Multimedia Tools and Applications (2024) 83:29969–29995
https://doi.org/10.1007/s11042-023-16649-3

Convolutional neural network and 2D logistic-adjusted-Chebyshev-based zero-watermarking of color images

Mohamed M. Darwish[1] · **Amal A. Farhat**[1] · **T. M. El-Gindy**[2]

Received: 17 August 2022 / Revised: 20 July 2023 / Accepted: 23 August 2023 /
Published online: 14 September 2023

Abstract

Robust zero-watermarking is a protection of copyright approach that is both effective and distortion-free, and it has grown into a core of research on the subject of digital watermarking. This paper proposes a revolutionary zero-watermarking approach for color images using convolutional neural networks (CNN) and a 2D logistic-adjusted Chebyshev map (2D-LACM). In this algorithm, we first extracted deep feature maps from an original color image using the pre-trained VGG19. These feature maps were then fused into a featured image, and the owner's watermark sequence was incorporated using an XOR operation. Finally, 2D-LACM encrypts the copyright watermark and scrambles the binary feature matrix to ensure security. The experimental results show that the proposed algorithm performs well in terms of imperceptibility and robustness. The BER values of the extracted watermarks were below 0.0044 and the NCC values were above 0.9929, while the average PSNR values of the attacked images were 33.1537 dB. Also, it is superior to other algorithms in terms of robustness to conventional image processing and geometric attacks.

Keywords Zero-Watermarking · Convolutional Neural Network · Chebyshev map

1 Introduction

Image security has grown increasingly critical as information and communication technologies have simplified digital image transmission and transfer. The technology of digital image watermarking is an effective technique to protect privacy and intellectual property protection [26, 50], content authentication [5], ensuring owner identification [32, 34], and digital image validity [8, 16, 18, 21]. The goal of traditional watermarking algorithms is to create a protected image by inserting information of the watermark into the digital image [1, 12, 24, 31, 59]. As well as this, these algorithms have several drawbacks that

✉ Amal A. Farhat
amal_bio@aun.edu.eg

[1] Department of Computer Science, Faculty of Computers and Information, Assiut University, Assiut, Egypt

[2] Department of Mathematics, Faculty of Science, Assiut University, Assiut, Egypt

are challenging for traditional watermarking methods, for example, embedded information contaminates the original image data, while traditional digital watermarking distorts the watermarked image. In some applications, image distortion is undesirable, such as medical diagnosis, artwork scanning, and military imaging systems. Moreover, due to its inherent contradictions, this strategy is challenging to balance in terms of robustness and imperceptibility.

Using zero-watermarking technology now increases copyright protection for digital multimedia information and visual quality, especially images, to solve the embedding watermarking problem. In the zero-watermarking approach [11, 28, 51, 56–58], the watermark sequence is logically associated with the original image instead of being physically embedded in it, which maintains the image's integrity. Therefore, it has a high level of imperceptibility. Zero-watermarking techniques have the following advantages: (1) Good imperceptibility due to these techniques preserves the quality of the original images without changing; (2) A proper balance of robustness, imperceptibility, and capacity; (3) Copyright authentication authority is involved in zero-watermarking techniques.

Zero-watermarking takes some intrinsic information from the host image instead of applying a sequence of watermarks. The watermark sequence of the owner connects these inherent properties to create a master share that is securely stored [35, 39, 60]. The owner can demonstrate ownership of the protected images using the zero watermark, which can be conveyed via any unsecured public communication channel. This is done by using the intrinsic features and the master share that were extracted from the analyzed image. Extraction of the host image's significant intrinsic features is the most significant problem for the zero-watermarking technique's desirable performance.

Based on the significant features of the image [19, 51], the approaches of zero-watermarking can be divided into four categories: features in the spatial domain based [2, 3], features in the frequency domain based [42, 43, 53], moments features based [10, 36], and CNN features based methods [9, 14].

In the first category, spatial domain features are directly used to obtain imagine features. However, when geometric and image processing attacks occur, spatial domain features exhibit great sensitivity regardless of the use of edge or texture information [2, 3]. The image features in the second category are constructed using frequency-domain features. The features in the frequency domain, on the other hand, suffer from a lack of invariance of rotation and scaling, resulting in poor performance [42, 43, 53]. Image features in the third category are generated using moments and moments invariants that have helpful invariance properties [10, 36]. In the fourth category, features for the color image are extracted from the CNN layers by merging a deep feature map to form the feature image [9, 14].

Sun et al. [40] presented an algorithm of zero-watermarking by using a combination of the quantization embedding role, the generalized Arnold transform, and the spread spectrum technique. Using orthogonal Fourier-Mellin moments (OFMMs), Shao et al. [37] introduced a robust double scheme of zero-watermarking to protect the copyright of two images simultaneously. Thanh et al. [41] introduced a robust algorithm of zero-watermarking by using a QR decomposition, and a visual map featuring their permutation features to reduce the computational cost and improve the robustness. Liu et al. [29] introduced a zero-watermarking technique that has higher robustness to geometric attacks. Through this algorithm, a timestamp is added to the sequence of watermark images to solve the problems caused by attacks of interpretation. Later, researchers developed a zero-watermarking technique using polar complex exponential transforms (PCETs) and logistic maps [47]. In addition, radial harmonic Fourier moments (RHFMs) in ternary representation are employed to create an algorithm of zero-watermarking for stereo images [48]. Using a modified logistic map and SCA, Daoui et al. [7] suggested strong image encryption as well as a zero-watermarking approach. Image zero-watermarking and encryption are integrated in this approach, to provide a higher level of security while sending images over the internet.

Xia et al. [55] suggested an algorithm for zero-watermarking using fractional-order RHFMs for lossless copyright protection of the medical gray-scale images. Hosny and Darwish [19] proposed a new scheme for zero-watermarking using multi-channel orthogonal Legendre-Fourier moments of fractional orders (MFrLFMs) to deal with color images. Based on new multi-channels shifted Gegenbauer moments of fractional orders (FrMGMs), Hosny and Darwish [17] presented a zero-watermarking scheme to protect the color images in the medical field. Roček et al. [33] presented an assessment analysis of zero-watermarking approaches ensuring the integrity and authorship of medical images.

The purpose of this research is to evaluate the effectiveness of chosen so-called zero-watermarking approaches in protecting the integrity and verifying the authorship of these medical image investigations. Wang et al. [46] proposed an octonion orthogonal moments theory applicable to zero-watermarking and color stereoscopic images. Xia et al. [54] proposed a zero-watermarking approach based on novel quaternion PCETs for color images. Gao et al. [13] combined PCETs with a self-organizing map and deep CNN to introduce a video zero-watermarking. Han et al. [15] presented a federated learning-based approach of zero-watermarking for protecting healthcare data. Hu et al. [20] developed an algorithm of zero-watermarking, which effectively protects the copyright of medical images and detects the tampering regions simultaneously. Ma et al. [30] utilized the ternary polar complex exponential transform and the chaotic system to propose an algorithm for zero-watermarking to protect two medical images simultaneously. Based on accurate quaternion generalized OFMMs, Wang et al. [49] introduced a color images zero-watermarking approach.

A new direction of research in zero-watermarking is presented by Fierro-Radilla [9] recently where they presented a zero-watermarking technique using CNN. According to [9], Han et al. [14] used VGG19 deep convolution neural network to introduce a robust algorithm for zero-watermarking.

Despite massive research work devoted to zero-watermarking, most present approaches have some issues and limitations. Some issues with existing zero-watermarking approaches can be summarized as follows:

(1) Most of these approaches show less resistance against geometric attacks.
(2) Most of these approaches show less resistance to combining common signal-processing attacks with geometric attacks.
(3) Approaches have features that are extracted in the frequency domain, these features are not robust to geometric attacks. Furthermore, their applicability and time complexity are inferior.
(4) While most traditional approaches for zero-watermarking are typically suitable for grayscale image protection, color image protection is more common.
(5) Most approaches based on moments for zero-watermarking usually use the approximation method to compute these moments, which suffer from instability, are inaccurate, and are inefficient, which has a considerable influence on the performance of these moment-based approaches for the extraction of the features of host images.

Taking the above challenges into consideration and to overcome the significant limitations listed above, in this paper, we present a new zero-watermarking algorithm for color images using the VGG19 deep convolution neural network [14] which has excellent intrinsic properties. When compared to other forms of VGG19, CNNs improve network depth and use an alternate structure with numerous nonlinear activation layers and convolution layers [25]. This is beneficial for extracting precise features. As a result, and the preprocessing method, to extract deep feature maps from color images, we simply use the max pooling layers and convolution layers from the pre-trained VGG19, as opposed to image classification tasks. In contrast to other zero-watermarking methods, the proposed algorithm can extract high-level features from color images to increase anti-geometric zero-watermarking's attack capability. Additionally, a chaotic system based on a Chebyshev map known as 2D-LACM [27] is used to ensure the proposed algorithm's high level of security. The overall contributions of this work can be listed as:

- The proposed algorithm used a deep conventional neural network called VGG19 to extract the robust essential features of the host color images for zero-watermarking.
- The proposed algorithm achieves higher resistance against geometric attacks and common signal-processing attacks.
- To enhance the zero-watermarking security, this paper uses a novel approach (2D-LACM) [27] to scramble the feature matrix and binary logo image.
- Extensive experiments are conducted using standard color images to verify that the proposed technique has high resilience against various attacks and outperforms algorithms of zero-watermarking.

The proposed approach consists of four main stages. *First*, the feature map for the original color image is created during the CNN (VGG19) training process using the output of the second fully connected layer (fc_2); *then*, using a security chaos sequence created by a 2D-LACM, the extracted features are transformed to the binarized features. *Next*, the owner's watermark sequence is combined with the binarized features. *Finally*, an XOR operation is performed on the encrypted version of the binarized features of the image and the encrypted version of the binary watermark digits to create a verification key of ownership, also known as a zero-watermark.

We prove experimentally that our algorithm can successfully resist various types of attacks and this algorithm outperformed other algorithms.

The rest of the paper is organized as follows. Section 2 presents the definitions of extraction of VGGnet features, then gives 2D-LACM. Section 3 describes the suggested zero-watermarking scheme in detail, and in Section 4, we present the experimental results in detail. Finally, Section 5 concludes the paper.

2 Preliminaries

This section explains VGGnet feature extraction and the 2D-LACM in detail.

2.1 Extraction of VGGnet feature

VGGnet is a deep CNN type, that is frequently used in learning and feature extraction techniques [9]. VGG19, which Simonyan and Zisserman established in 2015, is the most commonly used VGGnet [38], which comprises 3 fully linked layers and 16 convolution layers (19 hidden layers). Convolution layers are used by VGG19 to extend the number of feature channels and extract image features using a sequence of 3×3 convolution kernels. If T_j and δ_j are the weights and biases of the jth layer of convolution, the features may be retrieved as follows:

$$Z_j^{out} = S\left(T_j * Z_j^{in} + \delta_j\right), \tag{1}$$

where Z_j^{in} and Z_j^{out} indicate the feature maps input and output, respectively, and S denotes the rectified linear unit (ReLU). The stride in each layer of convolution is set to 1. VGG19 adopts layers of max pooling to reduce the size of the feature map to avoid calculation explosion. It is possible to translate the representation of the feature to the sample label space by connecting each node of the given layer to every node of the preceding layers in the fully linked layers as follows:

$$L = F_3\left(F_2\left(F_1\left(MP\left(Z_{16}^{out}\right)\right)\right)\right). \tag{2}$$

Where MP signifies the max-pooling operation and F_k means the operation of the k^{th} fully linked layer. A SoftMax layer generates the image categorization outcome at the end of VGG19:

$$L_i = \frac{e^{x_i}}{\sum_{c=1}^{C} e^{x_c}}, \tag{3}$$

in which C is the classification number, x_i is the output of the i^{th} node and L_i is the probability of the i^{th} node.

In our zero-watermark approach, we use the VGG19 network architecture shown in Fig. 1. The feature map was extracted using the second layer's output, as depicted in Fig. 1.

2.2 Chebyshev map with a two-dimensional logistic adjustment

The logistic map is described as follows:

$$u_{i+1} = \alpha u_i(1 - u_i), \tag{4}$$

where $u_i \in [0, 1]$ and $\alpha \in [0, 4]$. The Logistic map behaves chaotically when $\alpha \in (3.569945972, ..., 4]$.

The Chebyshev map is a one-parameter chaotic low-dimensional system. This chaotic map's mathematical expression is given in Eq. (5).

$$u_{i+1} = cos\left(\mu cos^{-1}\left(u_i\right)\right), \tag{5}$$

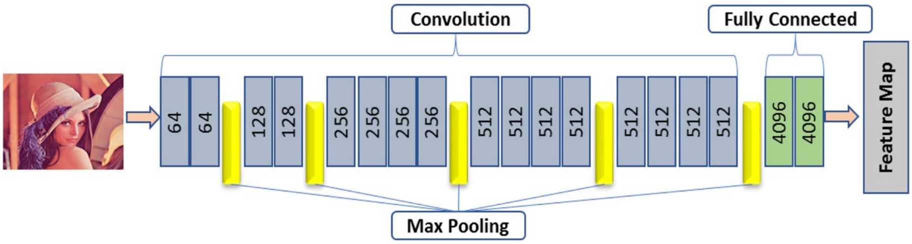


Fig. 1 The VGG19 network architecture utilized to extract the feature map

where μ is the Chebyshev map's control parameter, and when μ is larger than one, this map begins to display chaotic behavior. The mathematical expression of the 2D-LACM is given in [27] as follows:

$$\begin{cases} u_{i+1} = \pi e^{(\beta\times u_i\times(1-u_i)+v_i)} cos^{-1}\left(u_i\right) mod 1, \\ v_{i+1} = \pi e^{(\beta\times v_i\times(1-v_i)+u_{i+1})} cos^{-1}\left(v_i\right) mod 1. \end{cases} \tag{6}$$

In Eq. (6), β is the enhanced chaotic map's control parameter, which corresponds to the range [0, 4]. To improve the security of zero-watermarking, three chaotic sequences created by 2D-LACM are employed in this study to encrypt the watermark and scramble the binary feature sequences. To produce chaotic sequences, the secret key refers to the starting states (u_0, v_0) and parameter β of 2D-LACM.

3 Zero-watermarking algorithm for color image

The suggested technique is divided into two stages: generation of zero-watermarks and verification. The objective of zero-watermark generation is to utilize the essential features of the host image to produce a zero-watermark, and the goal of zero-watermark verification is to authenticate the original image's copyright. First, we discuss the encryption of the watermark in the proposed algorithm before describing the two steps in detail.

3.1 2D-LACM based encryption

The architecture for applying 2D-LACM in the proposed approach is depicted in Fig. 2. Encryption of a watermark (seen in the bottom portion of Fig. 2) randomly confuses the coordinates of a pixel and modifies a binary watermark image's bit values using bit operation diffusion and pixel-level scrambling. The process for encryption of the watermark via 2D-LACM is as follows, assuming the watermark W is $P \times Q$ in size:

(1) The chaotic system (6) is performed for $P \times Q$ iterations using the secret keys $SK_1 = (u_0^1, v_0^1, \beta^1)$ and obtain $P \times Q$ values of v_{i+1}.

(2) It is possible to generate a chaotic decimal sequence CS_1 of length $P \times Q$. Similarly, the chaotic decimal sequences CS_2 and CS_3 are constructed using $SK_2 = (u_0^2, v_0^2, \beta^2)$ and $SK_3 = \left(u_0^3, v_0^3, \beta^3\right)$.

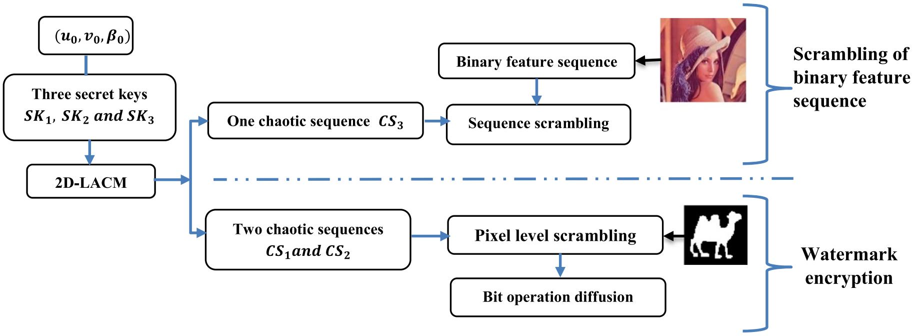

(u_0, v_0, β_0)
Three secret keys SK_1, SK_2 and SK_3
2D-LACM
One chaotic sequence CS_3
Binary feature sequence
Sequence scrambling
Scrambling of binary feature sequence
Two chaotic sequences $CS_1$and CS_2
Pixel level scrambling
Bit operation diffusion
Watermark encryption

Fig 2. 2D-LACM framework application in zero-watermarking

(3) According to the following equation, the chaotic sequence in decimal representation CS_2 is turned into a binarized chaotic sequence.

$$CS_2(i) = \{1, when\, CS_2(i) \geq ME, 0, when\, CS_2(i) < ME, 1 \leq i \leq P \times Q. \tag{7}$$

wherein ME is the average of CS_2.

(4) Based on the pixel location in CS_1, CS_1 is sorted ascendingly, and the associated index vector $L = (L_1, L_2, \ldots, L_{P\times Q})$ is computed.

(5) To acquire the shuffled watermark W_c, the watermark, W is rearranged as a vector W_b of length, $P \times Q$, and then scrambled at the level of pixel based on the index vector's elements L.

(6) XOR in the bit level is performed on the shuffled watermark W_c to obtain the encrypted version W_{sc} using the formula below:

$$W_{sc} = W_c \oplus CS_2. \tag{8}$$

(7) To generate the $P \times Q$ sized image of the encrypted watermark W_s, the sequence W_{sc} of the encrypted watermark is rearranged.

In the watermark descrambling process, an inverse scrambling operation is done following a back-diffusion action, which is the inverse of watermark encryption. In addition, a binary feature sequence that has been scrambled in the upper portion of Fig. 2 by a decimal chaotic sequence CS_3 of length, $P \times Q$ generated from $SK_3 = \left(u_0^3, v_0^3, \beta^3\right)$ in varying bit locations. The process of scrambling is the same as in steps 3 and 4 of watermark encryption.

3.2 Zero-watermark generation and verification

3.2.1 Generation of zero-watermark

Choosing a binary image with a specific meaning W with size $P \times Q$, as the original watermark, and color image I with size $M \times N$ as the original image. For the convenience of calculation, let $P = Q = 64$, $M = N = 512$ in the experiments. Figure 3 shows the watermarking generation procedure.

(1) We used the pre-trained VGG19 to extract the original color image's deep feature maps, $FM(k, l, p)$:

$$I(i,j) \rightarrow VGG19 \rightarrow FM(k, l, p), \tag{9}$$

where k, l, and p are the matrix dimensions of the feature map FM resulting from VGG19 the network architecture shown in Fig. 1, $1 \leq k \leq 10, 1 \leq l \leq 10$, and $1 \leq p \leq 4096$.

(2) We construct feature sequence BF by choosing $P \times Q$ randomly features from the feature maps and then making binarization operations on each one.

$$BF(r) = \begin{cases} 1, if\, FM(r) \geq ME, \\ 0, otherwise. \end{cases} \tag{10}$$

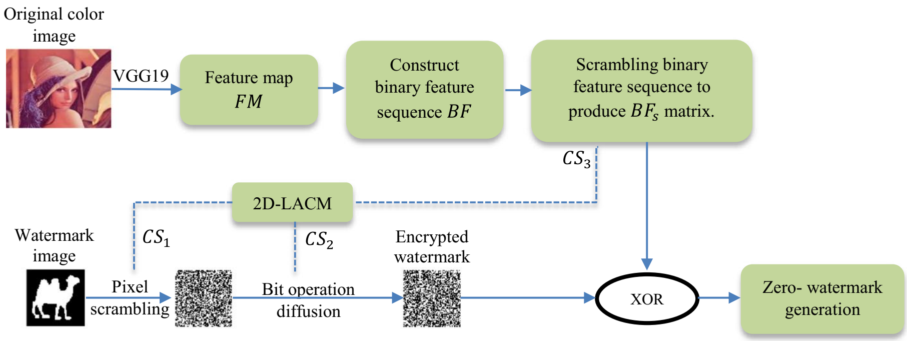


Fig. 3 Process flow for zero-watermark generation

Where r is a random number, $1 \leq r \leq 10 \times 10 \times 4096$ and ME is the average of feature maps chosen.

(3) Binary feature sequence permutation. With the secret key SK_3, we scramble the binary feature sequence, BF to generate BF_s using a chaotic sequence in decimal representation CS_3 constructed from 2D-LACM, as discussed at the end of Section 3.1, in addition, we transformed BF to the matrix BF_s with size $P \times Q$.

(4) Binary image encryption. With two secret keys SK_1 and SK_2 and to improve the zero-watermarking security, we encrypted the binary watermark to W_s using chaotic sequences CS_1 and CS_2 generated from 2D-LACM as described in Section 3.1.

(5) Zero-watermark construction. We apply XOR on the encrypted watermark image, W_s and the scrambled binary feature matrix BF_s to construct the zero-watermark W_{zero}, as follows:

$$W_{zero} = W_s \oplus BF_s. \tag{11}$$

(6) Finally, we store the secret keys SK_1, SK_2 in Step 3 and SK_3 in Step 4 and the zero-watermark, W_{zero}, in the database of copyright verification.

3.2.2 Verification of zero-watermark

The zero-watermark verification phase is mainly utilized to detect an image's watermark information. The verification process of the zero-watermark is depicted in Fig. 4, and the specific steps of the verification of the zero-watermark are presented here.

Step 1: Using Eq. 9, $FM'(k, l, p)$ was extracted from the attacked color image using VGG19.

Step 2: The feature sequence BF' was created by randomly selecting $P \times Q$ features from the feature maps FM' and then binarizing each one using Eq. 10.

Step 3: The binary feature sequence, BF', was scrambled using a chaotic sequence CS_3 used in the generation process, and then translate BF' to the matrix BF'_s of size $P \times Q$.

Step 4: The encrypted watermark image generation. We apply XOR on the reserved zero-watermark, W_{zero} and the scrambled matrix of binary feature BF'_s of the verified image to construct the encrypted watermark image W'_s as follows:

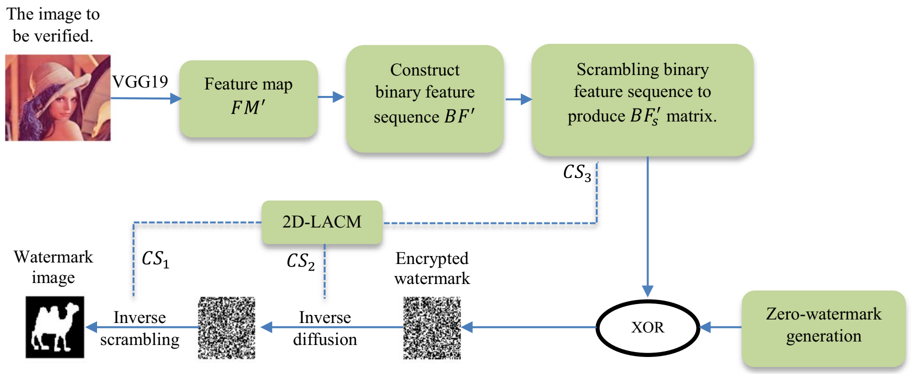

The image to be verified.
VGG19
Feature map FM'
Construct binary feature sequence BF'
Scrambling binary feature sequence to produce BF_s' matrix.
CS_3
2D-LACM
CS_1
CS_2
Watermark image
Encrypted watermark
Inverse scrambling
Inverse diffusion
XOR
Zero-watermark generation

Fig. 4 Process flow for the verification of zero-watermark

$$W'_s = W_{zero} \oplus BF'_s. \tag{12}$$

Step 5: Retrieving the watermark image. Finally, we utilize 2D-LACM decryption of the encrypted watermark image W_s' to retrieve the verifiable watermark image W'.

The first step in the descrambling process, as shown in Fig. 4, is to use a binary chaotic sequence CS_2 to execute a back-diffusion operation on the scrambled image W_s'. This chaotic sequence is formed via a 2D-LACM with the secret key SK_2. The goal of this procedure is to provide unpredictability and confusion to the image. Following the back-diffusion technique, an inverse scrambling operation is used to retrieve the original watermarking image W'. This process employs the ascending order index of a chaotic decimal sequence CS_1, which is also created using a 2D-LACM with a secret key SK_1. The goal of this procedure is to restore the original structure and appearance of the watermarking image. If the secret keys SK_1 and SK_2 are correct and match the ones used in the scrambling procedure, the reverse scrambling of the watermarking image will be identical to the original watermarking image. Figure 4 depicts a graphic illustration of this reverse scrambling process, illustrating how precisely it can retrieve the original watermarking image when exact secret keys are used.

4 Experimental results and analysis

In general, robust zero-watermarking requires both robustness and imperceptibility. Zero-watermarking has outstanding imperceptibility by nature, and robustness is a significant requirement. In this section, we will conduct five sets of experiments to validate the effectiveness of the zero-watermarking algorithm proposed in this paper.

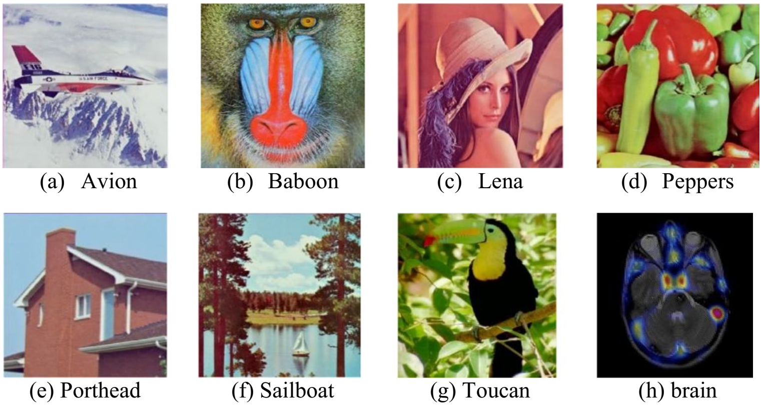

(a) Avion (b) Baboon (c) Lena (d) Peppers (e) Porthead (f) Sailboat (g) Toucan (h) brain

Fig. 5 Seven test original color images (**a**-**g**) and a medical image for the brain (**h**)

4.1 Experimental setup

4.1.1 Data sets

From the well-known standard color image datasets, USC-SIPI [44] and Computer Vision Group (CVG) [6], we chose seven color images having 512×512 pixels as the host images, as displayed in Fig. 5(a-g). We used ten binary images with 64×64 pixels as the watermark, which are shown in Fig. 6(a–j). Table 1 shows the experimental parameters. Control parameters γ and three secret keys $SK_1, SK_2 \ and \ SK_3$ consisting of the initial values u_0, v_0 for logistic Chebyshev map are given in chaotic sequence generation as follows: $SK_1 = \left(u_0^1 = 0.8633, v_0^1 = 0.9234, \beta^1 = 0.9956\right)$,$SK_2 = \left(u_0^2 = 0.897, v_0^2 = 0.9985, \beta^2 = 0.9049\right)$, and $SK_3 = \left(u_0^3 = 0.8622, v_0^3 = 0.9028, \beta^3 = 0.7112\right)$.

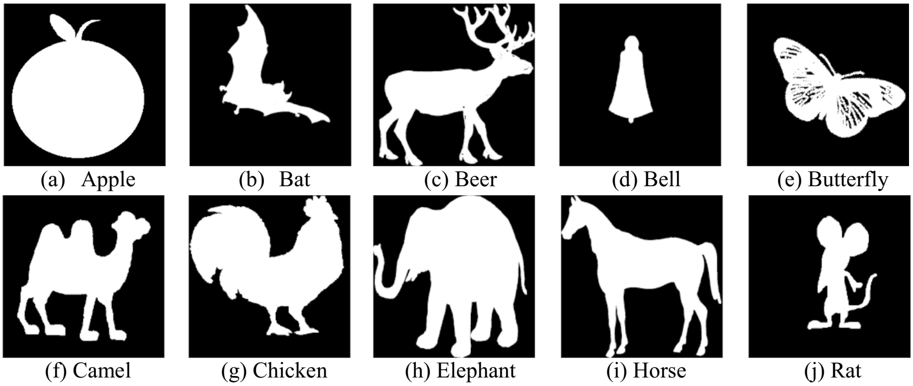

(a) Apple (b) Bat (c) Beer (d) Bell (e) Butterfly (f) Camel (g) Chicken (h) Elephant (i) Horse (j) Rat

Fig. 6 Ten original watermarks (**a**-**j**)

Table 1 Experimental parameters

Parameters	Meaning	Value
$M = N$	The original image's size	256 and 512
$P = Q = 64$	The watermark image's size	64
$SK_1 = \left(u_0^1, v_0^1, \beta^1\right)$	CS_1 chaotic sequence secret key for scrambling watermark	$(0.8633, 0.9234, 0.9956)$
$SK_2 = \left(u_0^2, v_0^2, \beta^2\right)$	CS_2 chaotic sequence secret key for scrambling watermark	$(0.8970, 0.9985, 0.9049)$
$SK_3 = (u_0^3, v_0^3, \beta^3)$	CS_3 chaotic sequence secret key for scrambling binary feature sequence	$(0.8622, 0.9028, 0.7112)$

4.1.2 Performance evaluation metrics

The peak signal-to-noise ratio (PSNR) was utilized to assess the attacked image's quality as:

$$PSNR = 10log\left(\frac{255^2 \times M \times N \times 3}{\sum_{k=1}^{3}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_k(x,y) - I'_k(x,y)\right]^2}\right), \tag{13}$$

where $I'(x,y)$ and $I(x,y)$ refer to the attacked and original image of size $M \times N$, respectively, $k\epsilon\{R, G, B\}$.

We evaluated the robustness of the proposed method using the bit error rate (BER) and normalized cross-correlation (NCC) of the retrieved watermark image. The following are the definitions of (BER) and (NCC):

$$BER = \frac{1}{P \times Q}\sum_{i=1}^{P}\sum_{j=1}^{Q}[W(i,j) \oplus W'(i,j)], \tag{14}$$

$$NCC = \frac{\sum_{i=1}^{P}\sum_{j=1}^{Q}[W(i,j) * W'(i,j)]}{\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q}\left[W(i,j)\right]^2}\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q}\left[W'(i,j)\right]^2}}, \tag{15}$$

where $W(i,j)$ and $W'(i,j)$ are the original and retrieved watermark images with the size $P \times Q$, respectively.

Obviously, the lower the BER, the higher the NCC, the better the robustness, and the higher the PSNR, the higher the quality.

4.2 Anti-attack performance analysis

Several experiments are carried out using conventional attacks to assess the robustness of the proposed zero-watermark technique. Different attacks such as rotation, scaling, brightness adjustment, filtering, additive noise, and JPEG compression are used on the color images in this subsection. Table 1 shows detailed parameter descriptions. The values of BER and NCC are calculated between the extracted watermark and the original one. Table 2 shows detailed descriptions of geometric and conventional signal-processing attacks.

Table 2 Attack types with varying parameters

Types of attack		Attacks parametric description
Filtering	Average filtering	3×3, 5×5, and 7×7
	Gaussian filtering	3×3, 5×5, and 9×9
	Median filtering	3×3, 5×5, and 7×7
	Wiener filtering	3×3, 5×5, and 7×7
Noise addition	Gaussian noise	0.1, 0.2, 0.3, 0.5
	Salt & pepper noise	0.1, 0.2, 0.3, 0.5
JPEG compression		Q = 5, 10, 20, 30, 40, 50, 60, 70, 80, 90
Brightness adjustment		Histogram equalization
		Sharpening (Unsharp masking)
Geometric transform	Rotation	10, 30, 50, 100
	Scaling	0.25, 0.5,2.0, 4.0
Combined attacks		Scaling 200% and 50% + JPEG compression (10)
		JPEG compression (10) + Rotation 2°
		Gaussian noise (0.3) + JPEG compression (10)
		JPEG compression (10) + Median filtering (5×5)
		Salt & pepper noise (0.3) + Wiener filtering (5×5)
		Median filtering (5×5) + Gaussian noise (0.3)

The robustness of the proposed technique is examined in this section for common image processing and geometric attacks. The conducted experiments can be divided into two main parts according to the size of the original image:

- We started with a standard color image called 'Baboon' of a size 256 × 256, shown in Fig. 6. In Tables 3, 4, 5, 6, and 7, the values of PSNR, BER, and NCC of the proposed technique are computed for each attack and listed with the associated retrieved watermark image. A binary image, 'Camel', was selected and used as the watermark. The retrieved watermarks from the proposed approach are closer to the original, as demonstrated in Table 3. The resulting values of BER and NCC are close to optimal. The obtained results clearly show that the retrieved watermarks remained detectable.
- Second experiments were carried out on the seven selected standard color images of size 512 × 512, as depicted in Fig. 6. Tables 8, 9, 10, 11, 12, and 13 exhibit the PSNR and NCC values of the suggested method for each attack.

Table 3 Robustness against rotation attacks

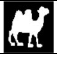
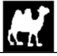
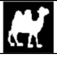
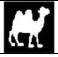
Attack	Rotation 1°	Rotation 3°	Rotation 5°	Rotation 10°
Attacked image				
Retrieved watermark				
BER	0	0. 0004	0	0
NCC	1	0. 9992	1	1
PSNR	29.082	27.9643	27.4529	26.7042

Table 4 Robustness against scaling attacks

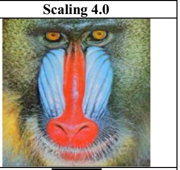
Attack	Scaling 0.25	Scaling 0.5	Scaling 2.0	Scaling 4.0
Attacked image				
Retrieved watermark				
BER	0	0.0004	0	0
NCC	1	0.9992	1	1
PSNR	30.1967	31.5144	37.7813	37.9869

Table 5 Robustness against filtering attack

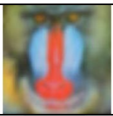
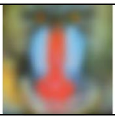
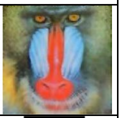
Attack	Average filtering			Median Filter		
	3x3	5x5	7x7	3x3	5x5	7x7
Attacked Image						
Retrieved watermark						
BER	0	0.0002	0. 0009	0.0002	0.0002	0.0002
NCC	1	0.9996	0. 9984	0.9996	0.9996	0.9996
PSNR	30.5574	29.8525	29.5626	32.1306	30.7814	30.2822
Attack	**Gaussian filtering**			**Wiener Filter**		
	3x3	5x5	9x9	3x3	5x5	7x7
Attacked Image						
Retrieved watermark						
BER	0	0.0005	0.0005	0.0005	0.0002	0.0005
NCC	1	0.9992	0.9992	0.9992	0.9996	0.9992
PSNR	29.6586	29.1112	28.5291	32.3540	30.9322	30.3236

Table 6 Robustness against noise attack

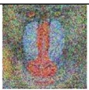
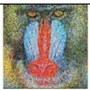
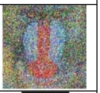
Attack	Salt & pepper noise				Gaussian noise			
Factor	(0.1)	(0.2)	(0.3)	(0.5)	(0.1)	(0.2)	(0.3)	(0.5)
Attacked Image								
Retrieved watermark								
BER	0	0	0	0. 0004	0. 0002	0. 0004	0	0. 0002
NCC	1	1	1	0. 9992	0.9996	0. 9992	1	0. 9996
PSNR	37.0567	34.0623	32.3290	30.0865	27.5601	27.4285	27.3506	27.2940

Table 7 Robustness against JPEG compression and conventional combined attacks

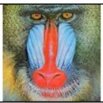
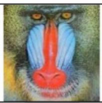
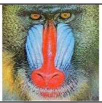
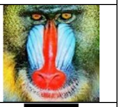
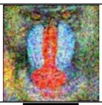
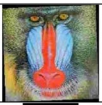
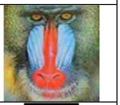
Attack	**JPEG Compression (5)**	**JPEG Compression (10)**	**JPEG Compression (50)**	**JPEG Compression (90)**	**Sharpening (Unsharp masking)**	**Histogram equalization**
Attacked image						
Retrieved watermark						
BER	0	0.0002	0	0	0	0
NCC	1	0.9996	1	1	1	1
PSNR	29.3322	29.7449	31.4096	34.2989	32.6184	28.5937
Attack	**Salt & pepper noise(0.3)+ Wiener filtering (5 × 5)**	**Median filtering (5 × 5) + Gaussian noise (0.3)**	**Median filtering (5 × 5) +JPEG compression (10)**	**JPEG compression (10)+ Gaussian noise (0.3)**	**JPEG compression (10) + Rotation 2°**	**Scaling 200% and 50%+JPEG compression (10)**
Attacked Image						
Retrieved watermark						
BER	0	0.0002	0	0.0005	0.0002	0
NCC	1	0.9996	1	0.9992	0.9996	1
PSNR	28.4922	27.9909	29.4838	27.7313	28.2170	29.6946

Table 8 Robustness to rotation attacks

Rotation Attack		Rotation							
		Bilinear interpolation				Bicubic interpolation			
Factor		1°	3°	5°	10°	1°	3°	5°	10°
Lena	PSNR	30.8970	28.7774	27.8730	26.8224	30.8355	28.7583	27.8622	26.8194
	NCC	1	0.9996	0.9984	0.9996	1	0.9996	0.9988	0.9988
Peppers	PSNR	30.9002	28.7124	27.8854	26.8734	30.8540	28.7023	27.8815	26.8731
	NCC	0.9992	0.9996	1	0.9976	0.9996	1	0.9996	0.9984
Baboon	PSNR	28.6955	27.8239	27.3698	26.6522	28.6481	27.8035	27.3577	26.6451
	NCC	0.9984	0.9988	0.9977	0.9973	0.9977	0.9988	0.9973	0.9961
Avion	PSNR	31.7638	29.7367	28.8379	27.5618	31.7314	29.7341	28.8410	27.5677
	NCC	1	1	1	0.9996	0.9996	0.9996	1	1
Sailboat	PSNR	30.1571	28.6563	27.9703	27.0277	30.0975	28.6342	27.9585	27.0226
	NCC	0.9996	0.9988	0.9988	0.9984	0.9988	0.9976	0.9984	0.9965
Porthead	PSNR	32.4308	30.1688	29.2069	27.7619	32.3773	30.1481	29.1942	27.7557
	NCC	0.9992	0.9996	0.9984	0.9992	0.9996	0.9992	0.9996	0.9992
Toucan	PSNR	30.4279	28.7394	28.1689	27.4116	30.4248	28.7383	28.1686	27.4122
	NCC	0.9996	0.9996	0.9996	0.9992	0.9996	0.9996	0.9996	0.9996

Table 9 Robustness to scaling attack

Interpolation		"Nearest"				"Bilinear"				"Bicubic"			
Scaling factor		0.5+2	2+0.5	0.25+4	4+0.25	0.5+2	2+0.5	0.25+4	4+0.25	0.5+2	2+0.5	0.25+4	4+0.25
	PSNR	35.0002	Inf	33.3291	Inf	36.0665	40.2875	33.9706	40.5647	37.0634	45.2892	34.6727	45.5700
Lena	NCC	0.9992	1	0.9984	1	0.9992	0.9988	0.9992	0.9988	0.9996	1	0.9984	1
	PSNR	34.2678	Inf	32.9645	Inf	35.4923	38.8299	33.4653	39.0442	36.0867	42.2831	34.1645	42.4787
Peppers	NCC	0.9996	1	0.9984	1	1	0.9996	0.9992	0.9992	0.9992	1	0.9996	1
	PSNR	30.7370	Inf	29.5453	Inf	30.4251	32.3272	29.7106	32.5467	30.7058	35.7239	29.8809	35.9024
Baboon	NCC	0.9938	1	0.9957	1	0.9973	0.9980	0.9957	0.9980	0.9965	1	0.9953	0.9996
	PSNR	36.0454	Inf	34.2342	Inf	35.9792	41.0237	33.5279	41.3388	37.4623	46.2194	34.2027	46.5612
Avion	NCC	0.9992	1	0.9996	1	0.9996	1	0.9992	0.9996	0.9992	0.9992	0.9980	1
	PSNR	32.7591	Inf	31.5071	Inf	33.1864	36.2117	31.7778	36.4816	33.7986	40.1902	32.1883	40.3911
Sailboat	NCC	1	1	0.9984	1	0.9996	0.9996	0.9992	0.9996	0.9992	1	0.9988	0.9996
	PSNR	39.0284	Inf	35.7365	Inf	40.7395	49.1285	35.8359	49.3445	46.2508	57.3491	37.0242	59.1332
Porthead	NCC	0.9996	1	0.9996	1	0.9996	1	1	1	1	1	0.9996	1
	PSNR	35.9066	Inf	33.6206	Inf	37.6199	44.5794	33.6470	44.9322	40.6326	52.4288	35.4697	53.1034
Toucan	NCC	0.9988	1	0.9969	1	0.9988	0.9992	0.9984	1	0.9996	1	0.9996	0.9996

Table 10 Robustness to filtering attacks

Filtering Attack		Average			Gaussian			Median			Wiener		
Factor		3×3	5×5	7×7	3×3	5×5	9×9	3×3	5×5	7×7	3×3	5×5	7×7
Lena	PSNR	36.1977	34.3053	33.3104	42.1488	42.1279	42.1278	37.6124	35.7095	34.8527	38.8426	36.8971	35.7405
	NCC	0.9980	0.9988	0.9984	1	0.9992	0.9996	0.9984	0.9992	0.9965	1	0.9992	0.9984
Peppers	PSNR	35.6325	34.1948	33.2558	41.1037	41.0912	41.0912	36.7373	35.6925	34.9798	37.5244	36.4202	35.5351
	NCC	0.9992	0.9996	0.9992	1	0.9996	0.9996	0.9996	0.9992	0.9988	0.9992	1	1
Baboon	PSNR	30.5574	29.8525	29.5626	34.5939	34.5789	34.5789	31.3107	30.1915	29.8868	31.5090	30.4939	30.1315
	NCC	0.9973	0.9961	0.9969	0.9996	0.9996	0.9988	0.9980	0.9957	0.9933	0.9980	0.9969	0.9965
Avion	PSNR	36.9519	34.3550	33.1878	42.5468	42.5266	42.5266	40.0758	37.1999	35.9914	40.6999	37.5362	35.8220
	NCC	0.9992	0.9988	1	0.9992	1	1	1	1	0.9992	0.9996	0.9988	0.9992
Sailboat	PSNR	33.4433	32.2868	31.6239	38.8819	38.8646	38.8646	34.1446	32.9641	32.3433	35.2778	34.0086	33.1500
	NCC	0.9996	0.9988	0.9988	1	1	0.9992	0.9992	0.9984	0.9980	0.9992	0.9992	0.9996
Porthead	PSNR	41.4144	36.7763	34.7859	45.6489	45.6319	45.6319	46.5120	40.1981	37.7902	46.0645	40.5581	38.1893
	NCC	1	0.9988	0.9996	1	1	1	1	1	0.9996	1	1	0.9996
Toucan	PSNR	39.1117	35.2629	33.4541	44.9340	44.9053	44.9053	42.5259	38.0370	35.8703	43.3591	38.7993	36.2604
	NCC	0.9992	0.9996	0.9996	1	0.9996	1	0.9992	0.9988	0.9996	1	0.9988	1

Table 11 Robustness to noise attack

Noise Attack		Gaussian noise				Salt & Pepper noise			
Factor		0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Lena	PSNR	27.5655	27.4217	27.3642	27.3006	37.1285	34.0859	32.2762	30.0933
	NCC	0.998	0.9988	0.9976	0.9984	0.9984	0.9969	0.9984	0.9992
Peppers	PSNR	27.8293	27.6693	27.6255	27.5704	37.3444	34.3024	32.5847	30.3436
	NCC	0.9996	0.9992	1	0.9992	0.9996	0.9984	0.9988	0.9988
Baboon	PSNR	27.5588	27.4161	27.3567	27.2996	37.0975	34.0669	32.3055	30.0763
	NCC	0.9949	0.9949	0.9973	0.9942	0.9973	0.9976	0.9949	0.9977
Avion	PSNR	27.5612	27.4118	27.3569	27.2944	37.1221	34.0561	32.3144	30.0788
	NCC	0.9996	1	0.9988	0.9996	0.9996	1	1	0.9996
Sailboat	PSNR	27.6036	27.4707	27.4007	27.3441	37.0693	34.1077	32.352	30.1447
	NCC	0.9992	0.9988	0.9996	0.9988	0.9996	0.9992	0.9988	1
Porthead	PSNR	27.5666	27.4203	27.3393	27.2877	37.0907	34.0589	32.2949	30.1034
	NCC	1	1	0.9996	1	0.9996	0.998	0.9996	1
Toucan	PSNR	28.973	28.8401	28.7591	28.6943	38.4904	35.4423	33.7259	31.5086
	NCC	0.9988	0.9984	0.9992	0.9996	0.9969	0.9965	0.9973	0.9992

4.2.1 Anti-geometric attacks performance

The most common geometric attacks such as scaling and rotation attacks cause losing synchronization of the watermark detection. In this experiment, the test image is rotated by the rotation angles of 1°, 3°, 5°, and 10°, where Table 3 shows the results of the rotation attacks. At most angles, the NCC and BER values are 1.0 and 0, respectively, indicating that this approach can exhibit perfect resilience to rotation attacks. Then, the original image is scaled in this experiment by employing different cases such as reducing and magnifying with scaling factors of 0.25, 0.5, 2.0, and 4.0°, where Table 4 shows the results of the scaling attacks. The results in Table 4 show that after a scaling attack of 0.5, the BER and NCC values are 0.0004 and 0.9992, respectively, which are close to the ideal values of 0 and 1. And for other scaling factors, the BER and NCC are the optimal values, 1.0 and 0, respectively, indicating that this approach can exhibit perfect resilience to scaling attacks.

4.2.2 Anti-image processing attacks performance

Here, we use the most common image processing attacks such as filtering, noise, JPEG compression, sharpening (Unsharp masking), and histogram equalization attacks to assess the robustness performance of the proposed algorithm through the following experiments on the test color image ‘‘Baboon’’. First, the test color image was subjected to filtering attacks where Table 5 shows the filtered images and their corresponding PSNR values, together with the retrieved watermark images and their corresponding values BER and NCC. We can observe the NCC and BER values are near to the ideal values of 1.0 and 0, respectively and the original and extracted watermarks are extremely close. These results demonstrate that the proposed algorithm can effectively survive image-filtering attacks. Secondly, the noise attacks for the test color image were conducted with Salt & pepper noise and Gaussian noise. Table 6 shows the noisy images and their corresponding PSNR values, together with the retrieved watermark images and their corresponding values BER

Table 12 Robustness against JPEG compression

JPEG compression											
Factors		5	10	20	30	40	50	60	70	80	90
Lena	PSNR	30.0835	32.4243	34.0711	34.9000	35.3567	35.6583	35.9848	36.3942	36.9843	37.9835
	NCC	0.9980	0.9980	0.9992	0.9984	0.9977	0.9996	1	0.9992	1	0.9996
Peppers	PSNR	30.4650	32.0267	33.2147	33.7817	34.1204	34.3979	34.6507	34.9637	35.3727	36.1375
	NCC	0.9992	0.9992	0.9992	1	0.9996	0.9996	0.9992	0.9996	0.9996	1
Baboon	PSNR	29.1392	29.5760	30.0987	30.4482	30.6521	30.8591	31.0708	31.3743	31.8324	32.7037
	NCC	0.9969	0.9953	0.9961	0.9984	0.9977	0.9977	1	0.9992	0.9992	0.9996
Avion	PSNR	31.3763	32.8542	34.3613	35.2268	35.7641	36.1267	36.5733	37.0820	37.8384	39.1233
	NCC	0.9992	0.9996	0.9980	0.9988	0.9984	0.9996	0.9984	1	0.9996	0.9984
Sailboat	PSNR	29.9892	30.9633	31.9154	32.3637	32.6343	32.8419	33.0350	33.2845	33.6402	34.3908
	NCC	0.9980	0.9984	0.9992	0.9988	0.9996	0.9988	0.9988	1	1	0.9992
Porthead	PSNR	32.1182	34.0507	35.8001	37.3113	37.7567	38.5223	39.2416	39.9404	40.7650	42.3187
	NCC	1	0.9988	1	1	1	1	1	1	1	1
Toucan	PSNR	31.5534	33.0981	34.9162	36.0207	36.6762	37.2848	37.8979	38.6132	39.6327	41.7206
	NCC	0.9973	0.9980	0.9984	0.9980	0.9992	0.9996	0.9996	0.9996	0.9996	0.9996

Table 13 Robustness against conventional combined attacks

Attack		Gaussian noise (0.3) + Median filtering (5×5)	Salt & pepper noise (0.3) + Wiener filtering (5×5)	JPEG compression (10) + Median filtering (5×5)	Gaussian noise (0.3) + JPEG compression (10)	JPEG compression (10) + Rotation 2°	Scaling 200% and 50% + JPEG compression (10)
Lena	PSNR	28.16697	28.7270	32.1945	27.7179	29.1848	32.3796
	NCC	0.997649	0.9969	0.9984	0.9965	0.9973	0.9992
Peppers	PSNR	28.384	29.4533	31.9408	28.3072	29.2044	32.0224
	NCC	0.999608	0.9988	0.9992	0.9988	0.9984	0.9996
Baboon	PSNR	28.02741	28.32854	29.32345	27.7166	28.0350	29.4649
	NCC	0.996487	0.993748	0.996482	0.9930	0.9965	0.9965
Avion	PSNR	28.08556	26.12836	32.92495	26.4792	29.9711	32.8035
	NCC	0.999608	0.998434	0.997649	0.9996	1	0.9988
Sailboat	PSNR	28.14442	28.4736	30.81212	27.8507	28.8169	30.9633
	NCC	0.998042	0.999608	0.999217	0.9992	0.9992	0.9992
Porthead	PSNR	28.15072	28.38254	33.88009	27.3950	30.4698	34.0509
	NCC	0.999216	0.999608	0.998827	0.9988	0.9992	1
Toucan	PSNR	29.48787	30.01508	32.75428	29.0851	29.2461	33.0425
	NCC	0.997261	0.996085	0.996865	0.9988	0.9980	0.9977

and NCC. From Table 6, we can observe that the BER and NCC values are close to the ideal value 0 and 1. And especially for the Salt & pepper noise attacks, three values of the BER and NCC are the optimal values, 1.0 and 0, respectively. The retrieved watermarks are extremely similar and identical to the original ones, demonstrating that this approach has strong resilience to noise attacks. Then, the test color image was subjected to JPEG compression attacks with the compression-quality factors as Q = 5, 10, 50, and 90, where Table 7 shows the compressed images and their corresponding PSNR values, together with the retrieved watermark images and their corresponding values BER and NCC. The results in Table 7 show that after a compression attack of Q = 10, the BER and NCC values are 0.0002 and 0.9996, respectively, which are close to the ideal value 0 and 1. And for other quality factors, the BER and NCC are the optimal values, 1.0 and 0, respectively and the retrieved watermarks are identical to the original ones, demonstrating that this approach can exhibit perfect resilience to JPEG compression attacks.

Finally, the sharpening, histogram equalization, and conventional combined attacks for the test color image were conducted where the attacked images and their corresponding PSNR values are displayed in Table 7, along with the watermark images that were retrieved and their corresponding BER and NCC values. Because the BER and NCC values are close to the ideal value 0 and 1 in some cases and in other cases are the same as the optimal values, 1.0 and 0, respectively. Additionally, the retrieved watermarks are extremely similar and identical to the original ones, demonstrating that this approach is capable of resisting these attacks.

4.3 Comparison of robustness with existing works

To fully assess how well the presented method performed, we conducted two comparisons in this section. In the first comparison, we first conduct various types of attacks on the seven selected standard color images of size 512×512, whose parameters are listed in Table 2. Then, we compute and select the minimum NCC values and the average of the PSNR values, as shown in Tables 8, 9, 10, 11, 12, and 13.

We summarized the obtained results in Table 14. And for readability and simplicity, the obtained results are depicted in Fig. 7. Finally, in Fig. 7 and Table 14, we compare these values with the results obtained by the zero-watermarking method [22]. The PSNR values of the proposed algorithm and method [22] for various attacks are presented in Table 14 and Fig. 7a. The results obtained from Fig. 7a and Table 14 show that the proposed method

Table 14 PSNR averages and NCC minimums against various attacks

Attacks type	Proposed Method	Method [22]	Proposed Method	Method [22]
	Average value of PSNR		Minimum value of NCC	
Filter (average, median, winner, gaussian)	37.1699	30.811	0.9949	0.96
Noisy (gaussian, Salt and Pepper)	30.6423	11.6486	0.9942	0.8321
JPEG Compression (5, 10, 20, 30, 40, 50, 60, 70, 80, 90)	34.6193	29.7121	0.9953	0.92
Rotation (bilinear, bicubic)	28.7528	15.3149	0.9961	0.67
Scaling (Nearest, bilinear, bicubic)	38.0713	31.4808	0.9937	0.9169
Conventional combined attacks	29.6665	17.1641	0.9929	0.90

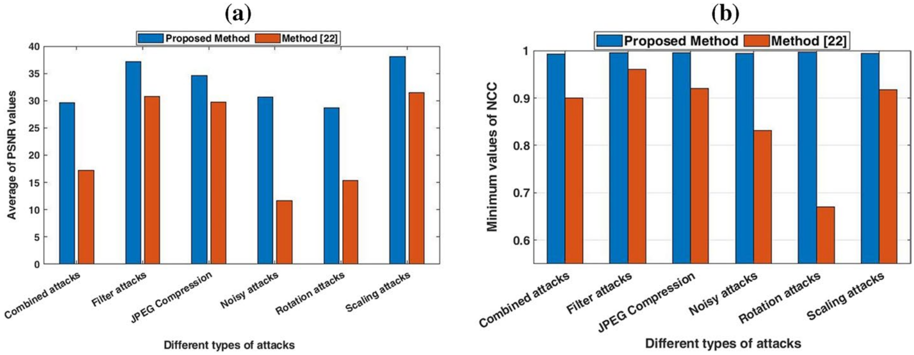


Fig. 7 Comparison between the proposed algorithm and the algorithm [22] under various attacks: (**a**) Represents the comparison for the average of PSNR values, and (**b**) Represents the comparison for the minimum values of NCC

has higher image quality than the zero-watermarking approach [22]. Moreover, the results of Fig. 7b and Table 14 show that the proposed algorithm's NCC values for various attack results are very close to the ideal value of 1.0 and higher than the compared values in [22]. These results demonstrate that the proposed approach can effectively survive image various attacks compared to the zero-watermarking approach [22].

In another comparison, the robustness of the proposed algorithm is compared with the zero-watermarking algorithms [4, 22, 45, 52, 61]. For various attacks, results of the comparison experiment are shown in Fig. 8 and Table 15, where the attacks include scaling (0.5, 2.0), rotation (3 °, 5 °, 10 °), median filtering (3×3, 5×5), Gaussian filtering (3×3, 5×5), average filtering (3×3, 5×5), Gaussian noise (0.001, 0.005, 0.01), Salt & pepper noise (0.01, 0.02), and JPEG compression (30, 50, 75). In comparison to the zero-watermarking algorithms [4, 22, 45, 52, 61], the proposed algorithm performs better, as can be observed from the comparison experiment results in Fig. 8 and Table 15. According to the experimental findings, the BER values of the proposed algorithm are very close to the optimal 0. This demonstrates the evident increase in resilience against various attacks of the proposed algorithm over the zero-watermarking algorithms [4, 22, 45, 52, 61].

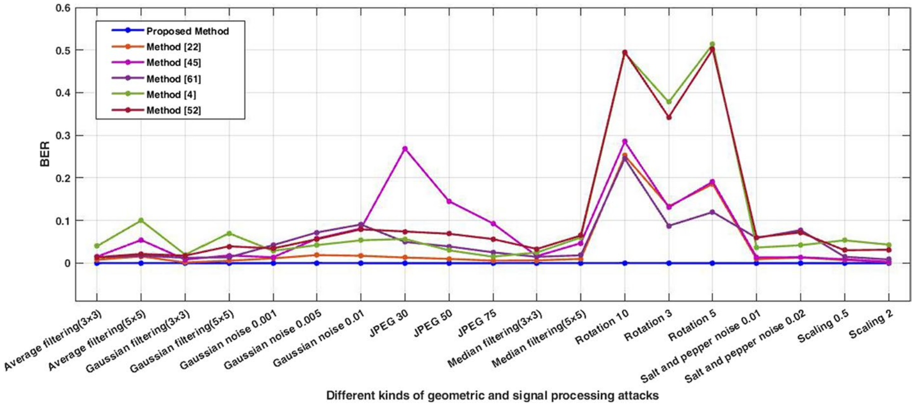


Fig. 8 BER value comparison between the proposed algorithm and existing algorithms [4, 22, 45, 52, 61] against different attacks

Table 15 BER value comparison between the proposed algorithm and existing algorithms [4, 22, 45, 52, 61] against different attacks

Attacks		BER					
		Proposed algorithm	Algorithm [22]	Algorithm [45]	Algorithm [61]	Algorithm [4]	Algorithm [52]
JPEG compression	QF = 30	0	0.0131	0.2687	0.0497	0.0567	0.0742
	QF = 50	0	0.0099	0.1454	0.0392	0.03	0.0688
	QF = 75	0	0.0058	0.0933	0.0253	0.0151	0.0564
Rotation	3°	0	0.1333	0.1311	0.0875	0.3778	0.3426
	5°	0	0.1858	0.1912	0.1198	0.5134	0.5016
	10°	0.0002	0.2522	0.285	0.2453	0.4926	0.4956
Scaling	0.5	0	0.0071	0.0091	0.0149	0.0535	0.0301
	2	0	0.0024	0.003	0.0086	0.0433	0.0315
Salt and pepper noise	0.01	0	0.0091	0.0136	0.0591	0.0365	0.0605
	0.02	0	0.0131	0.0138	0.0774	0.042	0.072
Gaussian noise	0.001	0	0.0112	0.0138	0.0422	0.0289	0.0345
	0.005	0.0002	0.0189	0.0576	0.072	0.0423	0.0562
	0.01	0	0.0172	0.0812	0.0906	0.0535	0.0793
Average filtering	3×3	0	0.0079	0.0153	0.0134	0.04	0.0138
	5×5	0.0002	0.0157	0.0542	0.0173	0.1	0.0215
	9×9	0.0005	0.0322	0.1289	0.0272	–	–
Gaussian filtering	3×3	0	0.0015	0.0085	0.0125	0.02	0.0176
	5×5	0	0.0056	0.0179	0.0136	0.07	0.0394
	9×9	0	0.014	0.0188	0.0137	–	–
Median filtering	3×3	0.0002	0.0064	0.0165	0.0147	0.025	0.0332
	5×5	0	0.0099	0.0467	0.0183	0.06	0.0647
	9×9	0	0.0201	0.1634	0.0271	–	–

4.4 PSNR-based comparative analysis

This experiment aims to examine the efficiency of the proposed algorithm for various host images and watermark sizes. Table 16 gives a comparative analysis based on PSNR. For the same attacks used in Table 15, this experiment employs different host image sizes according to different watermark sizes. The results show that when the watermark sizes are changed, the PSNR values of the attacked image remain very close or the same. This implies that the quality of the original image is unaffected by the size of the watermark. This emphasizes the significance of adopting the suggested method to protect intellectual property rights while maintaining image quality.

4.5 Comparative analysis for medical images

The robustness of the proposed technique is examined in this section for common image processing and geometric attacks. We select from the 'Whole-Brain Atlas' [23] a medical color MRI image of size 256 × 256, shown in Fig. 5 (h). In Table 17, the values of PSNR, BER,

Table 16 Comparison of PSNR values for different host image sizes according to different watermark sizes

Attacks		PSNR			
		256×256		512×512	
		32×32	64×64	32×32	64×64
JPEG compression	QF = 10	29.7450	29.7449	29.5760	29.5760
	QF = 50	31.4097	31.4096	30.8591	30.8591
	QF = 90	34.2989	34.2989	32.7037	32.7037
Rotation	3°	27.9643	27.9643	27.8239	27.8239
	5°	27.4529	27.4529	27.3698	27.3698
	10°	26.7042	26.7042	26.6522	26.6522
Scaling	0.5	31.5144	31.5144	30.7058	30.7058
	2	37.7813	37.7813	35.7239	35.7239
Salt and pepper noise	0. 1	37.0729	37.0567	37.0891	37.0975
	0. 2	34.0314	34.0623	34.0815	34.0669
Gaussian noise	0.1	27.5622	27.5601	27.5758	27.5588
	0.3	27.3357	27.3506	27.3581	27.3567
	0.5	27.2877	27.2940	27.2911	27.2996
Average filtering	3×3	31.2057	30.5574	30.5574	30.5574
	5×5	30.0856	29.8525	29.8525	29.8525
	7×7	29.5769	29.5626	29.5626	29.5626
Gaussian filtering	3×3	29.6586	29.6586	29.5415	34.5939
	5×5	29.1112	29.1112	29.1626	34.5789
	9×9	28.7740	28.5291	28.9135	34.5789
Median filtering	3×3	32.1306	32.1306	31.3107	31.3107
	5×5	30.7814	30.7814	30.1915	30.1915
	7×7	30.2822	30.2822	29.8868	29.8868

Table 17 Robustness of medical image against various attack

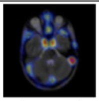
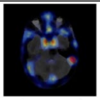
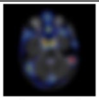
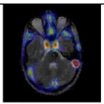
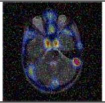
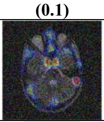
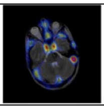
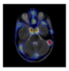
Attack	**Average filtering** 7x7	**Median Filter** 7x7	**Gaussian filtering** **9x9**	**JPEG Compression** **(10)**
Attacked Image				
Retrieved watermark				
BER	0.0002	0.0002	0.0002	0
NCC	0.9996	0.9996	0.9996	1
PSNR	31.7240	32.0012	31.6309	33.0250
Attack	**Salt & pepper noise** (0.1)	**Gaussian noise** **(0.1)**	**Rotation 10°**	**Scaling 0.5**
Attacked Image				
Retrieved watermark				
BER	0	0	0	0
NCC	1	1	1	1
PSNR	42.8346	33.3548	30.95541	31.6504

Table 18 Comparative analysis between the standard color image and medical image for the proposed algorithm

Attacks		BER		NCC		PSNR	
		standard	medical	standard	medical	Standard	medical
JPEG compression	QF = 10	0.0002	0	0.9996	1	29.7449	33.0251
	QF = 50	0	0	1	1	31.4096	35.4481
	QF = 90	0	0	1	1	34.2989	42.7211
Rotation	3°	0.0004	0	0.9992	1	27.9643	31.4016
	5°	0	0	1	1	27.4529	31.198
	10°	0	0	1	1	26.7042	30.9554
Scaling	0.5	0.0004	0	0.9992	1	31.5144	31.6504
	2	0	0.0002	1	0.999608	37.7813	31.7614
Salt and pepper noise	0. 1	0	0	1	1	37.0567	42.8346
	0. 2	0	0.0005	1	0.9992	34.0623	39.8714
Gaussian noise	0.1	0.0002	0	0.9996	1	27.5601	33.3548
	0.3	0	0.0005	1	0.9992	27.4285	33.1635
	0.5	0.0002	0.0002	0.9996	0.9996	27.2940	33.1094
Average filtering	3 × 3	0	0	1	1	30.5574	31.7485
	5 × 5	0.0002	0	0.9996	1	29.8525	31.7214
	7 × 7	0.0009	0.0002	0.9984	0.9996	29.5626	31.724
Gaussian filtering	3 × 3	0	0.0002	1	0.9996	29.6586	31.7161
	5 × 5	0.0005	0	0.9992	1	29.1112	31.6799
	9 × 9	0.0005	0.0002	0.9992	0.9996	28.5291	31.6309
Median filtering	3 × 3	0.0002	0	0.9996	1	32.1306	33.8574
	5 × 5	0.0002	0.0002	0.9996	0.9996	30.7814	32.4058
	7 × 7	0.0002	0.0002	0.9996	0.9996	30.2822	32.0012

and NCC of the proposed algorithm for a medical image are computed, and in Table 18, these values are compared with the standard color images for different attacks. The results highlight the effectiveness and robustness of our proposed algorithm against various types of attacks and its capability of extracting the watermark from medical images effectively.

5 Conclusion

A robust zero-watermark algorithm for the color images based on the VGG19 and 2D chaos map was proposed in this paper. The novelty of the proposed algorithm includes:

- The construction of a zero-watermark is obtained by using a VGG19 deep CNN.
- 2D-LACM is utilized to confuse and diffuse the original image feature matrix and watermark image.
- The proposed algorithm for zero-watermarking in this paper can be used to protect the copyright of color images under the requirements for strong robustness against geometric and signal processing attacks and the excellent visual quality of images.

We demonstrated experimentally that the proposed algorithm outperformed zero-watermarking algorithms. In the future, we will improve the proposed algorithm's security and robustness performance to extend its application to e-healthcare, telemedicine, stereoscopic, and real-time captured images.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability The datasets supporting this study's findings are available online and have proper citations within the paper.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bhatti UA, Yuan L, Yu Z, Li J, Nawaz SA, Mehmood A, Zhang K (2021) New watermarking algorithm utilizing quaternion Fourier transform with advanced scrambling and secure encryption. Multimed Tools Appl 80:13367–13387
2. Chang C-C, Chuang J-C (2002) An image intellectual property protection scheme for gray-level images using visual secret sharing strategy. Pattern Recognit Lett 23:931–941
3. Chang C-C, Lin P-Y (2008) Adaptive watermark mechanism for rightful ownership protection. J Syst Softw 81:1118–1129
4. Chen H, Luo T, Yu M, Jiang G, Zhou H, Mo K (2012) A zero-watermark method based on texture characteristic of image blocks for stereo images. In: 2012 International Conference on Industrial Control and Electronics Engineering. IEEE, pp 490–493
5. Chen Y-H, Huang H-C (2015) Coevolutionary genetic watermarking for owner identification. Neural Comput Appl 26:291–298
6. CVG Bonn (2014) Computer Vision Group. https://pages.iai.uni-bonn.de/gall_juergen/index.html. Accessed 9 Jan 2022
7. Daoui A, Karmouni H, Sayyouri M, Qjidaa H (2022) Robust image encryption and zero-watermarking scheme using SCA and modified logistic map. Expert Syst Appl 190:116193
8. Evsutin OO, Melman AS, Meshcheryakov RV (2020) Digital steganography and watermarking for digital images: a review of current research directions. IEEE Access 8:166589–166611. https://doi.org/10.1109/ACCESS.2020.3022779
9. Fierro-Radilla A, Nakano-Miyatake M, Cedillo-Hernandez M, Cleofas-Sanchez L, Perez-Meana H (2019) A robust image zero-watermarking using convolutional neural networks. In: 2019 7th International Workshop on Biometrics and Forensics, IWBF. Cancun, Mexico, pp 1–5
10. Gao G, Jiang G (2015) Bessel-Fourier moment-based robust image zero-watermarking. Multimed Tools Appl 74:841–858. https://doi.org/10.1007/s11042-013-1701-8
11. Gao J, Li Z, Fan B (2022) An efficient robust zero watermarking scheme for diffusion tensor-Magnetic resonance imaging high-dimensional data. J Inf Secur Appl 65:103106
12. Gao X, Deng C, Li X, Tao D (2010) Geometric distortion insensitive image watermarking in affine covariant regions. IEEE Trans Syst Man Cybern Part C Appl Rev 40:278–286

13. Gao Y, Kang X, Chen Y (2021) A robust video zero-watermarking based on deep convolutional neural network and self-organizing map in polar complex exponential transform domain. Multimed Tools Appl 80:6019–6039. https://doi.org/10.1007/s11042-020-09904-4
14. Han B, Du J, Jia Y, Zhu H (2021) Zero-Watermarking Algorithm for Medical Image Based on VGG19 Deep Convolution Neural Network. J Healthc Eng 2021. https://doi.org/10.1155/2021/5551520
15. Han B, Wang H, Qiao D, Xu J, Yan T (2023) Application of zero-watermarking scheme based on swin transformer for securing the metaverse healthcare data. IEEE J Biomed Heal Informatics. https://doi.org/10.1109/JBHI.2023.3257340
16. Hosny KM, Darwish MM (2021) Reversible Color Image Watermarking Using Fractional - Order Polar Harmonic Transforms and a Chaotic Sine Map. Circuits Syst Signal Process 40:6121–6145. https://doi.org/10.1007/s00034-021-01756-z
17. Hosny KM, Darwish MM (2021) New geometrically invariant multiple zero-watermarking algorithm for color medical images. Biomed Signal Process Control 70:103007
18. Hosny KM, Darwish MM, Fouda MM (2021) Robust Color Images Watermarking Using New Fractional-Order Exponent Moments. IEEE Access 9:47425–47435. https://doi.org/10.1109/ACCESS.2021.3068211
19. Hosny KM, Darwish MM, Fouda MM (2021) New Color Image Zero-Watermarking Using Orthogonal Multi-Channel Fractional-Order Legendre-Fourier Moments. IEEE Access 9:91209–91219. https://doi.org/10.1109/ACCESS.2021.3091614
20. Hu K, Wang X, Hu J, Wang H, Qin H (2021) A novel robust zero-watermarking algorithm for medical images. Vis Comput 37:2841–2853. https://doi.org/10.1007/s00371-021-02168-5
21. Iwendi C, Jalil Z, Javed AR, Thippa Reddy G, Kaluri R, Srivastava G, Jo O (2020) KeySplitWatermark: Zero Watermarking Algorithm for Software Protection against Cyber-Attacks. IEEE Access 8:72650–72660. https://doi.org/10.1109/ACCESS.2020.2988160
22. Kang X, Lin G, Chen Y, Zhao F, Zhang E, Jing C (2020) Robust and secure zero-watermarking algorithm for color images based on majority voting pattern and hyper-chaotic encryption. Multimed Tools Appl 79:1169–1202
23. Keith A. Johnson and J. Alex Becker The Whole Brain Atlas. http://www.med.harvard.edu/AANLIB/home.html. Accessed 16 Dec 2021
24. Li T, Li J, Liu J, Huang M, Chen Y-W, Bhatti UA (2022) Robust watermarking algorithm for medical images based on log-polar transform. EURASIP J Wirel Commun Netw 2022:1–11
25. Liao X, Li K, Zhu X, Liu KJR (2020) Robust detection of image operator chain with two-stream convolutional neural network. IEEE J Sel Top Signal Process 14:955–968
26. Lin SD, Chen C-F (2000) A robust DCT-based watermarking for copyright protection. IEEE Trans Consum Electron 46:415–421
27. Liu L, Jiang D, Wang X, Rong X, Zhang R (2021) 2D Logistic-Adjusted-Chebyshev map for visual color image encryption. J Inf Secur Appl 60:102854
28. Liu W, Li J, Shao C, Ma J, Huang M, Bhatti UA (2022) Robust Zero Watermarking Algorithm for Medical Images Using Local Binary Pattern and Discrete Cosine Transform. In: International Conference on Artificial Intelligence and Security. Springer, pp 350–362
29. Liu Y, Yang F, Gao K, Dong W, Song J (2017) A zero-watermarking scheme with embedding timestamp in vector maps for Big Data computing. Cluster Comput 20:3667–3675
30. Ma B, Chang L, Wang C, Li J, Li G, Xia Z, Wang X (2021) Double Medical Images Zero-Watermarking Algorithm Based on the Chaotic System and Ternary Accurate Polar Complex Exponential Transform. J Math Imaging Vis 63:1160–1178
31. Nandini DU, Divya S (2017) A literature survey on various watermarking techniques. In: 2017 International Conference on Inventive Systems and Control (ICISC). IEEE, pp 1–4
32. Qi X, Xin X (2015) A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization. J Vis Commun Image Represent 30:312–327
33. Roček A, Javorník M, Slavíček K, Dostál O (2021) Zero watermarking: critical analysis of its role in current medical imaging. J Digit Imaging 34:204–211
34. Rosales-Roldan L, Cedillo-Hernandez M, Nakano-Miyatake M, Perez-Meana H, Kurkoski B (2013) Watermarking-based image authentication with recovery capability using halftoning technique. Signal Process Image Commun 28:69–83
35. Seenivasagam V, Velumani R (2013) A QR code based zero-watermarking scheme for authentication of medical images in teleradiology cloud. Comput Math Methods Med 2013. https://doi.org/10.1155/2013/516465
36. Shao Z, Shang Y, Zeng R, Shu H, Coatrieux G, Wu J (2016) Robust watermarking scheme for color image based on quaternion-type moment invariants and visual cryptography. Signal Process Image Commun 48:12–21
37. Shao Z, Shang Y, Zhang Y, Liu X, Guo G (2016) Robust watermarking using orthogonal Fourier-Mellin moments and chaotic map for double images. Signal Process 120:522–531

38. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc
39. Singh A, Dutta MK (2018) Lossless and robust digital watermarking scheme for retinal images. In: 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, pp 1–5
40. Sun L, Xu JC, Zhang XX, Dong W, Tian Y (2015) A novel generalized Arnold transform-based zero-watermarking scheme. Appl Math Inf Sci 4:2023–2035
41. Thanh TM, Tanaka K (2017) An image zero-watermarking algorithm based on the encryption of visual map feature with watermark information. Multimed Tools Appl 76:13455–13471
42. Tsai H-H, Lai Y-S, Lo S-C (2013) A zero-watermark scheme with geometrical invariants using SVM and PSO against geometrical attacks for image protection. J Syst Softw 86:335–348
43. Tsai H-H, Tseng H-C, Lai Y-S (2010) Robust lossless image watermarking based on α-trimmed mean algorithm and support vector machine. J Syst Softw 83:1015–1028
44. University of Southern California (2020) USC-SIPI Image Database. In: USC-SIPI Image Database. http://sipi.usc.edu/database/. Accessed 22 Nov 2021
45. Vellaisamy S, Ramesh V (2014) Inversion attack resilient zero-watermarking scheme for medical image authentication. IET Image Process 8:718–727
46. Wang C, Hao Q, Ma B, Wu X, Li J, Xia Z, Gao H (2021) Octonion continuous orthogonal moments and their applications in color stereoscopic image reconstruction and zero-watermarking. Eng Appl Artif Intell 106:104450
47. Wang CP, Wang XY, Chen XJ, Zhang C (2017) Robust zero-watermarking algorithm based on polar complex exponential transform and logistic mapping. Multimed Tools Appl 76:26355–26376. https://doi.org/10.1007/s11042-016-4130-7
48. Wang C, Wang X, Xia Z, Zhang C (2019) Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm. Inf Sci (Ny) 470:109–120. https://doi.org/10.1016/j.ins.2018.08.028
49 Wang XY, Wang L, Tian JL, Niu PP, Yang HY (2021) Color Image Zero-Watermarking Using Accurate Quaternion Generalized Orthogonal Fourier-Mellin Moments. J Math Imaging Vis 63:708–734. https://doi.org/10.1007/s10851-020-01002-2
50. Wang Y, Doherty JF, Van Dyck RE (2002) A wavelet-based watermarking algorithm for ownership verification of digital images. IEEE Trans Image Process 11:77–88
51. Wen Q, Sun T-F, Wang S-X (2003) Concept and application of zero-watermark. Acta Electron Sin 31:214–216
52. Wu-Jie Z, Mei Y, Si-Min Y, Gang-Yi J, Ding-Fei G (2012) A zero-watermarking algorithm of stereoscopic image based on hyperchaotic system. Acta Phys Sin 61
53. Wu X, Sun W (2013) Robust copyright protection scheme for digital images using overlapping DCT and SVD. Appl Soft Comput 13:1170–1182
54. Xia Z, Wang X, Wang C, Ma B, Zhang H, Li Q (2021) Novel quaternion polar complex exponential transform and its application in color image zero-watermarking. Digit Signal Process A Rev J 116. https://doi.org/10.1016/j.dsp.2021.103130
55. Xia Z, Wang X, Wang C, Wang C, Ma B, Li Q, Wang M, Zhao T (2022) A robust zero-watermarking algorithm for lossless copyright protection of medical images. Appl Intell 52:607–621
56. Xiao X, Li J, Yi D, Fang Y, Cui W, Bhatti UA, Han B (2021) Robust Zero Watermarking Algorithm for Encrypted Medical Images Based on DWT-Gabor. In: Innovation in Medicine and Healthcare: Proceedings of 9th KES-InMed 2021. Springer, pp 75–86
57. Xiyao L, Zhang Y, Du S, Zhang J, Jiang M, Fang H (2022) DIBR Zero-watermarking based on Invariant Feature and Geometric Rectification. IEEE Multimed. https://doi.org/10.1109/MMUL.2022.3148301
58. Yi D, Li J, Fang Y, Cui W, Xiao X, Bhatti UA, Han B (2021) A robust zero-watermarkinging algorithm based on PHTs-DCT for medical images in the encrypted domain. In: Innovation in Medicine and Healthcare: Proceedings of 9th KES-InMed 2021. Springer, pp 101–113
59. Zeng C, Liu J, Li J, Cheng J, Zhou J, Nawaz SA, Xiao X, Bhatti UA (2022) Multi-watermarking algorithm for medical image based on KAZE-DCT. J Ambient Intell Humaniz Comput 1–9. https://doi.org/10.1007/s12652-021-03539-5
60. Zhou WJ, Yu M, Yu SM, Jiang GY, Ge DF (2012) A zero-watermarking algorithm of stereoscopic image based on hyperchaotic system. Wuli Xuebao/Acta Phys Sin 61. https://doi.org/10.7498/aps.61.080701
61. Zou B, Du J, Liu X, Wang Y (2018) Distinguishable zero-watermarking scheme with similarity-based retrieval for digital rights Management of Fundus Image. Multimed Tools Appl 77:28685–28708

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.