Signal Processing 224 (2024) 109582

Contents lists available at ScienceDirect

Signal Processing

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

Robust reversible watermarking of JPEG images☆

Xingyuan Liang, Shijun Xiang *

The College of Information Science and Technology/ College of Cyber Security, Jinan University, Guangzhou, 510632, China

ARTICLE INFO

Keywords:
Robust reversible watermarking
Discrete cosine transform coefficient
JPEG image
Copyright protection
Integrity authentication

ABSTRACT

Robust reversible watermarking (RRW) is a potential technology for copyright protection and integrity authentication due to its ability in both robustness and reversibility. Currently, there is no detailed report on RRW of JPEG images. In this paper, we propose an RRW algorithm of JPEG images by selecting quantized discrete cosine transform coefficients to construct robust features for watermarking. In the proposed algorithm, a watermark sequence can be embedded into a JPEG image by shifting the histogram of its constructed robust features. For less embedding distortion, smaller file size, and higher structural similarity, an evaluation method has been designed to select those appropriate frequency bands for watermarking. On the receiver side, the watermark can be accurately extracted by reconstructing the robust features, and the original JPEG image can be restored losslessly by performing the inverse operation of the histogram shifting. Experimental results show that the watermark is robust to common image processing operations (e.g., JPEG recompression, JPEG2000 compression, WebP compression, and additive white Gaussian noise), and can effectively resist those attacks from the lossy channels in real life, such as the most popular apps Instagram and WeChat.

1. Introduction

With the vigorous development of network and computer technologies, multimedia communication has become more convenient and extensive. Since digital multimedia can easily be maliciously copied, forged, and tampered with during transmission, many problems concerning the information security of digital multimedia have arisen, especially the copyright protection and integrity authentication of digital multimedia. Reversible watermarking (RW) technology can effectively address these problems by using the redundancy in the digital carrier to embed a watermark, and the receiver can extract the watermark and restore the original digital carrier losslessly [1]. This technology can be used for content identification, integrity certification, and copyright protection of digital multimedia, and has been widely used in fields with high requirements for the confidentiality, security, and fidelity of digital multimedia, such as legal certification, military images, remote sensing images, medical images, etc. [2]. RW technology has been developed rapidly in recent years, and the existing RW algorithms can be separated into four categories: lossless compression [3,4], difference expansion [5–7], histogram shifting [8–12], and prediction error expansion [13–18].

The existing RW technologies are mainly designed by using uncompressed images as carriers, while research on compressed images as carriers is in its infancy. As a compressed image format, JPEG has become the most widely used image format in our lives due to its excellent compression performance. Most of the images generated by the camera or scanner are saved as JPEG images. Thus, watermarking of JPEG images has more important practical value. A few existing RW algorithms of JPEG images can be mainly divided into the following three categories by modifying: the quantization table [19,20], the Huffman table [21–24], and the quantized discrete cosine transform (DCT) coefficients [25–32]. For the first category, Fridrich et al. [19] first proposed an RW algorithm of JPEG images based on modifying the quantization table. The algorithm divides some elements in the quantization table by 2 and the corresponding quantized DCT coefficient is multiplied by 2 to generate the embedding space. In [20], Wang et al. improved the algorithm proposed in [19] by extending the modification of the quantization table to k times. The works in [19,20] performed well in visual quality, but caused an excessive increase in file size. For the second category, Mobasseri et al. [21] modified the Huffman table to map the used variable-length code (VLC) to the unused VLC for watermarking. Next, Qian et al. [22] and Hu et al. [23] improved the algorithm proposed in [21] to increase the embedding capacity. These algorithms can maintain the file size well and do not cause any distortion to the carrier image. For the third category, Chang et al. [25] modified the continuous zero-valued coefficients in the middle frequency for watermarking. Then, Xuan et al. [26] and Huang

☆ This document is the results of the research project funded by the National Natural Science Foundation of China (No. 62272197), the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011928), and the National Key Research and Development Program of China (No. 2023YFF0905000).

* Corresponding author.
E-mail address: shijun_xiang@qq.com (S. Xiang).

https://doi.org/10.1016/j.sigpro.2024.109582
Received 14 February 2024; Received in revised form 25 April 2024; Accepted 13 June 2024
Available online 18 June 2024

et al. [27] both embedded the watermark into the JPEG image by shifting the histogram of quantized DCT coefficients. In [27], Huang et al. preferentially shifted the bins of + 1 and −1 in the histogram from smooth blocks to reduce embedding distortion and file extension. Hou et al. [28] established a cost function to select quantized DCT coefficients that cause less embedding distortion for watermarking. In [29], He et al. constructed the negative influence models based on the encoding mechanism of JPEG compression to select quantized DCT coefficients for watermarking. In [30], Yin et al. proposed a multi-objective optimization strategy to realize the balance of the rate–distortion and the file size expansion. Xiao et al. [32] designed a multi-histogram modification framework for RW of JPEG images.

RW technology, including RW technology of JPEG images, is sensitive to any attack, meaning that when the watermarked carrier is attacked by an additive noise or a common signal processing operation, the embedded watermark will be lost. In practical applications, the watermarked carrier may suffer some common signal processing operations. For example, when an image is transmitted in a public channel, the image may be compressed to improve the transmission efficiency, e.g., the apps WeChat and Instagram will perform a JPEG compression on the image, and Google will perform a WebP compression on the image. To effectively extract the watermark from the attacked carrier, a desired RW technology with some extent of robustness is required, which is named as robust reversible watermarking (RRW) technology. With the RRW technology, the receiver can correctly extract the watermark from the intact watermarked carrier and restore the original carrier losslessly. Most importantly, the watermark can still be correctly extracted from the distorted watermarked carrier that has been attacked to a certain degree. Notice that, RRW technology is different from robust watermarking technology. In robust watermarking algorithms, such as [33–37], the operation performed on the carrier for watermarking is irreversible, and it is difficult to compress the modifications on the carrier, which leads to insufficient redundancy to achieve reversibility, resulting in the original carrier being unable to be losslessly restored and the inevitable permanent loss on the original carrier. In fields where high fidelity is required for images, such as medical image diagnosis, judicial image transmission, high-definition photographic work publishing, and micro image processing, data loss caused by watermarking is expected to be avoided. The RRW has both robustness and reversibility. The robustness is useful for copyright protection, while the reversibility is valuable for integrity authentication and recovery of the carrier in case of no attacks. Thus, RRW technology is more suitable for copyright protection and integrity authentication in the above fields.

The existing RRW algorithms can be divided into the following two categories: histogram shifting-based methods [38–47] and two-stage embedding-based methods [48–52]. For the first category, Vleeschouwer et al. [38] first proposed an RRW algorithm based on the circular interpretation of bijective transformations. In this algorithm, the watermark is embedded into the uncompressed image by cyclically shifting pixels, and the embedded watermark is robust to JPEG compression. Next, Ni et al. [39] proposed an RRW framework based on histogram shifting. This paper embeds the watermark into the uncompressed image by shifting the histogram of arithmetic statistics calculated by pixels, which can resist JPEG compression and improve the visual quality of the watermarked image, but may embed error bits. Then, based on the RRW framework proposed in [39] and the 5/3 integer wavelet transform, Zou et al. [41] proposed an RRW algorithm of uncompressed images. This algorithm can achieve robustness against JPEG2000 compression and can be integrated into the JPEG2000 standard. In [42], Gao et al. embedded the watermark into the uncompressed image by using more robust statistics and setting embedding conditions to improve robustness and avoid embedding error bits. In [43], Zeng et al. proposed a new embedding strategy and introduced two thresholds to control the robustness. Next, for better robustness and imperceptibility of the watermark, An et al. [44] and Thabit et al. [45] adopted integer wavelet transform and Slantlet transform, respectively. For the second category, Coltuc et al. [48] proposed an RRW framework based on two-stage embedding. In the first stage, the robust watermark is embedded into the carrier, and then in the second stage, the distortion caused in the first stage is embedded into the carrier by using an RW algorithm for reversibility. Wang et al. [50] improved the algorithm proposed in [48] by using independent embedding domains for the two-stage embedding. In [51], Hu et al. proposed an RRW algorithm based on Zernike moments to achieve robustness against geometric attacks.

Most of the existing RRW technologies have been designed by using uncompressed images as carriers, but there is no detailed report on the RRW of JPEG images. The RRW of JPEG images refers to the RRW technology designed by using JPEG images as carriers. Based on the work on RW technology of JPEG images, it can be concluded that the file structure of JPEG images is different from that of uncompressed images. When the existing RRW algorithm for uncompressed images is applied to JPEG images, the JPEG image must first be decoded into a decoded image composed of pixels, and then the watermark is embedded into the decoded image, and the embedded watermark is extracted from the watermarked decoded image on the receiver side. But when the watermarked decoded image or even the restored decoded image is re-saved as a JPEG image, the image will be distorted by the quantization, rounding, and truncation operations in JPEG compression, resulting in the loss of the embedded watermark and the inability to restore the original JPEG image. Furthermore, the redundancy in JPEG images is less than that in uncompressed images, and the prior knowledge about embedding distortion derived from uncompressed images cannot be directly applied to JPEG images. Moreover, in JPEG images, the modification of quantized DCT coefficients may aggravate the blocking artifacts. Besides, the file size of JPEG images is often increased due to the embedded watermark, and an excessive increase in file size results in poor imperceptibility of the embedded watermark. Thus, the RRW algorithm of JPEG images must consider the file size expansion that is not considered in the RRW algorithm of uncompressed images. Therefore, the existing RRW algorithms of uncompressed images cannot be directly applied to the RRW of JPEG images, and it is more difficult to design an RRW algorithm of JPEG images than to design an RRW algorithm of uncompressed images. Considering that JPEG images have become the most commonly used images, it is necessary to research the RRW of JPEG images for copyright protection and integrity authentication of JPEG images.

Based on the above reasons, a new RRW algorithm of JPEG images is proposed in this paper. First, we utilize quantized DCT coefficients to construct robust features. Then, we propose a watermark embedding strategy based on histogram shifting to achieve the robustness of the watermark and the reversibility of the algorithm. Next, we analyze the effects of modifying the quantized DCT coefficients on JPEG images, including embedding distortion, file size expansion, and structural similarity, and design an evaluation method to select the appropriate frequency bands for watermark embedding. With the selected frequency bands, the watermark bits can be embedded into the JPEG image by shifting the histogram of the robust features. The experimental results show that the proposed algorithm has satisfactory performance in visual quality and file size expansion, and has strong robustness against common image processing operations, such as JPEG recompression, JPEG2000 compression, WebP compression, and additive white Gaussian noise (AWGN). The main contributions of our work can be summarized as follows:

- In the literature, there is no detailed report on the RRW of JPEG images. In this paper, we propose an RRW algorithm of JPEG images. And we design a method to construct robust features in JPEG images, and propose a watermark embedding strategy to achieve the robustness of the watermark and the reversibility of the algorithm;

- In order to better hide bits in JPEG images, we design an evaluation method to select the appropriate frequency bands for watermarking by considering robustness, embedding distortion, file size expansion, and structural similarity;
- The proposed algorithm can restore the original JPEG image losslessly and is robust to common image processing operations. It is suitable for the protection of JPEG images.

The remainder of this paper is organized as follows. Preliminary knowledge of the proposed algorithm is given in Section 2. Then, the detail of the proposed RRW algorithm of JPEG images is elaborated in Section 3. Next, how to select the appropriate frequency bands for watermarking is introduced in Section 4. Experimental results and analysis are given in Section 5. Finally, we conclude in Section 6.

2. Preliminary

In this section, the preliminary knowledge of the proposed RRW algorithm of JPEG images is introduced. First, the JPEG compression technique is reviewed. Then, the RRW framework based on histogram shifting is introduced.

2.1. Review of JPEG compression

In the encoding process of JPEG compression, first, an $M_1 \times N_1$ sized uncompressed image I is divided into $M_2 \times N_2$ non-overlapping blocks of 8×8 pixels (we only consider that $M_1 = 8 \times M_2$ and $N_1 = 8 \times N_2$). The (i, j)th block in I is denoted as $P_{i,j} = \{p_{i,j}(a, b) | a, b = 0, 1, \dots, 7\}$, where $i \in \{1, 2, \dots, M_2\}$, $j \in \{1, 2, \dots, N_2\}$, and $p_{i,j}(a, b)$ is the pixel value in $P_{i,j}$. Then, the two-dimensional DCT is performed on each block, and the resulting transform block corresponding to $P_{i,j}$ is denoted as $S_{i,j} = \{s_{i,j}(u, v) | u, v = 0, 1, \dots, 7\}$, where $s_{i,j}(u, v)$ is the DCT coefficient in $S_{i,j}$ and can be calculated by

$$s_{i,j}(u, v) = \sum_{a+b=0}^{14} \zeta_{u,v}^{a,b} p_{i,j}(a, b), \tag{1}$$

where $\zeta_{u,v}^{a,b}$ is the DCT basis and can be calculated by

$$\zeta_{u,v}^{a,b} = \frac{1}{4} C(u) C(v) cos \frac{(2a+1)u\pi}{16} cos \frac{(2b+1)v\pi}{16}. \tag{2}$$

Here, $C(0) = 1/\sqrt{2}$ and $C(u), C(v) = 1$ for $u, v > 0$. And $P_{i,j}$ can be obtained by performing inverse DCT (IDCT) on $S_{i,j}$, then $p_{i,j}(a, b)$ can be calculated by

$$p_{i,j}(a, b) = \sum_{u+v=0}^{14} \zeta_{u,v}^{a,b} s_{i,j}(u, v). \tag{3}$$

Since the DCT coefficients in each transform block are arranged in a zigzag scanning order after being quantized, for simplicity, $S_{i,j}$ can be denoted as:

$$\begin{aligned} S_{i,j} &= \begin{bmatrix} s_{i,j}(0,0) & s_{i,j}(0,1) & \cdots & s_{i,j}(0,7) \\ s_{i,j}(1,0) & \ddots & \ddots & s_{i,j}(1,7) \\ \vdots & \ddots & \ddots & \vdots \\ s_{i,j}(7,0) & s_{i,j}(7,1) & \cdots & s_{i,j}(7,7) \end{bmatrix} \\ &= \begin{bmatrix} s_{i,j}(1) & s_{i,j}(2) & \cdots & s_{i,j}(29) \\ s_{i,j}(3) & \ddots & \ddots & s_{i,j}(43) \\ \vdots & \ddots & \ddots & \vdots \\ s_{i,j}(36) & s_{i,j}(37) & \cdots & s_{i,j}(64) \end{bmatrix}. \end{aligned} \tag{4}$$

In this way, we have $S_{i,j} = \{s_{i,j}(f) | f = 1, 2, \dots, 64\}$, where $s_{i,j}(f)$ is the DCT coefficient in the frequency band f in $S_{i,j}$. Next, with the quantization table $Q = \{q(f) | f = 1, 2, \dots, 64\}$, $S_{i,j}$ is quantized to obtain the quantized block $D_{i,j} = \{d_{i,j}(f) | f = 1, 2, \dots, 64\}$. Here, $d_{i,j}(f)$ is the quantized DCT coefficient in the frequency band f in $D_{i,j}$ and can be computed by $d_{i,j}(f) = \lfloor s_{i,j}(f)/q(f) \rceil$, where $\lfloor \cdot \rceil$ is the rounding function and $q(f)$ is the quantization step in Q. Finally, all quantized DCT coefficients are encoded into a bitstream, and after prepending the header, a JPEG image can be obtained.

In the decoding process, first, the quantized DCT coefficients are obtained from the JPEG image by decoding the bitstream, and these quantized DCT coefficients are divided into $M_2 \times N_2$ non-overlapping quantized blocks of size 8×8. Then for each quantized block $D_{i,j}$, each coefficient $d_{i,j}(f)$ in $D_{i,j}$ is dequantized by $s_{i,j}^*(f) = d_{i,j}(f) \times q(f)$ to obtain a dequantized block $S_{i,j}^* = \{s_{i,j}^*(f) | f = 1, 2, \dots, 64\}$. According to Eq. (4), the elements in $S_{i,j}^*$ are reordered to $\{s_{i,j}^*(u, v) | u, v = 0, 1, \dots, 7\}$. Next, IDCT is performed on $S_{i,j}^*$ to obtain a decoded block $P_{i,j}^* = \{p_{i,j}^*(a, b) | a, b = 0, 1, \dots, 7\}$, where $p_{i,j}^*(a, b)$ is the decoded pixel. Finally, all decoded blocks are combined to obtain the decoded image I^*.

2.2. RRW framework based on histogram shifting

In [39], Ni et al. proposed an RRW framework based on histogram shifting. In this framework, to embed a watermark sequence $W = \{w(k) | k = 1, 2, \dots, L\}$ $(w(k) \in \{0, 1\})$ into an image I, the image I is first divided into L blocks, and then a robust feature is constructed in each block. The kth block in the image I is denoted as P_k, and the robust feature of P_k is denoted as $\lambda(k)$. Next, a histogram h of the robust features $\Lambda = \{\lambda(k) | k = 1, 2, \dots, L\}$ is constructed by

$$h(\theta) = \#\{k = 1, 2, \dots, L : \lambda(k) = \theta,\ \theta \in \Lambda\}, \tag{5}$$

where $h(\theta)$ counts the number of robust features with a value θ. Then, a threshold T that satisfies $T > max(|\Lambda|)$ is selected, where $max(|\Lambda|)$ is the maximum absolute value of the robust features in Λ. Finally, the watermark bit $w(k)$ can be embedded by shifting $\lambda(k)$. The operation of shifting $\lambda(k)$ is formulated as

$$\widetilde{\lambda}(k) = \begin{cases} \lambda(k) + T > 0, & if\ w(k) = 1, \\ \lambda(k) - T < 0, & if\ w(k) = 0, \end{cases} \tag{6}$$

where $\widetilde{\lambda}(k)$ is the watermarked version of $\lambda(k)$. The operation of shifting $\lambda(k)$ must be reversible.

According to Eq. (6), the ranges $[0, +\infty)$ and $(-\infty, 0)$ are defined as bit-1-region and bit-0-region, respectively. Fig. 1 shows the histogram $\widetilde{h}$ of the watermarked robust features $\{\widetilde{\lambda}(k) | k = 1, 2, \dots, L\}$, where the red bins are composed of the watermarked robust features $\widetilde{\lambda}(k)$ corresponding to $w(k) = 0$ and the blue bins are composed of those $\widetilde{\lambda}(k)$ corresponding to $w(k) = 1$. Note that in the histograms shown in this paper, the horizontal axis represents the value θ of the robust feature, and the vertical axis represents the number of robust features with a value θ. Accordingly, the watermark bit $w(k)$ can be extracted by

$$w(k) = \begin{cases} 1, & if\ \widetilde{\lambda}(k) \geq 0, \\ 0, & if\ \widetilde{\lambda}(k) < 0. \end{cases} \tag{7}$$

Then, the original image I can be restored losslessly by using the inverse operation of shifting $\lambda(k)$.

As shown in Fig. 1, the bins in the bit-0-region and the bins in the bit-1-region are separated by a robust region $G = g_1 + g_2$, where $g_1, g_2 \geq T - max(|\Lambda|)$. Therefore, if the watermarked robust feature $\widetilde{\lambda}(k)$ is not distorted to the wrong region due to the image processing operations, $w(k)$ can still be extracted correctly by Eq. (7). The larger T is set, the larger G can be obtained, the stronger the robustness will be, but the greater the embedding distortion will be, resulting in the decrease of imperceptibility. Thus, it is necessary to set an appropriate value for T to strike a balance between imperceptibility and robustness.

Most of the existing RRW algorithms are designed by using uncompressed images as carriers. As mentioned earlier, the file structure of JPEG images is different from that of uncompressed images, and the redundancy in JPEG images is less than that in uncompressed images. Moreover, the RRW algorithm of JPEG images must consider file size expansion and blocking artifacts. Therefore, the existing RRW algorithms of uncompressed images cannot be directly applied to the RRW of JPEG images. And it is necessary to design a new RRW algorithm of JPEG images.

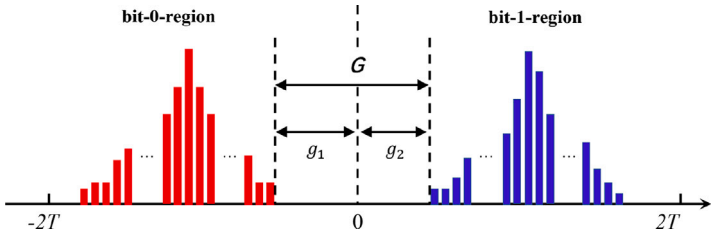


Fig. 1. The histogram of the watermarked robust features.

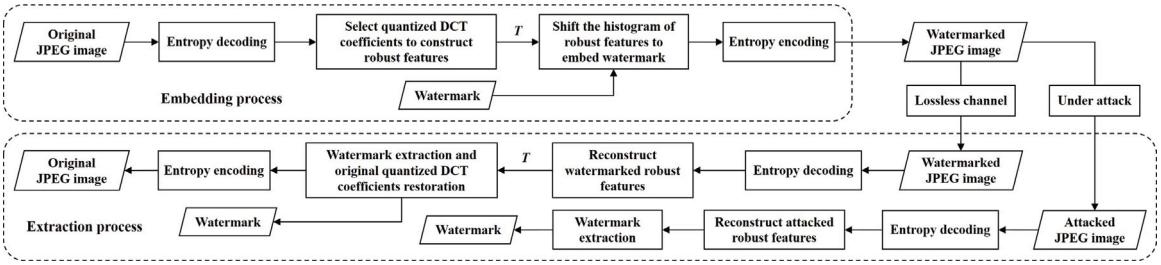


Fig. 2. Sketch of the proposed robust reversible watermarking algorithm of JPEG images.

3. Proposed algorithm

In this section, we introduce the proposed RRW algorithm of JPEG images in detail. Fig. 2 shows the sketch of the proposed algorithm.

In the embedding process, a JPEG image is first entropy-decoded to obtain the quantized DCT coefficients, where these quantized DCT coefficients are divided into $M \times N$ non-overlapping quantized blocks of size 8×8. Then, some quantized DCT coefficients are selected from each quantized block to construct robust features. Next, with a threshold T, the watermark can be embedded into the JPEG image by shifting the histogram of the robust features. Finally, the modified quantized DCT coefficients are entropy-encoded to obtain a watermarked JPEG image. The specific watermark embedding strategy is introduced in Section 3.1.

In the extraction process, the watermarked JPEG image is first entropy-decoded to obtain the modified quantized DCT coefficients. Secondly, the watermarked robust features are reconstructed by the same method as the robust features are constructed in the embedding process. Then, the watermark can be extracted based on the watermarked robust features, and with the threshold T, the quantized DCT coefficients can be restored by applying the inverse operation of histogram shifting. Finally, the restored quantized DCT coefficients are entropy-encoded to recover the original JPEG image for integrity verification. If the watermarked JPEG image is intact, the original JPEG image can be recovered losslessly. The details of how to extract the watermark from the intact watermarked image and recover the original JPEG image are described in Section 3.2. In addition, if the watermarked JPEG image is attacked by some image processing operations (such as JPEG recompression, format conversion, AWGN, and filtering), the watermark can still be effectively extracted based on the attacked robust features for copyright protection. The method of extracting the watermark under attacks is introduced in Section 3.3.

3.1. Watermark embedding strategy

According to the RRW framework based on histogram shifting, we design a watermark embedding strategy for RRW of JPEG images. The watermark embedding strategy includes two phases: constructing robust features and shifting the histogram of robust features.

3.1.1. Constructing robust features

The works reported in [42,47] have shown that the difference statistic has strong robustness to some common signal processing operations (such as lossy compression, filtering, AWGN, etc.), and is suitable for constructing robust features. In [42], Gao et al. proposed a method for calculating the difference statistic in uncompressed images, which is described below. For an image block P_k, all pixels in P_k are divided equally into two sets A and $\bar{A}$ according to a random mapping, then the difference statistic $\eta(k)$ of P_k can be calculated by

$$\eta(k) = \sum_{i=1}^{mn/2} (a_i - b_i), \tag{8}$$

where $a_i \in A$ and $b_i \in \bar{A}$. Due to the correlation between pixels, part of the noise introduced by the attack can be offset by calculating the difference between pixels to achieve robustness. Moreover, by calculating the sum of differences (i.e., difference statistic), more pixels can be utilized to improve robustness and reduce embedding distortion on each pixel.

The file structure of JPEG images is different from that of uncompressed images, and the redundancy in JPEG images is less than that in uncompressed images. And for the RRW algorithm of JPEG images, embedding distortion, file size expansion, structural similarity, and robustness must be considered in the construction of robust features. Thus, the method of calculating the difference statistic in uncompressed images cannot be directly applied to constructing robust features in JPEG images. Since there is redundancy in the quantized DCT coefficients in JPEG images and the quantized DCT coefficients possess a certain degree of robustness to common image processing operations, we design a method to construct robust features in JPEG images by utilizing quantized DCT coefficients. The designed method is described below.

First, the original JPEG image Y is entropy-decoded to obtain the quantized blocks $\mathbf{D} = \{D_{i,j} | i = 1, 2, \dots, M,\ j = 1, 2, \dots, N\}$. Secondly, considering that the frequency bands $\Omega_L = \{1, 2, \dots, 36\}$ are more robust to image processing operations than the high frequency bands $\Omega_H = \{37, 38, \dots, 64\}$, we select R frequency bands as $\{\sigma_r | \sigma_r \in \Omega_L,\ r = 1, 2, \dots, R\}$, where σ_r is the selected frequency band, R is the number of selected frequency bands, $R \in \{1, 2, \dots, 36\}$, $r = 1$ if $R = 1$, and $\sigma_a \neq \sigma_b$ if $a \neq b$. And then, we select the quantized DCT coefficients $\{d_{i,j}(\sigma_r) | r = 1, 2, \dots, R\}$ from each quantized block $D_{i,j}$ to construct robust features. The detail on how to select the appropriate frequency bands $\{\sigma_r | r = 1, 2, \dots, R\}$ for watermark embedding is described in

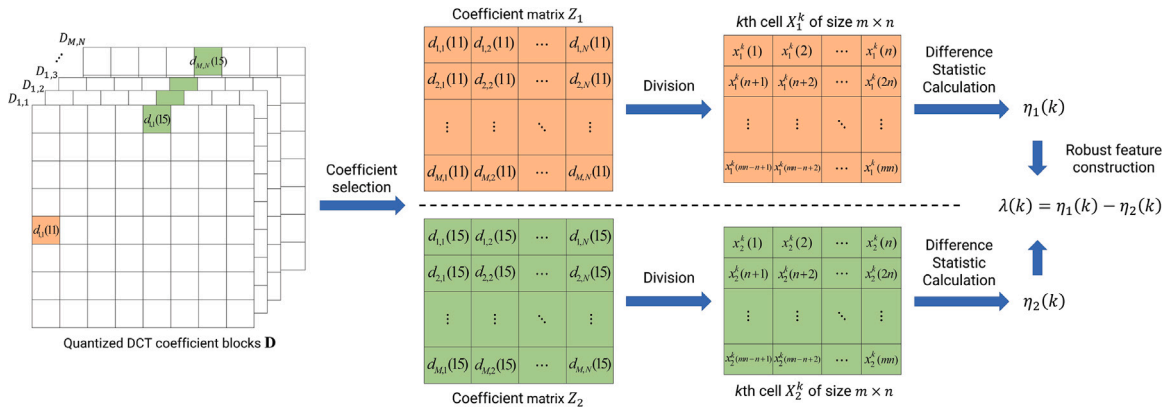


Fig. 3. Process of constructing the robust feature $\lambda(k)$ when $R = 2$ and $\{\sigma_1, \sigma_2\} = \{11, 15\}$.

Section 4. Next, in order to utilize the correlation among the inter-block quantized DCT coefficients in the same frequency band, we divide the selected quantized DCT coefficients into R coefficient matrices $\{Z_r | r = 1, 2, \dots, R\}$ according to the frequency bands $\{\sigma_r | r = 1, 2, \dots, R\}$, where the coefficient matrix $Z_r = \{z_r(i, j) | i = 1, 2, \dots, M,\ j = 1, 2, \dots, N\}$ for each $r \in \{1, 2, \dots, R\}$, and the element $z_r(i, j)$ in Z_r is defined by

$$z_r(i, j) = d_{i,j}(\sigma_r). \tag{9}$$

Then, for each $r \in \{1, 2, \dots, R\}$, the coefficient matrix Z_r is divided into L non-overlapping cells of size $m \times n$. Here, L is the embedding capacity calculated by $L = \lfloor M/m \rfloor \times \lfloor N/n \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. The kth cell in Z_r is denoted as $X_r^k = \{x_r^k(\varepsilon) | \varepsilon = 1, 2, \dots, mn\}$. Next, we define a random integer bijection as $\varphi_\varepsilon : \{\varphi_1, \varphi_2, \dots, \varphi_{mn}\} \Leftrightarrow \{1, 2, \dots, mn\}$, which is used as the private key K_s. And then, with K_s, the difference statistic $\eta_r(k)$ of the cell X_r^k in the frequency band σ_r can be calculated by

$$\eta_r(k) = \sum_{\varepsilon=1}^{mn} (-1)^{\varepsilon-1} x_r^k(\varphi_\varepsilon). \tag{10}$$

Finally, considering the correlation among the intra-block quantized DCT coefficients in different frequency bands, the robust feature $\lambda(k)$ of the group $\{X_r^k | r = 1, 2, \dots, R\}$ can be constructed by

$$\lambda(k) = \sum_{r=1}^{R} (-1)^{r-1} \eta_r(k). \tag{11}$$

Fig. 3 shows the process of constructing the robust feature $\lambda(k)$ when $R = 2$ and $\{\sigma_1, \sigma_2\} = \{11, 15\}$. By using the designed method, the robust features $\Lambda = \{\lambda(k) | k = 1, 2, \dots, L\}$ can be constructed in JPEG images.

3.1.2. Shifting the histogram of robust features

The histogram h of the robust features $\Lambda = \{\lambda(k) | k = 1, 2, \dots, L\}$ can be constructed by Eq. (5). A watermark sequence $W = \{w(k) | k = 1, 2, \dots, L\}$ $(w(k) \in \{0, 1\})$ can be embedded into the JPEG image Y by shifting the histogram h of the robust features Λ.

First, A positive integer is selected as the threshold T for watermark embedding and controlling robustness. The threshold T must satisfy $T > max(|\Lambda|)$ to achieve reversibility. Next, for each $k \in \{1, 2, \dots, L\}$, since the robust feature $\lambda(k)$ is constructed from the elements in the group $\{X_r^k | r = 1, 2, \dots, R\}$, by using T and K_s, a watermark bit $w(k)$ can be embedded by performing an integer transformation on the elements $\{x_r^k(\varepsilon) | r = 1, 2, \dots, R,\ \varepsilon = 1, 2, \dots, mn\}$. The integer transformation is formulated as

$$\widetilde{x}_r^k(\varphi_\varepsilon) = \begin{cases} x_r^k(\varphi_\varepsilon) + (2 \times w(k) - 1) \left\lfloor \dfrac{t_r + \varepsilon - 1}{m \times n} \right\rfloor, & if\ mod(\varepsilon, 2) = 1, \\ x_r^k(\varphi_\varepsilon) - (2 \times w(k) - 1) \left\lfloor \dfrac{t_r + \varepsilon - 1}{m \times n} \right\rfloor, & else, \end{cases} \tag{12}$$

and

$$t_r = (-1)^{r-1} \left\lfloor \frac{T + r - 1}{R} \right\rfloor, \tag{13}$$

where $\lfloor \cdot \rfloor$ is the floor function, and $\widetilde{x}_r^k(\varphi_\varepsilon)$ is the watermarked version of $x_r^k(\varphi_\varepsilon)$. The watermarked cell corresponding to X_r^k is denoted as $\widetilde{X}_r^k = \{\widetilde{x}_r^k(\varepsilon) | \varepsilon = 1, 2, \dots, mn\}$, and the difference statistic of $\widetilde{X}_r^k$ is denoted as $\widetilde{\eta}_r(k)$. With Eq. (12), the robust feature $\lambda(k)$ is shifted to $\widetilde{\lambda}(k)$, where $\widetilde{\lambda}(k)$ is the watermarked robust feature of the group $\{\widetilde{X}_r^k | r = 1, 2, \dots, R\}$. According to Eqs. (10), (11), (12), and (13), the watermarked robust feature $\widetilde{\lambda}(k)$ can be calculated by

$$\begin{aligned} \widetilde{\lambda}(k) &= \sum_{r=1}^{R} (-1)^{r-1} \widetilde{\eta}_r(k) \\ &= (2 \times w(k) - 1) \times T + \sum_{r=1}^{R} (-1)^{r-1} \eta_r(k) \\ &= \begin{cases} \lambda(k) + T > 0, & if\ w(k) = 1, \\ \lambda(k) - T < 0, & if\ w(k) = 0. \end{cases} \end{aligned} \tag{14}$$

According to Eq. (14), it can be seen that if $w(k) = 1$, $\lambda(k)$ is shifted into the range $(0, 2T)$, and if $w(k) = 0$, $\lambda(k)$ is shifted into the range $(-2T, 0)$. Thus, the ranges $[0, +\infty)$ and $(-\infty, 0)$ are defined as bit-1-region and bit-0-region, respectively.

For example, when we set $m = 8$, $n = 4$, $R = 1$, and $\{\sigma_1\} = \{11\}$, the histogram h of the robust features $\Lambda = \{\lambda(k) | k = 1, 2, \dots, 128\}$ of the 512×512 sized image *Lena* with a quality factor (QF) of 70 is shown in Fig. 4(a). Then, since $max(|\Lambda|) = 11$, we set $T = 50$. After a 128-bit watermark sequence is embedded into the image *Lena*, the histogram $\widetilde{h}$ of the watermarked robust features $\{\widetilde{\lambda}(k) | k = 1, 2, \dots, 128\}$ is shown in Fig. 4(b), where the red bins are composed of the watermarked robust features $\widetilde{\lambda}(k)$ corresponding to $w(k) = 0$ and the blue bins are composed of those $\widetilde{\lambda}(k)$ corresponding to $w(k) = 1$.

After W is embedded, the watermarked cells $\{\widetilde{X}_r^k | r = 1, 2, \dots, R,\ k = 1, 2, \dots, L\}$ can be obtained. Then, for each $r \in \{1, 2, \dots, R\}$ and $k \in \{1, 2, \dots, L\}$, X_r^k in Z_r is replaced with $\widetilde{X}_r^k$ to obtain the modified coefficient matrices $\{\widetilde{Z}_r | r = 1, 2, \dots, R\}$, where $\widetilde{Z}_r = \{\widetilde{z}_r(i, j) | i = 1, 2, \dots, M,\ j = 1, 2, \dots, N\}$ for each $r \in \{1, 2, \dots, R\}$. Next, for each $i \in \{1, 2, \dots, M\}$ and $j \in \{1, 2, \dots, N\}$, each modified quantized DCT coefficient $\widetilde{d}_{i,j}(f)$ in the modified quantized block $\widetilde{D}_{i,j} = \{\widetilde{d}_{i,j}(f) | f = 1, 2, \dots, 64\}$ can be obtained by

$$\widetilde{d}_{i,j}(f) = \begin{cases} \widetilde{z}_r(i, j), & if\ f = \sigma_r, \\ d_{i,j}(f), & if\ f \notin \{\sigma_1, \sigma_2, \dots, \sigma_R\}, \end{cases} \tag{15}$$

where σ_r is the selected frequency band and $r = 1, 2, \dots, R$.

Finally, all modified quantized blocks $\widetilde{\mathbf{D}} = \{\widetilde{D}_{i,j} | i = 1, 2, \dots, M,\ j = 1, 2, \dots, N\}$ are entropy-encoded to obtain the watermarked JPEG image $\widetilde{Y}$, and the threshold T is reversibly embedded into the header of $\widetilde{Y}$.

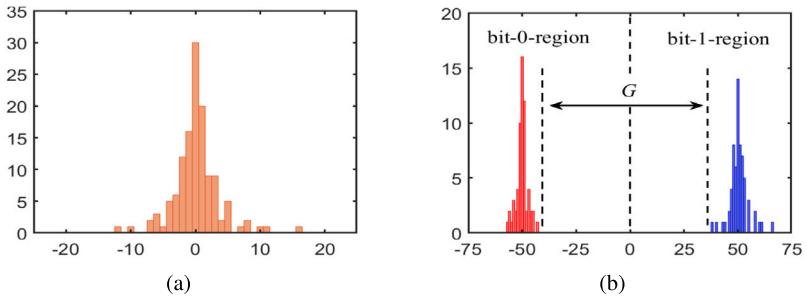


Fig. 4. The histograms of the robust features of the image *Lena*: (a) histogram before watermarking, (b) histogram after watermarking.

3.2. Watermark extraction and JPEG image recovery

First, the threshold T is extracted from the header of the watermarked JPEG image $\widetilde{Y}$, and $\widetilde{Y}$ is entropy-decoded to obtain the modified quantized blocks $\widetilde{\mathbf{D}}$. Secondly, according to the same frequency bands $\{\sigma_r | r = 1,2,\dots,R\}$, we select the modified quantized DCT coefficients $\{\widetilde{d}_{i,j}(\sigma_r) | i = 1,2,\dots,M,\ j = 1,2,\dots,N,\ r = 1,2,\dots,R\}$. Thirdly, by using the method designed in Section 3.1 to construct robust features in JPEG images, we can obtain the modified coefficient matrices $\{\widetilde{Z}_r | r = 1,2,\dots,R\}$. Then, with the private key K_s, the watermarked robust features $\{\widetilde{\lambda}(k) | k = 1,2,\dots,L\}$ can be reconstructed from $\{\widetilde{Z}_r | r = 1,2,\dots,R\}$ by using Eqs. (10) and (11). Next, for each $k \in \{1,2,\dots,L\}$, the watermark bit $w(k)$ can be extracted by

$$w(k) = \begin{cases} 1, & if\ \widetilde{\lambda}(k) \geq 0, \\ 0, & if\ \widetilde{\lambda}(k) < 0. \end{cases} \tag{16}$$

After that, if the watermarked JPEG image $\widetilde{Y}$ is intact, for each $k \in \{1,2,\dots,L\}$, by using T and K_s, the original elements $\{x_r^k(\varepsilon) | r = 1,2,\dots,R,\ \varepsilon = 1,2,\dots,mn\}$ in the group $\{X_r^k | r = 1,2,\dots,R\}$ can be restored by

$$x_r^k(\varphi_\varepsilon) = \begin{cases} \widetilde{x}_r^k(\varphi_\varepsilon) - (2 \times w(k) - 1) \left\lfloor \dfrac{t_r + \varepsilon - 1}{m \times n} \right\rfloor, & if\ mod(\varepsilon, 2) = 1, \\ \widetilde{x}_r^k(\varphi_\varepsilon) + (2 \times w(k) - 1) \left\lfloor \dfrac{t_r + \varepsilon - 1}{m \times n} \right\rfloor, & else, \end{cases} \tag{17}$$

where t_r can be computed by Eq. (13). Next, the original matrices $\{Z_r | r = 1,2,\dots,R\}$ can be restored. And then, for each $i \in \{1,2,\dots,M\}$, $j \in \{1,2,\dots,N\}$ and $f \in \{1,2,\dots,64\}$, the original quantized DCT coefficient $d_{i,j}(f)$ can be restored by

$$d_{i,j}(f) = \begin{cases} z_r(i,j), & if\ f = \sigma_r, \\ \widetilde{d}_{i,j}(f), & if\ f \notin \{\sigma_1, \sigma_2, \dots, \sigma_R\}, \end{cases} \tag{18}$$

where σ_r is the selected frequency band and $r = 1,2,\dots,R$.

Finally, the original quantized DCT coefficients are entropy-encoded to obtain the original JPEG image Y.

3.3. Watermark extraction under attacks

When JPEG images are transmitted over the network, JPEG images may be attacked by some common image processing operations, including lossy compression, filtering, AWGN, etc. After being attacked, the watermarked JPEG image $\widetilde{Y}$ is changed to the JPEG image $\widehat{Y}$. Although the original JPEG image cannot be restored under an attack, the embedded watermark can still be effectively extracted. To verify the effectiveness of the proposed watermark embedding strategy, we analyze the impact of attacks on the constructed robust features. After the watermarked JPEG image $\widetilde{Y}$ is attacked, the modified quantized DCT coefficient $\widetilde{d}_{i,j}(f)$ is distorted to the attacked quantized DCT coefficient $\widehat{d}_{i,j}(f)$. According to the designed method of constructing robust features, we assume that the modified quantized DCT coefficient $\widetilde{d}_{i,j}(f)$ corresponds to the watermarked element $\widetilde{x}_r^k(\varepsilon)$. Due to the attack, the watermarked element $\widetilde{x}_r^k(\varepsilon)$ is distorted into $\widehat{x}_r^k(\varepsilon) = \widetilde{x}_r^k(\varepsilon) + \delta_r^k(\varepsilon)$, where $\widehat{x}_r^k(\varepsilon)$ corresponds to $\widehat{d}_{i,j}(f)$, and $\delta_r^k(\varepsilon)$ is the noise imposed on $\widetilde{d}_{i,j}(f)$ by the attack. The attacked version of the watermarked robust feature $\widetilde{\lambda}(k)$ is denoted as $\widehat{\lambda}(k)$, and the attacked version of the difference statistic $\widetilde{\eta}_r(k)$ is denoted as $\widehat{\eta}_r(k)$. According to Eqs. (9), (10), (11), (12), (13), and (14), we can derive that

$$\begin{aligned} \widehat{\lambda}(k) &= \sum_{r=1}^{R} (-1)^{r-1} \widehat{\eta}_r(k) \\ &= \left(\sum_{r=1}^{R} (-1)^{r-1} \widetilde{\eta}_r(k) \right) + \sum_{r=1}^{R} (-1)^{r-1} \left(\sum_{\varepsilon=1}^{mn} (-1)^{\varepsilon-1} \delta_r^k(\varphi_\varepsilon) \right) \\ &= \widetilde{\lambda}(k) + \sum_{r=1}^{R} (-1)^{r-1} \left(\sum_{\varepsilon=1}^{mn} (-1)^{\varepsilon-1} \delta_r^k(\varphi_\varepsilon) \right), \end{aligned} \tag{19}$$

where the index φ_ε of the noise $\delta_r^k(\varphi_\varepsilon)$ is obtained from the private key K_s. The noise imposed by the attack on the watermarked robust feature $\widetilde{\lambda}(k)$ is denoted as $\Delta(k)$. According to Eq. (19), we have

$$\Delta(k) = \sum_{r=1}^{R} (-1)^{r-1} \left(\sum_{\varepsilon=1}^{mn} (-1)^{\varepsilon-1} \delta_r^k(\varphi_\varepsilon) \right). \tag{20}$$

Consequently, according to Eqs. (14), (19), and (20), it can be seen that for each $k \in \{1,2,\dots,L\}$, if the threshold T satisfies:

$$\begin{cases} T > -\lambda(k) - \Delta(k), & if\ w(k) = 1, \\ T > \lambda(k) + \Delta(k), & if\ w(k) = 0, \end{cases} \tag{21}$$

the watermark bit $w(k)$ can still be accurately extracted from the attacked robust feature $\widehat{\lambda}(k)$.

We take the JPEG image *Lena* as an example to further illustrate the effectiveness of the proposed watermark embedding strategy. After the watermarked JPEG image *Lena* obtained in Section 3.1 is attacked by common image processing operations, the histograms of the attacked robust features $\{\widehat{\lambda}(k) | k = 1,2,\dots,128\}$ are shown in Fig. 5, where the red bins are composed of the attacked robust features $\widehat{\lambda}(k)$ corresponding to $\widetilde{\lambda}(k) \in (-\infty, 0)$ and the blue bins are composed of those $\widehat{\lambda}(k)$ corresponding to $\widetilde{\lambda}(k) \in [0, +\infty)$. As shown in Figs. 5 and 4(b), after being attacked, the attacked robust features can still remain in the region where they were before the attack. And due to the robust region G, the watermarked robust feature $\widetilde{\lambda}(k)$ is not modified to the wrong region, then the watermark bit $w(k)$ can still be extracted accurately. Therefore, the watermark has good resistance to common image processing operations (such as JPEG recompression, JPEG2000 compression, AWGN, and Gaussian low-pass filtering).

The watermark extraction process under attacks is similar to the watermark extraction process in Section 3.2. The attacked JPEG image $\widehat{Y}$ is first entropy-decoded. Then, according to the same frequency bands $\{\sigma_r | r = 1,2,\dots,R\}$, the attacked quantized DCT coefficients $\{\widehat{d}_{i,j}(\sigma_r) | i = 1,2,\dots,M,\ j = 1,2,\dots,N,\ r = 1,2,\dots,R\}$ are selected, and the attacked coefficient matrices $\{\widehat{Z}_r | r = 1,2,\dots,R\}$ can be obtained. Next, with the private key K_s, the attacked robust features $\{\widehat{\lambda}(k) | k = 1,2,\dots,L\}$ can be reconstructed from $\{\widehat{Z}_r | r = 1,2,\dots,R\}$ by using

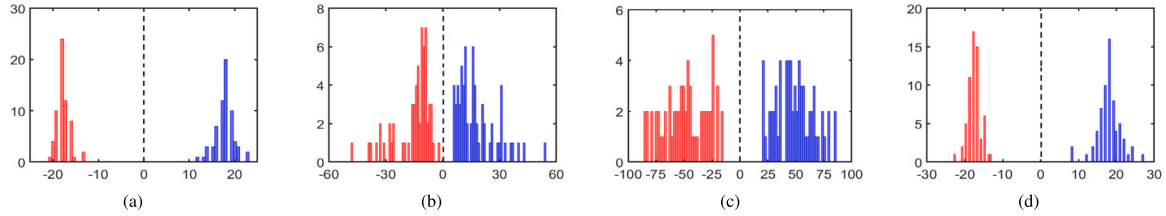


Fig. 5. The histograms of the robust features of the watermarked JPEG image *Lena* after being attacked: (a) JPEG recompression with a quality factor of 25, (b) JPEG2000 compression with a compression ratio of 13, (c) AWGN with a mean of 0 and a variance of 0.02, (d) Gaussian low-pass filtering with a window of 3 × 3 and a standard deviation of 1.

Eqs. (10) and (11). Finally, for each $k \in \{1, 2, \dots, L\}$, the watermark bit $w(k)$ can be extracted by

$$w(k) = \begin{cases} 1, & if\ \hat{\lambda}(k) \geq 0, \\ 0, & if\ \hat{\lambda}(k) < 0. \end{cases} \tag{22}$$

In this way, the embedded watermark bits can be extracted for copyright protection of JPEG images.

3.4. Prevention of overflow

It is worth noting that, unlike uncompressed images, JPEG images save quantized DCT coefficients rather than directly saving pixels, so it is necessary to prevent the overflow of quantized DCT coefficients to avoid encoding failure. According to the description in Section 4, the quantized AC coefficients are utilized to construct robust features in the proposed algorithm. For an 8-bit JPEG image, the range of quantized AC coefficients is $[-1023,\ 1023]$ in baseline sequential encoding mode. To prevent the overflow of the quantized AC coefficients, the quantized AC coefficients must be preprocessed before watermark embedding.

According to Eqs. (12) and (13), we can derive that

$$\left\lfloor \frac{\lfloor T/R \rfloor}{m \times n} \right\rfloor \leq \left| \widetilde{d}_{i,j}(f) - d_{i,j}(f) \right| \leq \left\lfloor \frac{\lfloor (T+R-1)/R \rfloor - 1}{m \times n} + 1 \right\rfloor. \tag{23}$$

We denote the maximum value of the modification on the quantized AC coefficient as Φ, then according to Eq. (23), we have

$$\Phi = \left\lfloor \frac{\lfloor (T+R-1)/R \rfloor - 1}{m \times n} + 1 \right\rfloor. \tag{24}$$

According to Eqs. (23) and (24), it can be seen that after watermark embedding, the range of the modified quantized AC coefficient $\widetilde{d}_{i,j}(f)$ is $[d_{i,j}(f) - \Phi,\ d_{i,j}(f) + \Phi]$. In order to prevent the overflow of the quantized AC coefficient, the original quantized AC coefficients must be adjusted to within the range $[-1023 + \Phi,\ 1023 - \Phi]$.

Before watermark embedding, the quantized AC coefficients used to construct robust features are scanned, and the quantized AC coefficients whose values are not within the range $[-1023+\Phi,\ 1023-\Phi]$ are adjusted by

$$d'_{i,j}(f) = \begin{cases} d_{i,j}(f) + \Phi, & if\ d_{i,j}(f) \in [-1023,\ -1023 + \Phi), \\ d_{i,j}(f) - \Phi, & if\ d_{i,j}(f) \in (1023 - \Phi,\ 1023], \end{cases} \tag{25}$$

where $d'_{i,j}(f)$ is the adjusted quantized AC coefficient that can be used for watermark embedding. Then, the positions of the adjusted quantized AC coefficients are marked to generate a location map. The location map can be embedded into the unselected frequency band by an RW algorithm of JPEG images.

On the receiver side, in the case of no attacks, the location map can be recovered losslessly to locate the adjusted quantized AC coefficient. After watermark extraction, the adjusted quantized AC coefficients can be recovered. Then, since Φ usually satisfies $\Phi < 20$ to ensure good image quality, the adjusted quantized AC coefficient $d'_{i,j}(f)$ can be re-adjusted by Eq. (26) to obtain the original quantized AC coefficient.

$$d_{i,j}(f) = \begin{cases} d'_{i,j}(f) - \Phi, & if\ d'_{i,j}(f) < 0, \\ d'_{i,j}(f) + \Phi, & if\ d'_{i,j}(f) > 0. \end{cases} \tag{26}$$

In fact, in order to reduce embedding distortion, the values of Φ and the selected quantized AC coefficients are very small in the proposed algorithm. Thus, in the vast majority of JPEG images, even JPEG images with $QF = 100$, the quantized AC coefficient is in the range $[-1023 + \Phi,\ 1023 - \Phi]$, which means that the overflow of the quantized AC coefficient will not occur.

4. Frequency band selection

In this section, how to select the appropriate frequency bands for watermark embedding is introduced in detail. In Section 3.1, we initially select $\Omega_L = \{1, 2, \dots, 36\}$ for watermark embedding. The modifications of the quantized DCT coefficients in different frequency bands have different effects on embedding distortion, file size expansion, and structural similarity in the spatial domain. Thus, we design an evaluation method that can comprehensively consider embedding distortion, file size expansion, and structural similarity to select the appropriate frequency bands for watermark embedding. In the remainder of this section, we analyze the impact of modifying the quantized DCT coefficient on embedding distortion, file size expansion, and structural similarity, and describe the designed evaluation method in detail.

4.1. Embedding distortion

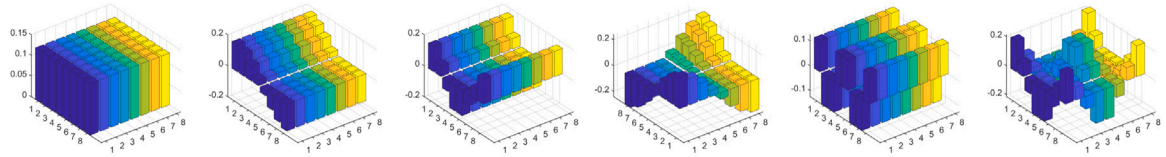
We apply the mean square error to evaluate the embedding distortion in the spatial domain caused by modifying the quantized DCT coefficients. The watermarked version of the original decoded block $P^*_{i,j}$ is denoted as $\widetilde{P}_{i,j} = \{\widetilde{p}_{i,j}(a,b) | a, b = 0, 1, \dots, 7\}$. The embedding distortion ED in the spatial domain can be computed by

$$ED = \frac{1}{64MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{a+b=0}^{14} [\widetilde{p}_{i,j}(a,b) - p^*_{i,j}(a,b)]^2. \tag{27}$$

The watermarked version of the original dequantized block $S^*_{i,j}$ is denoted as $\widetilde{S}_{i,j} = \{\widetilde{s}_{i,j}(f) | f = 1, 2, \dots, 64\}$, and according to the description in Section 2.1, we have $\widetilde{s}_{i,j}(f) = \widetilde{d}_{i,j}(f) \times q(f)$ and $s^*_{i,j}(f) = d_{i,j}(f) \times q(f)$. After the watermark is embedded, the modification on $d_{i,j}(f)$ is denoted as $\psi_{i,j}(f)$, and $\psi_{i,j}(f) = \widetilde{d}_{i,j}(f) - d_{i,j}(f)$. According to Eqs. (3) and (4), we can derive that

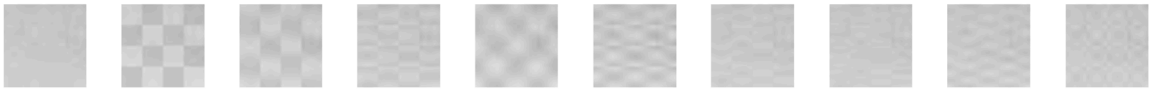
$$\begin{aligned} ED &= \frac{1}{64MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{a+b=0}^{14} \left\{ \sum_{f=1}^{64} \xi_f^{a,b} [\widetilde{s}_{i,j}(f) - s^*_{i,j}(f)] \right\}^2 \\ &= \frac{1}{64MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{a+b=0}^{14} \left\{ \sum_{f=1}^{64} \xi_f^{a,b} q(f) [\widetilde{d}_{i,j}(f) - d_{i,j}(f)] \right\}^2 \\ &= \frac{1}{64MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{a+b=0}^{14} \left\{ \sum_{f=1}^{64} \xi_f^{a,b} q(f) \psi_{i,j}(f) \right\}^2. \end{aligned} \tag{28}$$

According to Eq. (28), the larger the quantization step $q(f)$ or the modification $\psi_{i,j}(f)$, the greater the embedding distortion ED. In general, $q(f)$ increases as f increases. According to Eq. (13), $\psi_{i,j}(f)$ is mainly related to the threshold T determined by $max(|\Lambda|)$. The larger $max(|\Lambda|)$ is, the larger the value must be set for T, resulting in a larger $\psi_{i,j}(f)$. Thus, it is necessary to avoid selecting those frequency bands with



(a) (b) (c) (d) (e) (f)

Fig. 6. Modifications on $P^*_{i,j}$ in the spatial domain when $q(f)\psi_{i,j}(f) = 1$: (a) $f = 1$, (b) $f = 3$, (c) $f = 4$, (d) $f = 5$, (e) $f = 11$, (f) $f = 13$.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 7. The original image block and the watermarked image blocks under different σ_1: (a) original block, (b) $\sigma_1 = 1$, $T = 117$, (c) $\sigma_1 = 3$, $T = 119$, (d) $\sigma_1 = 4$, $T = 64$, (e) $\sigma_1 = 5$, $T = 121$, (f) $\sigma_1 = 9$, $T = 89$, (g) $\sigma_1 = 10$, $T = 32$, (h) $\sigma_1 = 11$, $T = 17$, (i) $\sigma_1 = 12$, $T = 38$, (j) $\sigma_1 = 13$, $T = 30$.

larger $q(f)$ or larger calculated $max(|\Lambda|)$ for watermark embedding. For the frequency band $f \in \{1, 2, \dots, 6\}$, since the amplitude of $d_{i,j}(f)$ is usually larger, a larger $max(|\Lambda|)$ is calculated by Eqs. (9), (10), and (11). For $f \in \{22, 23, \dots, 36\}$, $q(f)$ is larger. Hence, frequency bands $\{1, 2, \dots, 6, 22, 23, \dots, 36\}$ are not suitable for watermark embedding.

4.2. File size expansion

The modification of the quantifying DCT coefficients may expand the file size of JPEG images. As described in [27,29], and [53], modifications to the zero-valued quantized DCT coefficients will significantly increase the file size of JPEG images. Thus, we should avoid using the frequency bands with more zero-valued quantized DCT coefficients for watermark embedding. Since $d_{i,j}(f) = \left[s_{i,j}(f)/q(f)\right]$, the larger $q(f)$ is, the smaller $d_{i,j}(f)$ is, there may be more zero-valued quantized DCT coefficients in the frequency band with a larger $q(f)$. For the frequency band $f \in \{22, 23, \dots, 36\}$, as $q(f)$ is larger, these frequency bands are not suitable for watermark embedding. In this paper, the file size expansion is measured by the growth rate GR, and GR can be formulated as

$$GR = \frac{FS(\widetilde{Y}) - FS(Y)}{FS(Y)} \times 100\%, \tag{29}$$

where $FS(\widetilde{Y})$ is the file size of the watermarked JPEG image $\widetilde{Y}$, and $FS(Y)$ is the file size of the original JPEG image Y.

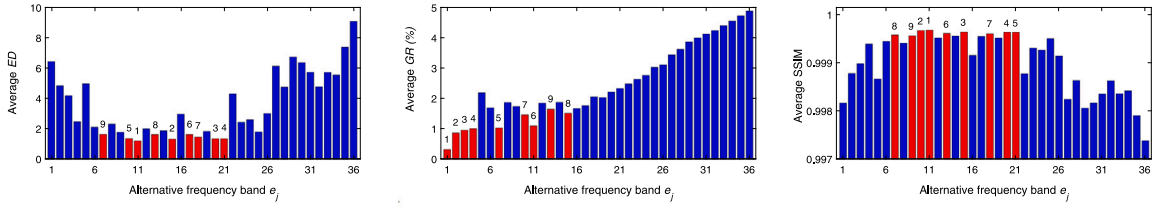
4.3. Structural similarity

Due to the DCT basis $\xi^{a,b}_f$, the modifications of the quantized DCT coefficients in different frequency bands cause different damages to the image structure in the spatial domain. Specifically, for a quantized block $D_{i,j}$, when only one quantized DCT coefficient $d_{i,j}(f)$ is modified and the modification $\psi_{i,j}(f)$ satisfies $q(f)\psi_{i,j}(f) = 1$, the corresponding modifications on decoded block $P^*_{i,j}$ are shown in Fig. 6, where the frequency band $f \in \{1, 3, 4, 5, 11, 13\}$. Note that for each case of f, when $q(f)\psi_{i,j}(f) = 1$, the embedding distortion ED on $P^*_{i,j}$ is $1/64$. As shown in Fig. 6, under the same embedding distortion ED, for different frequency bands, the modifications on the pixels in $P^*_{i,j}$ are different, resulting in different degrees of damage to the image structure. The greater the damage to the image structure, the worse the visual quality of the image. Thus, we introduce structural similarity (SSIM) [54] to evaluate the damage to the image structure. The closer the SSIM is to 1, the less damage is done to the image structure and the better the visual quality is.

In addition, the blocking artifacts may be aggravated due to the modification of the quantized DCT coefficients. The more serious the blocking artifacts are, the greater the damage to the image structure. For ease of explanation, we use the image *Airplane* with $QF = 70$ for discussion, and select a 32×32 sized image block as shown in Fig. 7(a) from this image for display. In the discussion, we set $m = n = 4$, $R = 1$, and $\{\sigma_1\} \in \{1, 3, 4, 5, 9, 10, 11, 12, 13\}$. Then a 256-bit watermark sequence is embedded into the image *Airplane* with different σ_1, and for each case of σ_1, T is set to $max(|\Lambda|) + 1$. Figs. 7(b)–(j) show the watermarked image blocks under different (σ_1) and the corresponding SSIM values between the original block and the watermarked block. As shown in Fig. 7, for $\sigma_1 \in \{1, 3, 4, 9\}$, the blocking artifacts are more serious, which causes more damage to the image structure, resulting in a smaller SSIM value. The reason is that for these frequency bands, due to the larger value of T and the corresponding DCT basis, the boundary pixels of block $P^*_{i,j}$ are greatly modified, which aggravates the blocking artifacts. For $\sigma_1 = 5$, the blocking artifacts are not obvious, but due to the large value of T and the corresponding DCT base $\xi^{a,b}_5$, the pixels at the corners of block $P^*_{i,j}$ are greatly modified, causing great damage to the image structure. Thus, it is necessary to avoid selecting those frequency bands with larger calculated $max(|\Lambda|)$ for watermark embedding, so that T can be set to a smaller value for higher structural similarity.

4.4. Evaluation method

Based on the above analysis, we design an evaluation method that can comprehensively consider the embedding distortion, file size expansion, and structural similarity to select the appropriate frequency bands for watermark embedding. Assuming that there are $\mathbb{N}$ options for $\{\sigma_r | r = 1, 2, \dots, R\}$ under a given R, and the jth option is denoted as $\mathbf{e}_j = \{\sigma^j_r | r = 1, 2, \dots, R\}$, where $j \in \{1, 2, \dots, \mathbb{N}\}$, σ^j_r is the selected frequency band in $\mathbf{e}_j$. Under the same conditions, including carrier, watermark, private key K_s, and cell size $m \times n$, when the option $\mathbf{e}_j$ is selected for watermark embedding, the resulting embedding distortion, file size expansion, and structural similarity are denoted as ED_j, GR_j, and $SSIM_j$, respectively. Note that for each option $\mathbf{e}_j$, the embedded watermark must be resistant to the same degree of attacks. Next, we sort $(ED_1, ED_2, \dots, ED_{\mathbb{N}})$ and $(GR_1, GR_2, \dots, GR_{\mathbb{N}})$ in ascending order to obtain two ranking sequences $(\gamma_1(1), \gamma_1(2), \dots, \gamma_1(\mathbb{N}))$ and $(\gamma_2(1), \gamma_2(2), \dots, \gamma_2(\mathbb{N}))$, and sort $(SSIM_1, SSIM_2, \dots, SSIM_{\mathbb{N}})$ in descending order to obtain a ranking sequence $(\gamma_3(1), \gamma_3(2), \dots, \gamma_3(\mathbb{N}))$. For each $i \in \{1, 2, 3\}$, we have $\gamma_i(u) = j$, where $u, j \in \{1, 2, \dots, \mathbb{N}\}$, $\gamma_1(u)$ satisfies $ED_{\gamma_1(1)} \le ED_{\gamma_1(2)} \le \cdots \le ED_{\gamma_1(\mathbb{N})}$ and $\gamma_1(a) < \gamma_1(b)$ if $ED_{\gamma_1(a)} = ED_{\gamma_1(b)}$ with $a < b$; $\gamma_2(u)$ satisfies $GR_{\gamma_2(1)} \le GR_{\gamma_2(2)} \le \cdots \le GR_{\gamma_2(\mathbb{N})}$ and $\gamma_2(a) < \gamma_2(b)$ if $GR_{\gamma_2(a)} = GR_{\gamma_2(b)}$ with $a < b$; and $\gamma_3(u)$ satisfies $SSIM_{\gamma_3(1)} \ge SSIM_{\gamma_3(2)} \ge \cdots \ge SSIM_{\gamma_3(\mathbb{N})}$ and $\gamma_3(a) < \gamma_3(b)$ if $SSIM_{\gamma_3(a)} = SSIM_{\gamma_3(b)}$ with $a < b$. For each $j \in \{1, 2, \dots, \mathbb{N}\}$, ED_j, GR_j, and $SSIM_j$ can be represented by their ranks RE_j, RG_j, and RS_j,



(a) (b) (c)

Fig. 8. Average ED, GR, and $SSIM$ of 1,000 JPEG images with $QF = 70$ for each optional frequency band $\mathbf{e}_j$ when $R = 1$, $\mathbf{e}_j = \{\sigma_1^j\} = \{j\} \in \Omega_L$, and the embedded watermark with 64 bits can resist JPEG recompression with a quality factor of 50: (a) Average ED, (b) Average GR, (c) Average SSIM.

(a) (b) (c) (d) (e) (f)

Fig. 9. Six standard test images: (a) *Baboon*, (b) *Bridge*, (c) *Goldhill*, (d) *Airplane*, (e) *Lena*, (f) *Boat*.

respectively, where $RE_j = {\gamma_1}^{-1}(j)$, $RG_j = {\gamma_2}^{-1}(j)$, and $RS_j = {\gamma_3}^{-1}(j)$. The evaluation factor $EF_{\mathbf{e}_j}$ for the option $\mathbf{e}_j$ can be computed by

$$EF_{\mathbf{e}_j} = \alpha RE_j + \beta RG_j + (1 - \alpha - \beta) RS_j. \tag{30}$$

$EF_{\mathbf{e}_j}$ is used to evaluate the comprehensive performance of the option $\mathbf{e}_j$ in terms of embedding distortion, file size expansion, and structural similarity. The option with a smaller $EF_{\mathbf{e}_j}$ has better comprehensive performance. In Eq. (30), $\alpha, \beta \in (0, 1)$ are two weighting factors and satisfy $\alpha + \beta < 1$. Specifically, if the performance in embedding distortion is more important, α should be larger, if less file size expansion is expected, β should be larger, while smaller α and β can achieve better performance in structural similarity. According to our observations, for α and β, the suggested ranges are $[0.4, 0.5]$ and $[0.2, 0.3]$, respectively. Referring to Section 5, when $(\alpha, \beta) = (0.45, 0.25)$, with the selected frequency bands, the proposed algorithm has satisfactory performance in embedding distortion, file size expansion, and structural similarity.

With the evaluation method, we test on 1,000 standard 512×512 sized JPEG images to select the appropriate frequency bands for watermark embedding, in which for each image, quality factors ranging from 10 to 100 with an interval of 10 are tested, and $(\alpha, \beta) = (0.45, 0.25)$. In testing, the robustness is evaluated by the bit error rate (BER), which is defined as the ratio of the number of extracted error bits to the number of embedded watermark bits. When the BER is less than 20%, it means that the embedded watermark is detectable and can resist the attack [51]. For ease of illustration, Fig. 8 shows the average ED, GR, and SSIM of 1,000 images with $QF = 70$ for each optional frequency band $\mathbf{e}_j$, when $R = 1$, $\mathbf{e}_j = \{\sigma_1^j\} = \{j\} \in \Omega_L$, the cell size $m \times n$ is set to 8×8, and for each $\mathbf{e}_j$, T is set to a minimum value that enables the embedded watermark with 64 bits to resist JPEG recompression with a quality factor of 50. In Fig. 8, the optional frequency bands with the top 9 ranks are marked in red and their corresponding ranks are marked above the bins. As shown in Fig. 8, in this case, the appropriate frequency band is $e_{11} = \{11\}$. After the extension experiments, according to our observations, the suggested range for R is $\{1, 2, 3\}$, and when $R = 1$, we consider selecting the frequency band $\{\sigma_1\} = \{11\}$ for watermark embedding; when $R = 2$, we consider selecting the frequency bands $\{\sigma_1, \sigma_2\} = \{11, 15\}$; and when $R = 3$, we consider selecting the frequency bands $\{\sigma_1, \sigma_2, \sigma_3\} = \{13, 11, 20\}$.

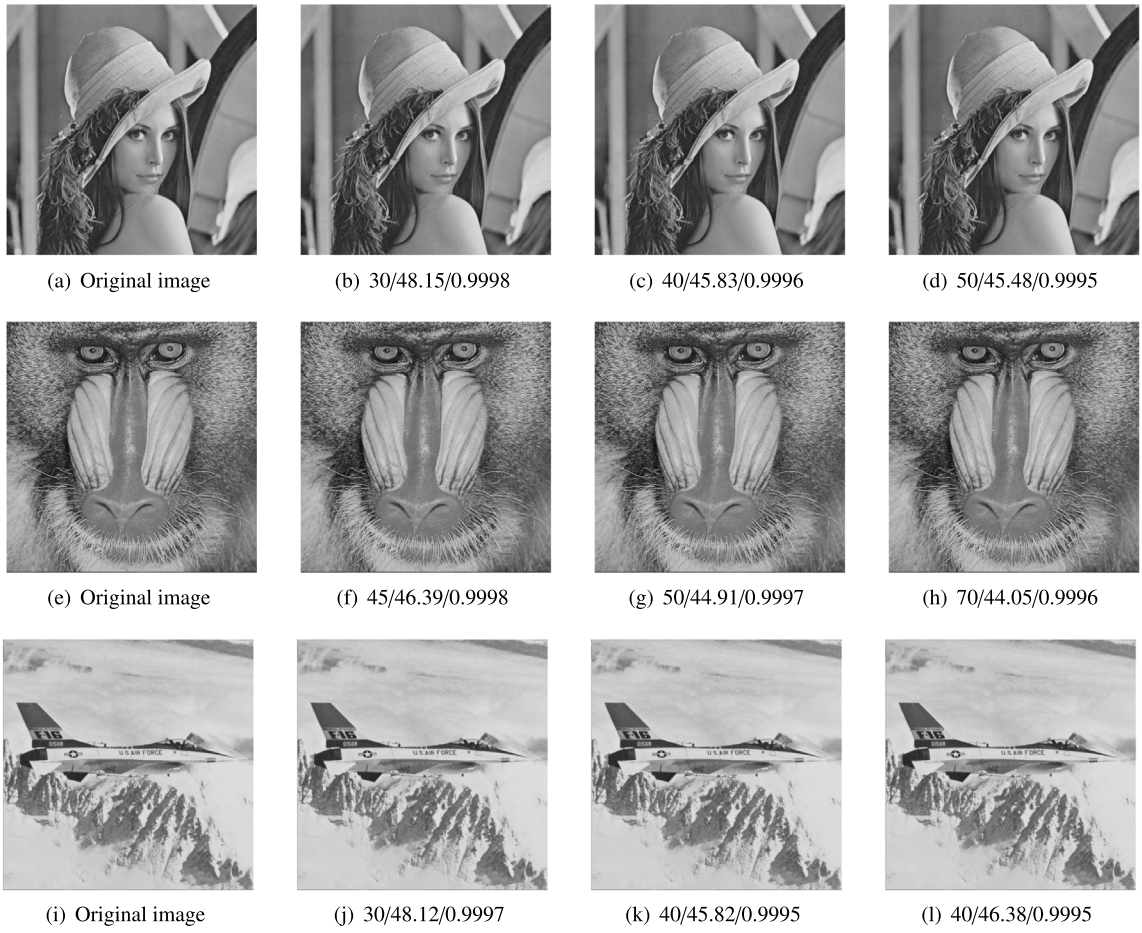
5. Experimental results

In this section, we evaluate the performance of the proposed algorithm. We randomly select 100 images from the USC-SIPI database and 100 images from the BOSSbase [55] database as test samples, and each image is a standard 512×512 sized 8-bit grayscale image. Fig. 9 shows the six classic standard images included in the test samples, where images *Baboon*, *Bridge*, and *Goldhill* represent texture images, and images *Airplane*, *Lena*, and *Boat* represent smooth images. The IJG toolbox [56] is used to compress the selected 200 images into JPEG images with different quality factors (QF = 70, 80, 90, and 100). In the experiments, the length of the watermark sequence corresponds to the embedding capacity, and we set the cell size $m \times n$ to 16×8, 8×8, 8×4, 4×4, and 4×2 for the embedding capacities of 32, 64, 128, 256, and 512 bits, respectively. Peak signal-to-noise ratio (PSNR) and SSIM are utilized to evaluate the visual quality of watermarked JPEG images. PSNR can measure embedding distortion, the higher the PSNR, the less the embedding distortion. SSIM can evaluate the structural similarity between the original image and the watermarked image. The higher the SSIM, the less damage is done to the image structure. The file size expansion is evaluated by the GR, which is calculated by Eq. (29) and is expected to be as small as possible. And the robustness is evaluated by the BER.

5.1. Effect on visual quality

In this section, we first use the three JPEG images with $QF = 70$ shown in the first column of Fig. 10 as example images to show the performance of the proposed algorithm in visual quality, including embedding distortion and structural similarity. In this experiment, a watermark composed of a 64-bit random sequence is embedded into each image with $R = 1$, 2, and 3, respectively. As shown in Fig. 10, for each example, the watermarked JPEG images generated with $R = 1, 2$, and 3 are shown in the second, third, and last columns, respectively. The corresponding threshold T and the PSNR and SSIM between the original JPEG image and the watermarked JPEG image are listed in the subtitles. From Fig. 10, we can see that PSNR higher than 44 dB and SSIM higher than 0.9995 can be obtained, which means that the proposed algorithm can generate watermarked JPEG images with satisfactory visual quality.

In the proposed algorithm, the visual quality is affected by the threshold T. To investigate the relationship between visual quality and T, we use the selected 200 images with $QF = 70$, 80, 90, and 100 for testing. A 64-bit watermark sequence is embedded into each image with $R = 1, 2$, and 3, respectively. Figs. 11 and 12 show the average PSNR values and average SSIM values obtained by the proposed algorithm for 200 test images with different values of T, respectively. From Figs. 11



(a) Original image (b) 30/48.15/0.9998 (c) 40/45.83/0.9996 (d) 50/45.48/0.9995

(e) Original image (f) 45/46.39/0.9998 (g) 50/44.91/0.9997 (h) 70/44.05/0.9996

(i) Original image (j) 30/48.12/0.9997 (k) 40/45.82/0.9995 (l) 40/46.38/0.9995

Fig. 10. Visualization results of the proposed algorithm with $R = 1, 2$, and 3. The first column is the original JPEG images. The second, third, and fourth columns show the watermarked JPEG images generated by the proposed method when $R = 1, 2$, and 3, respectively. T/PSNR/SSIM are shown in subtitles.

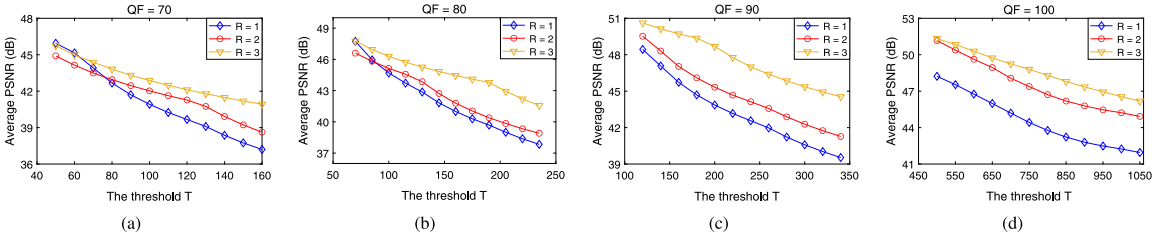


Fig. 11. Relation between the threshold T and PSNR when $R = 1, 2$, and 3.

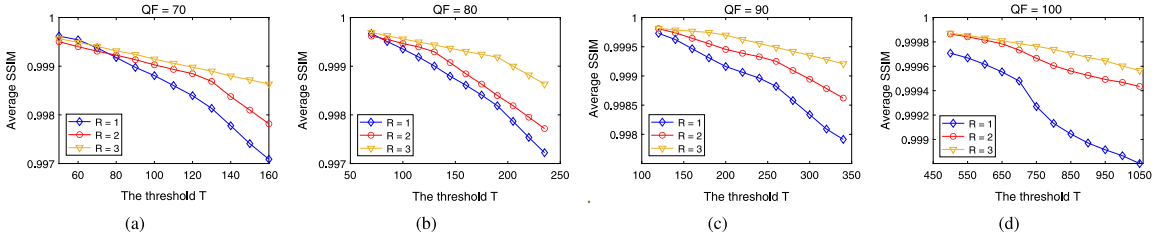


Fig. 12. Relation between the threshold T and SSIM when $R = 1, 2$, and 3.

and 12, we can see that for different QF and R, the average PSNR value greater than 38 dB and the average SSIM value higher than 0.9970 can be achieved. Since the embedding distortion increases as the threshold T increases, the average PSNR and average SSIM decrease as T increases. In addition, under the same QF and threshold T, compared to the cases of $R = 1$ and 2, the modification on a single quantized DCT coefficient is minimal in the case of $R = 3$. Thus, the proposed algorithm can perform better on PSNR and SSIM when $R = 3$.

Then, we investigate the performance of the proposed algorithm on visual quality under different embedding capacities. The average PSNR

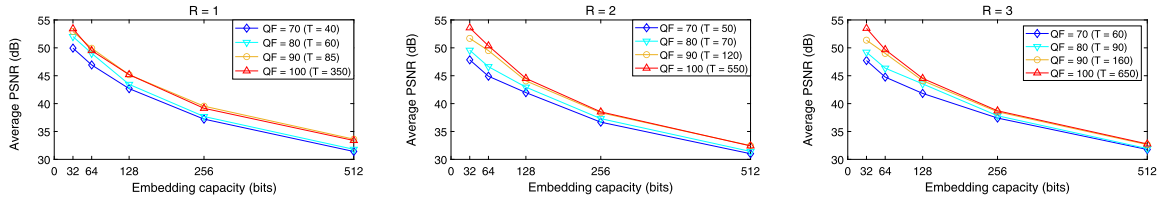


Fig. 13. Performance of the proposed algorithm on PSNR under different embedding capacities: (a) when $R = 1$, (b) when $R = 2$, (c) when $R = 3$.

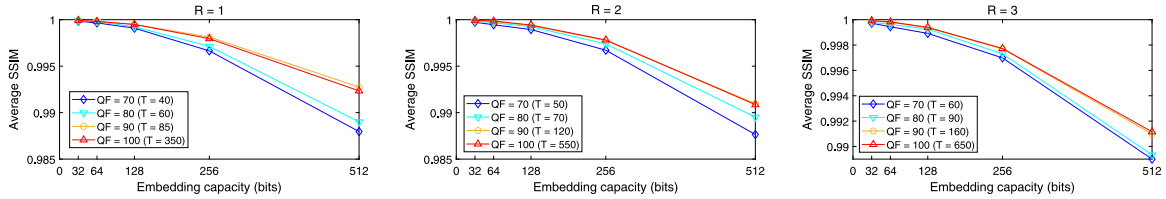


Fig. 14. Performance of the proposed algorithm on SSIM under different embedding capacities: (a) when $R = 1$, (b) when $R = 2$, (c) when $R = 3$.

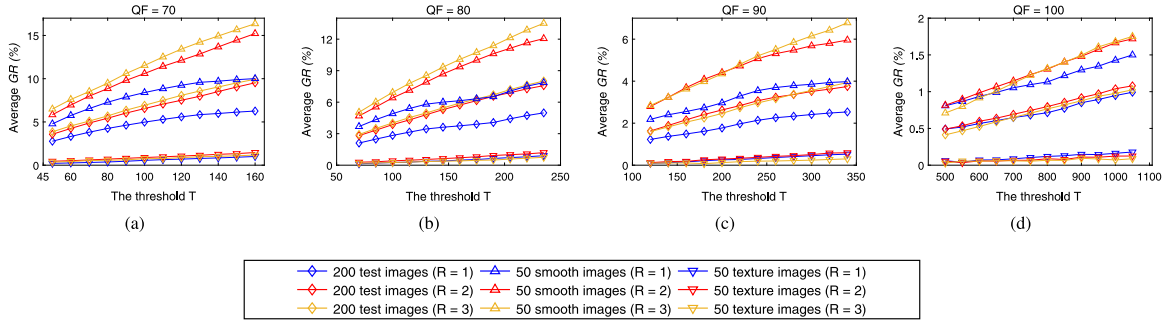


Fig. 15. Relation between the threshold T and GR when $R = 1, 2$, and 3.

values and average SSIM values obtained by the proposed algorithm for 200 test images with embedding capacities of 32, 64, 128, 256, and 512 bits are shown in Figs. 13 and 14, respectively. In this experiment, under the same R, for test images with the same QF, we fix the value of T. From Figs. 13 and 14, we can see that for different QF and R, when the embedding capacity is 256 bits, the average PSNR value greater than 35 dB and the average SSIM value greater than 0.9950 can still be achieved. The average PSNR value and the average SSIM value both decrease as the embedding capacity increases. The reason is that as the embedding capacity increases, the cell size must be reduced to obtain sufficient payload, and according to Eq. (12), the modification on the quantized DCT coefficients increases as the cell size decreases, which leads to an increase in embedding distortion.

5.2. Effect on file size expansion

File size expansion is an important indicator for evaluating JPEG image information hiding algorithms. In order not to affect the use of JPEG images, the file size expansion is expected to be small. Notice that the file size expansion for smooth images is different from that for texture images. Therefore, we select 200 images with $QF = 70$, 80, 90, and 100 for the experiment, including 50 texture images and 50 smooth images.

In the proposed algorithm, the file size expansion is affected by the threshold T. To research the relationship between the file size expansion and T, a 64-bit watermark sequence is embedded into each image with $R = 1$, 2, and 3, respectively. Fig. 15 shows the average GR values obtained by the proposed algorithm with different values of T for 200 test images, 50 smooth images, and 50 texture images. As shown in Fig. 15, for different QF and R, the average GR values for 200 test images are less than 10%. Also, the larger the threshold T, the greater the modification on the quantized DCT coefficients, which increases the length of the JPEG bitstream, increasing the average GR value. Moreover, when $R = 1$, only the quantized DCT coefficients in the frequency band $\{11\}$ are modified and the number of modified zero-valued quantized DCT coefficients is the smallest, so except for the images with $QF = 100$, under the same T, the average GR value is the smallest when $R = 1$. For images with $QF = 100$, since there are fewer zero-value quantized DCT coefficients in the selected frequency bands, the average GR value for images with $QF = 100$ is smaller, and the proposed algorithm has similar performance in file size expansion for different R. Fig. 16(a) shows the average number of modified zero-valued quantized DCT coefficients for 200 images with $QF = 70$, 80, 90, and 100 when $R = 1$, 2, and 3. Besides, the average GR value for texture images is smaller than that for smooth images, the reason is that there are more zero-value quantized DCT coefficients in the smooth image, then more zero-value quantized DCT coefficients will be modified during the watermark embedding process, which increases the file size. Fig. 16(b) shows the average number of modified zero-valued quantized DCT coefficients for 50 smooth images and 50 texture images when $R = 1$. In Fig. 16, the threshold T is set to 100, 150, 200, and 750 for test images with $QF = 70$, 80, 90, and 100, respectively.

Next, we investigate the performance of the proposed algorithm on file size expansion under different embedding capacities. The average GR values obtained by the proposed algorithm for 200 test images with embedding capacities of 32, 64, 128, 256, and 512 bits are shown

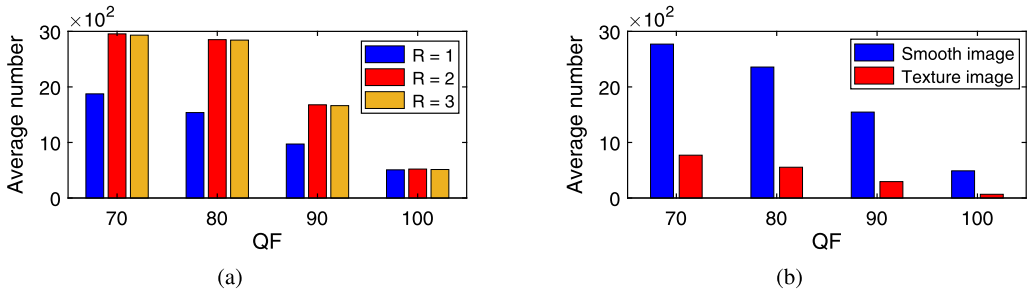


Fig. 16. The average number of modified zero-valued quantized DCT coefficients for test images with $QF = 70$, 80, 90, and 100: (a) for 200 test images when $R = 1, 2$, and 3, (b) for 50 smooth images and 50 texture images when $R = 1$.

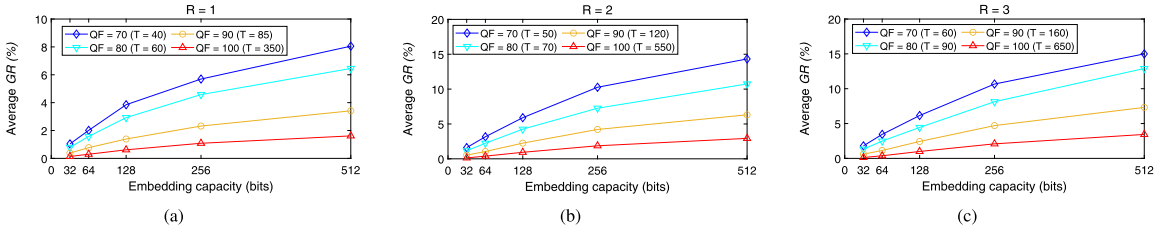


Fig. 17. Performance of the proposed algorithm on GR under different embedding capacities: (a) when $R = 1$, (b) when $R = 2$, (c) when $R = 3$.

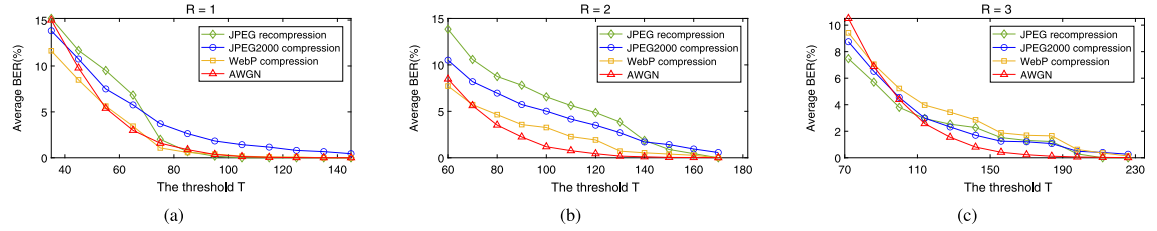


Fig. 18. Effect of the threshold T on the robustness against JPEG recompression, JPEG2000 compression, WebP compression, and AWGN: (a) when $R = 1$, (b) when $R = 2$, (c) when $R = 3$.

in Fig. 17. In this experiment, under the same R, for test images with the same QF, the value of T is fixed. As shown in Fig. 17, the average GR values can be less than 15% under different embedding capacities. Also, the average GR values rise as the embedding capacity increases, because the cell size must be reduced as the embedding capacity increases, which increases the modification on the quantized DCT coefficients, increasing the file size.

5.3. Robustness analysis

In the process of JPEG image transmission on the network, there are common attacks such as JPEG recompression and format conversion. Therefore, the embedded watermark is expected to be robust to these common attacks to a certain extent. In the experiments, we use the software MATLAB and Libwebp-1.2.0 as attack tools to test the robustness. Notice that, for the attacks except JPEG recompression, the attacked watermarked image must be saved as a JPEG image for watermark extraction.

In the proposed algorithm, the robustness of the watermark is mainly controlled by the threshold T. To research the effect of the threshold T on robustness, we use 200 JPEG images with $QF = 70$ as the test samples. A 64-bit watermark sequence is embedded into each image with $R = 1$, 2, and 3, respectively. Fig. 18 shows the average BER values of extracted watermark bits obtained by the proposed algorithm for 200 test images with different values of T under four common attacks, in which the quality factor of JPEG recompression is 20, the compression ratio of JPEG2000 compression is 15:1, the quality factor of WebP compression is 35, and the variance of AWGN with a mean of 0 is 0.03. As can be seen from Fig. 18, under different R, the average BER values for 200 test images decrease with the increase of T for different attacks, which means that the robustness rises as T increases. However, as T increases, the PSNR value and SSIM value decrease, and the GR value increases. Thus, it is necessary to adjust the value of T to strike a balance between visual quality, file size expansion, and robustness.

Next, in order to investigate the robustness of the proposed algorithm under different embedding capacities, we use 200 JPEG images with $QF = 70$, 80, 90, and 100 for testing. Considering that the PSNR value decreases as the embedding capacity increases, the PSNR values of watermarked images are adjusted to around 38 dB, and the values of T for different QF and embedding capacities are listed in Table 1. Table 1 shows the average BER values of extracted watermark bits obtained by the proposed algorithm for 200 test images with $R = 1$, 2, 3, and embedding capacities of 32, 64, 128, and 256 bits under four common attacks. The quality factor of JPEG recompression is 20, the compression ratio of JPEG2000 compression is 15:1, the quality factor of WebP compression is 25, and the variance of AWGN is 0.03. As shown in Table 1, for different QF, the proposed algorithm can achieve satisfactory robustness against these four attacks with different embedding capacities. And for each image, under the same PSNR, QF, and R, the average BER value for the embedding capacity of 32 bits is the smallest. The reason is that under the same PSNR, QF, and R, the value of T is the largest when the embedding capacity is 32 bits, which improves the robustness. To strike a balance between visual quality, file size expansion, embedding capacity, and robustness, we recommend that the embedding capacity be in the range of 64 to 128 bits. In addition, it can be seen from Table 1 that under the same embedding capacity and QF, compared with the cases of $R = 2$ and 3, the robustness is strongest when $R = 1$, because according to Table 1

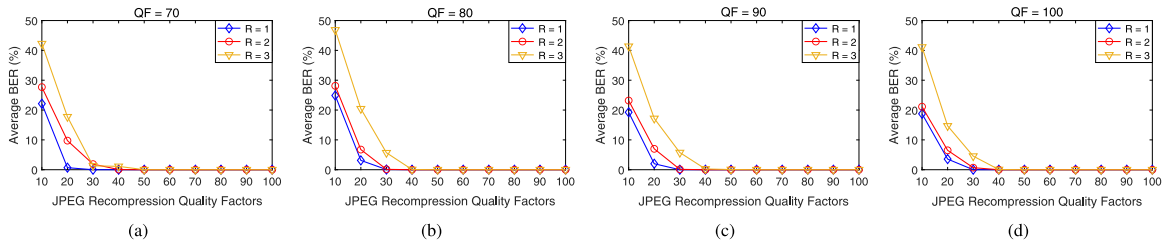


Fig. 19. Robustness to JPEG recompression when $R = 1, 2$, and 3.

Table 1
Average BER (%) of extracted watermark bits obtained by the proposed algorithm for 200 test images with $R = 1$, 2, 3, and embedding capacities (EC) of 32, 64, 128, and 256 bits under four common attacks (JPEG recompression (JPEGR), JPEG2000 compression (JP2C), WebP compression (WebPC), and AWGN).

R	EC (bits)	$QF = 70$					$QF = 80$					$QF = 90$					$QF = 100$				
		T	JPEGR	JP2C	WebPC	AWGN	T	JPEGR	JP2C	WebPC	AWGN	T	JPEGR	JP2C	WebPC	AWGN	T	JPEGR	JP2C	WebPC	AWGN
1	32	291	0	0	0	0	460	0	0.39	0.26	0	816	0	0.39	0.52	0	3319	0	0.26	0.52	0
	64	146	0	0.52	0.13	0.01	230	0	0.52	0.65	0.01	408	0	0.52	0.91	0.01	1660	0	0.52	1.17	0
	128	73	0.13	2.02	0.98	0.36	115	0.07	2.28	2.34	0.34	204	0.07	2.47	2.21	0.27	830	0.07	2.08	3.19	0.20
	256	37	0.46	4.36	3.16	2.13	58	0.39	4.88	4.59	2.11	102	0.33	5.4	4.62	2.07	415	0.33	3.55	4.88	1.71
2	32	345	0	0	0.78	0	529	0	0.39	0.78	0	1021	0.78	0.52	1.04	0	4620	0.26	0.26	0.78	0
	64	173	0	0.52	2.08	0.01	264	0	0.65	1.95	0.01	510	1.43	0.65	2.73	0.01	2310	1.56	0.52	3.13	0.01
	128	86	0.39	2.02	3.65	0.39	132	1.11	2.34	3.91	0.35	253	3.71	2.47	4.62	0.28	1155	1.89	2.28	4.49	0.20
	256	43	3.48	4.88	6.67	2.61	66	1.63	5.11	6.84	2.59	126	6.28	5.44	7.39	2.19	577	5.11	4.82	7.03	1.76
3	32	454	0	0	0.78	0	703	0.52	0.39	0.78	0	1370	0.78	0.52	1.04	0	5675	0.52	0.52	0.78	0
	64	277	0	0.65	2.60	0.02	352	1.69	0.65	2.73	0.02	685	1.82	0.65	3.26	0.02	2837	1.69	0.65	3.13	0.01
	128	113	0.33	2.08	5.01	0.42	175	3.61	2.54	5.34	0.42	345	3.91	2.67	5.92	0.39	1418	2.34	1.76	5.79	0.21
	256	56	1.56	5.21	7.32	2.62	88	6.93	5.57	8.69	2.68	171	7.03	5.96	8.89	2.79	709	5.73	3.45	8.85	1.86

Table 2
Average SSIM and average GR obtained by the proposed algorithm for 200 test images with $R = 1$, 2, 3, and an embedding capacity of 64 bits at PSNR $\approx$ 42 dB.

QF	Average SSIM			Average GR (%)		
	R = 1	R = 2	R = 3	R = 1	R = 2	R = 3
70	0.9991	0.9991	0.9991	4.27	6.04	6.14
80	0.9990	0.9991	0.9991	3.37	5.05	5.67
90	0.9989	0.9990	0.9990	1.95	2.86	3.31
100	0.9991	0.9991	0.9991	0.94	1.27	1.54

Table 3
Attack types and corresponding parameters.

Attack types	Parameters
JPEG recompression	Quality factor = 20
WebP compression	Quality factor = 20
JPEG2000 compression	Compression ratio = 15:1
AWGN	Mean = 0, Variance = 0.035
Gaussian low-pass filtering	Window = 3 × 3, Standard deviation = 1
Median filtering	Window = 3 × 3
Histogram equalization	–
Sharpening	Prewitt operator

and Eqs. (12) and (13), under the same PSNR and embedding capacity, as R increases, the robust region obtained on each quantized DCT coefficient decreases, resulting in a decrease in robustness.

To further evaluate the robustness of the proposed algorithm, we use 200 JPEG images with $QF = 70$, 80, 90, and 100 for the experiment, and test the robustness against three common compression attacks (JPEG recompression, WebP compression, and JPEG2000 compression). In this experiment, a 64-bit watermark sequence is embedded into each image with $R = 1$, 2, and 3, respectively. The PSNR values of watermarked JPEG images are adjusted to around 42 dB, and the corresponding average SSIM value and average GR value are listed in Table 2. The quality factors of JPEG recompression range from 10 to 100 with an interval of 10, the quality factors of WebP compression range from 10 to 100 with an interval of 10, and JPEG2000 compression ratios range from 5:1 to 50:1 with an interval of 5. Experimental results are plotted in Figs. 19 to 21.

As shown in Figs. 19 and 20, for different QF, the proposed algorithm can effectively resist JPEG recompression with a quality factor greater than 20 and WebP compression with a quality factor greater than 20. Also, under the attack with the same quality factor, compared with $R = 2$ and 3, the average BER value is the smallest when $R = 1$. And the proposed algorithm can even resist WebP compression with a quality factor of 10 when $R = 1$. As for JPEG recompression with a quality factor of 10, the robustness of the proposed algorithm is lower, because the quantized DCT coefficients in the selected frequency band are significantly distorted with the quantization table adopted in this case. In Fig. 21, for different QF, the average BER values of the proposed algorithm are still below 20% even when the compression ratio is 30:1. The above results show that the proposed algorithm can achieve strong robustness to JPEG recompression, WebP compression, and JPEG2000 compression.

Specifically, we use the six standard images shown in Fig. 9 to illustrate the robustness of the proposed algorithm against several attacks, where images *Baboon*, *Bridge*, and *Goldhill* represent texture images, and images *Airplane*, *Lena*, and *Boat* represent smooth images. The attack types and corresponding parameters are shown in Table 3.

Table 4 shows the BER values of extracted watermark bits obtained by the proposed algorithm for six standard images under three common compression attacks. As shown in Table 4, for the JPEG recompression, the BER value for each image is lower than 15%. And under the attack of the WebP compression, the BER value for each image is lower than 20%. Also, for JPEG recompression and WebP compression, the BER value for texture images is smaller than that for smooth images. For the JPEG2000 compression, the BER value for each image is lower than 20%. The results in Table 4 indicate that the proposed algorithm can effectively resist JPEG recompression, JPEG2000 compression, and WebP compression. In addition, due to the different image processing methods employed by these three compression attacks, the resultant distortion to quantized DCT coefficients varies correspondingly, resulting in different performances of the proposed algorithm against these three compression attacks. Specifically, DCT is performed on

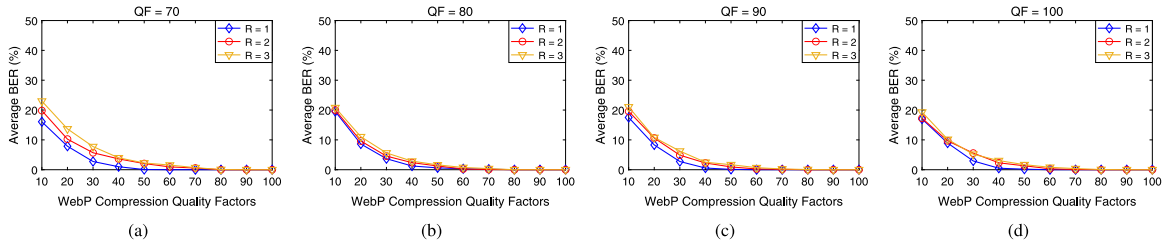


Fig. 20. Robustness to WebP compression when $R = 1, 2$, and 3.

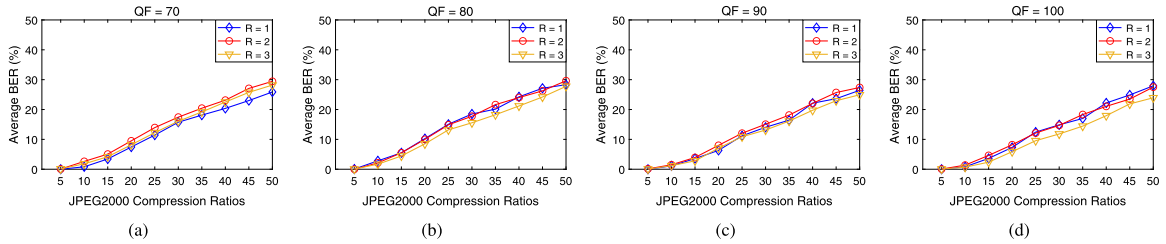


Fig. 21. Robustness to JPEG2000 compression when $R = 1, 2$, and 3.

Table 4
BER (%) of extracted watermark bits obtained by the proposed algorithm for six standard images with $R = 1$, 2, 3, and an embedding capacity of 64 bits under three common compression attacks (JPEG recompression, WebP compression, and JPEG2000 compression).

Attacks	Images	Baboon				Bridge				Goldhill				Airplane				Lena				Boat			
	R \ QF	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100
JPEG recompression	1	0	0	0	0	0	0	0	0	0	0	0	0	7.81	4.69	4.69	4.69	0	1.56	0	4.69	0	4.69	6.25	6.25
	2	1.56	0	0	0	0	0	0	0	3.13	1.56	0	0	12.50	7.81	12.50	10.94	10.94	9.38	9.38	12.50	10.94	7.81	9.38	7.81
	3	0	0	0	0	0	0	0	0	1.56	4.69	0	0	12.50	14.06	14.06	12.50	4.69	12.50	10.94	9.38	10.94	9.38	7.81	7.81
WebP compression	1	0	0	0	0	0	0	0	0	1.56	1.56	0	0	10.94	10.94	12.50	12.50	10.94	12.50	10.94	12.50	6.25	7.81	6.25	9.38
	2	0	1.56	0	0	0	0	0	0	4.69	3.13	0	1.56	17.19	14.06	15.63	14.06	15.63	14.06	15.63	14.06	10.94	9.38	14.06	10.94
	3	1.56	1.56	1.56	0	0	1.56	0	0	6.25	4.69	0	0	17.19	15.63	15.63	14.06	18.75	15.63	17.19	18.75	15.63	12.50	14.06	12.50
JPEG2000 compression	1	7.81	7.81	4.69	4.69	3.13	3.13	3.13	6.25	0	0	0	0	7.81	7.81	7.81	9.38	4.69	4.69	7.81	6.25	10.94	9.38	9.38	6.25
	2	18.75	10.94	7.81	9.38	7.81	9.38	6.25	4.69	0	1.56	0	0	12.50	10.94	9.38	10.94	4.69	4.69	4.69	3.13	12.50	9.38	9.38	10.94
	3	15.63	10.94	7.81	3.13	7.81	4.69	6.25	3.13	1.56	0	0	0	7.81	6.25	6.25	4.69	4.69	1.56	3.13	1.56	9.38	7.81	12.50	6.25

Table 5
BER (%) of extracted watermark bits obtained by the proposed algorithm for six standard images with $R = 1$, 2, 3, and an embedding capacity of 64 bits under other attacks (AWGN, Gaussian low-pass filtering, Median filtering, Histogram equalization, and Sharpening).

Attacks	Images	Baboon				Bridge				Goldhill				Airplane				Lena				Boat			
	R \ QF	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100	70	80	90	100
AWGN	1	1.70	1.17	1.00	0.94	1.45	1.09	0.89	0.84	1.19	0.95	0.76	0.78	1.09	0.79	0.64	0.59	0.86	0.53	0.39	0.36	0.79	0.79	0.59	0.42
	2	2.73	2.41	1.44	0.83	2.95	2.55	1.64	1.08	1.28	1.09	0.78	0.48	1.44	1.20	0.70	0.47	1.59	1.27	0.69	0.44	1.61	1.17	0.78	0.52
	3	4.50	2.81	2.50	1.19	3.98	2.70	2.36	1.20	2.73	1.22	1.14	0.48	2.91	1.30	1.11	0.45	3.55	1.78	1.56	0.76	2.98	1.34	1.09	0.55
Gaussian low-pass filtering	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	1.56	0	0	0	0	0	0	0	1.56	0	0	0
	3	0	1.56	0	0	1.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.56	1.56	1.56	0
Median filtering	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	3.13	1.56	1.56	1.56	3.13	0	1.56	0	0	0	0	0	3.13	0	0	0	0	0	0	0	1.56	1.56	0	0
	3	1.56	1.56	1.56	0	4.69	0	0	4.69	0	0	0	0	0	0	0	0	1.56	0	0	1.56	1.56	1.56	1.56	1.56
Histogram equalization	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sharpening	1	1.56	1.56	1.56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	3.13	3.13	1.56	1.56	1.56	1.56	1.56	1.56	1.56	0	0	1.56	0	0	0	0	0	0	0	0	1.56	0	0	0
	3	4.69	3.13	3.13	1.56	3.13	1.56	1.56	1.56	1.56	0	1.56	1.56	0	0	0	0	0	0	0	0	1.56	0	1.56	0

pixels in JPEG recompression, discrete wavelet transform is performed on pixels in JPEG2000 compression, and DCT is performed on the prediction error of pixels in WebP compression. Compared with JPEG recompression with a quality factor of 20, both JPEG2000 compression with a compression ratio of 15:1 and WebP compression with a quality factor of 20 cause greater distortion to the quantized DCT coefficients in the selected frequency band. Thus, under the parameter conditions in Table 3, the performance of the proposed algorithm for JPEG2000 compression and WebP compression is somewhat lower than that for JPEG recompression.

The BER values for AWGN, Gaussian low-pass filtering, Median filtering, Histogram equalization, and Sharpening are shown in Table 5. As shown in Table 5, for the AWGN, the BER value for each image is lower than 5%, which means that the proposed algorithm has strong robustness to AWGN. For the attacks caused by Gaussian low-pass filtering and Median filtering, the BER values of the proposed algorithm are lower than 2% and 5%, respectively, which means that the proposed algorithm has satisfactory robustness to Gaussian low-pass filtering and Median filtering. For the Histogram equalization, the proposed algorithm can achieve the BER value of 0. For the Sharpening, the BER values of the proposed algorithm are lower than 5%, and when $R = 1$, the BER values for Sharpening are lower than 2%. Furthermore, according to the results in Table 5, for the five above-mentioned image processing attacks, when $R = 1$, the proposed algorithm has better performance in robustness. Based on the performance of the proposed algorithm in visual quality, file size expansion, and robustness, we can observe that as R increases, PSNR and SSIM improve, while the file size expansion increases and the robustness decreases. For this reason, we recommend that the range for R is $\{1, 2, 3\}$.

Table 6
Performance under public lossy channels.

Channels	Instagram			WeChat		
Downloaded images						
PSNR (dB)	41.84	32.77	41.91	36.40	29.87	36.88
BER (%)	0	0	0	0	0	0

Table 7
PSNR (in dB) of six watermarked standard images generated by the proposed algorithm and the algorithms proposed in [43,45,49–51].

Images	[43]	[45]	[49]	[50]	[51]	Proposed
Baboon	36.99	37.67	37.73	37.71	37.43	38.00
Bridge	36.99	37.47	37.59	37.65	37.74	38.02
Goldhill	36.99	37.76	37.57	37.79	37.33	38.00
Airplane	36.99	37.65	37.99	37.86	37.70	38.00
Lena	36.99	37.74	37.74	37.60	37.95	38.00
Boat	36.99	37.67	37.73	37.99	37.69	38.00

Moreover, in order to test the robustness against public lossy channels, we apply the instant communication tools Instagram and WeChat to provide lossy channels to transmit watermarked JPEG images. In this experiment, we use 200 JPEG images as the test samples, a 64-bit watermark sequence is embedded into each image, and the PSNR values of watermarked images are adjusted to around 42 dB. Then, we upload all watermarked images through Instagram and WeChat, and extract the watermark sequence from the downloaded images. Table 6 shows the results for images *Lena*, *Baboon*, and *Airplane* with $QF = 90$ when $R = 1$, where the PSNR value is computed from the watermarked image and the downloaded image. The experimental results show that the watermark can be accurately extracted from the downloaded image, which means that the watermark embedded by the proposed algorithm can effectively resist the attacks from the lossy channels provided by Instagram and WeChat.

5.4. Performance comparison

In the literature, there is no detailed report on the RRW algorithm of JPEG images. In order to further illustrate the robustness of the proposed algorithm, several classic and advanced RRW algorithms of uncompressed images proposed in [43,45,49–51] are used for comparison. Note that when the RRW algorithm of uncompressed images is applied to JPEG images, the watermark is embedded into the decoded image. In the comparative experiment, the embedding capacity is 128 bits, and JPEG images with $QF = 90$ are used as test images, including the six standard images shown in Fig. 9. The experimental results of the proposed algorithm and the five compared algorithms for the six standard images are shown, and the experimental results for other images are similar to the experimental results for the six standard images. The hyper-parameters of the proposed algorithm are $(m, n) = (8, 4)$, $R = 1$, and $\{\sigma_1\} = \{11\}$. For the algorithm in [43], the block size is set to 64×32. For the algorithms in [45,49], the block size is set to 32×32. In [50], the block size is set to 64×16. In [51], the max order N of Zernike moments is set to 31, and the maximum value T of the normalized moments is set to 1000. For a fair comparison, the PSNR values of watermarked images are adjusted to around 38 dB, as shown in Table 7.

We first investigate the robustness of the proposed algorithm and the five compared algorithms to JPEG recompression, JPEG2000 compression, WebP compression, and AWGN. In the experiments, the quality factors of JPEG recompression range from 10 to 100 with an interval of 10, JPEG2000 compression ratios range from 1:1 to 20:1 with an interval of 1, and the quality factors of WebP compression range from 10 to 100 with an interval of 10. Additionally, for AWGN, the mean is 0 and the variance ranges from 0.005 to 0.035 with an interval of 0.0025. Experimental results are plotted in Figs. 22 to 25.

As can be seen from Fig. 22, the proposed algorithm can achieve a BER value of 0 for JPEG recompression with a quality factor greater than 20. For JPEG recompression with a quality factor of 10, the proposed algorithm achieves BER values below 20% for most images. From Fig. 23, it can be seen that for JPEG2000 compression with a compression ratio of 20:1, the BER values of the proposed algorithm are below 20%, indicating that the proposed algorithm can effectively resist JPEG2000 compression. In Fig. 24, for WebP compression with a quality factor greater than 20, the BER values of the proposed algorithm are below 5%, and the proposed algorithm can achieve BER values below 20% for most images under WebP compression with a quality factor of 10. Furthermore, from Fig. 25, we can observe that for AWGN with a mean of 0 and a variance of 0.035, the proposed algorithm can still achieve BER values below 1%, and can reach or approach the best results for six standard images. With the comparative experiments, we can see that the proposed algorithm has satisfactory robustness to JPEG recompression, JPEG2000 compression, WebP compression, and AWGN.

Next, we test the robustness of the proposed algorithm and the five compared algorithms to Gaussian low-pass filtering, Median filtering, Histogram equalization, and Sharpening. In the experiments, the window size and standard deviation of Gaussian low-pass filtering are 3×3 and 1, respectively. The window size of Median filtering is 3×3, and the Prewitt operator is used for Sharpening. The average BER (%) values of watermarks extracted from 200 watermarked images generated by the proposed algorithm and five compared algorithms under different attacks are listed in Table 8. As can be seen from Table 8, for these four attacks, the average BER values obtained by the proposed algorithm are below 0.3%. Compared with the algorithms in [43,45,49–51], the proposed algorithm achieves the best results for Histogram equalization and Sharpening. For Gaussian low-pass filtering and Median filtering, the average BER values obtained by the proposed algorithm are close to the best results, which means that the proposed algorithm has satisfactory robustness to Gaussian low-pass filtering and Median filtering.

It is worth noting that the algorithms proposed in [43,45,49–51] are RRW algorithms of uncompressed images. When the RRW algorithm of uncompressed images is applied to JPEG images, the JPEG image must first be decoded into pixels, and then the watermark is embedded into the decoded image, and the embedded watermark is extracted from the watermarked decoded image on the receiver side. However, when the watermarked decoded image or even the restored decoded image is re-saved as a JPEG image, the image will be distorted by the quantization, rounding, and truncation operations in JPEG compression, resulting in the loss of the embedded watermark and the inability to restore the original JPEG image. In the proposed algorithm, the watermark is directly embedded into the JPEG image without decoding the JPEG image into pixels. Thus, the proposed algorithm can accurately extract the embedded watermark from the JPEG image and restore the original JPEG image losslessly.

Moreover, in the existing RW algorithms of JPEG images, after the watermarked JPEG image is attacked by a common image processing

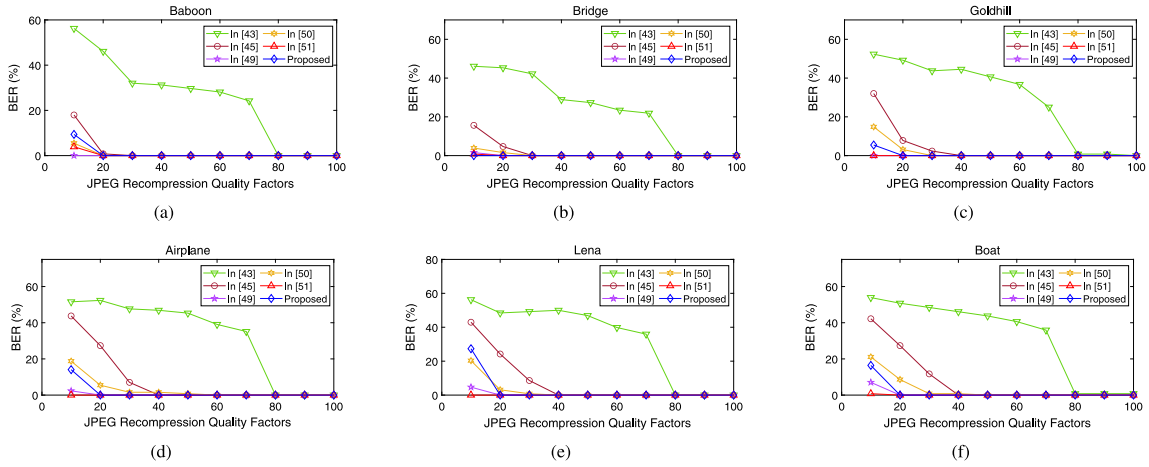


Fig. 22. Robustness to JPEG recompression compared with the algorithms proposed in [43,45,49–51].

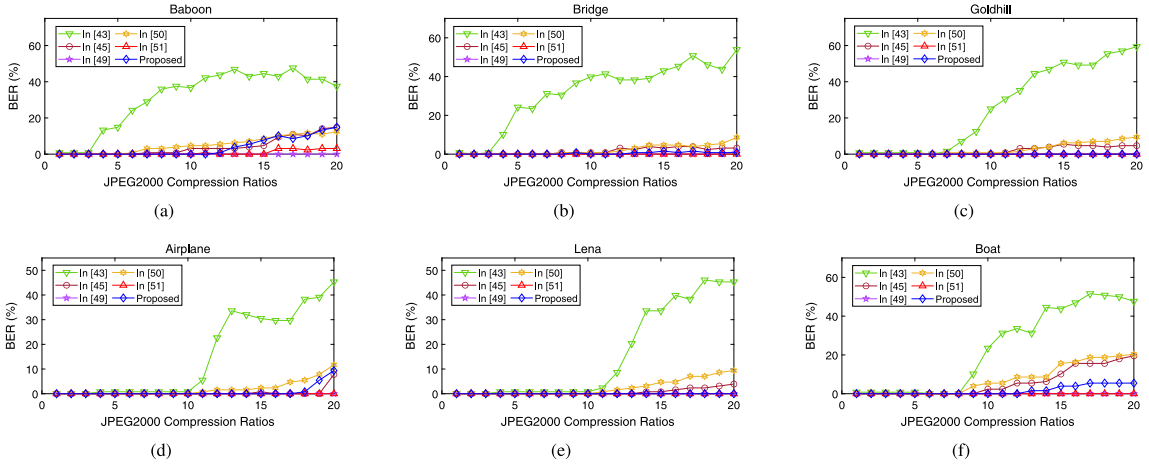


Fig. 23. Robustness to JPEG2000 compression compared with the algorithms proposed in [43,45,49–51].

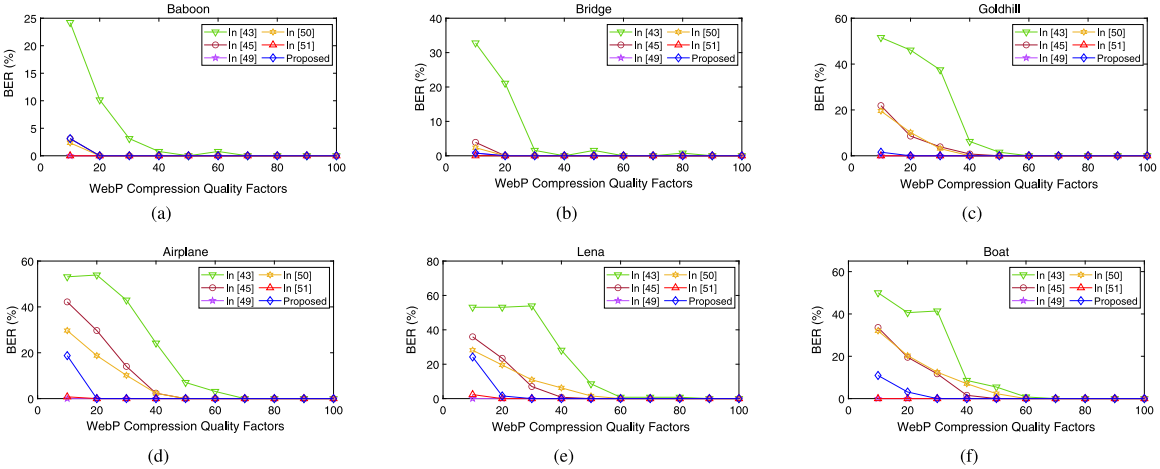


Fig. 24. Robustness to WebP compression compared with the algorithms proposed in [43,45,49–51].

Table 8
Robustness to other attacks compared with the algorithms proposed in [43,45,49–51].

Attacks	[43]	[45]	[49]	[50]	[51]	Proposed
Gaussian low-pass filtering	11.26	0	0.39	2.47	0	0.13
Median filtering	1.69	0.20	3.26	2.67	0.59	0.26
Histogram equalization	0.46	0	48.63	0.07	47.14	0
Sharpening	0.59	0.42	50.91	0.09	36.07	0.07

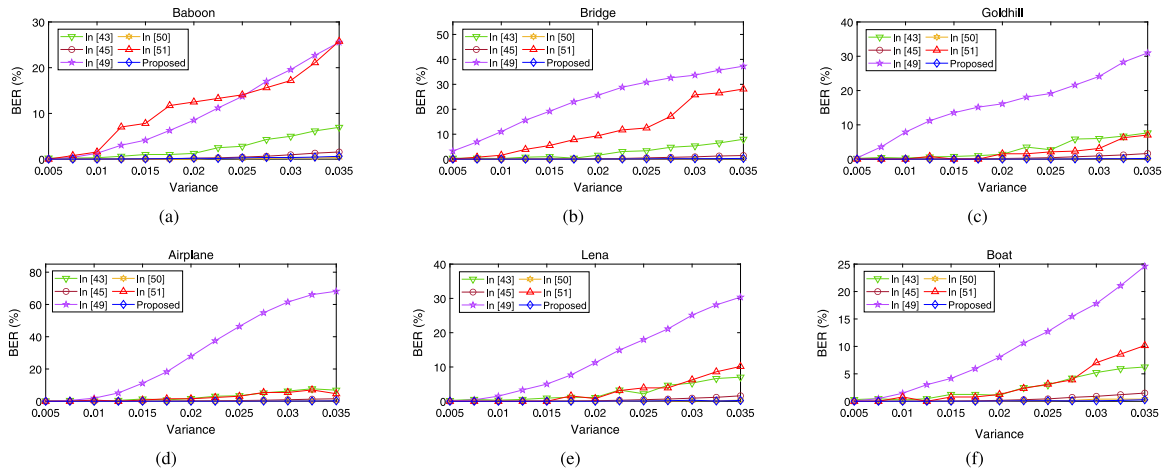


Fig. 25. Robustness to AWGN compared with the algorithms proposed in [43,45,49–51].

operation, since the quantization table, Huffman table, and quantization DCT coefficients are greatly modified, the embedded watermark will be lost. In the proposed algorithm, the embedded watermark achieves robustness against some common image processing operations, and without attack, the watermark can be extracted correctly and the original JPEG image can be restored losslessly. Therefore, the proposed algorithm is more suitable for copyright protection and integrity authentication of JPEG images.

6. Conclusion

In this paper, we proposed an effective robust reversible watermarking algorithm of JPEG images. The robustness of the watermark is achieved by selecting those quantized DCT coefficients in middle-frequency bands to construct robust features for watermarking. Since the constructed robust features can well exploit the correlation of quantized DCT coefficients, the embedded watermark achieves strong robustness. Furthermore, we designed an evaluation method to select appropriate frequency bands for watermarking by considering embedding distortion, file size expansion, structural similarity, and robustness, which effectively improved the performance of the proposed algorithm. The proposed algorithm has both robustness and reversibility. The robustness is useful for copyright protection, while the reversibility is valuable for integrity authentication and recovery of the original JPEG image in case of no attacks. Therefore, the proposed algorithm is more suitable for copyright protection and integrity authentication of JPEG images such as valuable news photographs, fine art photography works, forensic photographs, and scanned copies of important documents.

Experimental results have shown that in the three cases of $\{\sigma_1\} = \{11\}$ for $R = 1$, $\{\sigma_1, \sigma_2\} = \{11, 15\}$ for $R = 2$, and $\{\sigma_1, \sigma_2, \sigma_3\} = \{13, 11, 20\}$ for $R = 3$, the proposed algorithm still has satisfactory performance in terms of visual quality and file size expansion. When $R = 3$, the algorithm has better performance in PSNR value and SSIM value. In addition, the algorithm has strong robustness to those common image processing operations, including JPEG recompression with a quality factor of 20, JPEG2000 compression with a compression ratio of 15:1, WebP compression with a quality factor of 20, and AWGN with a mean of 0 and a variance of 0.035. In the case of $R = 1$, the algorithm has better performance in robustness. Compared with existing RRW algorithms of uncompressed images, the proposed algorithm maintains satisfactory robustness to common image processing operations while being able to restore the original JPEG image without loss. Moreover, our experimental results on public channels (Instagram and WeChat) show that the proposed algorithm is practical and suitable for the copyright protection of JPEG images.

The proposed algorithm is an effective attempt at RRW of JPEG images, and there is room for improvement in visual quality and robustness when the embedding capacity is greater than 256 bits. And the performance of the proposed algorithm in embedding capacity is expected to be further improved. In future works, we will strive to design more effective robust features and watermark embedding strategies for JPEG images, so as to achieve the RRW algorithm of JPEG images with better performance in terms of visual quality, file size expansion, embedding capacity, and robustness.

CRediT authorship contribution statement

Xingyuan Liang: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Shijun Xiang:** Validation, Methodology, Conceptualization.

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

The data that has been used is confidential.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272197), the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011928), and the National Key Research and Development Program of China (No. 2023YFF0905000).

References

[1] C.W. Honsinger, P.W. Jones, M. Rabbani, J.C. Stoffel, Lossless recovery of an original image containing embedded data, in: US Patent 6 278, 2001, p. 791.
[2] J.-B. Feng, I.-C. Lin, C.-S. Tsai, Y.-P. Chu, Reversible watermarking: Current status and key issues, Int. J. Netw. Secur. 2 (3) (2006) 161–170.
[3] J. Fridrich, M. Goljan, R. Du, Lossless data embedding-new paradigm in digital watermarking, EURASIP J. Adv. Signal Process. 2002 (2) (2002) 185–196.
[4] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-lsb data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.
[5] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[6] D.M. Thodi, J.J. Rodríguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.

[7] C. Qin, Z. He, H. Yao, F. Cao, L. Gao, Visible watermark removal scheme based on reversible data hiding and image inpainting, Signal Process.-Image Commun. 60 (2018) 160–172.
[8] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[9] X. Li, W. Zhang, X. Gui, B. Yang, Efficient reversible data hiding based on multiple histograms modification, IEEE Trans. Inf. Forensic Secur. 10 (9) (2015) 2016–2027.
[10] W. Wang, J. Ye, T. Wang, W. Wang, A high capacity reversible data hiding scheme based on right-left shift, Signal Process. 150 (2018) 102–115.
[11] Y. Jia, Z. Yin, X. Zhang, Y. Luo, Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting, Signal Process. 163 (2019) 238–246.
[12] H. Zheng, C. Wang, J. Wang, S. Xiang, A new reversible watermarking scheme using the content-adaptive block size for prediction, Signal Process. 164 (2019) 74–83.
[13] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process. 20 (12) (2011) 3524–3533.
[14] B. Ou, X. Li, Y. Zhao, R. Ni, Y.-Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, IEEE Trans. Image Process. 22 (12) (2013) 5010–5021.
[15] H.-T. Wu, J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process. 92 (12) (2012) 3000–3009.
[16] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Process. 93 (1) (2013) 198–205.
[17] M. Xiao, X. Li, Y. Wang, Y. Zhao, R. Ni, Reversible data hiding based on pairwise embedding and optimal expansion path, Signal Process. 158 (2019) 210–218.
[18] W. Qi, T. Zhang, X. Li, B. Ma, Z. Guo, Reversible data hiding based on prediction-error value ordering and multiple-embedding, Signal Process. 207 (2023) 108956.
[19] J. Fridrich, M. Goljan, R. Du, Lossless data embedding for all image formats, in: Proc. SPIE, Security Watermarking Multimedia Contents, SPIE, San Jose, CA, USA, 2002, pp. 572–583.
[20] K. Wang, Z.-M. Lu, Y.-J. Hu, A high capacity lossless data hiding scheme for jpeg images, J. Syst. Softw. 86 (7) (2013) 1965–1975.
[21] B.G. Mobasseri, R.J. Berger, M.P. Marcinak, Y.J. NaikRaikar, Data embedding in jpeg bitstream by code mapping, IEEE Trans. Image Process. 19 (4) (2010) 958–966.
[22] Z. Qian, X. Zhang, Lossless data hiding in jpeg bitstream, J. Syst. Softw. 85 (2) (2012) 309–313.
[23] Y. Hu, K. Wang, Z.-M. Lu, An improved VLC-based lossless data hiding scheme for jpeg images, J. Syst. Softw. 86 (8) (2013) 2166–2173.
[24] Y. Du, Z. Yin, X. Zhang, High capacity lossless data hiding in jpeg bitstream based on general VLC mapping, IEEE Trans. Dependable Secure Comput. 19 (2) (2020) 1420–1433.
[25] C.-C. Chang, C.-C. Lin, C.-S. Tseng, W.-L. Tai, Reversible hiding in DCT-based compressed images, Inform. Sci. 177 (13) (2007) 2768–2786.
[26] G. Xuan, Y.Q. Shi, Z. Ni, P. Chai, X. Cui, X. Tong, Reversible data hiding for jpeg images based on histogram pairs, in: Proc. Int. Conf. Image Anal. Recognit, Springer, Montreal, QC, Canada, 2007, pp. 715–727.
[27] F. Huang, X. Qu, H.J. Kim, J. Huang, Reversible data hiding in jpeg images, IEEE Trans. Circuits Syst. Video Technol. 26 (9) (2016) 1610–1621.
[28] D. Hou, H. Wang, W. Zhang, N. Yu, Reversible data hiding in jpeg image based on DCT frequency and block selection, Signal Process. 148 (2018) 41–47.
[29] J. He, J. Chen, S. Tang, Reversible data hiding in jpeg images based on negative influence models, IEEE Trans. Inf. Forensic Secur. 15 (2019) 2121–2133.
[30] Z. Yin, Y. Ji, B. Luo, Reversible data hiding in jpeg images with multi-objective optimization, IEEE Trans. Circuits Syst. Video Technol. 30 (8) (2020) 2343–2352.
[31] N. Li, F. Huang, Reversible data hiding for jpeg images based on pairwise nonzero ac coefficient expansion, Signal Process. 171 (2020) 107476.
[32] M. Xiao, X. Li, B. Ma, X. Zhang, Y. Zhao, Efficient reversible data hiding for jpeg images with multiple histograms modification, IEEE Trans. Circuits Syst. Video Technol. 31 (7) (2021) 2535–2546.
[33] J.-S. Tsai, W.-B. Huang, Y.-H. Kuo, On the selection of optimal feature region set for robust digital image watermarking, IEEE Trans. Image Process. 20 (3) (2011) 735–743.
[34] A. Anand, A.K. Singh, Dual watermarking for security of covid-19 patient record, IEEE Trans. Dependable Secure Comput. 20 (1) (2023) 859–866.
[35] J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: Hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 657–672.
[36] Z. Jia, H. Fang, W. Zhang, Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 41–49.
[37] J. Huang, T. Luo, L. Li, G. Yang, H. Xu, C.-C. Chang, Arwgan: Attention-guided robust image watermarking model based on gan, IEEE Trans. Instrum. Meas. 72 (2023) 1–17.
[38] C. De Vleeschouwer, J.-F. Delaigle, B. Macq, Circular interpretation of bijective transformations in lossless watermarking for media asset management, IEEE Trans. Multimed. 5 (1) (2003) 97–105.
[39] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, Robust lossless image data hiding, in: Proc. IEEE Int. Conf. Multimedia Expo, IEEE, Taipei, Taiwan, China, 2004, pp. 2199–2202.
[40] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, Robust lossless image data hiding designed for semi-fragile image authentication, IEEE Trans. Circuits Syst. Video Technol. 18 (4) (2008) 497–509.
[41] D. Zou, Y.Q. Shi, Z. Ni, W. Su, A semi-fragile lossless digital watermarking scheme based on integer wavelet transform, IEEE Trans. Circuits Syst. Video Technol. 16 (10) (2006) 1294–1300.
[42] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol. 21 (8) (2011) 1061–1070.
[43] X.-T. Zeng, L.-D. Ping, X.-Z. Pan, A lossless robust data hiding scheme, Pattern Recognit. 43 (4) (2010) 1656–1667.
[44] L. An, X. Gao, X. Li, D. Tao, C. Deng, J. Li, Robust reversible watermarking via clustering and enhanced pixel-wise masking, IEEE Trans. Image Process. 21 (8) (2012) 3598–3611.
[45] R. Thabit, B.E. Khoo, A new robust lossless data hiding scheme and its application to color medical images, Digit. Signal Process. 38 (2015) 77–94.
[46] R. Kumar, K.-H. Jung, Robust reversible data hiding scheme based on two-layer embedding strategy, Inform. Sci. 512 (2020) 96–107.
[47] X. Liang, S. Xiang, Robust reversible audio watermarking based on high-order difference statistics, Signal Process. 173 (2020) 107584.
[48] D. Coltuc, Towards distortion-free robust image authentication, in: Journal of Physics: Conference Series, vol. 77, IOP Publishing, 2007, 012005.
[49] X. Liu, J. Lou, H. Fang, Y. Chen, P. Ouyang, Y. Wang, B. Zou, L. Wang, A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images, IEEE Access 7 (2019) 76580–76598.
[50] X. Wang, X. Li, Q. Pei, Independent embedding domain based two-stage robust reversible watermarking, IEEE Trans. Circuits Syst. Video Technol. 30 (8) (2019) 2406–2417.
[51] R. Hu, S. Xiang, Cover-lossless robust image watermarking against geometric deformations, IEEE Trans. Image Process. 30 (2020) 318–331.
[52] R. Hu, S. Xiang, Lossless robust image watermarking by using polar harmonic transform, Signal Process. 179 (2021) 107833.
[53] Information technology – Digital compression and coding of continuous – Tone still images: Requirements and guidelines, standard ITU-T.81, 1992.
[54] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[55] Bossbase-1.01-hugo-alpha=0.4.tar.bz2, 2014, http://agents.fel.cvut.cz/stegodata/.
[56] Jpegsr9e.zip, 2022, http://www.ijg.org/.