Signal Processing 238 (2026) 110152

Contents lists available at ScienceDirect

Signal Processing

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

A Robust Reversible Watermarking scheme using DC prediction and histogram shifting

Jiancheng Xiao [a,1], Shuaichao Wu [a,1], Bingwen Feng [a,*], Jilian Zhang [a], Bing Chen [b], Zhihua Xia [a,d], Wei Lu [c]

[a] College of Cyber Security, Jinan University, Guangzhou 510632, China
[b] School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, 510665, China
[c] School of Computer Science and Engineering, Ministry of Education Key Laboratory of Information Technology, Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China
[d] Engineering Research Center of Trustworthy AI, Ministry of Education, Guangzhou 510632, China

ARTICLE INFO

Keywords:
Robust Reversible Watermarking
JPEG compression
Histogram shifting
Robust reversibility
Prediction error expansion

ABSTRACT

Robust Reversible Watermarking (RRW) not only ensures the resilience of watermarked images under various attacks but also enables the exact recovery of the original host images from these watermarked versions. However, many existing RRW methods suffer from compromised reversibility when subjected to attacks, preventing successful restoration of the host image. In this paper, we explore the dual robustness of RRW—simultaneously enhancing both watermark resilience and reversibility. We propose a JPEG compression-resistant histogram-shifting algorithm that withstands targeted compression and exhibits strong robustness against common image manipulations. Building on this algorithm, we introduce two RRW schemes: one embeds watermark bits into the AC coefficients, and the other embeds them into the prediction error of DC coefficients. Furthermore, we design a convolutional neural network (CNN)-based DC predictor to infer DC coefficients from AC coefficients. Experimental results demonstrate that our approach achieves superior robustness and watermarked image quality, while reliably preserving reversibility under various distortions.

1. Introduction

The rapid advancement of high-quality multimedia technologies has greatly facilitated the creation and dissemination of information. Simultaneously, the protection of multimedia copyrights has become paramount. Reversible watermarking (RW) invisibly embeds data within host multimedia content to secure copyright, while uniquely allowing for the complete restoration of the original host data without any loss. This feature renders RW an ideal solution for safeguarding high-fidelity multimedia, where even minimal alterations to the host signal are unacceptable.

Digital images account for a substantial share of multimedia content; therefore, this paper targets RW techniques specifically crafted for images. RW methods are broadly divided into spatial-domain and transform-domain categories. Spatial-domain approaches harness pixel redundancy to carve out space for watermark bits—either by expanding inter-pixel differences [1–3] or by manipulating histograms of pixel prediction errors [4–6]. However, most digital images today are stored in compressed formats, particularly JPEG, rendering spatial-domain schemes sensitive to compression artifacts and underscoring the necessity of transform-domain RW [7–12]. To adapt to JPEG, Chen et al. [7] introduced a novel distortion metric based on the spatial-domain response to DCT (Discrete Cosine Transform) coefficient modifications; Xiao et al. [8] refined two-dimensional histograms by leveraging DCT's decorrelation property; Weng et al. [9] enhanced block-smoothness estimation by combining zero-valued AC coefficient counts with embedding distortion; and Yin et al. [10] embedded watermark bits into zero coefficients to optimize embedding cost and alleviate distortions from high-value coefficient shifts. Beyond the DCT domain, integer wavelet-based methods further mitigate embedding distortions [11,12]. Although these transform-domain techniques resist standard compression during storage and transmission, the embedded watermark bits remain vulnerable to distortions introduced by transmission channels or malicious attacks [13].

Robust Reversible Watermarking (RRW) techniques ensure that embedded watermark bits persist through various attacks while maintaining complete recoverability. A prominent category of RRW employs

* Corresponding author.
E-mail address: bingwenfeng@jnu.edu.cn (B. Feng).
[1] The two authors contribute equally to this work.

https://doi.org/10.1016/j.sigpro.2025.110152
Received 21 December 2024; Received in revised form 19 May 2025; Accepted 3 June 2025
Available online 21 June 2025

histogram shifting. Vleeschouwer et al. [14] first introduced the concept by rotating the circular histogram of pixel groups to embed watermark bits, demonstrating resilience against JPEG compression. Ni et al. [15] refined this approach by eliminating salt-and-pepper noise and improving robustness against both JPEG and JPEG2000 compression. Gao et al. [16] proposed a generalized statistical-quantity histogram that leverages histogram similarity and sparsity to achieve conditional robustness to JPEG compression under suitable scaling. Huang et al. [17] shifted the histogram of linear correlation values — commonly used in spread-spectrum coding — for watermark embedding. Despite these advances, spatial-domain methods still face degradation when JPEG saving is applied, threatening both watermark integrity and host image recovery. Liang et al. [18] addressed this by constructing robust features directly within the JPEG domain and embedding watermark bits via their histogram, thereby bypassing additional JPEG compression during saving; however, their reversibility remains vulnerable to distortions. Another class of RRW employs two-stage embedding strategies [12,19–23], which balance enhanced robustness with reversible reconstruction. In the second stage of these schemes, errors introduced by robust watermarking are embedded as auxiliary information to facilitate recovery of the host image. These methods achieve high robustness against various distortions, including geometric attacks. However, reversibility is easily compromised when the embedded auxiliary information is lost due to external distortions.

The reversibility of RW fundamentally requires the ability to remove embedding distortions from the watermarked image. Let $\mathbf{X}$ denote the original host image and $\mathbf{N}_w$ the embedding distortion. Then the watermarked image can be expressed as $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{N}_w$. After transmission through a noisy channel, the received image becomes $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{N}_w + \mathbf{N}_c$, where $\mathbf{N}_c$ represents channel noise. If reversibility is preserved, one can recover $\mathbf{X}' = \mathbf{X} + \mathbf{N}_c$. In other words, an ideal RRW scheme should provide dual robustness: it must ensure both the integrity of the watermark and reversibility. This dual robustness offers several benefits. First, embedding distortions are generally more significant than external distortions, so their removal greatly improves image quality, which is crucial for applications requiring high visual fidelity. Second, certain external distortions, such as JPEG compression, are inherent to image coding. Thanks to this robustness, the RRW algorithm maintains image quality even after recompression. Various techniques have been proposed to enhance steganography's resistance to JPEG compression [24–28]. Tao et al. [25] adjusted coefficients so that the channel-compressed version of the intermediate image matches the stego image exactly. Zhang et al. [26] devised a robustness model in the spatial domain derived from DCT coefficients and proposed a cost function quantifying the disparity between spatial pixels reconstructed from modified DCT coefficients and those adjusted by the model. Butora et al. [27] introduced an error-free robust JPEG steganography method based on the output of the target JPEG encoder. Huang et al. [28] proposed a DCT residual modulation algorithm to mitigate residual DCT coefficients generated during compression. Motivated by these approaches, we investigate the stability of the histogram shifting algorithm to achieve robustness against JPEG compression.

This paper presents a JPEG recompression-resistant RRW framework. First, we introduce a histogram-shifting algorithm tailored to the JPEG quantization step, which achieves error-free robustness against JPEG compression. Based on this algorithm, two schemes are developed: the first embeds watermark bits into AC coefficients, while the second embeds them into the prediction error of DC coefficients. Additionally, we design a convolutional neural network (CNN)-based DC predictor to accurately estimate DC coefficients from AC components, even under noisy conditions. Experimental results demonstrate that our framework effectively withstands target JPEG compression and common attacks, offering dual robustness that further enhances reversibility. The main contributions of this work are as follows.

- We propose a histogram-shifting algorithm that remains error-free under targeted JPEG compression and exhibits strong robustness to common image manipulations.
- We develop a CNN-based DC predictor to estimate DC coefficients from AC components, achieving an improved balance between reversibility and watermark transparency.
- We establish two reversible and robust watermarking schemes: one leveraging AC coefficients for high embedding capacity, and the other leveraging DC coefficients for enhanced robustness. Experimental evaluations confirm their superior performance in watermark extraction and cover image recovery.

The rest of the paper is structured as follows. Section 2 details the embedding and extraction processes of our histogram-shifting algorithm. Building upon this, Section 3 describes the reversible watermarking scheme for AC coefficients, and Section 4 presents the corresponding scheme for DC coefficients. Section 5 discusses parameter selection and reports experimental results on robustness and reversibility. Finally, Section 6 concludes the paper.

2. JPEG compression resilient histogram shifting algorithm (JCRHS)

JPEG compression reduces the storage size of an image by quantizing its DCT coefficients. This operation incurs quantization distortion, potentially hindering watermark extraction. Given a quality factor $\overline{\mathrm{QF}} \in [1, 100]$, the DCT coefficient $a_i[j,k]$ at position (j,k) within the ith DCT block undergoes quantization as

$$Q(a_i[j,k]) \triangleq \left\lfloor \frac{a_i[j,k]}{q^{\overline{\mathrm{QF}}}[j,k]} \right\rfloor \times q^{\overline{\mathrm{QF}}}[j,k] \tag{1}$$

where $q^{\overline{\mathrm{QF}}}[j,k]$ denotes the quantization step for the DCT coefficient at position (j,k) at quality factor $\overline{\mathrm{QF}}$. Noting the rounding operation in Eq. (1), a common practical challenge in JPEG quantization-resist watermarking is how to minimize rounding errors.

Suppose the watermark embedding is achieved by $b_i[j,k] = w[j,k] + a_i[j,k]$ where w represents the watermarking amplitude. Ideally, to ensure that watermark extraction is unaffected by JPEG compression, $w[j,k]$ should be a multiple of $q^{\overline{\mathrm{QF}}}[j,k]$, allowing

$$Q(b_i[j,k]) = Q(w[j,k]) + Q(a_i[j,k]) \tag{2}$$

Under these conditions, the watermark remains intact during JPEG recompression. Inspired by this observation, we propose a reversible watermarking algorithm based on histogram shifting that is resilient to JPEG compression. The embedding and extraction procedures are outlined below

2.1. Watermark embedding

Let the DCT coefficient vector $\mathbf{a}[j,k]$ consist of $a_i[j,k]$ from all DCT blocks, and for brevity, we denote this vector simply as $\mathbf{a}$. The corresponding quantization factor is concisely denoted as q. We proceed to compute the histogram of $\mathbf{a}$, represented by $\mathbf{h}$. It is noteworthy that DCT coefficients are often non-integral, thus allowing for decimal indices in the histogram bins.

The embedding process commences by displacing histogram bins indexed by i within the intervals $(-\infty, -(\beta - 1/2)q]$ and $[(\beta + 1/2)q, \infty)$ away from the origin by a distance of $2\iota q$, where ι and β are parameters that govern the shape of the shifted histogram. In particular, β determines the histogram bins employed for embedding watermark bits, whereas ι regulates their shifting distance. This strategy creates two intervals, each of length $2\iota q$, which serve as vacant intervals for watermark bits.

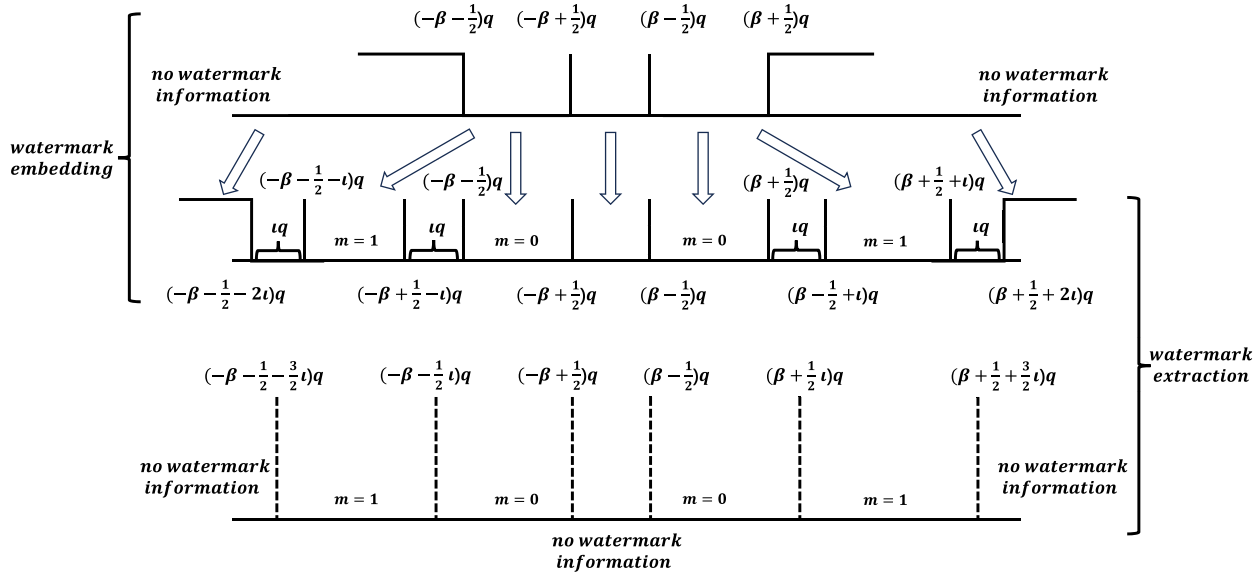


Fig. 1. The demonstration of embedding and extraction algorithm.

Subsequently, watermark bits are embedded into coefficients belonging to histogram bins indexed within $\left((-\beta-\frac{1}{2})q,(-\beta+\frac{1}{2})q\right] \cup \left[(\beta-\frac{1}{2})q,(\beta+\frac{1}{2})q\right)$. This embedding process can be formulated as

$$b=\begin{cases} a-2\iota q, & \text{if } a\le(-\beta-\frac{1}{2})q;\\ a-\iota q, & \text{if } (-\beta-\frac{1}{2})q<a\le(-\beta+\frac{1}{2})q \text{ and } m=1;\\ a+2\iota q, & \text{if } a\ge(\beta+\frac{1}{2})q;\\ a+\iota q, & \text{if } (\beta+\frac{1}{2})q>a\ge(\beta-\frac{1}{2})q \text{ and } m=1;\\ a, & \text{otherwise.}\end{cases} \tag{3}$$

where, a denotes a coefficient from $\mathbf{a}$, and m denotes the message bit intended for embedding. The total number of message bits that can be embedded into $\mathbf{a}$ is given by the sum of histogram counts within the intervals $\left((-\beta-\frac{1}{2})q,(-\beta+\frac{1}{2})q\right]$ and $\left[(\beta-\frac{1}{2})q,(\beta+\frac{1}{2})q\right)$. The embedding algorithm is demonstrated in Fig. 1.

2.2. Watermark extraction and host coefficient recovery

Using the proposed embedding algorithm, the watermarked coefficients in $\mathbf{b}$ satisfy

$$b\in\begin{cases} \left(-\infty,-(2\iota+\beta+\frac{1}{2})q\right] & \text{if } a\le(-\beta-\frac{1}{2})q\\ \left(-(\iota+\beta+\frac{1}{2})q,-(\iota+\beta-\frac{1}{2})q\right] & \\ \quad \text{if } (-\beta-\frac{1}{2})q<a\le(-\beta+\frac{1}{2})q \text{ and } m=1 & \\ \left(-(\beta+\frac{1}{2})q,-(\beta-\frac{1}{2})q\right] & \\ \quad \text{if } (-\beta-\frac{1}{2})q<a\le(-\beta+\frac{1}{2})q \text{ and } m=0 & \\ \left(-(\beta-\frac{1}{2})q,(\beta-\frac{1}{2})q\right) & \\ \quad \text{if } (-\beta+\frac{1}{2})q<a<(\beta-\frac{1}{2})q & \\ \left[(\beta-\frac{1}{2})q,(\beta+\frac{1}{2})q\right) & \\ \quad \text{if } (\beta+\frac{1}{2})q>a\ge(\beta-\frac{1}{2})q \text{ and } m=0 & \\ \left[(\iota+\beta-\frac{1}{2})q,(\iota+\beta+\frac{1}{2})q\right) & \\ \quad \text{if } (\beta+\frac{1}{2})q>a\ge(\beta-\frac{1}{2})q \text{ and } m=1 & \\ \left[(2\iota+\beta+\frac{1}{2})q,\infty\right) & \text{if } a\ge(\beta+\frac{1}{2})q \end{cases} \tag{4}$$

It is evident that the coefficients bearing watermark bit 1 have values that are separated by a distance of ι from their counterparts. This security buffer ensures resilience against alien distortions. In the absence of such distortions, the receiver can effortlessly extract the watermark bits from the received coefficients $\tilde{\mathbf{b}}$ and subsequently restore the host coefficients by inverting the process outlined in Eq. (3).

When JPEG compression is applied with the specified quantity factor $\overline{\mathrm{QF}}$, the quantized coefficients subsequent to watermark embedding adhere to the following property

$$a'=\left\lfloor\frac{b}{q}\right\rfloor\times q=\left\lfloor\frac{a+\omega}{q}\right\rfloor\times q=\left\lfloor\frac{a}{q}\right\rfloor\times q+\omega \tag{5}$$

This signifies that the quantization process does not impinge upon the watermark signal ω, as it is inherently a multiple of q. Consequently, the embedded watermark bits can be accurately extracted, and the original host coefficients can be flawlessly restored.

Under other types of external distortions, the watermarked coefficients may deviate from their original values. By identifying the margin ι that separates coefficients carrying a watermark bit of 1 from the rest, we can use the midpoint of this margin as a decision threshold, as depicted in the last line of Fig. 1. Consequently, watermark extraction and host coefficient restoration are performed as follows

$$\begin{cases} a'=\tilde{b}+2\iota q & \text{if } \tilde{b}\le-(\frac{3}{2}\iota+\beta+\frac{1}{2})q\\ m'=1, a'=\tilde{b}+\iota q & \text{if } -(\frac{3}{2}\iota+\beta+\frac{1}{2})q\\ & \quad<\tilde{b}\le-(\frac{1}{2}\iota+\beta-\frac{1}{2})q\\ m'=0, a'=\tilde{b} & \text{if } -(\frac{1}{2}\iota+\beta+\frac{1}{2})q\\ & \quad<\tilde{b}\le-(\beta-\frac{1}{2})q\\ a'=\tilde{b} & \text{if } -(\beta-\frac{1}{2})q\\ & \quad<\tilde{b}<(\beta-\frac{1}{2})q\\ m'=0, a'=\tilde{b} & \text{if } (\beta-\frac{1}{2})q\\ & \quad\le\tilde{b}<(\frac{1}{2}\iota+\beta+\frac{1}{2})q\\ m'=1, a'=\tilde{b}-\iota q & \text{if } (\frac{1}{2}\iota+\beta-\frac{1}{2})q\\ & \quad\le\tilde{b}<(\frac{3}{2}\iota+\beta+\frac{1}{2})q\\ a'=\tilde{b}-2\iota q & \text{if } (\frac{3}{2}\iota+\beta+\frac{1}{2})q\le\tilde{b} \end{cases} \tag{6}$$

It should be noted that coefficients carrying the watermark bit 0 are adjacent to those not utilized for watermarking. Alien distortions can easily shift these to the incorrect side, thereby increasing their vulnerability to erroneous extraction compared to those carrying the watermark bit 1. However, when restoring the host coefficients, these do not require modification, ensuring that error in extracting watermark bit 0 does not compromise the quality of the restored image. Moreover, we can predetermine the count of 0s in the watermark

sequence, denoted as N_0, which aids in extracting watermark bit 0. During the extraction phase, we select N_0 coefficients residing within the intervals $\left(-(\frac{1}{2}\iota+\beta+\frac{1}{2})q, -(\beta-\frac{1}{2})q\right] \cup \left[(\beta-\frac{1}{2})q, (\frac{1}{2}\iota+\beta+\frac{1}{2})q\right)$ and having the largest distance from the origin, assuming they were originally intended for embedding watermark bit 0. Notably, N_0 does not need to be transmitted to the receiver, as it can be preset as half the length of the watermark sequence, given the similar occurrence of 0s and 1s in an encrypted message sequence. Additionally, the message sequence can be truncated or appended to ensure the count of 0s matches N_0.

3. Scheme I: Robust revisable watermarking on DC coefficients (RRW-AC) base on JCRHS

Our scheme embeds watermark bits in the DCT domain. In standard JPEG compression, an 8×8 block DCT transforms each spatial image block into 63 AC coefficients and one DC coefficient. Leveraging the distinct characteristics of AC and DC coefficients, we develop two watermarking methods based on our histogram-shifting algorithm: one targets AC coefficients, and the other focuses on the DC coefficient.

For the AC-based scheme, we group coefficients at the same position across all DCT blocks into an AC vector, which carries a single watermark sequence. Although each block provides 63 potential vectors — offering correspondingly high embedding capacity — we prioritize image quality and observe that high-frequency coefficients are sparse. Consequently, we selectively embed watermarks only into the (1,2) and (1,3) positions of each DCT block.

3.1. Watermarking embedding procedure

The scheme is executed on a grayscale JPEG image. In the case of a color JPEG image, solely the luminance component is employed. Let $\mathbf{X}$, $\mathbf{Y}$, and $\tilde{\mathbf{Y}}$ denote the host, watermarked, and received images, respectively. Furthermore, we designate the watermark bits to be embedded as $\mathbf{m} \in \{0,1\}^{l_m}$. In the context of DCT processing, we denote the DCT coefficients in the ith DCT block as $\{a_i[1,1], a_i[1,2], a_i[1,3], \ldots, a_i[8,8]\}$ where $a_i[1,1]$ signifies the DC coefficient, and $a_i[1,2], a_i[1,3], \ldots, a_i[8,8]$ encompass the AC coefficients. The embedding procedure consists of the following steps.

1. Entropy decode $\mathbf{X}$ to obtain all its DCT blocks. Subsequently, reorganize the $(1,2)$-th AC coefficients within these blocks to form the AC vector $\mathbf{a}_{(1,2)} = \{a_1[1,2], a_2[1,2], \ldots\}$.
2. Employ the embedding algorithm outlined in Section 2.1 to embed half of the watermark $\mathbf{m}$ into $\mathbf{a}_{(1,2)}$, yielding watermarked vector $\mathbf{b}_{(1,2)}$. Subsequently, put each coefficient in $\mathbf{b}_{(1,2)}$ to its original position in the DCT blocks.
3. Repeat the process for the $(1,3)$-th AC coefficients by reorganizing them across all DCT blocks to form the AC vector $\mathbf{a}_{(1,3)}$. Then, repeat the embedding procedure from Step 2 to embed the remaining half of $\mathbf{m}$ into $\mathbf{a}_{(1,3)}$ to obtained $\mathbf{b}_{(1,3)}$.
4. Finally, entropy encode all the modified DCT blocks to generate the watermarked image $\mathbf{Y}$.

3.2. Watermarking extraction and host image restoration procedure

Upon receiving the image $\tilde{\mathbf{Y}}$, the receiver proceeds to decode the DCT blocks from it. Subsequently, he can extract the watermark sequence from the coefficients within these blocks, and restore the original host coefficients. The extraction procedure consists of the following steps.

1. Entropy decode $\tilde{\mathbf{Y}}$ to obtain all of its DCT blocks, and then construct the AC vector $\tilde{\mathbf{b}}_{(1,2)}$.
2. Utilize the extraction and restoration algorithm detailed in Section 2.2 to extract half of the watermark $\mathbf{m}'$ from $\tilde{\mathbf{b}}_{(1,2)}$, and restore the AC vector, resulting in $\mathbf{a}'_{(1,2)}$. Afterwards, reposition each coefficient in $\mathbf{a}'_{(1,2)}$ within its original location within the DCT blocks.
3. Form AC vector $\tilde{\mathbf{b}}_{(1,3)}$, and repeat the procedure from Step 2 to extract the remaining half of $\mathbf{m}'$ and restore $\mathbf{a}'_{(1,3)}$.
4. Conclusively, entropy encode all the modified DCT blocks to produce the restored image $\mathbf{X}'$.

4. Scheme II: Robust revisable watermarking on DC coefficients (RRW-DC) base on JCRHS

The distribution of DC coefficients varies significantly across different images, making it difficult to select universally optimal embedding parameters. To address this, we embed message bits into the prediction error of DC coefficients rather than the coefficients themselves. Specifically, we develop a DC predictor to estimate all DC coefficient values within an image. When this predictor performs well, its errors closely follow a zero-centered Gaussian distribution, which enhances the performance of our watermarking algorithm. This section first describes the chosen DC predictor and then provides a detailed explanation of the embedding, extraction, and restoration procedures.

4.1. DC predictor

We feed the AC coefficients of an image into our predictor, whose objective is to estimate the corresponding DC coefficients. The predictor adopts a U-Net architecture [29] comprising five downsampling and five upsampling modules, as illustrated in Fig. 2. The input image $\mathbf{X}$ undergoes an 8×8 block DCT to separate AC and DC components. The extracted AC coefficients are then passed to the network to predict the DC coefficients. To provide adversarial supervision, we incorporate a PatchGAN-based discriminator [30].

To improve robustness under noisy transmission, we augment the AC coefficients during training with Gaussian noise (mean $\mu = 0$, standard deviation $\sigma = 0.1$) and JPEG compression artifacts (quality factor uniformly sampled from 50 to 100). This augmentation enables the predictor to maintain high accuracy in DC coefficient estimation, even in the presence of noise and distortions.

The overall loss of the generator is a balanced composition of prediction loss, image loss, and adversarial loss, formulated as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{pred} + \lambda_2 \mathcal{L}_{img} + \lambda_3 \mathcal{L}_{adv}. \tag{7}$$

where λ_1, λ_2, and λ_3 are hyperparameters that regulate the relative importance of each loss component.

The prediction loss $\mathcal{L}_{pred}$ is defined as the L2 distance between the predicted DC coefficients $\mathbf{a}^p_{(1,1)}$ and their genuine values $\mathbf{a}_{(1,1)}$.

$$\mathcal{L}_{pred} = \|\mathbf{a}^p_{(1,1)} - \mathbf{a}_{(1,1)}\|_2. \tag{8}$$

The adversarial loss $\mathcal{L}_{adv}$ mirrors the formulation presented in [30], fostering a competitive learning environment between our generator and discriminator.

Moreover, we introduce an image loss $\mathcal{L}_{img}$ to uphold the coherence between the predicted DC coefficients and its corresponding AC coefficients. This loss metric measures the L2 distance between the original image $\mathbf{X}$ and the reconstructed image derived from the combination of predicted DC coefficients $\mathbf{a}^p_{(1,1)}$ and original AC coefficients $\{\mathbf{a}_{(1,2)}, \ldots, \mathbf{a}_{(8,8)}\}$, formed as

$$\mathcal{L}_{img} = \|IDCT(\{\mathbf{a}^p_{(1,1)}, \mathbf{a}_{(1,2)}, \ldots, \mathbf{a}_{(8,8)}\}) - \mathbf{X}\|_2. \tag{9}$$

This loss aids in the reconstruction of high-fidelity images by ensuring consistency between the predicted DC coefficients and the existing AC coefficients.

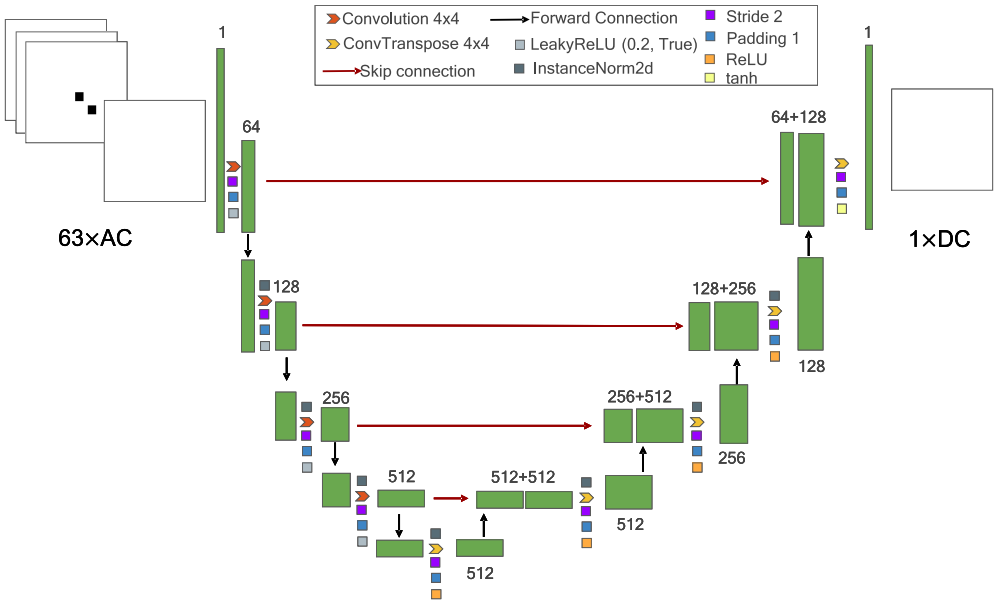


Fig. 2. The architecture of DC predictor.

4.2. Watermark embedding procedure

With the predicted DC coefficients $\mathbf{a}^{p}_{(1,1)}$, the prediction error $\mathbf{e}$ can be derived. We propose a robust reversible watermarking algorithm based on prediction error expansion to embed watermark bits into $\mathbf{a}_{(1,1)}$. It consists of the following steps.

1. Entropy decode $\mathbf{X}$ to obtain all its DCT blocks. Subsequently, reorganize these blocks to form a DC vector $\mathbf{a}_{(1,1)} = \{a_1[1,1], a_2[1,1], \dots, \dots\}$, and 63 AC vectors $\mathbf{a}_{(1,2)}, \dots, \mathbf{a}_{(8,8)}$.
2. Utilize the DC predictor to predict the DC vector $\mathbf{a}^{p}_{(1,1)}$ based on AC vectors $\mathbf{a}_{(1,2)}, \dots, \mathbf{a}_{(8,8)}$. Then, calculate the prediction error, $\mathbf{e}$, by subtracting the predicted vector from the actual DC vector: $\mathbf{e} = \mathbf{a}_{(1,1)} - \mathbf{a}^{p}_{(1,1)}$.
3. Implement the embedding algorithm detailed in Section 2.1 to embed the watermark $\mathbf{m}$ into the prediction error $\mathbf{e}$, yielding the watermarked prediction error $\mathbf{e}_w$.
4. Reconstruct the watermarked DC vector by $\mathbf{b}_{(1,1)} = \mathbf{a}^{p}_{(1,1)} + \mathbf{e}_w$. Subsequently, reposition each coefficient in $\mathbf{b}_{(1,1)}$ back to its original position within the respective DCT blocks.
5. Finally, entropy encode all the modified DCT blocks to generate the watermarked image $\mathbf{Y}$.

4.3. Watermarking extraction and host image restoration procedure

At the receiver, both DC and AC coefficients of the received image $\tilde{\mathbf{Y}}$ may be corrupted by transmission distortions. If $\tilde{\mathbf{Y}}$ undergoes JPEG compression with the specified quality factor $\overline{\mathrm{QF}}$, our DC predictor and the extraction-restoration algorithm remain invariant to this process. Under other distortions, accumulated prediction errors compound with DC-vector distortions, which can degrade our method's effectiveness. Nevertheless, the robustly trained DC predictor still generates a reasonably accurate DC vector under noisy conditions, mitigating overall performance loss. The watermark extraction and host-image restoration procedure is outlined below.

1. Entropy decode $\tilde{\mathbf{Y}}$ to obtain all of its DCT blocks, then construct the DC vector $\tilde{\mathbf{b}}_{(1,1)}$, and the AC vectors $\tilde{\mathbf{a}}_{(1,2)}, \dots, \tilde{\mathbf{a}}_{(8,8)}$.
2. Utilize $\tilde{\mathbf{a}}_{(1,2)}, \dots, \tilde{\mathbf{a}}_{(8,8)}$ to predict the DC vector $\tilde{\mathbf{a}}^{P}_{(1,1)}$. Subsequently, calculate the prediction error as $\tilde{\mathbf{e}} = \tilde{\mathbf{b}}_{(1,1)} - \tilde{\mathbf{a}}^{P}_{(1,1)}$.
3. Apply the extraction and restoration algorithm detailed in Section 2.2 to extract watermark $\mathbf{m}'$, and restore the prediction error $\mathbf{e}'$.
4. Restore the DC vector by $\mathbf{a}'_{(1,1)} = \tilde{\mathbf{a}}^{P}_{(1,1)} + \mathbf{e}'$. Following this, reposition each coefficient in $\mathbf{a}'_{(1,1)}$ back to its original position within the corresponding DCT blocks.
5. Finally, entropy encode all the DCT blocks to generate the restored image $\mathbf{X}'$.

5. Experiments

Several experiments were conducted to evaluate the performance of our scheme. For a comprehensive comparison, we benchmarked our approach against state-of-the-art RRW methods, including those by Huang et al. [17], X. Wang et al. [19], H. Wang et al. [31], Chen et al. [23], and Yang et al. [32]. Huang et al. [17] employ adaptive spread-spectrum coding to achieve attack resilience, enabling lossless recovery in attack-free scenarios and partial recovery under distortions. X. Wang et al. [19] decouple robust and reversible watermark embedding into separate transform domains for independent optimization. H. Wang et al. [31] embed watermark bits via histogram shifting within image blocks, augmented by a high-pass filter and a blind extraction strategy that jointly extracts data and restores the image. The deep-learning approach in Chen et al. [23] establishes an invertible mapping between cover and stego images using a robust integer network architecture. Distinctly, Yang et al. [32] introduce a CNN-based estimator that prioritizes AC coefficients with higher embedding efficiency for data hiding in the JPEG-compressed domain. Notably, except for Yang et al.'s domain-specific design [32], all the compared methods demonstrate exceptional robustness to common signal processing attacks, such as JPEG compression and AWGN.

5.1. Experiment setting

We evaluate our proposed scheme using the Bossbase dataset [33] and the COCO dataset [34]. Bossbase comprises 10,000 grayscale images, each of size 512 × 512 pixels, captured in RAW format by seven distinct cameras. COCO, a large-scale dataset of 330,000 color images, is also employed; however, since our experiments focus exclusively on grayscale inputs, all COCO images are first converted to luminance values using the standardized ITU-R BT.709 equation.

$$Y = 0.299R + 0.587G + 0.114B$$

Table 1
Robustness (1-BER) and reversibility (PSNR($\tilde{\mathbf{X}}, \tilde{\mathbf{X}}'$)) evaluation of RRW-AC using different parameters.

Para.		QF=100		QF=80		QF=60		QF=50		QF=40	
ι	β	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR
1	1	0	NaN	0.1107	53.0169	0.5585	44.9863	0	NaN	0.1252	39.3621
	2	0	NaN	0.1318	56.4229	0.4901	52.1481	0	NaN	0.3300	45.9660
	3	0	NaN	0.0850	59.8661	0.3284	54.9148	0	NaN	0.4276	53.7494
	4	0	NaN	0.0885	58.0450	0.0758	53.2004	0	NaN	0.3385	50.5786
	5	0	NaN	0.0281	58.7336	0.0443	53.7612	0	NaN	0.0087	49.0158
2	1	0	NaN	0.1108	53.9254	0.5585	45.4679	0	NaN	0.1264	45.1995
	2	0	NaN	0.1320	53.8721	0.4901	49.7846	0	NaN	0.3308	46.4387
	3	0	NaN	0.0846	57.4428	0.3271	50.4195	0	NaN	0.4278	47.1854
	4	0	NaN	0.0884	59.6021	0.0717	53.6810	0	NaN	0.3359	50.0367
	5	0	NaN	0.0281	58.6485	0.0443	56.8667	0	NaN	0.0087	53.6223
3	1	0	NaN	0.1105	51.8620	0.5585	46.8749	0	NaN	0.1280	41.5765
	2	0	NaN	0.1322	57.4037	0.4901	49.6348	0	NaN	0.3278	45.4446
	3	0	NaN	0.0851	56.6534	0.3291	52.9390	0	NaN	0.4277	51.4012
	4	0	NaN	0.0897	59.6507	0.0760	54.4319	0	NaN	0.3359	52.1372
	5	0	NaN	0.0281	59.4569	0.0443	54.6893	0	NaN	0.0087	51.0524
4	1	0	NaN	0.1088	52.4807	0.5585	47.6499	0	NaN	0.1265	45.4409
	2	0	NaN	0.1317	55.4261	0.4901	50.7940	0	NaN	0.3287	49.6246
	3	0	NaN	0.0852	56.8437	0.3287	51.8129	0	NaN	0.4276	50.2943
	4	0	NaN	0.0887	58.1740	0.0739	52.5263	0	NaN	0.3387	49.1489
	5	0	NaN	0.0281	58.6992	0.0443	55.4208	0	NaN	0.0087	53.2645

where (R, G, B) denote the color channels of an image. To ensure uniform dimensions, images are cropped to 512×512 pixels when necessary and then converted to JPEG using MATLAB's *imwrite* function. We examine JPEG-compressed channels with quality factors of $QF = 50$ and $QF = 100$. Accordingly, test images are stored at $QF = 100$, while our scheme is specifically optimized for $QF = 50$. It should be emphasized that this quality factor is illustrative—our method can accommodate other values, and parameters can be fine-tuned once channel characteristics are known.

For each experiment, we generate random binary messages of length $l_m = 1024$ for both the RRW-AC and RRW-DC schemes. To quantify the quality of watermarked and recovered images, we employ Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [35]. For two images $\mathbf{X}$ and $\mathbf{Y}$ of size $l_h \times l_w$, PSNR is defined as

$$\mathrm{PSNR}(\mathbf{X}, \mathbf{Y}) = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(\mathbf{X}, \mathbf{Y})} \tag{10}$$

$$\mathrm{MSE}(\mathbf{X}, \mathbf{Y}) = \frac{1}{l_h \times l_w} \sum_{i=1}^{l_h} \sum_{j=1}^{l_w} (\mathbf{X}[i, j] - \mathbf{Y}[i, j])^2 \tag{11}$$

while SSIM is computed as

$$\mathrm{SSIM}(\mathbf{X}, \mathbf{Y}) = \frac{(2\mu_X \mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{12}$$

where μ_X and μ_Y are the means of $\mathbf{X}$ and $\mathbf{Y}$, respectively. σ_X^2 and σ_Y^2 are their corresponding variances, and σ_{XY} is the covariance between $\mathbf{X}$ and $\mathbf{Y}$. C_1 and C_2 are constants to stabilize the division in cases of weak denominators. It should be note that the experiments also assess the quality of watermarked and recovered images in the presence of attacks. To illustrate, take PSNR as an example. We compute PSNR($\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}$) and PSNR($\tilde{\mathbf{X}}, \tilde{\mathbf{X}}'$), where $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ represent the distorted versions of $\mathbf{X}$ and $\mathbf{Y}$, respectively, and $\tilde{\mathbf{X}}'$ denotes the image recovered from $\tilde{\mathbf{Y}}$.

Furthermore, we employ the 1-Bit Error Ratio (1-BER) to quantify the accuracy of watermark extraction, defined as.

$$\text{1-BER} = \left(1 - \frac{\sum_{i=1}^{l_m} (\mathbf{m}[i] - \mathbf{m}'[i])^2}{l_m}\right) \times 100\% \tag{13}$$

where $\mathbf{m}'$ represents the extracted watermark bits.

In Section 4.1, the training of the DC predictor involves selecting 30,000 images from the COCO dataset, while a random sample of 100 images from the remaining dataset is utilized for testing. Both the generator and discriminator in our model are optimized using the Adam optimizer, with parameters set to $\beta_1 = 0.5$, $\beta_2 = 0.999$, a learning rate of $r = 1 \times 10^{-4}$, and a batch size of 30. Moreover, the hyperparameters in Eq. (7) are empirically set as $\lambda_1 = 10$, $\lambda_2 = 300$, and $\lambda_3 = 0.2$.

5.2. Evaluation of parameter setting

5.2.1. Parameter setting in RRW-AC

First, we discuss the selection of β and ι in the RRW-AC scheme. Here, β specifies the histogram bins used for watermark embedding, while ι controls the magnitude of the bin shifts. To investigate the effect of β, we vary it over the set $\{1, 2, 3, 4, 5\}$ with ι held constant, and evaluate its impact on embedding capacity, image quality, and robustness using 100 images randomly sampled from the COCO dataset.

Fig. 3(a) plots the average embedding capacity as a function of β, and Fig. 3(b) reports the corresponding PSNR of the watermarked images. As β increases, capacity decreases but PSNR improves, since bins at the histogram tails contain fewer coefficients. Even in the worst case, our scheme supports over 1,500 bits of payload. Table 1 summarizes robustness and reversibility under JPEG compression at QF = 50 and QF = 100, showing that $\beta = 5$ yields the highest resistance, likely due to the larger coefficient magnitudes being less susceptible to compression.

We similarly examine the effect of varying ι in Fig. 3 and Table 1. While increasing ι enhances robustness, it degrades PSNR, and the marginal gain in resilience does not justify the loss in image quality. Therefore, we adopt $\iota = 1$ and $\beta = 5$ in our final configuration.

5.2.2. Parameter setting in RRW-DC

Analogous experiments were conducted for the RRW-DC scheme to optimize parameters β and ι. Figs. 4(a) and 4(b) depict the embedding capacity and PSNR of watermarked images, respectively, as functions of β. The results indicate that capacity depends solely on β, and in comparison to RRW-AC, RRW-DC's capacity declines more rapidly—likely due to its coefficient distribution being more tightly clustered around the mean. Although increasing ι offers marginal robustness gains, it significantly degrades image quality.

Table 2 summarizes robustness and reversibility under JPEG compression at various quantization factors, demonstrating that RRW-DC outperforms RRW-AC in robustness. To achieve a capacity comparable to RRW-AC, we set $\iota = 1$ and $\beta = 1$ for the RRW-DC configuration.

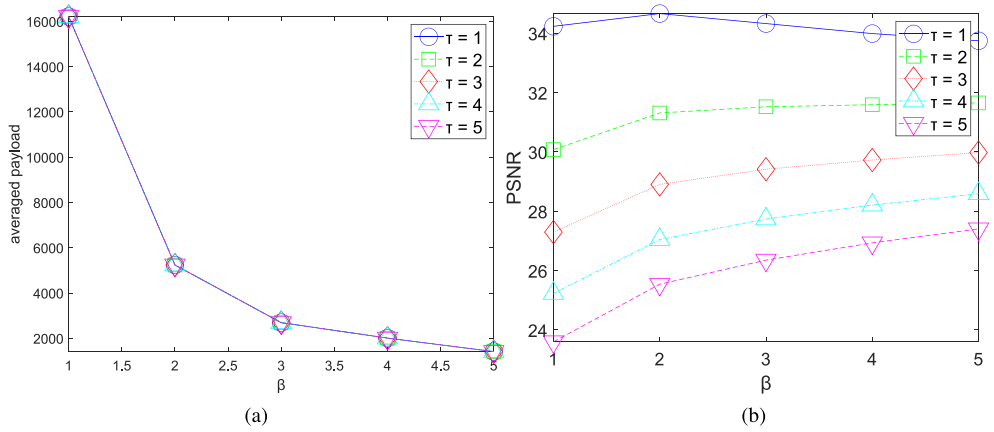


Fig. 3. The influence of changing β and ι in RRW-AC. (a) shows the embedding capacity and (b) shows the watermarked image quality.

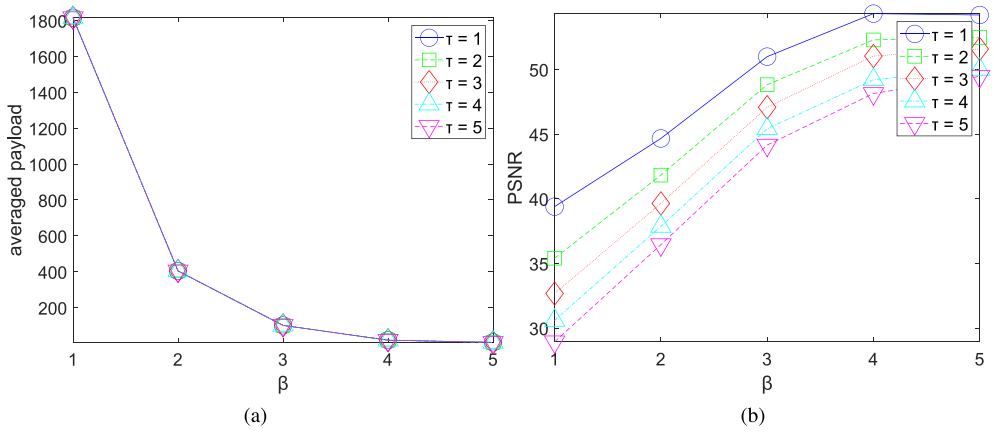


Fig. 4. The influence of changing β and ι in RRW-DC. (a) shows the embedding capacity and (b) shows the watermarked image quality.

Table 2
Robustness (1-BER) and reversibility (PSNR($\tilde{\mathbf{X}}, \tilde{\mathbf{X}}'$)) evaluation of RRW-DC using different parameters.

Para.	QF=100		QF=80		QF=60		QF=50		QF=40	
	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR	1-BER	PSNR
$\forall \iota \in \{1,\ldots,5\}$ $\forall \beta \in \{1,\ldots,5\}$	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN

Table 3
Comparison of maximum capacities of different schemes.

	Proposed		Huang et al. [17]	X. Wang et al. [19]	H. Wang et al. [31]	Chen et al. [23]	Yang et al. [32]
	RRW-AC	RRW-DC					
Capacity	16 218	1816	4096	1024	1024	256	15 000

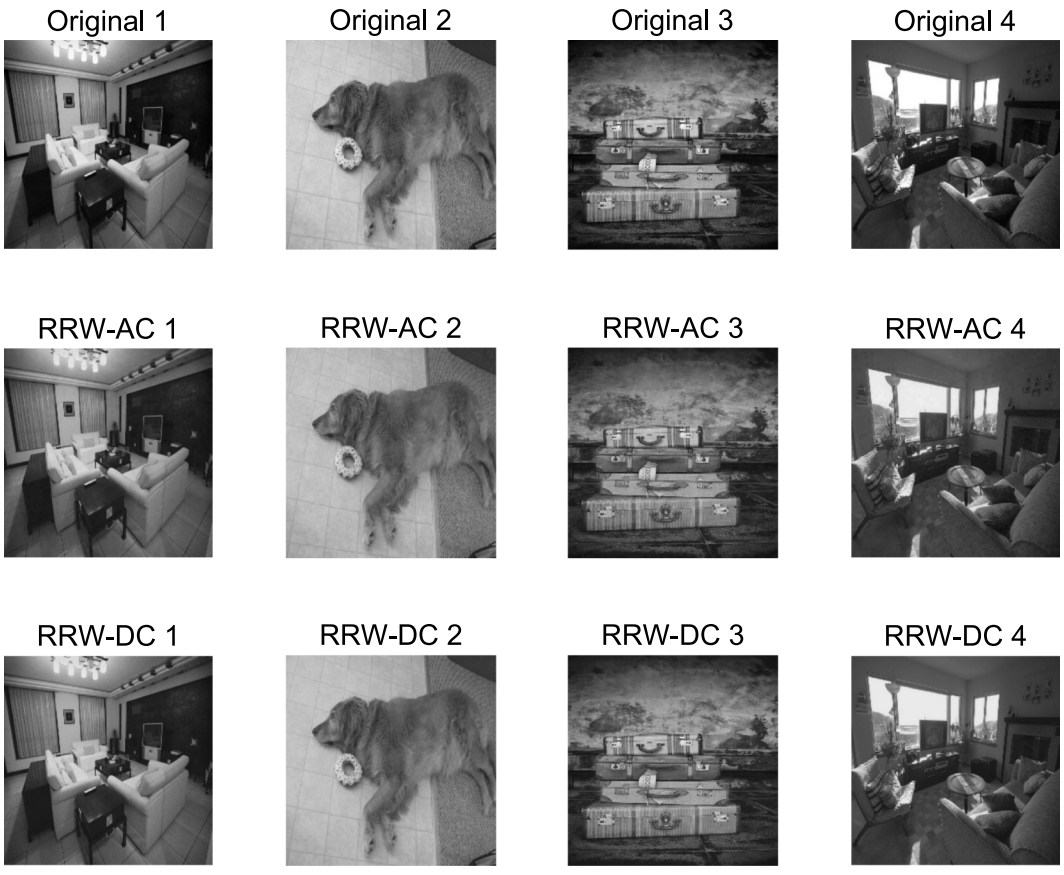
5.3. Evaluation of capacity

We assess the maximum potential capacity of our scheme. Considering a host image with dimensions $l_h \times l_w$, there are $l_h/8 \times l_w/8$ DC coefficients and $7l_h/8 \times 7l_w/8$ AC coefficients available for embedding watermark bits. This underscores our scheme could provide a considerable capacity. Table 3 outlines the maximum capacities of various schemes. It can be observed that the capacity offered by RRW-DC is on a par with other schemes, whereas the capacity of RRW-AC significantly surpasses the rest. It gives our scheme good flexibility. We can choose varying capacities tailored to specific applications.

5.4. Evaluation of watermarked image quality

This section assesses the visual quality of watermarked images by comparing our methods with those in [17,19,23,31,32]. Although our scheme supports a substantially higher payload, all methods were normalized to a 1024-bit payload for fair comparison through tailored implementation strategies. Specifically, except for Chen et al. [23], competing methods randomly select a subset of cover coefficients for embedding while preserving the remaining coefficients unchanged to meet the payload constraint. In contrast, Chen et al. [23] extend capacity by dividing images into smaller spatial units for bit-wise embedding.

Fig. 5 presents representative watermarked images generated by RRW-AC and RRW-DC, demonstrating high visual fidelity. To quantify this, we computed the average PSNR and SSIM over 100 images sampled from the COCO dataset. As shown in Table 4, our approach achieves image quality comparable to existing methods. Moreover, RRW-DC outperforms RRW-AC in visual quality, primarily because smaller values of β and ι introduce less embedding distortion.



(a)

Fig. 5. Demonstration of watermarked images. In the first rows are the test images from COCO datasets. The watermarked images obtained by RRW-AC are demonstrated in the second row, while those obtained by RRW-DC are demonstrated in the third row.

Table 4
The averaged PSNR and SSIM scores of watermarked images obtained by different schemes.

	Proposed		Huang et al. [17]	X. Wang et al. [19]	H. Wang et al. [31]	Chen et al. [23]	Yang et al. [32]
	RRW-AC	RRW-DC					
PSNR	31.8592	38.0919	35.9711	33.2979	37.8733	38.54	38.56
SSIM	0.9637	0.9972	0.9129	0.8768	0.9647	0.9780	0.9835

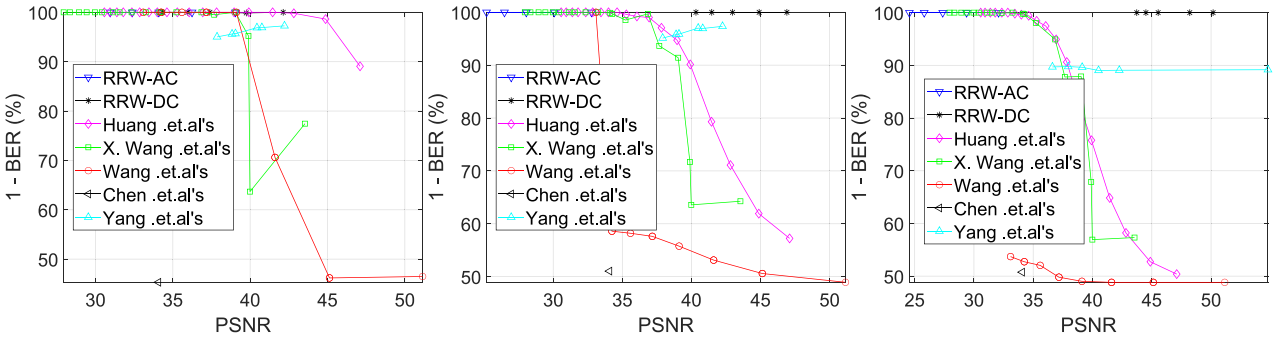
5.5. Evaluation of robustness

We compare the ability of different watermarking schemes to extract watermarks under several common attacks. The PSNR scores of each scheme are adjusted by varying their parameters, which allows us to fine-tune the trade-off between robustness and imperceptibility.

(1) under JPEG compression. Fig. 6 present the robustness of watermark extraction under JPEG compression attacks. The results illustrate the performance of various watermarking schemes, highlighting both extraction and image recovery capabilities. Our digital watermarking scheme demonstrates exceptional resistance, and consistently surpasses other approaches by maintaining image integrity under different attack conditions. Compared to other schemes with similar PSNR, our approach exhibits greater robustness and more accurate watermark extraction, especially at higher PSNR levels. Notably, Chen et al.'s method demonstrates suboptimal performance when subjected to JPEG compression, potentially arising from the inherent sensitivity of invertible flow to aggressive compression parameters.

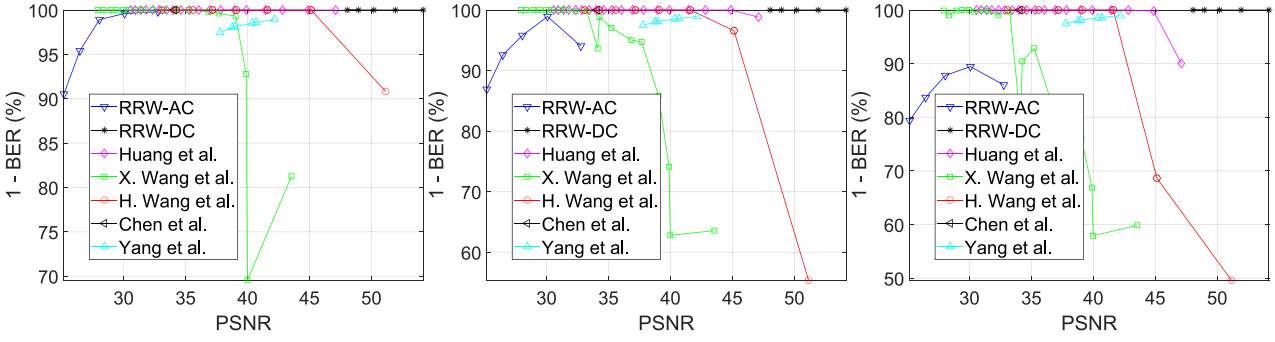
(2) under JPEG2000 compression. Fig. 7 evaluates the robustness of various schemes under JPEG2000 compression, contrasting domain-specific embedding strategies. Our AC coefficient embedding approach exhibits decreased extraction accuracy, due to fundamental domain discrepancies between JPEG2000's wavelet-based decomposition and the DCT framework, for which AC-based watermarking was initially optimized. Despite this domain mismatch, the AC method remains competitively effective compared to other schemes. Furthermore, it is evident that our DC embedding method demonstrates superior robustness, achieving the best 1-BER scores while simultaneously maintaining superior image fidelity across all tested compression ratios. These findings underscore domain compatibility as a crucial design criterion for compression-resistant watermarking architectures, and our RRW-DC demonstrates consistent resilience against various compression artifacts.

(3) under rotating attacks. Fig. 8 presents the extraction results of various watermarking schemes following rotation attacks. It shows that, compared with Chen et al.'s and Yang et al.'s schemes, our schemes do not perform very well under rotation attacks. It is because DC/AC coefficients are inherently sensitive to geometric distortion, and our watermark design did not specifically target this type of attacks. Nevertheless, RRW-DC still achieves 1-BER scores around 0.75. Potential solutions include incorporating template matching or leveraging neural networks to estimate and correct the rotation. For instance, by



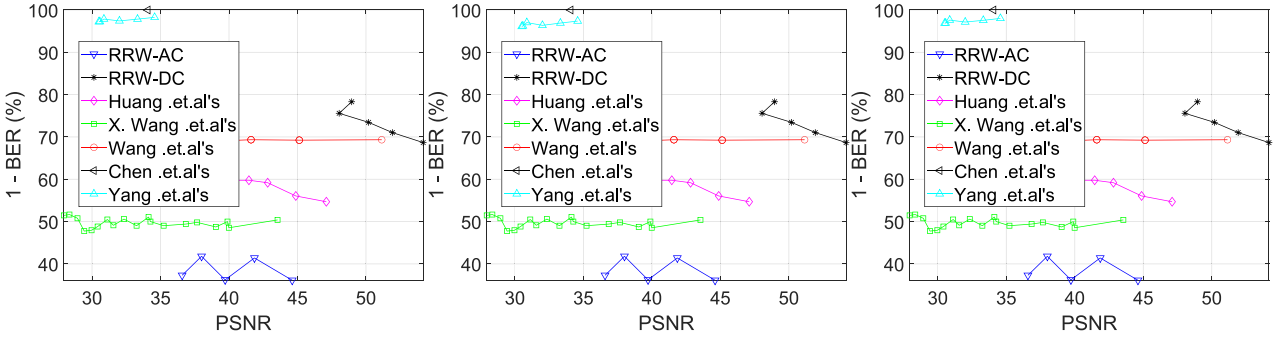
(a) QF=80 (b) QF=50 (c) QF=40

Fig. 6. The robustness of watermark extraction under JPEG attacks.



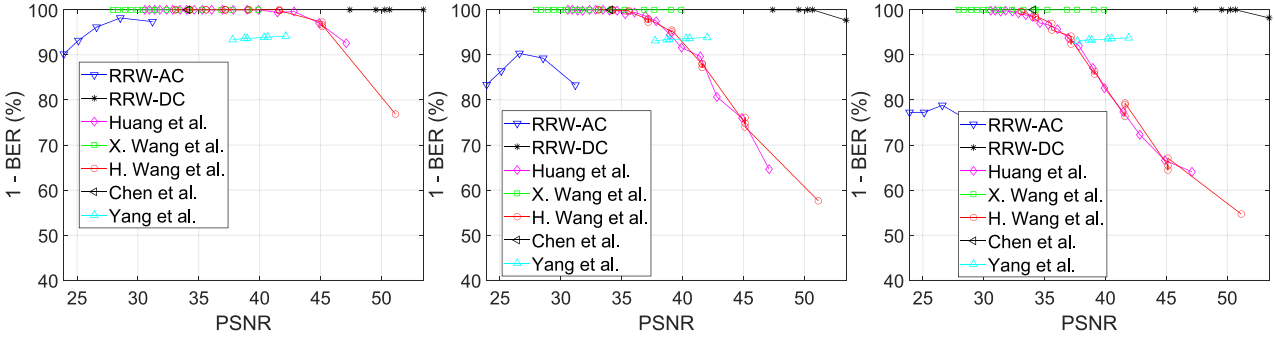
(a) QF=10 (b) QF=15 (c) QF=20

Fig. 7. The robustness of watermark extraction under JPEG2000 attacks.



(a) angle=0.001 (b) angle=0.05 (c) angle=0.01

Fig. 8. The robustness of watermark extraction under Rotating attacks.

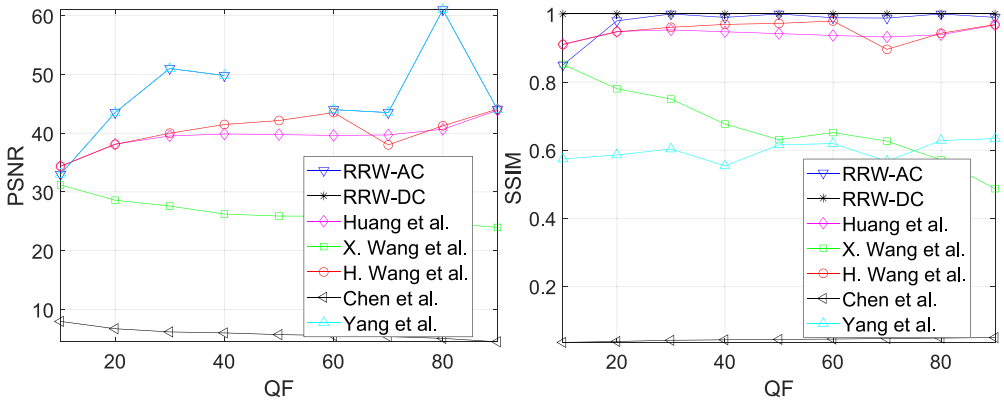


(a) σ=0.001 (b) σ=0.005 (c) σ=0.009

Fig. 9. The robustness of watermark extraction under AWGN attacks.

Table 5
The averaged PSNR and SSIM scores of watermarked images without attacks obtained by different schemes.

	Proposed		Huang et al. [17]	X. Wang et al. [19]	H. Wang et al. [31]	Chen et al. [23]	Yang et al. [32]
	RRW-AC	RRW-DC					
PSNR	inf	inf	72.3511	inf	inf	inf	inf
SSIM	1	1	0.9999	1	1	1	1



(a) PSNR (b) SSIM

Fig. 10. The robustness of image recovery under JPEG compression attacks.

training a neural network to recognize the rotation angle and adjust the watermark extraction accordingly.

(4) under AWGN. Fig. 9 demonstrates the performance of our watermarking schemes under AWGN attacks. The Gaussian noise used in the experiments has a mean of 0, with the variance serving as the parameter of interest. It is observed that under various parameter settings, our DC embedding outperforms both competing schemes and AC embedding. This may be attributed to the concentration of critical watermark energy in low-frequency coefficients, which are less susceptible to slight noise corruption. Furthermore, compared to AC coefficients, DC coefficients have larger magnitude values, providing stronger resistance to amplitude-modulated noise perturbations.

5.6. Evaluation of reversibility

5.6.1. In scenarios without attacks

We first assess the reversibility of the proposed schemes under attack-free conditions. Table 5 reports the quality metrics of the recovered images. The results demonstrate that, similar to existing methods, our schemes achieve outstanding preservation of host image integrity. They are capable of flawless reconstruction of the host image, ensuring that the original content remains intact and clearly discernible. This highlights the effectiveness of our approach in safeguarding the quality and integrity of watermarked images in the absence of any attacks.

5.6.2. In scenarios with attacks

This section evaluate the ability of image recovery under various attacks.

(1) under JPEG compression. Fig. 10 demonstrates the image recovery performance of different watermarking schemes under JPEG compression attacks. Our scheme consistently surpasses others in image recovery across different compression levels, emphasizing the robustness of our approach in preserving image integrity. It is worth noting that embedding within DC coefficients allows for perfect recovery from JPEG compression. Consequently, the PSNR scores of recovered images reach "*inf*", which is thus not depicted in Fig. 10(a).

(2) under JPEG2000 attacks. Fig. 11 illustrates the image recovery performance under JPEG2000 compression attacks. It can be observed that the proposed RRW-DC outperforms the other compared schemes. This superior robustness of RRW-DC in both JPEG and JPEG2000 compression scenarios can be attributed to the inherent resilience of DC coefficients against the effects of image compression.

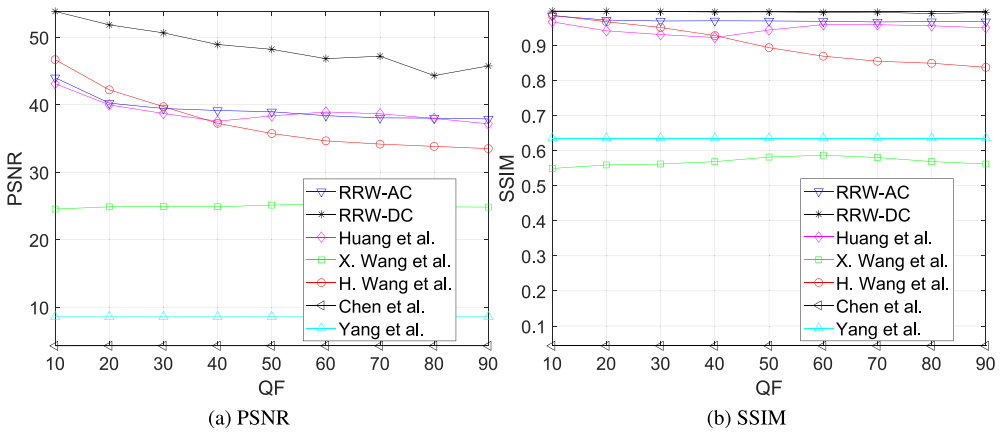
(3) under rotating attacks. Fig. 12 showcases the image recovery performance of various watermarking schemes when subjected to rotation attacks. It is apparent that, despite our schemes exhibiting somewhat diminished extraction accuracy after enduring such attacks, they still surpass other schemes in the realm of image recovery. Impressively, they are capable of attaining a PSNR score that exceeds 35, marking a significant achievement in this context.

(4) under AWGN. Fig. 13 illustrates the image recovery performance of various watermarking schemes when subjected to AWGN attacks. It is observed that all schemes exhibit comparable performance, with the exception of Chen et al.'s and Yang et al.'s schemes. This may be attributed to the lack of adversarial training or stochastic noise injection in the network architectures of these two schemes during optimization. Furthermore, the invertible flow amplifies the AWGN perturbations in the reverse path during the recovery of the cover images in Chen et al.'s scheme. In contrast, our RRW-AC/DC frameworks achieve noise-agnostic recovery. As a result, our scheme possesses robust image recovery capabilities, even amidst diverse attack types.

6. Conclusion

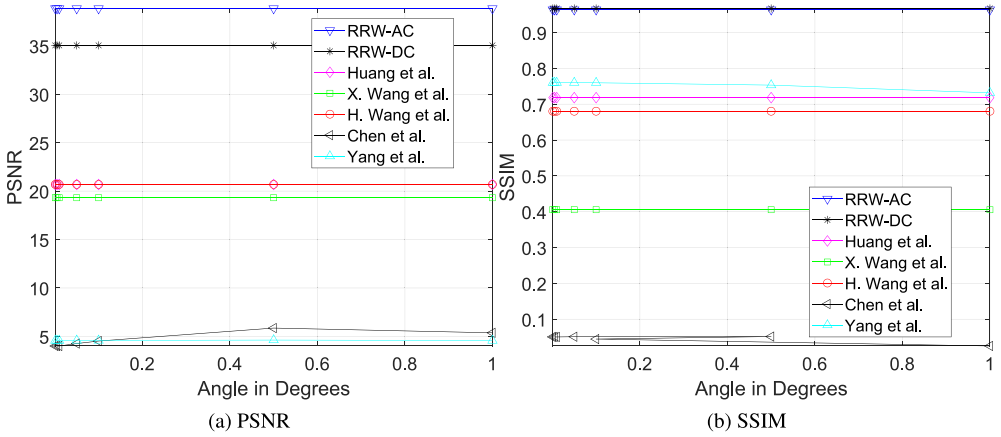
This paper presents a resilient reversible watermarking technique designed to withstand JPEG compression attacks. Leveraging DCT transformation, the proposed method employs histogram shifting in both low-frequency and high-frequency components for watermark embedding. Our histogram shifting strategy ensures that the watermarked image's histogram remains unchanged after JPEG compression. Furthermore, the specified shift distance enhances the robustness of embedded watermark bits against other common signal processing operations.

Based on this embedding algorithm, we introduce two schemes: RRW-AC, which embeds in AC coefficients, and RRW-DC, which embeds in DC coefficients. Owing to the abundance of AC coefficients, RRW-AC offers substantially higher embedding capacity, whereas RRW-DC delivers superior robustness. Each scheme is tailored to different applications: RRW-AC suits scenarios requiring high payloads, such as multimedia metadata embedding, integrity verification, and broadcast



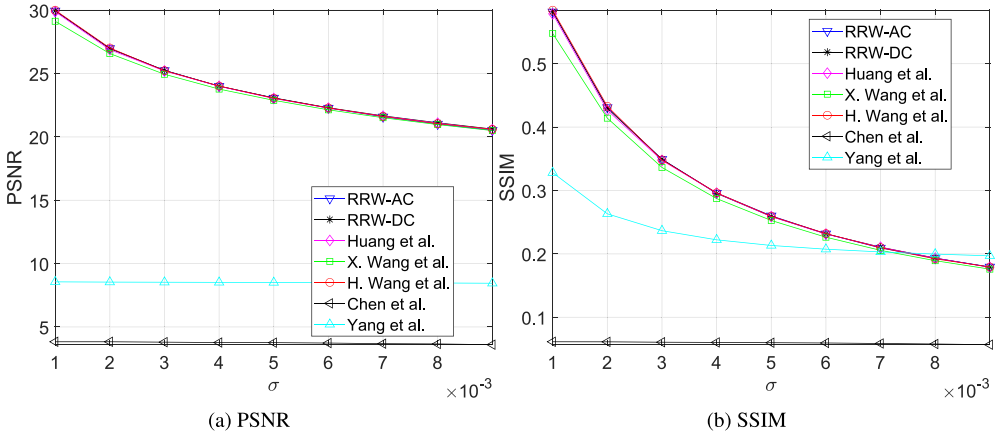
(a) PSNR (b) SSIM

Fig. 11. The robustness of image recovery under JPEG2000 compression attacks.



(a) PSNR (b) SSIM

Fig. 12. The robustness of image recovery under rotating attacks.



(a) PSNR (b) SSIM

Fig. 13. The robustness of image recovery under AWGN attacks.

monitoring, while RRW-DC is ideal for use cases demanding high robustness and imperceptibility, such as embedding digital copyright marks where payload can be smaller in exchange for enhanced reliability.

Experimental results confirm that both RRW-AC and RRW-DC achieve outstanding robustness, image quality, and reversibility under various distortions. However, our approach currently embeds only in the Y channel of JPEG-compressed color images, limiting its applicability. Future work will investigate color image properties and develop robust reversible watermarking algorithms specifically for full-color images.

CRediT authorship contribution statement

Jiancheng Xiao: Writing – review & editing, Writing – original draft, Software, Resources, Methodology. **Shuaichao Wu:** Writing – review & editing, Writing – original draft, Software, Resources, Methodology. **Bingwen Feng:** Writing – review & editing, Writing – original

draft, Validation, Methodology, Conceptualization. **Jilian Zhang:** Investigation, Data curation, Conceptualization. **Bing Chen:** Investigation, Data curation. **Zhihua Xia:** Validation, Conceptualization. **Wei Lu:** Investigation, 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.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant number 2022YFB3103100, National Natural Science Foundation of China (Grant No. 62472199, 62261160653, 62102101, 62122032, U23B2023), Natural Science Foundation of Guangdong Province, China (Grant No. 2025A1515011601), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2023B1212060036), the Opening Project of MoE Key Laboratory of Information Technology (Sun Yat-sen University) (Grant No. 2024ZD001).

Data availability

Data will be made available on request.

References

[1] I.-C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking, IEEE Trans. Image Process. 23 (4) (2014) 1779–1790.
[2] Y. Ke, M.-Q. Zhang, J. Liu, T.-T. Su, X.-Y. Yang, Fully homomorphic encryption encapsulated difference expansion for reversible data hiding in encrypted domain, IEEE Trans. Circuits Syst. Video Technol. 30 (8) (2020) 2353–2365.
[3] M. Jiang, Reversible data hiding algorithm in encrypted images using adaptive total variation and cross-cyclic shift, Int. J. Auton. Adapt. Commun. Syst. 16 (6) (2023) 611–631.
[4] W. He, G. Xiong, Y. Wang, Reversible data hiding based on adaptive multiple histograms modification, IEEE Trans. Inf. Forensics Secur. 16 (2021) 3000–3012.
[5] Q. Chang, X. Li, Y. Zhao, Reversible data hiding for color images based on adaptive three-dimensional histogram modification, IEEE Trans. Circuits Syst. Video Technol. 32 (9) (2022) 5725–5735.
[6] Z. Chen, J. Qin, Reversible data hiding in encrypted images based on histogram shifting and prediction error block coding, Int. J. Auton. Adapt. Commun. Syst. 18 (1) (2025) 45–66.
[7] K. Chen, H. Zhou, D. Hou, W. Zhang, N. Yu, Reversible data hiding in JPEG images under multi-distortion metric, IEEE Trans. Circuits Syst. Video Technol. 31 (10) (2020) 3942–3953.
[8] M. Xiao, X. Li, Y. Zhao, Reversible data hiding for JPEG images based on multiple two-dimensional histograms, IEEE Signal Process. Lett. 28 (2021) 1620–1624.
[9] S. Weng, Y. Zhou, T. Zhang, M. Xiao, Y. Zhao, Reversible data hiding for JPEG images with adaptive multiple two-dimensional histogram and mapping generation, IEEE Trans. Multimed. 25 (2023) 8738–8752.
[10] X. Yin, S. Wu, B. Chen, K. Wang, W. Lu, Reversible data hiding in JPEG document images based on zero coefficients embedding, Signal Process. 206 (2023) 108917.
[11] G. Gao, Y.-Q. Shi, Reversible data hiding using controlled contrast enhancement and integer wavelet transform, IEEE Signal Process. Lett. 22 (11) (2015) 2078–2082.
[12] G. Gao, M. Wang, B. Wu, Efficient robust reversible watermarking based on ZMs and integer wavelet transform, IEEE Trans. Ind. Inform. 20 (3) (2024) 4115–4123.
[13] W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, J. Sun, A comprehensive survey on robust image watermarking, Neurocomputing 488 (2022) 226–247.
[14] 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.
[15] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, Robust lossless image data hiding, in: IEEE International Conference on Multimedia & Expo, 2004, pp. 2199–2202.
[16] 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.
[17] Z. Huang, B. Feng, S. Xiang, Robust reversible image watermarking scheme based on spread spectrum, J. Vis. Commun. Image Represent. 93 (2023) 103808.
[18] X. Liang, S. Xiang, Robust reversible watermarking of JPEG images, Signal Process. 224 (2024) 109582.
[19] 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.
[20] R. Hu, S. Xiang, Cover-lossless robust image watermarking against geometric deformations, IEEE Trans. Image Process. 30 (2021) 318–331.
[21] D. Fu, X. Zhou, L. Xu, K. Hou, X. Chen, Robust reversible watermarking by fractional order zernike moments and pseudo-zernike moments, IEEE Trans. Circuits Syst. Video Technol. 33 (12) (2023) 7310–7326.
[22] Y. Tang, C. Wang, S. Xiang, Y.-M. Cheung, A robust reversible watermarking scheme using attack-simulation-based adaptive normalization and embedding, IEEE Trans. Inf. Forensics Secur. 19 (2024) 4114–4129.
[23] J. Chen, W. Wang, C. Shi, L. Dong, Y. Li, X. Hu, Deep robust reversible watermarking, 2025, arXiv preprint arXiv:2503.02490.
[24] Y. Zhang, X. Luo, J. Wang, W. Lu, C. Yang, F. Liu, Research progress on digital image robust steganography, J. Image Graph. 27 (01) (2022) 3–26.
[25] J. Tao, S. Li, X. Zhang, Z. Wang, Towards robust image steganography, IEEE Trans. Circuits Syst. Video Technol. 29 (2) (2018) 594–600.
[26] J. Zhang, X. Zhao, X. He, H. Zhang, Improving the robustness of JPEG steganography with robustness cost, IEEE Signal Process. Lett. 29 (2021) 164–168.
[27] J. Butora, P. Puteaux, P. Bas, Errorless robust JPEG steganography using outputs of JPEG coders, IEEE Trans. Dependable Secur. Comput. (2023).
[28] Y. Huang, Z. Liu, Q. Wu, X. Liu, Robust image steganography against JPEG compression based on DCT residual modulation, Signal Process. (2024) 109431.
[29] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[30] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[31] H. Wang, X. Li, M. Xiao, Y. Zhao, A novel robust reversible watermarking method against JPEG compression, in: Advances in Artificial Intelligence and Security: 7th International Conference, Springer, 2021, pp. 312–322.
[32] X. Yang, Y. Wang, F. Huang, CNN-based reversible data hiding for JPEG images, IEEE Trans. Circuits Syst. Video Technol. (2024).
[33] P. Bas, T. Filler, T. Pevnỳ, "Break our steganographic system": The ins and outs of organizing BOSS, in: International Workshop on Information Hiding, 2011, pp. 59–70.
[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision, 2014, pp. 740–755.
[35] 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.