and BBL. The PBL method can get the tamper detection result in a fast way. Because it avoids computing a new reference bit for comparison, thus lowering the number of computations and saving detection time. However, the premise of PBL obtaining accurate results is to ensure that the first extracted block label is (1,1). If the first extracted block label is not (1,1), the tamper detection

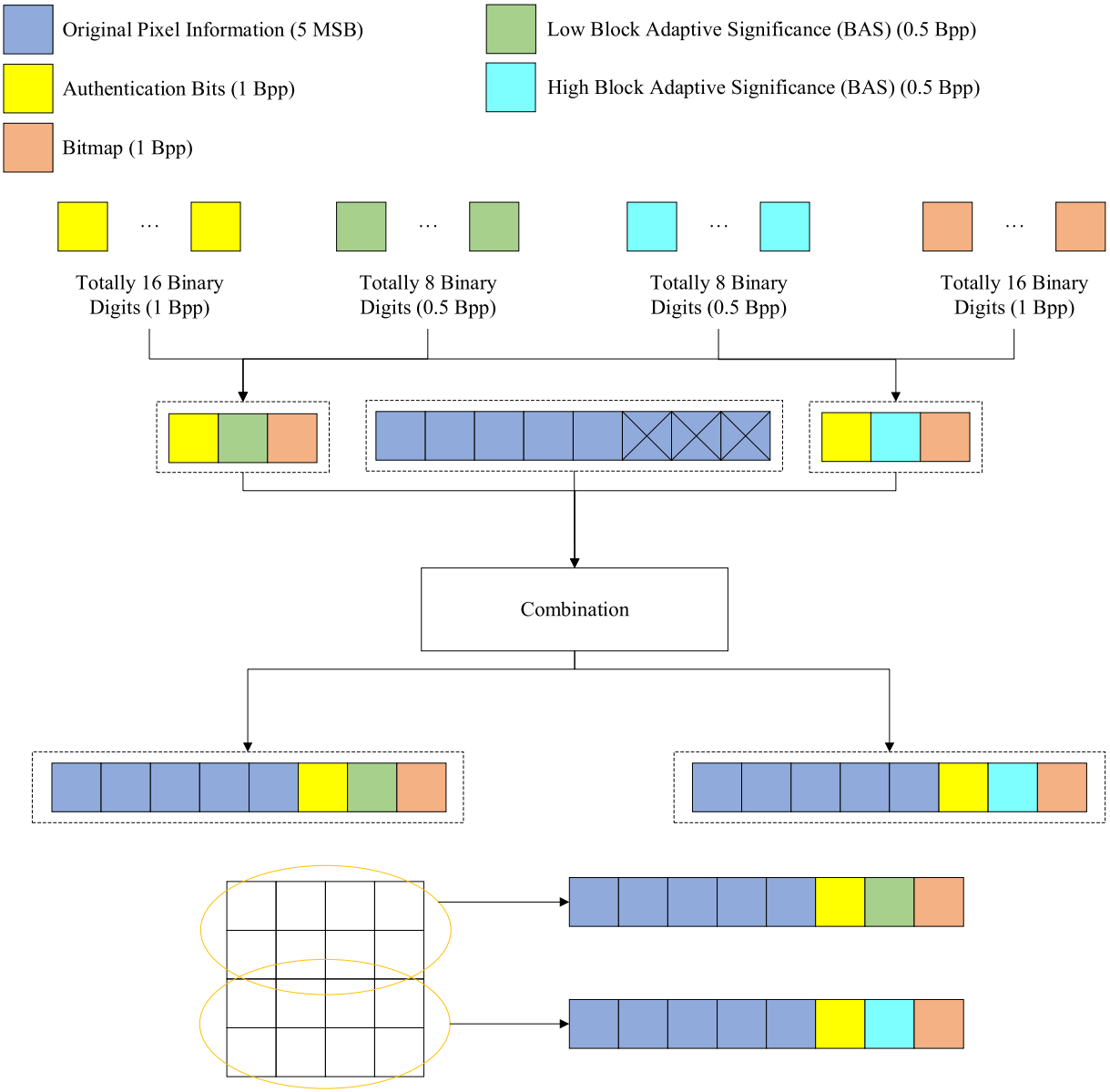


Fig. 2. Demonstration of Watermark Embedding.

result may be incorrect. In this case, we propose a second supplementary algorithm BBL that can perfectly solve this problem. Therefore, after the extracted block label $A_{extra}(i,j)$ of each block is obtained, we need to determine whether the first extracted block label is (1,1). If the block coordinate is (1,1), we conduct the PBL algorithm; otherwise perform the BBL algorithm.

For the PBL, the obtained $A_{extra}(i,j)$ need to be checked whether it conforms to Increase-Encoding. If a block label is not in this order, mark this block as a tampered block. For the BBL, we use the BCL algorithm to calculate the authentication bits, which will get the calculated block label $A_{cal\_block\_label}$. Then we compare the calculated block label $A_{cal\_block\_label}$ with the extracted block label $A_{extra}(i,j)$. And the block which has two different block labels is marked as a tampered block. Finally, we can obtain the tamper detection result D_T after all of the blocks are marked.

Algorithm 2: Procedure of Tamper Detection

Input: Received Image $I_R \rightarrow M \times N$;
Output: Detection Result D_T;
I_b = blocking$(I_R, M_T, N_T)\rightarrow$ Divide image into blocks size of $M_T \times N_T$.

(*continued on next column*)

(*continued*)

Algorithm 2: Procedure of Tamper Detection

For $i = 1$ to M/M_T do
  For $j = 1$ to N/N_T do
    Calculated Authentication Bits.$A_{cal\_block\_label} = BCL(I_R)$
    Extracted Authentication Bits form LSB-2 of I_R. $A_{temp_{extra}}(i,j) = Extract\_LSB2(I_R)$
    Inverse digital base conversion $A_{extra}(i,j) = 2\_10\_inverse(A_{temp_{extra}}(i,j))$
    If $A_{extra}(1,1) = (1,1)$
      If $A_{extra}(i,j) = (i,j)$
        Mark $A_{extra}(i,j)$ as not tampered
        Continue;
      Else Mark $A_{extra}(i,j)$ as tampered then
        $A_{temp_{locate}}(1, tempered\_block) = i$
        $A_{temp_{locate}}(2, tempered\_block) = j\rightarrow$ Locate and recording tampered block.
      End If
    Else If $A_{extra}(i,j) = A_{cal\_block\_label}(i,j)$
      Mark $A_{extra}(i,j)$ as not tampered
      Continue;
    Else Mark $A_{extra}(i,j)$ as tampered then
      $A_{temp_{locate}}(1, tempered\_block) = i$
      $A_{temp_{locate}}(2, tempered\_block) = j\rightarrow$ Locate and recording tampered block.
    End if

(*continued on next page*)

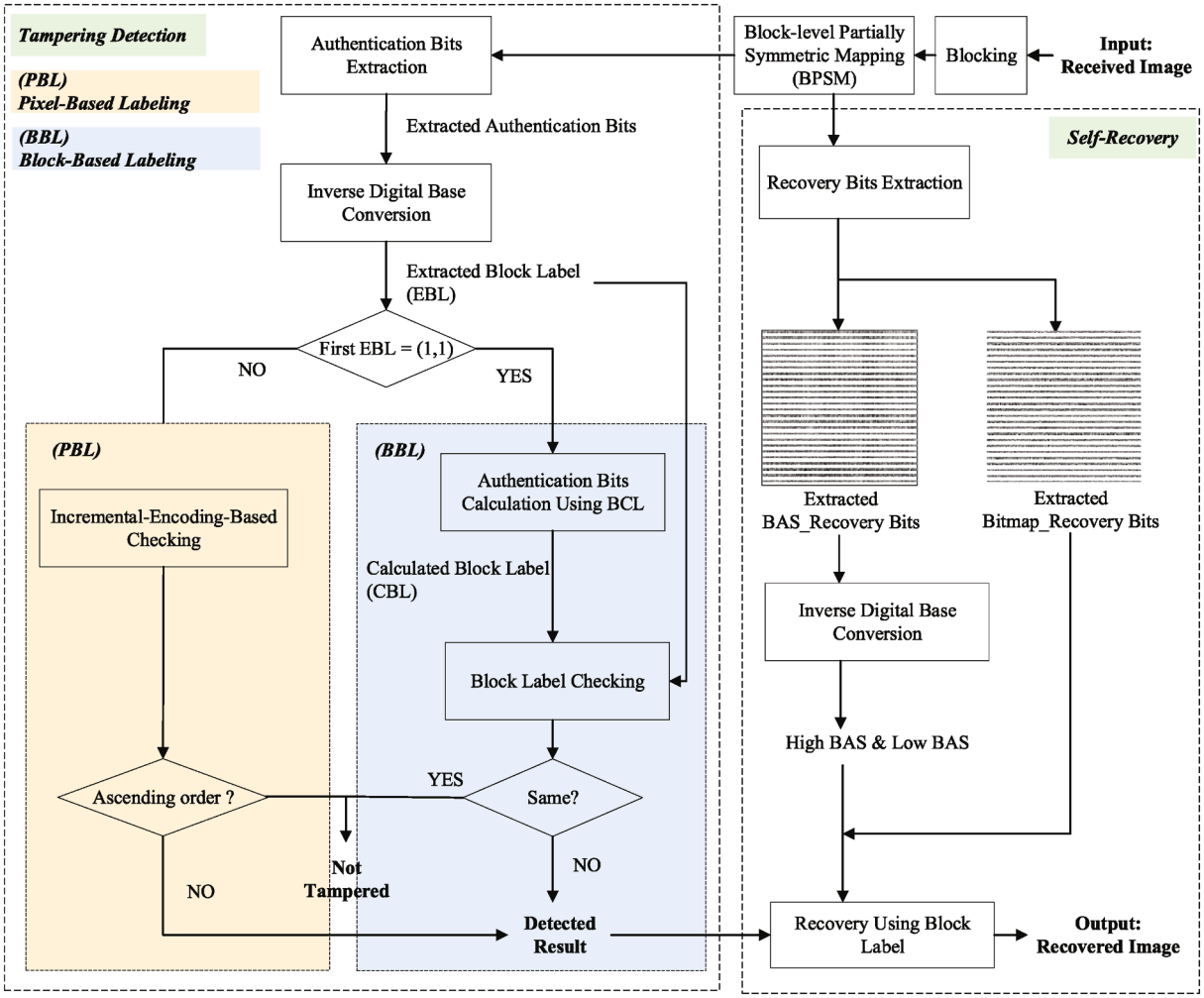


Fig. 3. Flowchart of Dual Tampering Detection and Self-Recovery Based on BAS.

(*continued*)

Algorithm 2: Procedure of Tamper Detection

End for
End for
Block-level detection result D_T=A_{extra}

After detecting the tampered area, we need to search and locate the position where the recovery bits of the corresponding blocks are stored. In the procedure of tamper detection, we record the coordinates of the blocks in tampered area. According to the coordinates of the tampered blocks, we then use (1) to find the blocks which record the recovery bits. The details of the proposed scheme are shown in Fig. 4. First, high BAS and low BAS are extracted from the penultimate digit of each block, and then convert them to decimal numbers. The second step is to extract the last bit of every pixel in the block, which will obtain a 4 × 4 bitmap matrix. Finally, we fill the extracted high BAS and low BAS into the block of the tampered position according to 0,1 position of the bitmap. The corresponding relationship is 0 corresponds to low adaptive significance, and 1 corresponds to high adaptive significance.

Algorithm 3: Image Self-recovery Using BAS

Input: Received Image I_R of size $(M \times N)$;
Output: Recovered Image I_{rec} of size $(M \times N)$;
1 $I_b = Blocking(I_R, M_T, N_T) \rightarrow$ Divide image into blocks size of $M_T \times N_T$.
2 We use Partially Symmetric Mapping to $A_{temp_{locate}} \rightarrow$ to get the location where tampered block recovery information is stored

(*continued on next column*)

(*continued*)

Algorithm 3: Image Self-recovery Using BAS

$A_{locate\_PSM} = PSM(A_{temp_{locate}}) \rightarrow$ Block-Level partially symmetric mapping
3 **procedure** IMAGE SELF-RECOVERY
α = (longth of A_{locate_PSM})
For $i = 1$ to α do
Set $A_{locate\_PSM}(1,i) \rightarrow$ recovery bits storage coordinate X_i
Set $A_{locate\_PSM}(2,i) \rightarrow$ recovery bits storage coordinate Y_i
$\rightarrow [X_i, Y_i]$ is the recovery bits storage coordinate label.
Set $A_{temp_{locate}}(1,i) \rightarrow$ tampered location coordinate X_{temp_i}
Set $A_{temp_{locate}}(2,i) \rightarrow$ tampered location coordinate Y_{temp_i}
$\rightarrow [X_{temp}, Y_{temp}]$ is tampered location coordinate label.
Then $[L_A[X_i, Y_i], H_A[X_i, Y_i]] = Extract\_LSB1(I_b[X_i, Y_i])$
$B_{map}[X_i, Y_i] = Extract\_LSB0(I_b[X_i, Y_i])$
For $j = 1$ to M_T do
For $k = 1$ to N_T do
If $B_{map}[X_i, Y_i]$ *of* $(j,k) == 1 \rightarrow (j,k)$ is the pixel coordinate label in the block.
$D_T[X_{temp\_i}, Y_{temp\_i}]$ *of* $(j,k) = H_A[X_i, Y_i]$
Else if $B_{map}[X_i, Y_i]$ *in* $(j,k) = = 0$
$D_T[X_{temp\_i}, Y_{temp\_i}]$ *of* $(j,k) = L_A[X_i, Y_i]$
End else if
End if
End for
End for
End for
Construct the recovery data I_{re} using (7)
$I_{re} = D_T[X_{temp\_1}, Y_{temp\_1}]||D_T[X_{temp\_2}, Y_{temp\_2}]||\cdots||D_T[X_{temp\_\alpha}, Y_{temp\_\alpha}]$(7)
$I_{rec} = I_R \bigcup I_{re}$(8)
End procedure

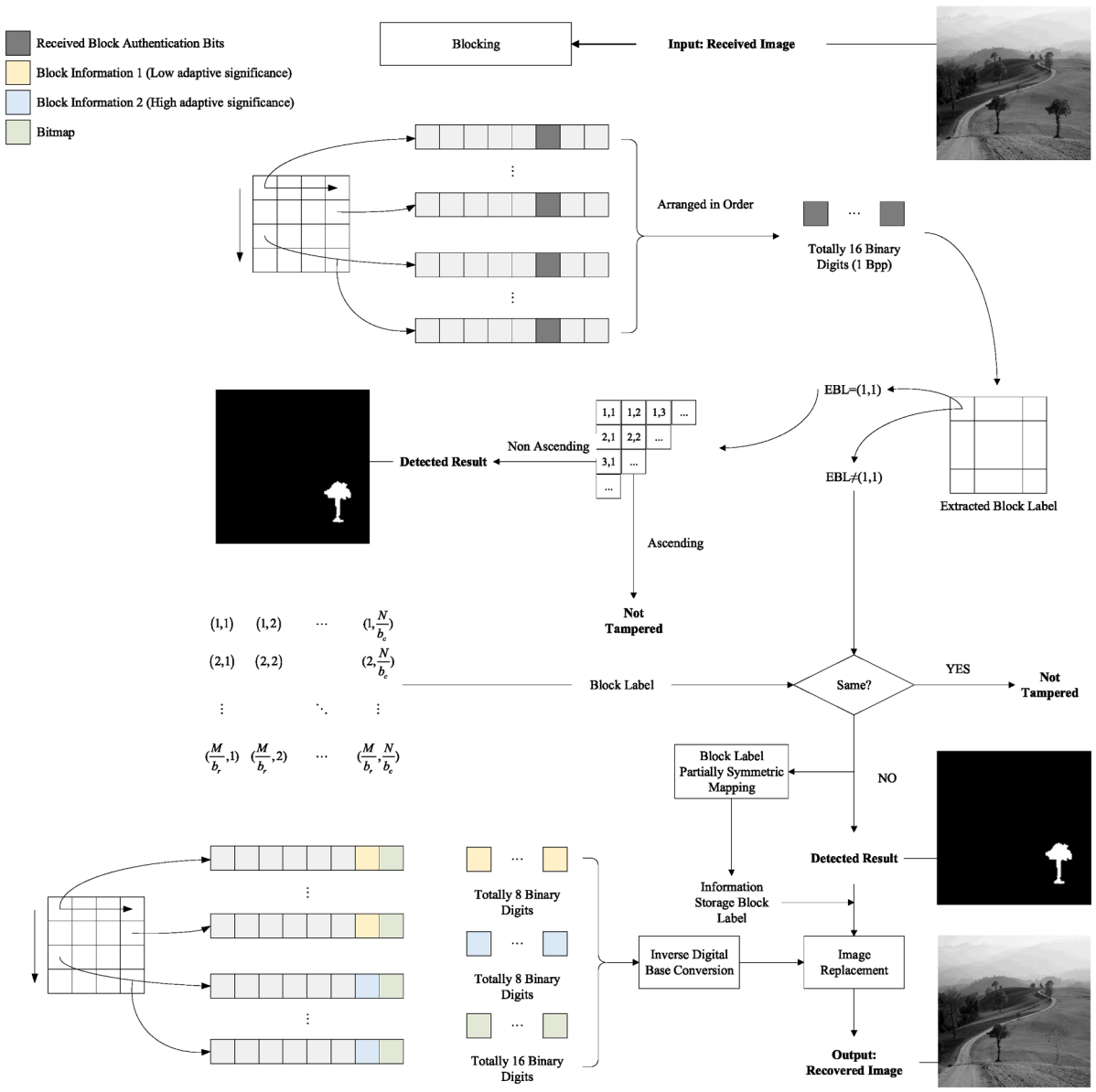


Fig. 4. Demonstration of Watermark Extraction and Tampered Region Detection and Self-Recovery.

4. Experimental results and discussions

In this section, we use two public datasets for practicality and efficiency assessments of our proposed scheme. One is the BOWS2 (Bas & Furon, 2007) dataset which contains a large number of manipulated images that are challenging to recognize with the naked eye. The other is the USC-SIPI (Weber 2006) dataset which includes several standard images of 512 × 512. In our experiments, we use MATLAB 2020b on a computer with a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz processor and 16.0 GB of RAM, to perform our analyses and generate our results. During experiments, we set the images to grayscale and 512 × 512 in size for consistency. To evaluate the accuracy of tampered region localization, *Precision*, *Recall*, and *F1 score*, which are defined in (9)~(11), are the metrics we use in this paper. The *Precision* indicates the accuracy of the manipulated pixels of predicted result. The *Recall* is the capacity to record really altered pixels. The *F1 score* is a comprehensive measurement that combines *Precision* and *Recall*. To evaluate quality of the watermarked image and recovered image, *PSNR* and Structural Similarity Index Measure *(SSIM)*, as defined in (12) and (14), are calculated respectively, to reveal the difference in quality and structural similarity between images. In addition, the Natural Image Quality Evaluator *(NIQE)* (Mittal et al. 2012), as defined in (15), is calculated to evaluate image quality, without relying on human-evaluated distorted images. It is a fully blind image quality analyzer that uses statistical regularities observed in natural images, lower NIQE scores indicate higher image quality.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1Score = \frac{2TP}{2TP + FP + FN} \tag{11}$$

where TP and TN represent the total amount of detected modified pixels and pixels which are not modified, respectively, and FP and FN represent the amount of wrongly labeled modified pixels and incorrectly detected non-modified pixels, respectively.

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{12}$$

where MAX is the highest value that pixels in the comparison image can choose from, and MSE stands for the mean square error between two images.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{13}$$

I and K are two images to be compared respectively, and m × n is the size of the image.

$$\text{SSIM} = \frac{(2\mu_1\mu_2 + c_1)(2\sigma_{12} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2)} \tag{14}$$

where μ and σ represent the average and variance values of the pixels. The σ_{12} means the covariance between two images. The c_1 and c_2 are constant values that are used to avoid the instability caused by denominators close to 0.

$$D(v_1, v_2, e_1, e_2,) = \sqrt{((v_1 - v_2)^T \times \left(\frac{e_1 - e_2}{2}\right)^{-1} \times v_1 - v_2))} \tag{15}$$

where v_1, v_2 is the mean vector of the natural Multivariate Gaussian (MVG) model (Mittal et al. 2012) and the distorted image MVG model. And the e_1, e_2 is the covariance matrix of the natural MVG model and the distorted image MVG model.

4.1. Evaluation of the proposed scheme

To demonstrate the superiority of our method, we evaluate our method using BOWS2 (Bas & Furon, 2007) database and USC-SIPI (Weber 2006) standard image database. The results obtained by applying our method to the BOWS2 dataset are used as the primary evaluation, and the specific experimental results are presented in section 4.1.1. The USC-SIPI dataset is mainly used to evaluate our proposed scheme under different tampering rates, and the visualization results under different tampering rates are shown in section 4.1.2.

4.1.1. Performance on BOWS2 dataset

To demonstrate the feasibility and completeness of our method, we select five representative examples from BOWS2 dataset with tampered parts of varying sizes and give the experiment results in Fig. 5. The 1st row lists the original images, 'hedge', 'jellyfish_chaos', 'knight_moves', 'kore', and 'no_beach'. The 2nd row is the watermarked images. According to the figure, we can see that our embedding method has no perceptual distortion on images, which indicates the good imperceptibility of our scheme. Moreover, for objective evaluation, the *PSNR*, *SSIM* and *NIQE* are calculated as: {*PSNR, SSIM, NIQE*} = {*37.79 dB, 0.8943, 5.046*}, {*37.86 dB, 0.9188, 6.039*}, {*37.93 dB, 0.9311, 4.507*}, {*37.70 dB, 0.9386, 4.905*}, and {*37.83 dB, 0.9520, 4.588*}. The 3rd row shows the tampered images where we marked the tampered area with a yellow circle for easy tracking and comparison. The 4th row is the ground truth, from which the tampering rates are calculated as 4.37%, 0.12%, 0.013%, 23.6%, and 0.64%. The 5th row presents the detected regions, and the corresponding *Precision, Recall*, and *F1 Scores* are shown as *{Precision, Recall, F1 Score} = {96%, 99%, 98%}, {79%, 100%, 88%}, {81%, 100%, 90%}, {91%, 99%, 95%},* and *{73%, 100%, 84%}*. Clearly, our method can localize the tampered area precisely, whether it is a large or a small area. In addition, no matter how smooth the edge is or the complexity of the texture, we can accurately find and mark the tampered part, which shows the sensitivity and universality of our detection algorithm. And the last row gives the recovered images of our method, and the *PSNR*, *SSIM and NIQE* used to evaluate the image quality are given as follows*: {PSNR, SSIM, NIQE} = {48.28 dB, 0.9916, 5.517}, {57.03 dB, 0.9992, 6.069}, {45.95 dB, 0.9994, 4.511}, {30.63 dB, 0.9377, 5.050}, and {41.08 dB, 0.9967, 4.820}*. The *PSNR and SSIM* demonstrate that our scheme can achieve high quality of recovered images. Furthermore, our scheme generates restored images with low NIQE values, indicating that tampered regions are effectively restored, meanwhile recovered images are highly natural. In conclusion, the reliability of our method is directly proved by the detection accuracy, recovery quality, and naked-eye observation.

Fig. 6 comprehensively shows the performance of our method in tampering detection and recovery using BOWS2 (Bas & Furon, 2007) dataset. In Fig. 6, the 1st column shows the results of tampering detection in terms of *Precision*, *Recall,* and *F1 Scores* respectively, as shown in (a1), (a2), and (a3). The obtained results are: {Max, Average} *Precision* = {*98.23%, 86.70%*}, {Max, Average} *Recall* = {*100%, 99.42%*}; {Max, Average} *F1* = {98.57%, 92.02%} for block size 4 × 4; while for block size 8 × 8, {Max, Average} *Precision* = {88.66%, 77.13%}, {Max, Average} *Recall* = {99.04%, 98.55%}; {Max, Average} *F1 Score* = {94.56%, 87.38%}. The 2nd column shows the results of image self-recovery in terms of *PSNR, SSIM, and NIQE respectively*, as shown in (b1), (b2), and (b3). The obtained results are: {Max, Average} *PSNR* = {57.03 dB, 41.66 dB}, {Max, Average} *SSIM* = {0.9787, 0.9398}, {Min, Average} *NIQE* = {3.460, 5.163} for block size 4 × 4; while for block size 8 × 8, {Max, Average} *PSNR* = {55.86 dB, 40.47 dB}, {Max, Average} *SSIM* = {0.9656, 0.9114}, {Min, Average} *NIQE* = {3.873, 6.783}. The results show that the proposed scheme can achieve a better performance in both tampering detection and image recovery, when the block size is 4 × 4. Therefore, in the experiments, we set the block size to be 4 × 4.

4.1.2. Performance on USC-SIPI dataset

In this section, we evaluate the proposed scheme on the USC-SIPI (Weber 2006) dataset by implementing different tampering rates. As shown in Fig. 7, the classic images 'Lena', 'Pepper' and 'plane' are selected as examples and tampering with different tampering rates varying from 10% to 50% with a step size of 10%. In Fig. 7, the first row is the horizontal tampering on 'Lena', the third row is the vertical tampering on 'Pepper', and the fifth row is the oblique tampering on 'Plane'. The corresponding recovered images are shown in the second, fourth, and sixth rows. The average *PSNR, SSIM and NIQE* of the recovered images are: {*PSNR, SSIM, NIQE*} = {*42.96 dB, 0.9869, 6.3134*} for 10% tampering in 1st column, {*PSNR, SSIM, NIQE*} = {*39.59 dB, 0.9749, 6.0798*} for 20% tampering in 2nd column, {*PSNR, SSIM, NIQE*} = {*37.52 dB, 0.9619, 6.0071*} for 30% tampering in 3rd column, {*PSNR, SSIM, NIQE*} = {*36.28 dB, 0.9502, 5.9669*} for 40% tampering in 4th column, {*PSNR, SSIM, NIQE*} = {*35.23 dB, 0.9401, 5.9452*} for 50% tampering in 5th column. The results show that our method is efficient in image recovery when under the various tampering rates.

4.2. Significance of BBL

In this section, we show the significance of the proposed BBL by comparing PBL and BBL methods. Fig. 8 shows the visualization of two examples where the PBL fails to detect the tampered region, while the BBL succeeds in the detection. In Fig. 8, the 1st column shows the original images, the 2nd column shows the corresponding tampered images, and the 3rd column shows the ground truth. The 4th and 5th columns respectively show the results of using PBL and BBL, where it is clearly that the PBL fails to detect the tampered region. In contrast, by leveraging the BBL method, we are able to obtain accurate detection results. This outcome serves to demonstrate that the BBL method serves as a viable remedy to the limitations of the PBL method, thus attesting to the universality of the BBL method.

Table 1 presents the results of our analysis of the tamper detection time, recovery time, and overall time consumption for different detection methods. The averaged detection time for tampering using BBL is 1.11 s, while the averaged detection time using PBL is 0.8 s. This difference in detection time can be attributed to the fact that BBL requires

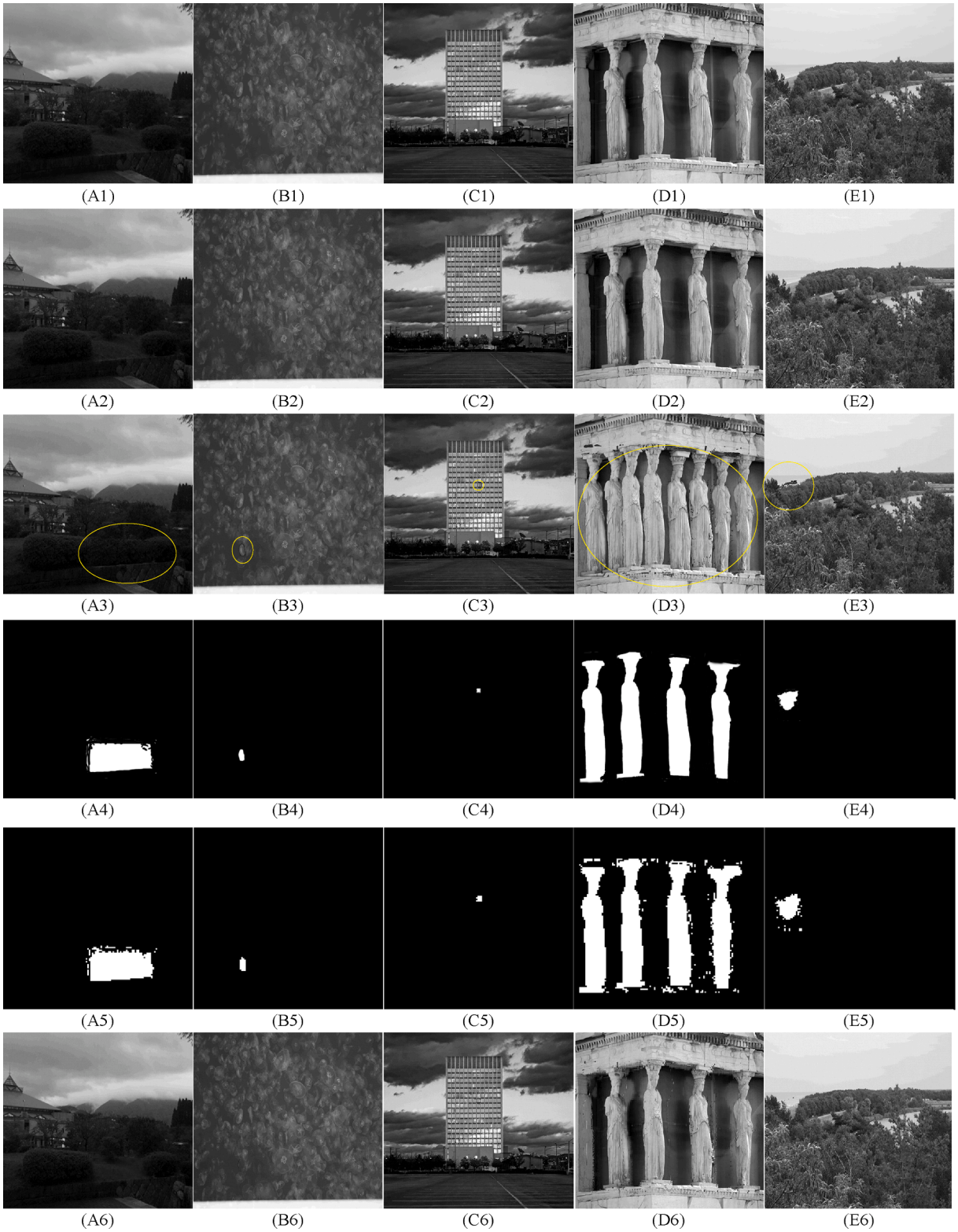


Fig. 5. Visualization of the results on BOWS2 (Lin et al., 2009) database. (A1)~(E1) are the original images; (A2)~(E2) are the corresponding watermarked images: {PSNR, SSIM, NIQE} = {37.79 dB, 0.8943, 5.046}, {37.86 dB, 0.9188, 6.039}, {37.93 dB, 0.9311, 4.507}, {37.70 dB, 0.9386, 4.905}, and {37.83 dB, 0.9520, 4.588}; (A3)~(E3) are the tampered images; (A4)~(E4) are the ground truth, with the tampering rate: 4.37%, 0.12%, 0.013%, 23.6%, 0.64%; (A5)~(E5) are the detected tampered results: {Precision, Recall, F1 Score} = {96%, 99%, 98%}, {79%, 100%, 88%}, {81%, 100%, 90%}, {91%, 99%, 95%}, {73%, 100%, 84%}; and (A6)~(E6) are the recovered image: {PSNR, SSIM, NIQE} = {48.28 dB, 0.9916, 5.517}, {57.03 dB, 0.9992, 6.069}, {45.95 dB, 0.9994, 4.511}, {30.63 dB, 0.9377, 5.050}, and {41.08 dB, 0.9967, 4.820}.

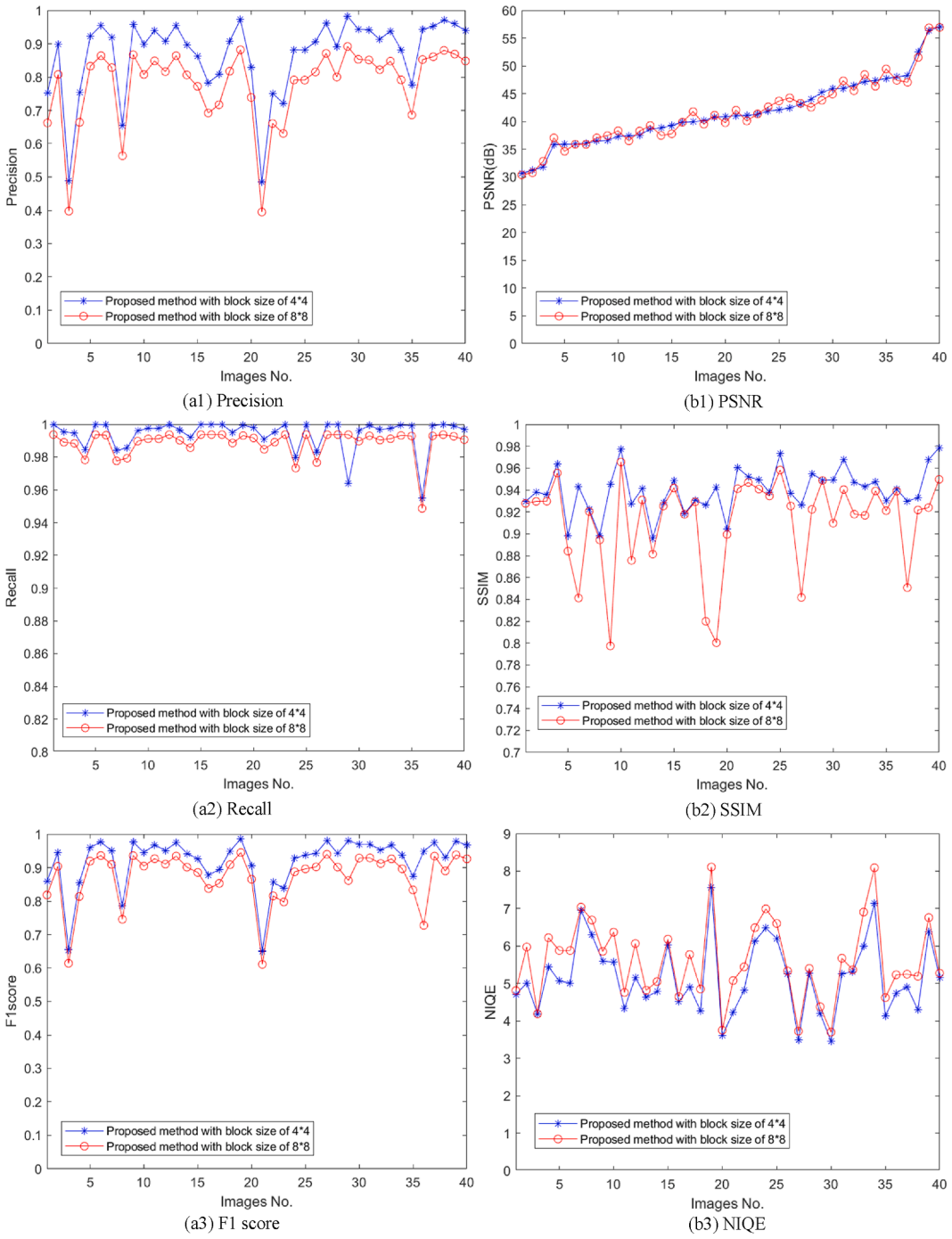


Fig. 6. Results of tampering detection and image self-recovery using different block sizes, 1st column: results of tampering detection in terms of (a1) Precision, (a2) Recall, and (a3) F1 score; 2nd column: results of self-recovery in terms of (b1) PSNR, (b2) SSIM, and (b3) NIQE.

recalculation of the Block Label, which takes more time than PBL. However, BBL demonstrates a higher degree of universality in detecting tampering, making it a more effective method for detecting tampering in a wider range of scenarios. Overall, our findings highlight the tradeoff between detection time and universality of detection when selecting a tamper detection method, with BBL providing a more accurate approach to tamper detection at the expense of longer detection time. While similar mechanisms have been proposed in the past, recent research by Liu et al. (Liu & Yuan, 2021) has also proposed a Dual-Tamper-Detection scheme that uses two check bits to reduce the probability of false-negative errors. Each image should be detected twice from both pixel level and block level. The corresponding time expense of their scheme with block size 4×4 was measured as 17.90 s. We propose the PBL tampering detection method that only requires us to extract authentication information and check if it conforms to the ascending sequence. This allows for rapid localization and only requires the image to be detected once. Compared to Liu et al.'s method, our approach is faster and more efficient, with a reduced computational time for self-recovery based on PBL and BBL, which are 0.813 s and 1.123 s respectively. In terms of detecting tampering accuracy, our proposed method has an average F1 Score of 92.02% for blocks of size 4×4. This is higher than the existing method (Liu & Yuan, 2021), which has an accuracy of 90.93%.

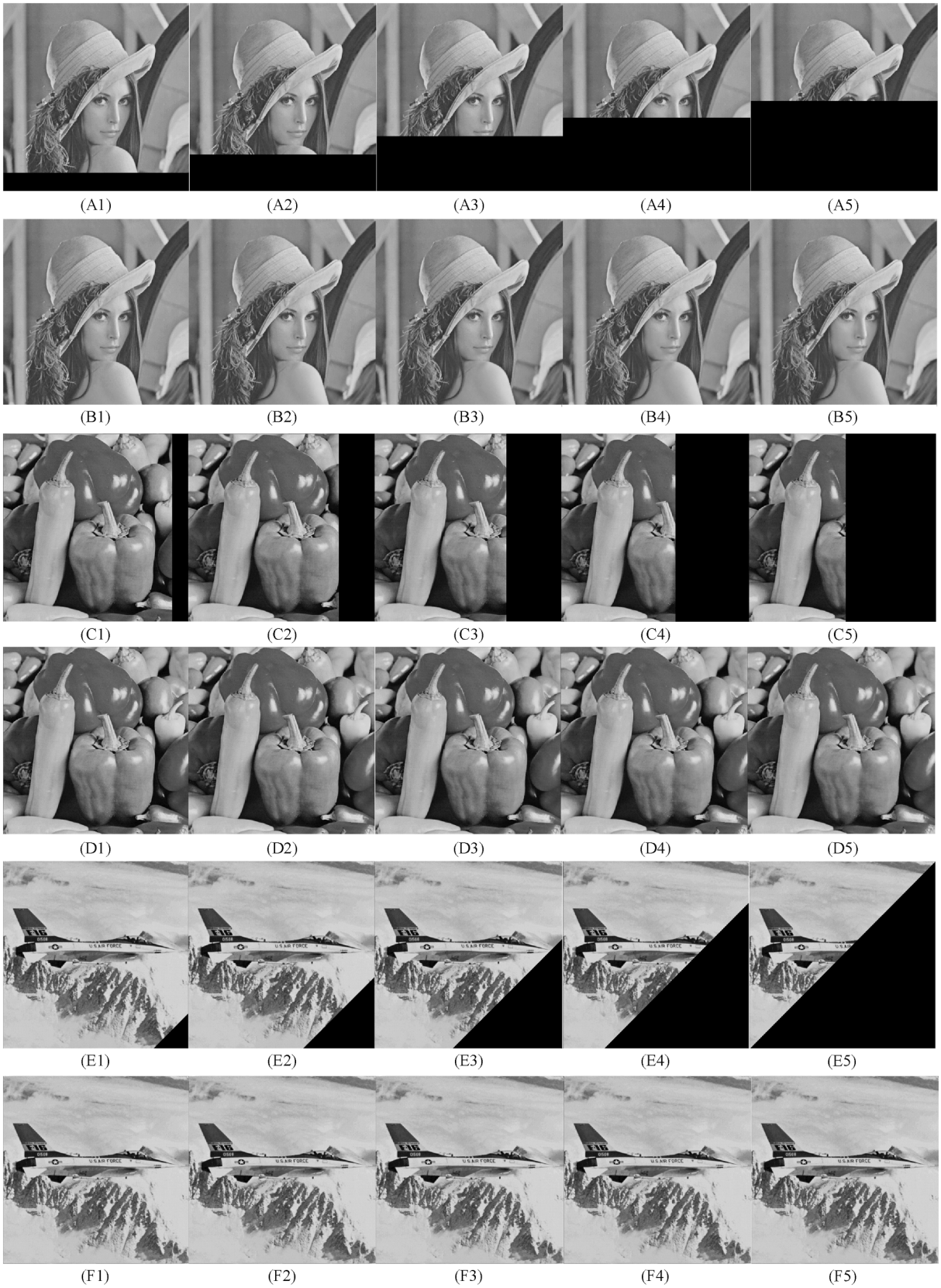


Fig. 7. Example of the results on USC-SIPI (Liu and Yuan, 2020) dataset under different tampering rates: 10% in the 1st column, 20% in the 2nd column, 30% in the 3rd column, 40% in the 4th column, and 50% in the 5th column; (A1)~(A5), (C1)~(C5), (E1)~(E5) are the tampered images; (B1)~(B5), (D1)~(D5), (F1)~(F5) are the corresponding recovered images.

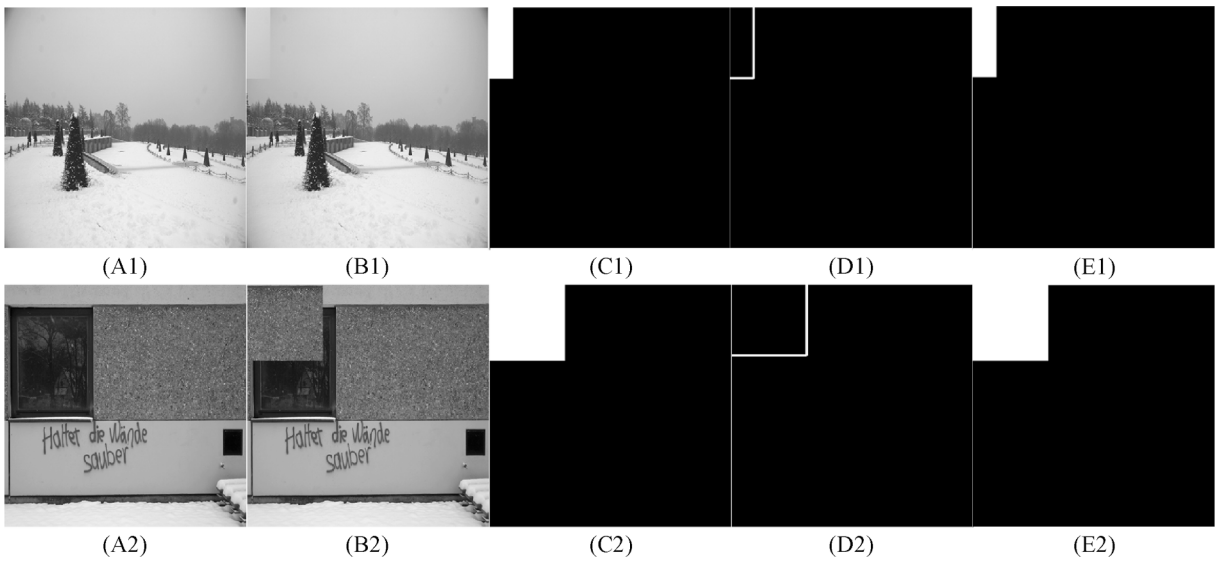


Fig. 8. Visulation of significance of BBL. (A1), (A2) are the original images; (B1), (B2) are the tampered images; (C1), (C2) are the ground truth; (D1), (D2) are the detection results using PBL method; (E1), (E2) are the detection results using BBL method.

Table 1
Comparison of runtime for tamper detection and recovery using BBL and PBL.

	Tamper Detection using BBL (s)	Tamper Detection using PBL (s)	Image Recovery (s)	Total of using BBL (s)	Total of using PBL (s)
MAX	1.243	0.957	0.037	1.280	0.994
MIN	1.061	0.766	0.001	1.062	0.767
Average	1.110	0.800	0.013	1.123	0.813

4.3. Robustness of the proposed scheme

As the content of images may be intentionally altered during transmission, a desirable tamper detection system should have strong robustness to resist different attacks. The collage attack and the copy-move attack are the two primary attacks that frequently arise in the areas of tampering localization and self-recovery. The collage attack creates attacked images by keeping portions of at least two images in the same relative spatial location. Under this attack, the tampered part also has the watermark content of its original image. The copy-move attack substitutes information from other locations in the same image for its own content. Since the blocks come from the same image and are independent of each other, copy-move attacks are often difficult to deal with.

In Figs. 9 and 10, we exhibit the performance of the proposed approach under collage and copy-move attacks to demonstrate the robustness of the proposed approach. The original images are listed in the 1st column. The 2nd column presents the tampered image with attacks, where the tampered area is marked prominently with yellow circles. Moreover, the arrow in Fig. 10 indicates the direction of replica movement. Next, the 3rd column is the ground truth of tampered area and the 4th column shows the detected regions with our method, the corresponding *Precision, Recall,* and *F1score* values are given accordingly. Last, the 5th column shows the recovered images, with the *PSNR*

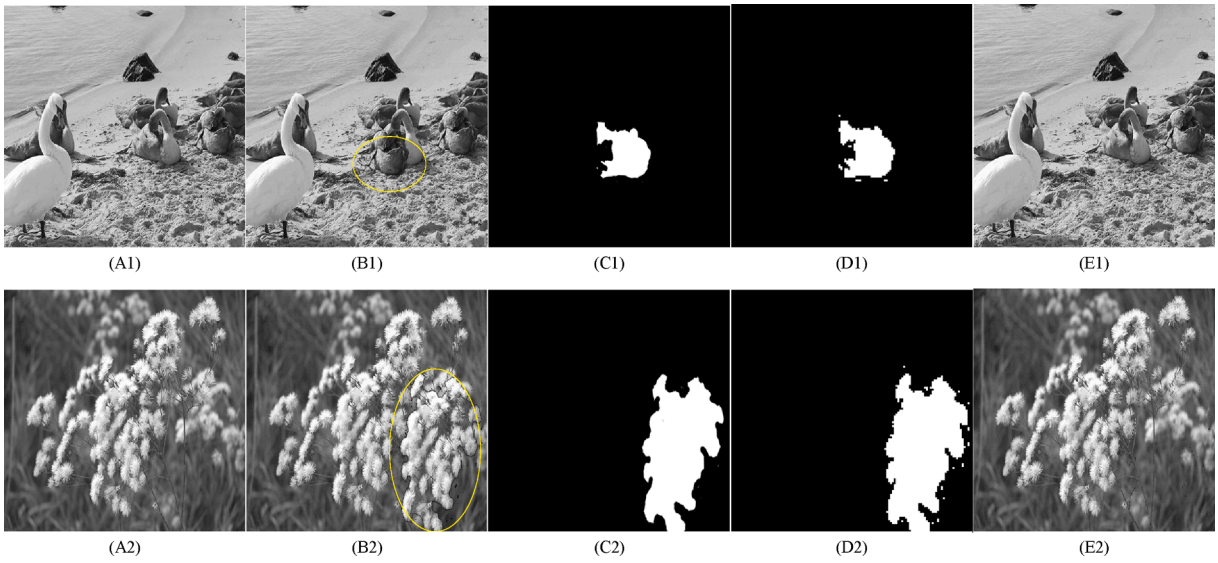


Fig. 9. Robustness of proposed method under the collage attack. (A1), (A2) are the original images; (B1), (B2) are the tampered images by collage attack; (C1), (C2) are the ground truth, with the tampering rate as 3.1% and 12.67%, respectively; (D1), (D2) are the detected tampered regions: {Precision, Recall, F1 Score} = {91%, 99%, 95%}, {95%, 99% 97%}; (E1), (E2) are the recovered images: {PSNR, SSIM, NIQE} = {37.14 dB, 0.9937, 5.2570}, {38.01 dB, 0.9753, 4.7432}.

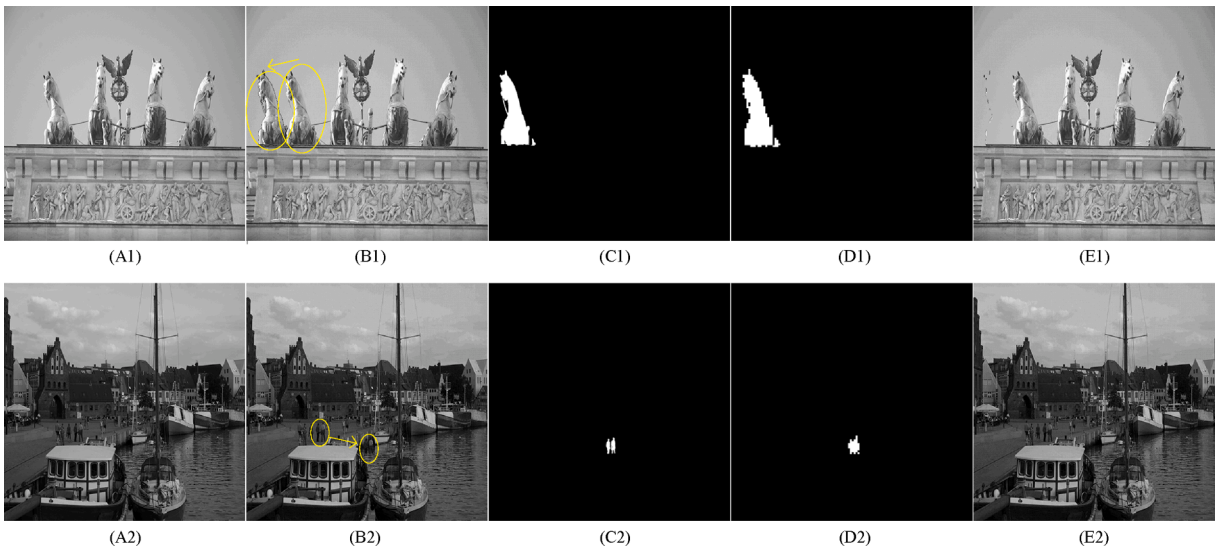


Fig. 10. Robustness of proposed method under the copy-move attack. (A1), (A2) are the original images; (B1), (B2) are the tampered images by copy-move; (C1), (C2) are the ground truth, with the tampering rate as 2.1%, 0.12%; (D1), (D2) are the detected tampered regions: {Precision, Recall, F1 Score} = {90%, 99%, 94%}, {77%, 99% 87%}; (E1), (E2) are the recovered images: {PSNR, SSIM, NIQE} = {39.30 dB, 0.9927, 4.6364}, {52.52 dB, 0.9995, 4.1296}.

and *SSIM* values calculated. For the examples in the two rows in Fig. 9, {*Precision, Recall, F1 Score*} = {*91%, 99%, 95%*} and {*95%, 99%, 97%*}, while {*PSNR, SSIM, NIQE*} = {*37.14 dB, 0.9937, 5.2570*} and {*38.01 dB, 0.9753, 4.7432*}. For the two examples in Fig. 10, {*Precision, Recall, F1 Score*} = {*90%, 99%, 94%*} and {*77%, 99%, 87%*}, while {*PSNR, SSIM, NIQE*} = {*39.30 dB, 0.9927, 4.6364*} and {*52.52 dB, 0.9995, 4.1296*}. The experiment results indicate that our method can still perform well under collage and copy-move attacks. In addition, different sizes of

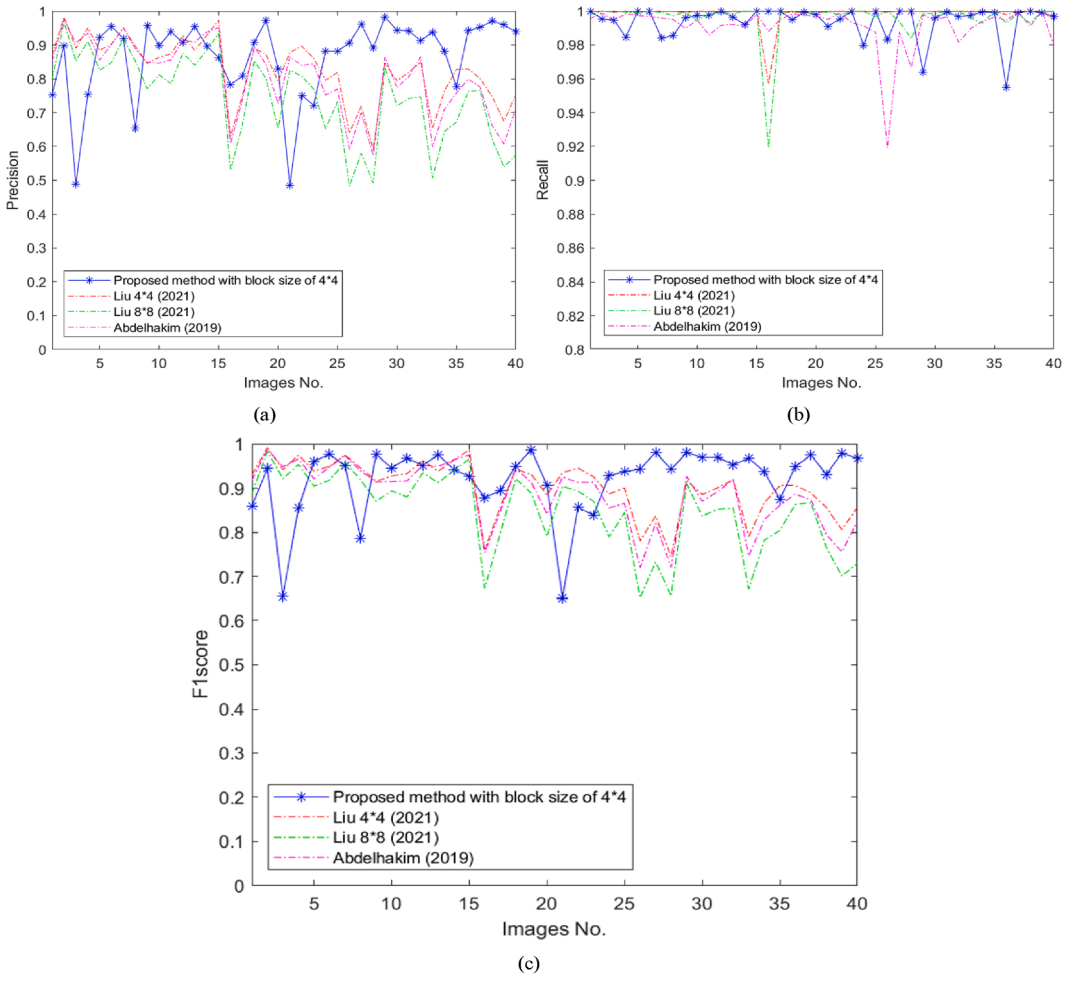


Fig. 11. Comparisons with existing works in tampering detection. (a) Precision, (b) Recall, (c) F1 score.

tampering are simulated in the examples, where tampering rates are 3.1%, 12.67%, 2.1%, and 0.12%, and the results indicate that no matter the tampered region is large or small, our method achieves satisfying results.

4.4. Comparison with the existing works

In this section, we compare our method with state-of-the-arts to demonstrate the superiority of our method. Fig. 12 presents the comparison in terms of tampering detection of our method with (Abdelhakim et al. 2019) and (Liu and Yuan, 2020) on BOWS2 (Bas & Furon, 2007) database. The average *Precision, Recall,* and *F1 Score* of Abdelhakim's (Abdelhakim et al. 2019) and Liu's (Liu and Yuan, 2020) methods are *80.52%*, *99.12%*, *88.47%* and *82.78%*, *99.86%*, *90.23%*, respectively; while the average *Precision, Recall* and *F1 Score* of the proposed method are ***86.70%, 99.42% and 92.02%***. In Fig. 11-(c), we present the results of a comparison between the comprehensive index F1score of our proposed scheme and that of other existing methods. The results indicate that our proposed scheme outperforms most existing methods in detecting tampered regions in digital images. This suggests that our scheme can be an effective solution for tamper detection in a wide range of applications. The experiment result shows that our method achieves improvements in *Precision* and *F1 Score*, which indicates that our method localizes the tampered region more accurately than others.

In addition to the performance in tampering detection, we also evaluate the recovery quality of tampered images by comparing the proposed method with the existing works on the USC-SIPI (Weber 2006) dataset. In Fig. 12, we show the *PSNR* and *SSIM* values of the recovered images, and compare our results with the existing works Al-Otum (Al-Otum & Ibrahim, 2021), Haghighi (Haghighi et al. 2019), Gul (Gul & Ozturk, 2021). The horizontal axis indicates the different tampering rates, the range from 10% to 50%, with the step of 10%. The restored *PSNR* values of the 'boat' with proposed method are calculated as *42.96 dB*, *41.59 dB*, *37.52 dB*, *36.28 dB*, and *35.13 dB*, under the different tampering rates; and the corresponding *PSNR* values are *44.46 dB*, *42.67 dB*, *38.46 dB*, *35.07 dB*, and *34.01 dB* for 'goldhill'. On the other hand, the *SSIM* values of the recovered 'boat' are *0.9861*, *0.9747*, *0.9624*, *0.9513*, and *0.9392*, under the different tampering rates; and the corresponding *SSIM* values are *0.9863*, *0.9743*, *0.9618*, *0.9500*, and *0.9364* for 'goldhill'. It can be seen that on the basis of ensuring the image structure, our method achieves higher quality of recovered image.

Table 2 comprehensively compares the average PSNR of our proposed scheme with existing work under different tampering rates. When the tampering rate is higher than 20%, our scheme can achieve significantly better restored image quality. Our scheme achieves an average PSNR of **39.59 dB** when the tampering rate is 20%. In addition, our scheme can still achieve an average PSNR of **35.23 dB** when the tampering rate is 50%.

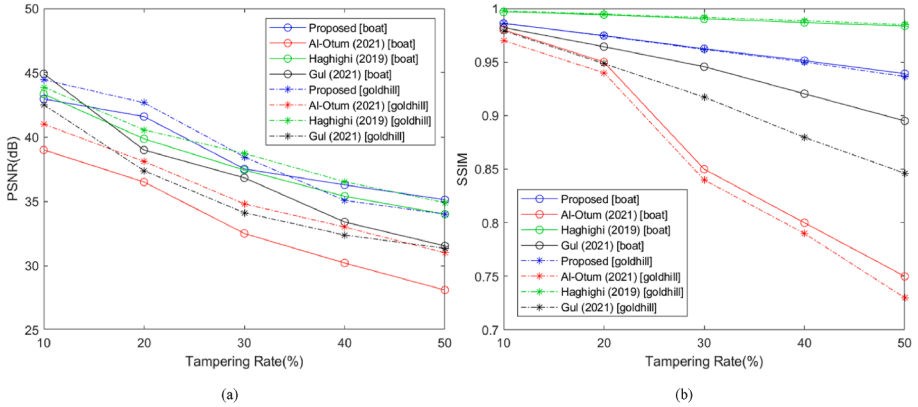
Table 2
PSNR (dB) comparison with existing work at different tampering rates.

	Tampering Rate				
	10%	20%	30%	40%	50%
Haghighi et al. (2019)	42.60	39.20	37.08	35.96	34.45
Al-Otum and Ibrahim (2021)	39.64	37.30	33.65	31.60	29.55
Gul and Ozturk (2021)	**43.71**	38.18	35.47	32.88	31.44
Proposed method	42.96	**39.59**	**37.52**	**36.28**	**35.23**

Table 3 provides an overview of the comparison between our proposed scheme and the state-of-the-art methods in terms of imperceptibility (Average PSNR of Watermarked Image), recovery quality (PSNR of Recovered Image [Min, Max]), and sustainable Tampering Rate. Our proposed scheme exhibits a high level of imperceptibility, as evidenced by the average PSNR of the watermarked images, the watermark payload is 1.5 bpb, and the average PSNR of the watermarked image is

Table 3
Overall comparison with the state-of-the-art works.

Methods	Average PSNR of Watermarked Images	PSNR of Recovered Images [Min, Max]	Tampering Rate
Qin (2017)	42 dB	[29, 41] dB	$\alpha < 45\%$
Qin (2018), 4 × 4	**44 dB**	[33, 42] dB	$\alpha < 45\%$
Qin (2018), 8 × 8	**44 dB**	[31, 40] dB	$\alpha < 50\%$
Shehab et al. (2018)	**44 dB**	[29, 41] dB	$\alpha < 45\%$
Garcia (2020)	42 dB	[27,44] dB	$\alpha < 50\%$
Kim (2021)	40 dB	[36, 40] dB	$\alpha < 45\%$
Gul and Ozturk (2021)	38 dB	[32, 44] dB	$\alpha < 50\%$
Tohidi et al. (2021), 5 × 5	**44 dB**	[32, 42] dB	$\alpha < 55\%$
Tohidi et al. (2021), 7 × 7	**44 dB**	[31, 41] dB	$\alpha <$ **60%**
Tohidi et al. (2021), 9 × 9	**44 dB**	[32, 40] dB	$\alpha <$ **60%**
Al-Otum and Ibrahim (2021)	43.5 dB	[30, 44] dB	$\alpha < 50\%$
Proposed method, 4 × 4	38 dB	**[35, 57] dB**	$\alpha < 50\%$
Proposed method, 8 × 8	**44 dB**	**[32, 54] dB**	$\alpha < 50\%$



(a) (b)

Fig. 12. Comparisons with existing works on quality of recovered images under different tampering rates. (a) PSNR, (b) SSIM.

44 dB, which is among the highest in comparison to other methods. This indicates that our watermarking scheme can preserve the quality of the original image while embedding the watermark. Furthermore, our method is capable of producing high-quality recovered images, as demonstrated by the high PSNR of the recovered image, which ranges from the minimum to the maximum value. This highlights the strong self-recovery capability of our method, which is an important point for tamper detection and image self-recovery. Moreover, our proposed scheme exhibits better adaptability to tampering rates below 50%, suggesting that it can effectively detect tampering and maintain the integrity of the image. The results of our study collectively demonstrate the effectiveness of our proposed scheme in achieving high imperceptibility, precise localization of tampered regions, and self-recovery capabilities. Our scheme successfully maintains the quality of the original image while embedding the watermark, ensuring high imperceptibility. Additionally, it accurately identifies the tampered regions in the image, providing precise localization. Moreover, our scheme is capable of self-recovery, which enables it to recover the original image even after it has been tampered with.

5. Conclusion

This paper proposes a new scheme for localization and restoration of image tampered regions, utilizing the block labelling and block adaptive significances. The watermark information to be embedded includes two parts, the authentication data and the recovery data. The BCL method is proposed to generate authentication data which consists of the coordinate information of each block. Recovery data is generated by calculating high BAS, low BAS and bitmap, and the method of BPSM is then proposed to effectively increase the security of data hiding. Since the accuracy of tamper detection will directly affect the quality of the restored image, to balance detection speed and accuracy, we propose a dual detection scheme that adaptively selects the proposed PBL detection and BBL detection approach depending on a predefined condition of the received image. Specifically, PBL detection will be adaptively employed to quickly detected the tampering with a small amount of calculation when the predefined condition is satisfied; otherwise BBL detection with higher success rate will be performed. In the recovery phase, we use BPSM to find the corresponding recovery information block, and then extract the recovery data to restore tampered blocks. In the experiments, standard evaluation metrics, including *Precision*, *Recall*, and *F1 Score,* are employed to evaluate the effectiveness of our tamper detection approach; while *PSNR*, *SSIM*, and *NIQE* are used to evaluate the quality of the recovered images. In addition, different types of attacks are simulated, and it can be seen that our scheme has good robustness against common attacks. Admittedly, our method also has limitations. The main limitation of our proposed method is that the recovery performance is not satisfied when tampering is extensive. In future work, we aim to improve image restoration performance by reducing the payload, enabling more accurate image localization and restoration. Furthermore, we plan to improve the proximity estimation method to occupy less LSB space and have higher recovery quality. We also aim to use more efficient restoration methods to increase the recoverable image area and achieve high restoration quality even if the tampered area of the image is greater than 50%.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgment

This work is supported by the Science and Technology Development Fund of Macau SAR [grant number 0045/2022/A], and the research project of the Macao Polytechnic University [Project No. RP/FCA-12/2022].

References

Abdelhakim, A., Saleh, H. I., & Abdelhakim, M. (2019). Fragile watermarking for image tamper detection and localization with effective recovery capability using K-means clustering. *Multimedia Tools and Applications, 78*(22), 32523–32563. https://doi.org/10.1007/s11042-019-07986-3

Ahmadi, S. B. B., Zhang, G., Rabbani, M., Boukela, L., & Jelodar, H. (2021). An intelligent and blind dual color image watermarking for authentication and copyright protection. *Applied Intelligence, 51*(3), 1701–1732. https://doi.org/10.1007/s10489-020-01903-0

Al-Otum, H. M., & Ibrahim, M. (2021). Color image watermarking for content authentication and self-restoration applications based on a dual-domain approach. *Multimedia Tools and Applications, 80*(8), 11739–11764. https://doi.org/10.1007/s11042-020-10368-9

Bas, P. and T. Furon (2007). BOWS-2 (July 2007). http://bows2.ec-lille.fr (accessed in 2016).

Bravo-Solorio, S., Calderon, F., Li, C.-T., & Nandi, A. K. (2018). Fast fragile watermark embedding and iterative mechanism with high self-restoration performance. *Digital Signal Processing, 73*, 83–92. https://doi.org/10.1016/j.dsp.2017.11.005

Chang, C.-C., Lin, C.-C., & Su, G.-D. (2020). An effective image self-recovery based fragile watermarking using self-adaptive weight-based compressed AMBTC. *Multimedia Tools Applications, 79*, 24795–24824. https://doi.org/10.1007/s11042-020-09132-w

Chen, W.-C., & Wang, M.-S. (2009). A fuzzy c-means clustering-based fragile watermarking scheme for image authentication. *Expert Systems with Applications, 36*(2), 1300–1307. https://doi.org/10.1016/j.eswa.2007.11.018

Chuang, J.-C., & Y.-C. J. J. o. V. C. Hu and I. Representation,. (2011). An adaptive image authentication scheme for vector quantization compressed image. *Journal of Visual Communication and Image Representation, 22*(5), 440–449. https://doi.org/10.1016/j.jvcir.2011.03.011

Dadkhah, S., Abd Manaf, A., Hori, Y., Hassanien, A. E., & Sadeghi, S. (2014). An effective SVD-based image tampering detection and self-recovery using active watermarking. *Signal Processing: Image Communication, 29*(10), 1197–1210. https://doi.org/10.1016/j.image.2014.09.001

Fridrich, J. and M. Goljan (1999). Images with self-correcting capabilities. Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), IEEE 10.1109/ICIP.1999.817228.

Gul, E., & Ozturk, S. (2021). A novel pixel-wise authentication-based self-embedding fragile watermarking method. *Multimedia Systems, 27*(3), 531–545. https://doi.org/10.1007/s00530-021-00751-3

Haghighi, B. B., Taherinia, A. H., & Mohajerzadeh, A. H. (2019). TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA. *Information Scientist, 486*, 204–230. https://doi.org/10.1016/j.ins.2019.02.055

He, H., Chen, F., Tai, H.-M., Kalker, T., & Zhang, J. (2011). Performance analysis of a block-neighborhood-based self-recovery fragile watermarking scheme. *IEEE Transactions on Information Forensics and Security, 7*(1), 185–196. https://doi.org/10.1109/TIFS.2011.2162950

Hong, W., Li, D., Lou, D.-C., Zhou, X., & Chang, C.-H. (2020). A bit toggling approach for AMBTC tamper detection scheme with high image fidelity. *PLoS One1, 15*(4), e0230997.

Hu, Y.-C., Lo, C.-C., Chen, W.-L., & Wen, C.-H. (2013). Joint image coding and image authentication based on absolute moment block truncation coding. *Journal of Electronic Imaging, 22*(1), Article 013012. https://doi.org/10.1117/1.JEI.22.1.013012

Hu, Y.-C., Lo, C.-C., Wu, C.-M., Chen, W.-L., & Wen, C.-H. (2013). Probability-based tamper detection scheme for BTC-compressed images based on quantization levels modification. Retrieved 3, 7, from *International Journal of Security and Its Applications* https://www.earticle.net/Article/A211016.

Huang, R., Liu, H., Liao, X., Sun, S. J. M. T., & Applications,. (2019). A divide-and-conquer fragile self-embedding watermarking with adaptive payload. *Multimedia Tools and Applications, 78*(18), 26701–26727. https://doi.org/10.1007/s11042-019-07802-y

Lee, T.-Y., & Lin, S. D. (2008). Dual watermark for image tamper detection and recovery. *Pattern Recognition, 41*(11), 3497–3506. https://doi.org/10.1016/j.patcog.2008.05.003

Lin, C.-C., Lee, T.-L., Chang, Y.-F., Shiu, P.-F., & Zhang, B. (2023). Fragile Watermarking for Tamper Localization and Self-Recovery Based on AMBTC and VQ. *Electronics, 12*(2), 415. https://doi.org/10.3390/electronics12020415

Lin, P.-Y., Lee, J.-S., & Chang, C.-C. (2009). Dual digital watermarking for internet media based on hybrid strategies. *IEEE Transactions on Circuits and Systems for Video Technology, 19*(8), 1169–1177. https://doi.org/10.1109/TCSVT.2009.2020263

Liu, T., & Yuan, X. (2020). Adaptive Feature Calculation and Diagonal Mapping for Successive Recovery of Tampered Regions. *IEEE Transactions on Circuits and Systems for Video Technology, 31*(7), 2617–2630. https://doi.org/10.1109/TCSVT.2020.3032455

Liu, T., & Yuan, X. (2021). A dual-tamper-detection method for digital image authentication and content self-recovery. *Multimedia Tools Applications, 80*, 29805–29826. https://doi.org/10.1007/s11042-021-11179-2

Lu, C.-S., & Liao, H.-Y.-M. (2000). Structural digital signature for image authentication: An incidental distortion resistant scheme. *IEEE Transactions on Multimedia, 5*(2), 161–173. https://doi.org/10.1109/TMM.2003.811621

Mahmood, T., Mehmood, Z., Shah, M., & Khan, Z. (2018). An efficient forensic technique for exposing region duplication forgery in digital images. *Applied Intelligence, 48*(7), 1791–1801. https://doi.org/10.1007/s10489-017-1038-5

Mittal, A., Soundararajan, R., & Bovik, A. C. (2012). Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters, 20*(3), 209–212. https://doi.org/10.1109/LSP.2012.2227726

Molina-Garcia, J., Garcia-Salgado, B. P., Ponomaryov, V., Reyes-Reyes, R., Sadovnychiy, S., & Cruz-Ramos, C. (2020). An effective fragile watermarking scheme for color image tampering detection and self-recovery. *Signal Processing: Image Communication, 81*, Article 115725. https://doi.org/10.1016/j.image.2019.115725

Puhan, N. B., & Ho, A. T. (2007). Secure authentication watermarking for localization against the Holliman-Memon attack. *Multimedia Systems, 12*(6), 521–532. https://doi.org/10.1007/s00530-006-0068-3

Roy, S. D., Li, X., Shoshan, Y., Fish, A., & Yadid-Pecht, O. (2012). Hardware implementation of a digital watermarking system for video authentication. *IEEE transactions on circuits and systems for video technology, 23*(2), 289–301. https://doi.org/10.1109/TCSVT.2012.2203738

Sarreshtedari, S., & Akhaee, M. A. (2015). A source-channel coding approach to digital image protection and self-recovery. *IEEE Transactions on Image Processing, 24*(7), 2266–2277. https://doi.org/10.1109/TIP.2015.2414878

Shehab, A., Elhoseny, M., Muhammad, K., Sangaiah, A. K., Yang, P., Huang, H., & Hou, G. (2018). Secure and robust fragile watermarking scheme for medical images. *IEEE Access, 6*, 10269–10278. https://doi.org/10.1109/Access.2018.2799240

Sinhal, R., Ansari, I. A., & Ahn, C. W. (2020). Blind image watermarking for localization and restoration of color images. *IEEE Access, 8*, 200157–200169. https://doi.org/10.1109/Access.2020.3035428

Tai, W.-L., & Liao, Z.-J. (2018). Image self-recovery with watermark self-embedding. *Signal Processing: Image Communication, 65*, 11–25. https://doi.org/10.1016/j.image.2018.03.011

Tohidi, F., Paul, M., & Hooshmandasl, M. R. (2021). Detection and Recovery of Higher Tampered Images Using Novel Feature and Compression Strategy. *IEEE Access, 9*, 57510–57528. https://doi.org/10.1109/Access.2021.3072314

Vali, M. H., Aghagolzadeh, A., & Baleghi, Y. (2018). Optimized watermarking technique using self-adaptive differential evolution based on redundant discrete wavelet transform and singular value decomposition. *Expert Systemswith Applications, 114*, 296–312. https://doi.org/10.1016/j.eswa.2018.07.004

Van Schyndel, R. G., A. Z. Tirkel and C. F. Osborne (1994). A digital watermark. Proceedings of 1st international conference on image processing, IEEE 10.1109/ICIP.1994.413536.

Weber, A. G. (2006). The USC-SIPI image database: Version 5. http://sipi. usc. edu/database/.

Yeung, M. M., & Mintzer, F. (1997). An invisible watermarking technique for image verification. *Proceedings of international conference on image processing, IEEE.* https://doi.org/10.1109/ICIP.1997.638587

Zhang, X., & Wang, S. (2007). Statistical fragile watermarking capable of locating individual tampered pixels. *IEEE Signal Processing Letters, 14*(10), 727–730. https://doi.org/10.1109/LSP.2007.896436

Zhang, X., & Wang, S. (2008). Fragile watermarking with error-free restoration capability. *IEEE Transactions on Multimedia, 10*(8), 1490–1499. https://doi.org/10.1109/TMM.2008.2007334