QDCT-based blind color image watermarking with aid of GWO and DnCNN for performance improvement

Ling-Yuan Hsu¹ and Hwai-Tsu Hu², Member, IEEE
¹Department of Information Management, St. Mary's Junior College of Medicine, Nursing and Management, I-Lan, Taiwan
²Department of Electronic Engineering, National I-Lan University, I-Lan, Taiwan

ABSTRACT: Artificial intelligence (AI) is of great potential for improving the performance of image processing and applications. In this study, we incorporate two AI techniques, namely, the grey wolf optimizer (GWO) and denoising convolutional neural network (DnCNN), within a framework developed based on the quaternion discrete cosine transform (QDCT). Binary embedding is formulated according to the attribute of each QDCT component and the distinctive properties of available modulation schemes. The GWO is responsible for performance optimization, while the DnCNN makes the extracted binary watermark more visually recognizable. Experiment results demonstrate the efficacy of the proposed scheme for resisting a variety of image processing attacks. The proposed scheme outperforms existing ones in terms of the robustness and intelligibility of the retrieved watermarks under the same payload capacity.

INDEX TERMS: Blind color image watermarking; Grey wolf optimizer; Denoising convolutional neural network; Quaternion discrete cosine transform; Mixed modulation.

I. INTRODUCTION

The widespread availability of editing tools for digital content has led to increasing levels of theft and manipulation of intellectual property. Digital watermarking is a common approach to protecting the rights of digital content owners. In case the digital content has been misappropriated or plagiarized, an examination of retrieved watermarks via the publicly available algorithm can easily reveal illicit attempts. Apart from the application to the authentication of audio, video, image, and data files [1–4], watermarking is also important for new imaging modalities such as light field, hologram [5], and point cloud [6]. Watermarks are judged according to their ability to resist tampering or removal while remaining imperceptible to the casual observer [7].

Image watermarking techniques can be divided into those implemented in the spatial domain and those implemented in the transform domain. Watermarking in the spatial domain is performed by adjusting image pixels directly. Watermarking in the transform domain involves converting image pixel values in the spatial domain into transform coefficients supposed to use in watermark embedding and extraction. Watermarking schemes in the transform domain impose computational requirements higher than those in the spatial domain; however, they tend to be far more robust against common attacks using a comparable amount of information. Typical methods in the transform domain include those based on the discrete cosine transform (DCT) [8–11], discrete wavelet transform (DWT) [12, 13], and discrete Fourier transform (DFT) [14, 15].

The quaternion is a useful mathematical tool that enables algebraic operations in a four-dimensional vector space. Because a color image can be regarded as a two-dimensional array with an extra depth constituted by three primary colors, the algebraic properties of the quaternion make it attractive to handle color image watermarking. When the quaternion is applied to image watermarking, there are two ways to process the image content: grayscale and color. In grayscale image watermarking, the quaternion wavelet transform (QWT) [16, 17] renders a shift-invariant magnitude and three accompanying components (with two of them representing local image shifts and the third one delivering textural information). The watermark is then embedded in the coefficient of magnitude component to improve imperceptibility and robustness. For the case of color image watermarking, a color image can be cast into a...
real component along with three imaginary components. The watermark is then embedded into the selected components. Such quaternion-based watermarking makes it possible to take into account three color channels as a whole and achieve a balanced improvement in the payload capacity, imperceptibility, and robustness. This sort of quaternion-based approach includes the quaternion discrete cosine transform (QDCT) [18], the quaternion discrete Fourier transform (QDFT) [3, 19], and the quaternion discrete fractional random transform (QDFRT) [20].

Most watermarking schemes rely on the control of adjustable parameters to achieve the desired performance. The optimization method is the most popular approach for this purpose. Optimization methods such as the teaching-learning-based optimization (TLBO) [21], support vector machine (SVM) [22, 23], particle swarm optimization (PSO) [24], and grey wolf optimizer (GWO) [25] are known for their capabilities of exploiting the collective behavior of organized systems to iteratively work through a large number of candidate solutions. The DCT-based watermarking scheme proposed by Moosazadeh and Ekbatanifard [26] adopted the low-frequency DCT coefficients as the basis for watermark embedding. The TLBO aimed at determining the optimal position and embedding strength for watermark embedding. Chen et al. [20] developed a blind color watermarking scheme based on the QDFRT. Their scheme exploited the properties of the human vision system (HVS) to adaptively adjust the watermark strength, along with the employment of a random number matrix to enhance security. The SVM was responsible for acquiring the watermark through the relationship between the embedding position and the neighboring coefficient. The scheme proposed by Li et al. [18] transformed the host image into the QDCT domain, where the watermark was embedded in the coefficient of the unitary matrix after singular value decomposition (SVD). In their scheme, the PSO helped to find the appropriate matrix embedding intensity factor for each image to obtain better imperceptibility and robustness. Similar to the preceding approach, Hsu and Hu [3] developed a blind watermarking scheme based on the QDFT. Their scheme used multi-bit partly sign-altered mean modulation to embed watermarks in each QDFT block. The PSO also played the role of optimizing the embedding strength and coefficients of selected components in the QDFT domain.

This paper is focused on the development of an efficient and effective blind color image watermarking scheme that exploits the merits of QDCT and optimization modeling. To promote the performance even further, we incorporate a denoising convolutional neural network (DnCNN) to eliminate Gaussian noise of unknown levels [31]. In this study, we resort to the DnCNN to render a watermark that is visually more recognizable.

The remainder of this paper is organized as follows. Section II outlines the technical backgrounds involved in implementing blind color image watermarking using QDCT. Section III details the procedures involved in the proposed watermark embedding and extraction processes. Section IV compares the performance of the proposed scheme with that of other DCT-related schemes in terms of imperceptibility and robustness. Section V addresses the watermarking enhancement through the DnCNN denoising. Finally, concluding remarks are summarized in Section VI.

II. QUATERNION DISCRETE COSINE TRANSFORM (QDCT) OF A COLOR IMAGE

The outstanding ability of DCT to transform the energy of an image signal into low-frequency coefficients makes it ideal for image compression and watermarking. For a monochrome image of size $M \times N$, the DCT conversion pair between the spatial-domain signal (termed $f(m,n)$) and spectral-domain representation (termed $F(p,q)\)$ are given as follows:

$$F(p,q) = \text{DCT}\{f(m,n)\} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \cos \frac{(2m+1)p\pi}{2M} \cos \frac{(2n+1)q\pi}{2N},$$

$$0 \leq p \leq M - 1, 0 \leq q \leq N - 1, \quad (1)$$

$$f(m,n) = \text{IDCT}\{F(p,q)\} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q F(p,q) \cos \frac{(2m+1)p\pi}{2M} \cos \frac{(2n+1)q\pi}{2N},$$

$$0 \leq m \leq M - 1, 0 \leq n \leq N - 1, \quad (2)$$

where $\text{DCT}\{\}$ and $\text{IDCT}\{\}$ respectively refer to the DCT and its inverse transformation. The notations $(m,n)$ and $(p,q)$ respectively correspond to the coordinates in the spatial and transform domains. $\alpha_p$ and $\alpha_q$ denote two multiplying factors defined as

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0; \\ \frac{1}{\sqrt{M}}, & 1 \leq p \leq M - 1 \end{cases}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0; \\ \frac{2}{\sqrt{N}}, & 1 \leq q \leq N - 1 \end{cases}$$

A color image is a composite representation of red, green, and blue components, the values of which indicate the light intensity required to describe the digital image. Accordingly, a digital color image is usually stored as a matrix of size $M \times N \times 3$. In a quaternion representation, a color image can
be arranged as an array of quaternion numbers of the following form:

\[ f_q(m,n) = 0 + f_k(m,n) \cdot \hat{i} + f_o(m,n) \cdot \hat{j} + f_p(m,n) \cdot \hat{k} \]  

(4)

where \( f_k(m,n) \), \( f_o(m,n) \), and \( f_p(m,n) \) correspond to the \((m,n)^{th}\) pixel values of the red, green, and blue components, respectively. Note that \( \hat{i}, \hat{j}, \) and \( \hat{k} \) denote three complex operators, which can be interpreted as unit vectors pointing along the three spatial axes. These three unit-vectors (i.e., \( \hat{i}, \hat{j}, \) and \( \hat{k} \)) hold the following relationships:

\[
\begin{align*}
\hat{i}^2 = \hat{k}^2 = \hat{i} \cdot \hat{j} \cdot \hat{k} &= -1 \\
\hat{i} \cdot \hat{k} &= \hat{j}, \hat{j} \cdot \hat{i} &= -\hat{k} \\
\hat{j} \cdot \hat{k} &= \hat{i}, \hat{k} \cdot \hat{j} &= -\hat{i} \\
\hat{k} \cdot \hat{i} &= \hat{j}, \hat{i} \cdot \hat{k} &= -\hat{j}
\end{align*}
\]  

(5)

Note that Eq. (4) distributes the color image into three imaginary parts and leaves out the scalar part as zero. According to the formulation in [32], the quaternion discrete cosine transform (QDCT) derived from quaternion algebra and classical 2-dimensional DCT can be expressed as

\[
F_0(p,q) = QDCT \left\{ f_q(m,n) \right\} = \mu_0 DCT \left\{ \mu DCT \left\{ f_k(m,n) \hat{i} \right\} + DCT \left\{ f_o(m,n) \hat{j} \right\} \\
+ DCT \left\{ f_p(m,n) \hat{k} \right\} \right\}
\]

(6)

where \( \mu_0 = \mu \cdot \hat{i} + \mu \cdot \hat{j} + \mu \cdot \hat{k} \) denotes a unit quaternion subject to the constraint that \( \mu_0^2 = -1 \). \( A(p,q), C(p,q), D(p,q) \) and \( E(p,q) \) respectively represent the vector components of the \((p,q)^{th}\) quaternion coefficient. Specifically,

\[
A(p,q) = -\mu_0 DCT \left\{ f_k(m,n) \right\} - \mu DCT \left\{ f_o(m,n) \right\} - \mu DCT \left\{ f_p(m,n) \right\}
\]

(7)

\[
C(p,q) = \mu DCT \left\{ f_k(m,n) \right\} - \mu_0 DCT \left\{ f_o(m,n) \right\}
\]

(8)

\[
D(p,q) = \mu DCT \left\{ f_k(m,n) \right\} - \mu_0 DCT \left\{ f_p(m,n) \right\}
\]

(9)

\[
E(p,q) = \mu_0 DCT \left\{ f_o(m,n) \right\} - \mu DCT \left\{ f_p(m,n) \right\}
\]

(10)

Following the formulation in [33], the inverse conversion from \( F_0(p,q) \) to \( f_q(m,n) \) takes the following form:

\[
f_q(m,n) = -\mu_0 \cdot IQDCT \left\{ F_0(p,q) \right\}
\]

\[
= f_k(m,n) + f_o(m,n) \hat{i} + f_p(m,n) \hat{j} + f_e(m,n) \hat{k}
\]

(11)

where \( f_k(m,n), f_o(m,n), f_p(m,n), \) and \( f_e(m,n) \) correspond to the four components of the resulting quaternion in a 4-dimensional vector space. Substituting \( \mu_0 \) into Eq. (11) results in the following:

\[
f_k(m,n) = \mu_1 \cdot IDCT \left\{ C(p,q) \right\}
\]

\[
+ \mu_1 \cdot IDCT \left\{ D(p,q) \right\} + \mu_2 \cdot IDCT \left\{ E(p,q) \right\}
\]

(12)

\[
f_o(m,n) = -\mu_1 \cdot IDCT \left\{ A(p,q) \right\}
\]

\[
+ \mu_1 \cdot IDCT \left\{ D(p,q) \right\} - \mu_2 \cdot IDCT \left\{ E(p,q) \right\}
\]

(13)

\[
f_p(m,n) = -\mu_1 \cdot IDCT \left\{ A(p,q) \right\}
\]

\[
- \mu_2 \cdot IDCT \left\{ C(p,q) \right\} + \mu \cdot IDCT \left\{ E(p,q) \right\}
\]

(14)

\[
f_e(m,n) = -\mu_1 \cdot IDCT \left\{ A(p,q) \right\}
\]

\[
+ \mu_2 \cdot IDCT \left\{ C(p,q) \right\} - \mu \cdot IDCT \left\{ D(p,q) \right\}
\]

(15)

Theoretically, \( f_k(m,n) \) shall remain at zero. \( f_o(m,n), f_p(m,n), \) and \( f_e(m,n) \) respectively coincide with the intensity levels in the red, green, and blue channels (namely, \( f_k(m,n), f_o(m,n), \) and \( f_p(m,n) \) ) of each color pixel.

III. PROPOSED BLIND WATERMARKING SCHEME

In this section, we discuss how to implement binary watermarking using suitable modulation schemes under the QDCT framework, hereafter referred to as watermark QDCT (wQDCT). The out-of-range errors problem encountered in watermark embedding can be settled through extreme pixel adjustment (EPA). The GWO is then employed to optimize the performance in imperceptibility and robustness. Finally, a DnCNN is introduced to make the extracted binary watermark more visually recognizable. Figure 1 illustrates the architecture of the proposed watermarking scheme.
A. BINARY WATERMARKING

Analogous to the manners seen in DCT-based watermarking, watermarking in the QDCT domain involves partitioning the host image into non-overlapping blocks of 8×8 pixels, taking the QDCT of each block via Eq. (6), and manipulating the selected coefficients according to prescribed rules. While converting the modified QDCT coefficients back to the spatial domain representation, it is important to ensure \( f_s(m,n) = 0 \) such that the QDCT coefficients remain intact throughout the course of inverse transformation. Because the derivation of \( f_s(m,n) \) does not involve \( A(p,q) \) in Eq. (12), the watermark embedding can be proceeded by either manipulating \( A(p,q) \) alone or modifying \( C(p,q) \), \( D(p,q) \) and \( E(p,q) \) jointly as long as \( f_s(m,n) \) is zero. To probe into the general characteristics of the four QDCT components, we examined the QDCT coefficients of all the non-overlapping 8×8 matrix blocks obtained from eight representative color images (namely, Lena, Baboon, Avion, Peppers, Bobcat, Fiore, Lighthouse, and Quietime accessible from the Computer Vision Group at the University of Granada [34]) with a diversity of brightness. Table I lists the statistical means and standard deviations of the QDCT coefficients situated in the second to fourth anti-diagonals, which roughly cover the low-to-medium frequency range. In general, embedding a watermark in the low-frequency region is robust against high-frequency attacks like lowpass filtering and image compression; however, it remains vulnerable to low-frequency attacks such as unsharp filtering and histogram equalization. The third and fourth anti-diagonals in mid-frequencies appear an acceptable compromise for a wide variety of attacks.

As shown in Table I, the statistical means of the QDCT coefficients under investigation are close to zero. Among the four components, \( A(p,q) \) presented the largest standard deviation for each QDCT location. Overall, the three imaginary components \( C(p,q) \), \( D(p,q) \), and \( E(p,q) \) have similar standard deviations with the smallest one associated with \( C(p,q) \). In light of these observations, we adopted different schemes to cope with the corresponding characteristics of the QDCT components.

There are two popular modulation schemes for DCT-based watermarking: quantization index modulation (QIM) [35] and relative modulation (RM) [26]. The QIM dichotomizes the selected DCT coefficients into alternating zones, whereas RM manipulates the coefficients according to a reference level. RM has proven highly effective in terms of robustness; however, it occasionally results in excessive distortion. Hu and Hsu [36] proposed a remedial scheme referred to as mixed modulation (MM), which grafts QIM onto RM to overcome the deficiency of RM.

The observation on the statistical distribution of quaternion components allows us to choose among the available modulation schemes to optimize overall performance. As revealed by Eq. (12), \( A(p,q) \) plays no role in the formation of \( f_s(m,n) \). In other words, \( f_s(m,n) \) remains unchanged no matter how \( A(p,q) \) changes. This makes \( A(p,q) \) an ideal choice for watermarking in the QDCT domain. Nonetheless, Eqs. (13)-(15) also indicate that any modification of \( A(p,q) \) simultaneously alters the RGB values. In view of the large standard deviation associated with \( A(p,q) \), we employed MM to embed binary bits into \( A(p,q) \). The use of MM to embed the watermark bit \( w_i \) (i.e., 0 or 1) into the coefficient is implemented in two steps. The first step involves the execution of RM and QIM using Eqs. (16) and (17), respectively:

\[
\hat{A}(\hat{p},\hat{q}) = \begin{cases} 
\max \{ \Delta, A(\hat{p},\hat{q}) \}, & \text{if } w_i = 1; \\
\min \{-\Delta, A(\hat{p},\hat{q})\}, & \text{if } w_i = 0.
\end{cases}
\]  

\[
\hat{A}(\hat{p},\hat{q}) = \begin{cases} 
\left[ \frac{A(\hat{p},\hat{q}) - \eta + (1-w_i)}{2\Delta} \right] \cdot 2\Delta + w_i \cdot \Delta + \eta, & \text{if } A(\hat{p},\hat{q}) > \rho \\
\left[ \frac{A(\hat{p},\hat{q}) + \eta - w_i}{2\Delta} \right] \cdot 2\Delta - (1-w_i) \cdot \Delta - \eta, & \text{if } A(\hat{p},\hat{q}) < -\rho \\
(1-w_i) \cdot \eta, & \text{otherwise}
\end{cases}
\]

where \([\cdot] \) and \([-\cdot] \) respectively denote the floor and ceiling functions, \((\hat{p},\hat{q})\) denotes the chosen location, \(\Delta\) refers to the quantization step of QIM, \(\eta\) is the ground level in the upward direction, and \(\rho\) denotes the threshold separating RM from QIM. Note that \(\eta\) and \(\rho\) set at 2\(\Delta\) and 1.5\(\Delta\) in this study. As shown in Eqs. (16) and (17), as long as \(A(\hat{p},\hat{q})\) falls within \(\pm\rho\), RM is used; otherwise, QIM is used. Once \(\hat{A}(\hat{p},\hat{q})\) and \(\tilde{A}(\hat{p},\hat{q})\) are available, the next step is to identify a better solution that induces smaller alteration to the block, as follows:

\[
\hat{A}(\hat{p},\hat{q}) = \begin{cases} 
\hat{A}(\hat{p},\hat{q}), & \text{if } \hat{A}(\hat{p},\hat{q}) - A(\hat{p},\hat{q}) \geq 0 \\
A(\hat{p},\hat{q}) - \tilde{A}(\hat{p},\hat{q}), & \text{otherwise}
\end{cases}
\]
Apart from the use of $A$ component in the MM, the other components $C$, $D$, and $E$ can also be used to embed binary bits as long as the constraint $f_A(m,n) = 0$ is satisfied. The small standard deviation and mean of $C(p,q)$ makes it an ideal candidate for RM, since the damage imposed by RM is mild for a QDCT component with a rather concentrated distribution. The regular patterns in Eqs. (13)-(15) provide a convenient pathway to implement RM on a selected $\hat{C}(\hat{p},\hat{q})$ subject to the constraint $f_A(m,n) = 0$ via compensatory adjustments on $\hat{D}(\hat{p},\hat{q})$ and $\hat{E}(\hat{p},\hat{q})$. In mathematical form, we obtain the following:

$$
\hat{C}(\hat{p},\hat{q}) = \begin{cases} 
\max\{\delta, C(\hat{p},\hat{q})\}, & \text{if } w_1 = 1; \\
\min\{-\delta, C(\hat{p},\hat{q})\}, & \text{if } w_1 = 0.
\end{cases}
$$

(19)

$$
\begin{align*}
\hat{D}(\hat{p},\hat{q}) &= D(\hat{p},\hat{q}) - 0.5 \cdot (\hat{C}(\hat{p},\hat{q}) - C(\hat{p},\hat{q})) \\
\hat{E}(\hat{p},\hat{q}) &= E(\hat{p},\hat{q}) - 0.5 \cdot (\hat{C}(\hat{p},\hat{q}) - C(\hat{p},\hat{q}))
\end{align*}
$$

(20)

where $\delta$ refers to a clearance threshold. The above RM formula (Eq. (19)) raises $C(\hat{p},\hat{q})$ to a positive level when $w_1 = 1$ and drags $C(\hat{p},\hat{q})$ to a negative level when $w_1 = 0$.

Watermark extraction for the RM is relatively straightforward. Once the QDCT coefficients of the watermarked color image are obtained, we examine the sign of the designated imaginary component. A positive value indicates the embedding of a binary “1”, whereas a negative value corresponds to a binary “0”. Specifically,

$$\tilde{w}_b = \begin{cases} 
1, & \text{if } \tilde{C}(\hat{p},\hat{q}) \geq 0; \\
0, & \text{otherwise}.
\end{cases}
$$

(21)

where the tilde indicates that the variable was acquired from a watermarked image following an attack.

By contrast, watermark extraction of the MM is relatively complicated. The watermark bit is determined by examining the retrieved $\tilde{A}(\hat{p},\hat{q})$ using a quadruple branching process. The MM is used when $\tilde{A}(\hat{p},\hat{q})$ falls within $\pm \rho$; otherwise, QIM is employed. The MM can be expressed in mathematical form as follows:

$$\tilde{w}_b = \begin{cases} 
\text{mod}\left(\left(\tilde{A}(\hat{p},\hat{q}) - \eta / A_0 + 0.5, 2\right), & \text{if } \tilde{A}(\hat{p},\hat{q}) > \rho \\
\text{mod}\left(\left(\tilde{A}(\hat{p},\hat{q}) + \eta / A_0 - 1.5, 2\right), & \text{if } \tilde{A}(\hat{p},\hat{q}) < -\rho \\
1, & \text{if } |\tilde{A}(\hat{p},\hat{q})| \leq \rho \text{ and } \tilde{A}(\hat{p},\hat{q}) > 0 \\
0, & \text{if } |\tilde{A}(\hat{p},\hat{q})| \leq \rho \text{ and } \tilde{A}(\hat{p},\hat{q}) \leq 0
\end{cases}
$$

(22)

where mod$(\cdot, 2)$ denotes the modulo function with a modulus of 2.

B. EXTREME PIXEL ADJUSTMENT (EPA)

After a watermark is embedded, all of the pixels are converted to 8-bit unsigned integer values between 0 and 255. Nonetheless, out-of-range errors can occur during the process of storing the watermarked image [37]. Pixels that are close to extreme pixel values in the original image block can easily exceed the legal interval [0, 255] after the watermark is embedded. The rounding up or down of these values to 0 or 255 when the watermarked image is stored as a digital file can lead to the loss of watermark information. Hsu and Hu [3] resolved the out-of-range error by adopting the following extreme pixel adjustment (EPA) function:

$$I = \begin{cases} 
I \times (\gamma_1 - \gamma_2) / \gamma_1 + \gamma_2, & \text{if } I < \gamma_1 \\
(255 - \gamma_2) - (255 - I) (\gamma_1 - \gamma_2) / \gamma_1, & \text{if } I > 255 - \gamma_1, \\
I, & \text{otherwise}
\end{cases}
$$

(23)

where $I$ denotes the pixel value in one channel. $\gamma_1$ and $\gamma_2$ respectively account for the adjustment range and boundary reduction, and $\gamma_1 \geq \gamma_2$. With the activation of the EPA, the values of $\gamma_1$ and $\gamma_2$ are both set at 1 initially and $\gamma_1$ will increase by 0.5 iteratively until a faultless watermark retrieval can be resolved.

C. EMBEDDING AND EXTRACTION OF WQDCT

Figures 2 and 3 detail the procedures involved in watermark embedding and extraction, respectively. All of the binary images used as watermarks were first permuted using the Arnold transform scrambling algorithm [38] with chaotic mapping [39] to enhance security. The host color image was divided into non-overlapping blocks of size 8x8. QDCT was performed on each color image block, wherein one real component ($A$) and three imaginary components (i.e., $C$, $D$, and $E$) were transformed using Eqs. (6)-(10). If the real component $A$ is the embedding target, then MM is used to embed the watermark bit as in Eqs. (16)-(18). If an imaginary coefficient component is selected, then RM is employed to embed the watermark according to Eq. (19), while the other imaginary components are adjusted per Eq. (20). After all of the watermarks have been embedded, the color block image is reconstructed using inverse QDCT following Eqs. (11)-(15) and all of the pixels are converted to 8-bit integers. Watermark extraction is synchronously performed to ensure that each watermark bit has been correctly embedded in the image block. If all of the watermark bits in the coefficient component of QDCT are correct, then the same procedure is repeated for the next color block. Otherwise, EPA is applied to the image block per Eq. (23).

Watermark extraction involves dividing the watermarked image into non-overlapping color image blocks of size 8x8. QDCT is performed on the blocks and real component $A$, whereupon the three imaginary components $\tilde{C}$, $\tilde{D}$, and $\tilde{E}$ are transformed per Eq. (11). If the real component $A$ is selected, then MM is used to extract the watermark bit in
accordance with Eq. (22). If the watermark bit is embedded in the imaginary coefficient component \( \tilde{C} \) (or \( \tilde{D} \), \( \tilde{E} \)), then RM is used to extract the watermark bit per Eq. (21). The watermark image is then reconstructed using the inverse scrambling algorithm, of course, with the secret key.

**D. PERFORMANCE OPTIMIZATION VIA THE GWO**

The GWO algorithm was inspired by the hunting mechanism and leadership hierarchy of grey wolves, in which the population is divided into four types: Alpha, Beta, Delta, and Omega [25]. In the original GWO, the most suitable solution is the Alpha, followed respectively by Beta, Delta, and Omega. The hunting mechanism involves searching for prey, encircling it, and then attacking. In searching for prey, the GWO seeks to discover new parts of the searching space by applying sudden changes to the solution. In encircling and attacking the prey, the main goal is to improve the estimated solution obtained during the exploration process by discovering the neighborhood of each solution.

In the present work, the representation of wolf Z in the GWO is as follows:

\[
\tilde{X}_z = (x_{z,1}, x_{z,2}, x_{z,3}^{\text{int}}, \ldots, x_{z,7}^{\text{int}}, x_{z,10}^{\text{int}}),
\]

where \( x_{z,1} \) represents the quantization step \( \Delta \) in the MM and \( x_{z,2} \) means the clearance threshold \( \delta \) in the RM. \( x_{z}^{\text{int}} \) stands for the coefficient index of the QDCT component in integer format (e.g., \( x_{z,3}^{\text{int}} \) and \( x_{z,4}^{\text{int}} \) are for the real component; \( x_{z,5}^{\text{int}}, \ldots, x_{z,10}^{\text{int}} \) are for the three imaginary components). The spacing parameters and the embedding coefficient locations determine the overall performance of the watermarking scheme. The optimal set of the parameters can be pursued by designating each candidate as a wolf in the GWO and monitoring a combined objective function \( f(\tilde{X}_z) \) of the imperceptibility and robustness defined as follows:

\[
\begin{align*}
\text{Minimize} & \quad f(\tilde{X}_z) = -f_1(\tilde{X}_z) \times f_2(\tilde{X}_z) \times f_3(\tilde{X}_z) , \\
\text{Subject to} & \quad \alpha \leq f_1(\tilde{X}_z) , \\
& \quad \beta \leq f_2(\tilde{X}_z) , \\
& \quad \varepsilon \leq f_3(\tilde{X}_z) , \\
\text{Variable range} & \quad 10 \leq x \leq 100 , \\
& \quad 2 \leq x^{\text{int}} \leq 64 ,
\end{align*}
\]

where \( \alpha \), \( \beta \), and \( \varepsilon \) respectively represent the acceptable lower bound values of PSNR (peak signal-to-noise ratio), MSSIM (mean structural similarity), and BER (bit error rate). The objective function comprises PSNR and MSSIM (to estimate image distortion between the original and watermarked images) and BER (to determine the

![FIGURE 2. Embedding procedure of the proposed wQDCT.](image1)

![FIGURE 3. Extraction procedure of the proposed wQDCT.](image2)
occurrence of an attack). Functions $f_1(\tilde{X}_z)$, $f_2(\tilde{X}_z)$, and $f_3(\tilde{X}_z)$ respectively denote the results of the PSNR and MSSIM, and BER using the wolf parameter $X_z$. The three functions are defined as follows:

$$f_1(\tilde{X}_z) = \min\left\{\frac{40}{\text{PSNR}(\tilde{I}, \tilde{I})} \right\},$$

(26)

$$f_2(\tilde{X}_z) = \text{MSSIM}(\tilde{I}, \tilde{I}),$$

(27)

$$f_3(\tilde{X}_z) = 1 - \text{BER}(W, \tilde{W}).$$

(28)

In the above equations, all the function results are distributed between 0 and 1. According to Eq. (25), the smaller the objective function, the better performance can be achieved; i.e., larger $f_1(\tilde{X}_z)$, $f_2(\tilde{X}_z)$, and $f_3(\tilde{X}_z)$ are preferable. Generally, PSNR of 40 dB is considered satisfactory; therefore, the fraction term $\frac{\text{PSNR}(\tilde{I}, \tilde{I})}{40}$ represents a normalized baseline of $f_1(\tilde{X}_z)$. The three indicators are defined as follows:

$$\text{PSNR}(\tilde{I}, \tilde{I}) = \frac{1}{O} \sum_{i=1}^{O} 10 \cdot \log_{10} \left\{ \frac{1}{M_i \cdot N_i} \sum_{m=1}^{M_i} \sum_{n=1}^{N_i} (I_{m,n,o} - \tilde{I}_{m,n,o})^2 \right\},$$

(29)

$$\text{MSSIM}(\tilde{I}, \tilde{I}) = \frac{1}{O \cdot M_B \cdot N_B} \sum_{i=1}^{O} \sum_{r=1}^{M_B} \sum_{s=1}^{N_B} \text{SSIM}(B_{r,s,o}, \tilde{B}_{r,s,o}),$$

(30)

$$\text{BER}(W, \tilde{W}) = \frac{1}{N_W \cdot N_W} \sum_{i=1}^{N_W} \sum_{j=1}^{N_W} |w_{i,j} - \tilde{w}_{i,j}|.$$

(31)

where $O$ denotes the number of channels in an RGB color image, which is normally set to 3. $M_i \times N_i$, $M_B \times N_B$, and $N_W \times N_W$ respectively denote the size of the host image, non-overlapping blocks, and watermark image. $I = \{I_{m,n,o}\}$ and $\tilde{I} = \{\tilde{I}_{m,n,o}\}$ respectively represent the host and watermarked images. $B_{r,s,o}$ and $\tilde{B}_{r,s,o}$ respectively represent the host and watermarked blocks. Function $\text{SSIM}(\cdot)$ is used to compute the degree of similarity between the host and watermarked image blocks. $W = \{w_{m,n}\}$ and $\tilde{W} = \{\tilde{w}_{m,n}\}$ respectively represent the original and extracted watermark images [40].

In the proposed wQDCT, the GWO seeks to identify the optimal spacing parameters and coefficient locations with the aim of achieving a suitable tradeoff among imperceptibility, robustness, and payload. The implementation of GWO in wQDCT for blind watermarking color images is illustrated in Fig. 4. Note that the acceptable lower-bounds of PSNR $\alpha$, MSSIM $\beta$, and BER $\varepsilon$ were respectively set at 0.925 (PSNR = 37dB), 0.95, and 0.8. The optimal spacing parameters and coefficient locations obtained through the GWO were then installed in the public algorithm for subsequent watermark extraction. To enhance security, the watermarks were permuted using the Arnold transform [38] with chaotic mapping [39], the encryption keys shall be agreed upon in advance by the sender and receiver.

To seek the optimal spacing parameters and coefficient locations of wQDCT, we examined the outcomes of four tentative images (namely, Lena, Baboon, Avion, and Peppers) in presence of three types of image attacks including JPEG compression with a quality factor (QF) of 30, 1% S&P noise, and histogram equalization while simulating the score of each wolf. For the hardware platform equipped with an Intel® Core(TM) i9-9900K CPU @ 3.60GHz, 32GB RAM, and RTX 2080 graphics card under the Matlab® environment, the average computation time required for each wolf to process the abovementioned four images and three attacks in each iteration is 69.51 seconds. As the number of searching agents (i.e., wolves) is set at 20, the total time required by the GWO for 100 iterations is thus 139,020 seconds. Figure 5 depicts the

![FIGURE 4](image.png)

**FIGURE 4.** The optimization process of the proposed wQDCT.

![FIGURE 5](image.png)

**FIGURE 5.** The convergence curve of the objective function for searching the best parameters by the GWO.
convergence curve of the GWO searching. As seen in Fig. 5, although we adopted the results of the 100th iteration, the objective function converged quickly after 20 iterations.

E. DENOISING CONVOLUTIONAL NEURAL NETWORK (DnCNN)

The DnCNN is the most popular deep learning architecture used for image denoising. The DnCNN consists of a series of Conv, BN, and ReLU layers. Conv refers to a convolutional layer used for the automatic extraction of features. BN denotes batch normalization aimed at improving the speed to convergence in training and reducing the influence of the network on initialization variables. ReLU signifies an activation function commonly used in artificial neural networks. The DnCNN model creates various combinations of the Conv, BN, and ReLU layers to perform image denoising tasks, such as Gaussian denoising, single image super-resolution, and JPEG image deblocking [30].

The role of the DnCNN is not intended to further improve the robustness of the proposed wQDCT. Rather, its function is focused on enhancing the texture of the extracted watermark following a possible attack, thus making the watermark more recognizable. The DnCNN considered here adopts a residual learning strategy to draw the latent clean image from noisy observation. During the training stage, the output of the network is just a residual image, and the optimization goal is to reduce the binary cross-entropy between the actual residual image and network output. When a well-trained DnCNN model is used in the testing phase, the information (i.e., weights and biases) stored in the model will be used to eliminate noise in the input noisy image followed by reconstructing an image closer to the original image. The inputs to the DnCNN were 81,184 image patches of size 64×64 acquired by partitioning binary images of plain-text articles into non-overlapping blocks of prescribed size. It is particularly pointed out that the training data set did not contain the watermarks later used in the testing phase. We trained the modified DnCNN for binary denoising using the following architecture and parameters. Next, noise with three different densities (i.e., 0%, 5%, and 10%) was added to these patches. Depth \( L_D \) was set at 16 and each layer included 64 filters. Figure 6 illustrates the DnCNN network employed to denoise the noisy watermark. We employed adaptive moment estimation (Adam) to optimize the DnCNN with the learning rate set as 0.001 for epochs 1-30, 0.0001 for epochs 31-60, and 0.00005 for epochs 61-210. Note that the modeling parameters involved in the above DnCNN were all tuned offline in advance. Once the DnCNN passed through the training and verification stages, it could cooperate online with the wQDCT to render a clearer and more distinguishable watermark recovered from the watermarked image.

IV. PERFORMANCE EVALUATION OF THE PROPOSED wQDCT

The proposed watermarking algorithm was evaluated in aspects of invisibility, robustness, and embedding capacity. The test dataset in the experiments included 64 different 512×512 24-bit color images obtained from [34]. In this study, we used binary images as the watermarks so that performance evaluation can also be achieved via visual inspection. The sizes of the watermark images were 64×64, 192×64, and 128×128, which respectively corresponds to 1, 2, and 4 pixels per image.
3, and 4-bit embedding in every color image block of size 8x8. The 1-bit watermark images and corresponding scrambled versions are shown in Fig. 7. As shown in Fig. 7, there were similar numbers of “1s” and “0s” in the scrambled watermarks; Nonetheless, the distribution of the scrambled binary bits was different from each other. Figure 8 illustrates the original and watermarked “Lena” images based on the proposed wQDCT. In Fig. 8, the naked eye visual inspection cannot discern whether the image contains a watermark. A variety of attacks were implemented in the experiment to examine the robustness of the proposed watermarking algorithm, as shown in Table II.

The performance of the proposed wQDCT watermarking scheme with the GWO incorporated was compared with that of different transform-based watermarking schemes. To facilitate the exposition in the subsequent discussion, we adopted the three symbols, , , and , to denote the proposed, non-optimized, and optimized watermarking schemes, respectively. embeds the watermark in the (1, 1) DCT coefficient of the luminance channel; resorts to the TLBO to optimize the DCT position and embedding strength; relies on the SVM to predict the watermark hidden in the QDFT domain; explores the PSO to optimize the embedding strength matrix for watermark embedding in the unitary matrix of SVD in the QDCT domain. Following the same denomination principle, we used (including and ) [8], (including and ) [19], and (including and ) [3] to represent 3-bit watermark embedding. represents the scenario of embedding a watermark of size 32x32x3 with entropy selection enabled, while corresponds to the case of embedding a watermark of size 64x64x3 with entropy selection disabled. and signify the use of PSO to optimize QDFT-based watermarking in components and , respectively, and represent similar processes resulting from another QDFT approach. Finally, respectively denote the cases of embedding 1 and 3 watermark bits in each block using the proposed wQDCT.

Table III summarizes the average imperceptibility and robustness obtained from 64 images using various blind watermarking schemes. As shown by the PSNR values in Table III, the image quality of the abovementioned schemes was similar under deliberate control. The PSNR of was good when entropy selection was enabled but dropped when entropy selection was disabled. The resultant MSSIMs generally remain roughly at the same level as seen in the PSNRs. The MSSIM values of most schemes were

![Figure 7](image1.png)

**FIGURE 7.** The watermark images and their 1-bit scrambled version with size: (a) 64x64, (b) 192x64, and (c) 128x128.

![Figure 8](image2.png)

**FIGURE 8.** (a) the original and watermarked Lena images, (b) 1-bit embedding in component A, (c) 3-bit in component A, (d) 4-bit in component A, (e) 1-bit in component C, (f) 3-bit in component C, and (g) 4-bit in component C.
between 0.940 and 0.960; however, those with \(a_{QDCT}^{(PSO)}\) and \(a_{QDCT}^{(PSO)}\), were relatively high, mainly because the watermark is embedded in the mid-frequency DCT coefficient.

Table III also presents the BER results obtained under various attack conditions. These compared watermarking schemes exhibited somewhat different performances when the attack was absent. \(a_{QDCT}^{(PSO)}, a_{QDCT}^{(PSO)}, a_{QDCT}^{(PSO)}, a_{QDCT}^{(PSO)}\) and wQDCT were able to extract the watermark perfectly; however, the others ended up with slightly defective watermarks due to out-of-range errors. In cropping image attacks (A1 and A2) where 25% of a watermarked image was missing, BER values close to 12.5% were normally acceptable. The BER value of \(a_{QDCT}^{(PSO)}\) appeared unreasonably high, because the BER has begun to accumulate even in the absence of an attack. \(a_{QDCT}^{(PSO)}\) merely chose certain parts of image blocks to embed the watermark; therefore, the BER value was relatively small.

In case of 1-bit watermark embedding, \(a_{QDCT}^{(PSO)}\) survived JPEG2000 compression (C1-C4), compressive sensing (CS) (D4), Gaussian noise (E1-E2), S&P noise (E3-E4), speckle noise (E5), Gaussian low-pass filter (F2), unsharp filter (F3), brightening (G1), darkening (G2), spotlight (G3), histogram equalization (H1), and rotation (H3) attacks. \(a_{QDCT}^{(PSO)}\) appeared more resistant to JPEG compression (B1-B4), JPEG2000 compression (C4), CS (D1-D3), Gaussian low-pass filter (F2), unsharp filter (F3), spotlight (G3), histogram equalization (H1), and scaling (H2) attacks than \(a_{QDCT}^{(PSO)}\), but worse than \(a_{QDCT}^{(PSO)}\) in other attacks. \(a_{QDCT}^{(PSO)}\) was better than \(a_{QDCT}^{(PSO)}\) in withstanding most attacks, but not as good as \(a_{QDCT}^{(PSO)}, a_{QDCT}^{(PSO)}, a_{QDCT}^{(PSO)}\), and \(a_{QDCT}^{(PSO)}\) could resist JPEG compression (B1-B4), JPEG2000 compression (C1-C4), CS (D2-D3). Gaussian noise (E1), S&P noise (E3), speckle noise (E5), median filter (F1), Gaussian low-pass filter (F2), adaptive Wiener filter (F4), brightening (G1), darkening (G2), spotlight (G3), scaling (H2), and rotation (H3); however, they (i.e., \(a_{QDCT}^{(PSO)}\) and \(a_{QDCT}^{(PSO)}\) were unable to resist unsharp filtering (F3) and histogram equalization (H1) attacks. Between these two compared objects, \(a_{QDCT}^{(PSO)}\) performed slightly better. In contrast to \(a_{QDCT}^{(PSO)}\), \(a_{QDCT}^{(PSO)}\) also exhibited exceptional robustness against most attacks listed in Table II, except for JPEG compression (B1-B4).
For those implemented with 3-bit watermark embedding, the $\sigma_{\text{wQDCT}}^3$ with entropy selection outperformed the $\sigma_{\text{wQDCT}}^3$ without entropy selection involved. The performance of $\sigma_{\text{wQDCT}}^3$ was also superior to that of $\bar{\sigma}_{\text{QDFT14}}^3$ and $\bar{\sigma}_{\text{QDFT20}}^3$ under most attack scenarios except JPEG2000 compression (C1-C4), CS (D3-D4), Gaussian noise (E1), brightening (G1), and spotlight (G3). Nonetheless, the payload capacity of $\sigma_{\text{wQDCT}}^3$ was only a quarter of that of $\bar{\sigma}_{\text{QDFT14}}^3$ and $\bar{\sigma}_{\text{QDFT20}}^3$ due to selective embedding. In most cases, $\sigma_{\text{wQDCT}}^3$ was better than $\bar{\sigma}_{\text{QDFT14}}^3$, except for CS with a higher ratio measurement (D2-D4) in component C. Also, except for JPEG (B1-B4), JPEG2000 compression (C1-C4), and CS (D1-D4), the watermark embedded in component C exhibited more robustness against attacks than that in component A. Unlike $\sigma_{\text{wQDCT}}^3$, $\bar{\sigma}_{\text{QDCT}}^3$ was able to resist JPEG compression (B2-B4), JPEG2000 compression (C1-C4), CS (D2-D4), Gaussian noise (E1), speckle noise (E5), Gaussian low-pass filter (F2), brightening (G1), darkening (G2), spotlight (G3), and rotation (H3) attacks. $\bar{\sigma}_{\text{QDCT}}^3$ proved ineffective against JPEG compression (B1-B4) attack; however, it often took the lead among the compared schemes in the presence of other attacks.

Note that the CS attack (D1-D4) considered in the experiment is the version developed by Metzler et al. [41]. As long as the CS was assigned with higher ratio measurements, all compared watermarking schemes were able to withstand the CS attack. However, the proposed wQDCT showed effective resistance against the CS with mid-to-high ratio measurements whether in the cases of $\bar{\sigma}_{\text{QDCT}}^3$ and $\bar{\sigma}_{\text{QDCT}}^3$ with 1-bit watermark embedding or $\bar{\sigma}_{\text{QDCT}}^3$ with 3-bit watermark embedding.

Table III also presents the results after the employment of wQDCT and DnCNN. For convenience, we used the abbreviation “wQDCT-D” to signify the presence of DnCNN during watermark extraction. Except for the extreme cases where the BER was nearly 0 or around 50%, the wQDCT-D considerably reduced the BER values attainable by the wQDCT with the highest reach around 10%. Nevertheless, the DnCNN might contribute a slight degradation of BER if the retrieved watermark was already error-free. In the category of wQDCT-D, both $\sigma_{\text{QDCT}}^3$ and $\bar{\sigma}_{\text{QDCT}}^3$ suffered a tiny loss of BER from 0.00% to 0.01%.

The normalized cross-correlation (NCC) metric represents another aspect of the robustness of the watermarking scheme against various attacks [42]. The larger the NCC value is, the better the robustness. Table IV lists the NCC results obtained under various attack conditions.
conditions. The NCC exhibited a similar tendency roughly equal to $1 - 2 \times BER$. Overall, the proposed wQDCT-D outperformed other compared schemes in most attacks. After applying the DnCNN, the wQDCT-D consistently showed better NCC values.

To conclude the above discussion, we note that $\tilde{\omega}_{QDCT}^{(SYM)}$ uses a random number matrix to enhance security but the random number matrix leads to a considerable impact on different images. $\tilde{\omega}_{QDCT}^{(UPD)}$ and $\tilde{\omega}_{QDCT}^{(PLOD)}$ cannot be as robust as DCT-based watermarking ($\tilde{\omega}_{QDCT}^{(HI)}$ and $\tilde{\omega}_{QDCT}^{(PLFO)}$) despite that these two are developed based on quaternion. Watermarking in the DCT domain demonstrates better robustness than that in other transform domains (i.e., FFT, DFT, and DWT). $\tilde{\omega}_{QDCT}^{(HI)}$ embeds the watermark in the low-frequency DCT coefficient, leading to the consequence that it cannot resist low-frequency attacks such as unsharp filtering (F3) and histogram equalization (H1). By contrast, $\tilde{\omega}_{QDCT}^{(PLFO)}$ embeds the watermark in the mid-frequency range, thus surviving the unsharp filtering (F3) and histogram equalization (H1). Amongst the quaternion transforms, QDCT retains the intrinsic characteristics of DCT, which make QDCT better than QDFT ($\tilde{\omega}_{QDFT}^{(HI)}$ and $\tilde{\omega}_{QDFT}^{(PLFO)}$) in terms of robustness against most attacks. QDFT resembles QDCT in the sense that they both hold higher energy for the real components at low frequencies and their imaginary components at mid-frequencies are comparable. For the same sake, the proposed wQDCT has an excellent performance in real or imaginary components. Watermarks embedded in imaginary components of the proposed wQDCT ($\tilde{\omega}_{QDCT}^{(HI)}$ and $\tilde{\omega}_{QDCT}^{(PLFO)}$) are vulnerable to JPEG compression attacks. The cause can be attributed to the fact that the JPEG takes effect on the luminance (i.e., Y), blue-difference chroma (i.e., Cb), and red-difference chroma (i.e., Cr) instead of the R, G, and B channels. In accordance with the conversion formula, the luminance is just the weighted sum gathered from RGB color primaries. As the production of the real component in QDCT (as in Eq. (7)) bears a resemblance to the RGB-to-YCbCr conversion, the watermark embedded in the real component remains secured as long as the embedding scheme can sustain the luminance modification. By contrast, Cb and Cr are the results of calculating the difference across color channels. Changes in Cb and Cr will be adversely propagated to the imaginary components, as the imaginary component is obtained by subtracting two DCT coefficients drawn from different color channels. This explains why the real component can withstand JPEG compression attacks, but the imaginary component cannot. In summary, except for

| Attack Type | $R_{QDCT}^{(HI)}$ | $R_{QDCT}^{(PLFO)}$ | $T_{QDCT}^{(HI)}$ | $T_{QDCT}^{(PLFO)}$ | $F_{QDCT}^{(HI)}$ | $F_{QDCT}^{(PLFO)}$ |
|-------------|------------------|------------------|------------------|------------------|------------------|------------------|
| None        | 0.00             | 0.68             | 0.00             | 0.68             | 0.00             | 0.68             |
| A1          | 12.60            | 3.83             | 12.40            | 3.59             | 13.19            | 3.72             |
| A2          | 12.59            | 3.79             | 12.49            | 3.83             | 13.07            | 3.74             |
| B1          | 30.05            | 53.79            | 49.09            | 50.89            | 42.31            | 51.74            |
| B2          | 4.38             | 3.75             | 49.37            | 49.34            | 37.15            | 34.20            |
| B3          | 1.50             | 1.19             | 47.76            | 48.50            | 36.14            | 31.69            |
| B4          | 0.56             | 0.75             | 37.75            | 35.58            | 31.43            | 24.28            |
| C1          | 0.06             | 0.68             | 0.04             | 0.68             | 1.20             | 0.60             |
| C2          | 0.73             | 0.72             | 0.57             | 0.70             | 1.82             | 0.73             |
| C3          | 1.74             | 0.98             | 2.44             | 1.02             | 3.80             | 1.05             |
| C4          | 3.50             | 1.81             | 4.81             | 2.82             | 6.08             | 2.84             |
| D1          | 32.97            | 27.18            | 29.18            | 21.98            | 30.21            | 23.32            |
| D2          | 6.67             | 5.63             | 7.08             | 5.39             | 7.30             | 5.38             |
| D3          | 2.00             | 1.24             | 2.09             | 1.26             | 2.18             | 1.30             |
| D4          | 0.20             | 0.70             | 0.11             | 0.69             | 0.14             | 0.69             |
| E1          | 1.98             | 1.30             | 1.06             | 0.84             | 2.30             | 0.88             |
| E2          | 32.47            | 25.97            | 21.97            | 12.55            | 24.60            | 15.44            |
| E3          | 12.74            | 4.79             | 8.15             | 2.58             | 10.09            | 3.10             |
| E4          | 39.83            | 36.69            | 26.24            | 17.64            | 29.55            | 21.90            |
| E5          | 6.81             | 3.34             | 4.23             | 1.69             | 5.81             | 1.93             |
| F1          | 17.94            | 9.62             | 2.88             | 1.50             | 7.37             | 2.34             |
| F2          | 3.75             | 3.49             | 0.06             | 0.69             | 1.90             | 0.79             |
| F3          | 3.59             | 2.53             | 0.08             | 0.69             | 9.04             | 2.42             |
| F4          | 13.92            | 8.33             | 0.54             | 0.75             | 4.60             | 1.34             |
| G1          | 0.55             | 0.94             | 0.31             | 0.74             | 1.51             | 0.78             |
| G2          | 3.33             | 1.47             | 2.01             | 1.24             | 3.57             | 1.26             |
| G3          | 1.43             | 0.89             | 0.67             | 0.79             | 1.99             | 0.82             |
| H1          | 25.40            | 17.55            | 0.75             | 0.70             | 8.15             | 2.35             |
| H2          | 18.77            | 9.84             | 0.68             | 0.79             | 6.19             | 1.71             |
| H3          | 0.76             | 0.96             | 0.08             | 0.69             | 1.29             | 0.70             |
| Mean        | 8.75             | 6.79             | 9.89             | 8.71             | 10.41            | 7.33             |
| PSNR        | 38.12            | 38.03            | 38.03            | 38.08            | 38.08            | 38.10            |
| MSSIM       | 0.943            | 0.945            | 0.944            | 0.944            | 0.944            | 0.944            |

Table V: Comparison results of BER with and without DnCNN of the proposed wQDCT.
JPEG compression attacks, watermarks embedded in the imaginary components are more reliable than those embedded in the real component. The incorporation of the DnCNN with the wQDCT will render better BERs, which subsequently allows the extracted watermarks to be more distinguishable.

V. WATERMARK ENHANCEMENT THROUGH DENOISING

In this experiment, a 4-bit watermark composition was adopted to demonstrate the competence of the DnCNN and to explore the feasibility of embedding the watermark into different sorts of components. Table V lists the BERs of $\tilde{F}^{A(GWO)}_{QDCT}$, $\tilde{F}^{B(GWO)}_{QDCT}$, $\tilde{F}^{C(GWO)}_{QDCT}$, $\tilde{F}^{D(GWO)}_{QDCT}$, and $\tilde{F}^{E(GWO)}_{QDCT}$ obtained from the wQDCT and wQDCT-D. Note that the $uAvC$ appearing in the denomination of $\tilde{F}^{A(GWO)}_{QDCT}$ reflects the situation that $u$ bits reside in real component $A$ and $v$ bits are in imaginary component $C$. The PSNR and MSSIM values for the five combinations in Table V were similar. The proposed wQDCT-D was shown to suppress the BER especially for the range between 10% and 40%. However, when the BER was above 40% or below 1%, the performance of the DnCNN deteriorated. The reason is due

![FIGURE 9](https://example.com/figure9.png)

**FIGURE 9.** The extracted watermark images and BERs of the $\tilde{F}^{A(GWO)}_{QDCT}$ before and after denoising (left blue box: extracted watermark, right cyan box: denoised watermark).
to that there was hardly any useful information left in the watermark once the BER exceeded 40%. On the other hand, artifact errors tended to occur in case the BER fell below 1%. The DnCNN improved the performance of $P_{QDCT}^{14AC(GWO)}$ and $P_{QDCT}^{14AC(GWO)}$, and it distinctly improved the performance of $P_{QDCT}^{14AC(GWO)}$ and $P_{QDCT}^{14AC(GWO)}$ for the BERs between 1% and 40%. The results in Table V suggest that when multiple watermark bits are embedded, allocating slightly more bits in the real component than in the imaginary component (e.g., $P_{QDCT}^{14AC(GWO)}$ and $P_{QDCT}^{14AC(GWO)}$) appears a wiser choice. Finally, Figure 9 illustrates the effect due to the incorporation of DnCNN. Within each grid, the left half image highlighted with a blue box shows the watermark extracted from the wQDCT with Lena, while the right half enclosed by a cyan box displays the result of $P_{QDCT}^{14AC(GWO)}$ after denoising. Clearly, most of the text images are easier to recognize after undergoing all attacks except for JPEG compression.

VI. CONCLUSION

This paper presents a QDCT-based watermarking scheme jointly exploiting the EPA, MM, and RM, and artificial intelligence techniques GWO and DnCNN. The GWO enables the proposed wQDCT scheme to identify the optimal control parameters as well as QDCT coefficient locations to achieve a satisfactory performance between robustness and imperceptibility. The DnCNN acts as an auxiliary appliance to refine the extracted watermark. In a series of experiments, the proposed wQDCT-D scheme not only demonstrated superior robustness but also enhanced the comprehensibility of the retrieved watermarks. The watermark embedding in the real component of QDCT proved effective in withstanding JPEG compression, while embedding in the imaginary components (i.e., $C$, $D$, and $E$) proved effective against unsharp filtering and histogram equalization. The distributive embedding in the real and imaginary components makes it possible to defend against a variety of attacks when embedding multiple watermarks. Overall, the proposed scheme outperforms other DCT-based schemes in terms of robustness when the embedding strength and payload capacity are set at the same level.

REFERENCES

[1] H.-T. Hu and L.-Y. Hsu, “Robust, transparent and high-capacity audio watermarking in DCT domain,” Signal Processing, vol. 109, pp. 226-235, 2015.
[2] W. Huan, S. Li, Z. Qian, and X. Zhang, “Exploring Stable Coefficients on Joint Sub-bands for Robust Video Watermarking in DT CWT Domain,” IEEE Transactions on Circuits and Systems for Video Technology, pp. 1-1, 2021.
[3] L.-Y. Hsu and H.-T. Hu, “Blind watermarking for color images using EMMQ based on QDFT,” Expert Systems with Applications, vol. 149, p. 113325, 2020/07/01/ 2020.
[4] M. R. Khosravi and S. Samadi, “Efficient payload communications for IoT-enabled VISAR vehicles using discrete cosine transform-based quasi-sparse bit injection,” EURASIP Journal on Wireless Communications and Networking, vol. 2019, p. 262, 2019/11/29 2019.
[5] S. Jiao, C. Zhou, Y. Shi, W. Zou, and X. Li, “Review on optical image hiding and watermarking techniques,” Optics & Laser Technology, vol. 109, pp. 370-380, 2019/01/01/ 2019.
[6] H. Luo, Z. Lu, and J. Pan, “A Reversible Data Hiding Scheme for 3D Point Cloud Model,” in 2006 IEEE International Symposium on Signal Processing and Information Technology, 2006, pp. 863-867.
[7] Y. Luo, L. Li, J. Liu, S. Tang, L. Cao, S. Zhang, et al., “A multi-scale image watermarking based on integer wavelet transform and singular value decomposition,” Expert Systems with Applications, vol. 168, p. 114272, 2021.
[8] P. Ermawan and M. N. Kahir, “A Robust Image Watermarking Technique With an Optimal DCT-Psychovisual Threshold,” IEEE Access, vol. 6, pp. 20464-20480, 2018.
[9] S. Byun, H. Son, and S. Lee, “Fast and Robust Watermarking Method Based on DCT Specific Location,” IEEE Access, vol. 7, pp. 100706-100718, 2019.
[10] L.-Y. Hsu and H.-T. Hu, “Robust blind image watermarking using crosstalk inter-block prediction in the DCT domain,” Journal of Visual Communication and Image Representation, vol. 46, pp. 33-47, 2017.
[11] H.-J. Ko, C.-T. Huang, G. Horng, and S.-J. Wang, “Robust and blind image watermarking in DCT domain using inter-block coefficient correlation,” Information Sciences, vol. 517, pp. 128-147, 2020/05/01/ 2020.
[12] T. K. Araghi and A. A. Manaf, “An enhanced hybrid image watermarking scheme for security of medical and non-medical images based on DWT and 2-D SVD,” Future Generation Computer Systems, vol. 101, pp. 1223-1246, 2019/12/01/ 2019.
[13] D. Rajani and P. R. Kumar, “An optimized blind watermarking scheme based on principal component analysis in redundant discrete wavelet domain,” Signal Processing, vol. 172, p. 107556, 2021/07/01/ 2020.
[14] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, “A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, pp. 776-786, 2003.
[15] M. Zhang, W. Li, Q. Du, L. Gao, and B. Zhang, “Feature Extraction for Classification of Hyperspectral and LiDAR Data Using Patch-to-Patch CNN,” IEEE Trans. on Cybernetics, vol. 50, pp. 100-111, 2020.
[16] B. Lei, D. Ni, S. Chen, T. Wang, and F. Zhou, “Optimal image watermarking scheme based on chaotic map and quaternion wavelet transform,” Nonlinear Dynamics, vol. 78, pp. 2897-2907, 2014.
[17] P.-p. Niu, L. Wang, X. Shen, S.-y. Zhang, and X.-y. Wang, “A novel robust image watermarking in quaternion wavelet domain based on superpixel segmentation,” Multidimensional Systems and Signal Processing, vol. 31, pp. 1509-1530, 2020/10/01/ 2020.
[18] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, “A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram,” Soft Computing, vol. 22, pp. 47-65, January 01, 2018.
[19] B. Chen, G. Coatireux, G. Chen, X. Sun, J. L. Coatireux, and H. Shu, “Full 4-D quaternion discrete Fourier transform based watermarking for color images,” Digital Signal Processing, vol. 28, pp. 106-119, 2014.
[20] B. Chen, C. Zhou, B. Jeeun, Y. Zheng, and J. Wang, “Quaternion discrete fractional random transform for color image adaptive watermarking,” Multimedia tools and applications, vol. 77, pp. 20809-20837, 2018/08/01/ 2018.
[21] A. Baykasoglu, A. Hamzadayi, and S. Y. Köse, “Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases,” Information Sciences, vol. 276, pp. 204-218, 2014/08/20/ 2014.
[22] T. Joachims, “Making large-scale SVM learning practical,” Technical report1998.
[23] C.-P. Wang, X.-Y. Wang, C. Zhang, and Z.-Q. Xia, “Geometric correction based color image watermarking using fuzzy least squares support vector machine and Bessel K form distribution,”
