Hindawi
Complexity
Volume 2024, Article ID 6618382, 18 pages
https://doi.org/10.1155/2024/6618382

Research Article

Image Robust Watermarking Method Based on DWT-SVD Transform and Chaotic Map

Weishuai Wu, Yujiao Dong, and Guangyi Wang

Institute of Modern Circuits and Intelligent Information, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Yujiao Dong; yjdong@hdu.edu.cn

Received 28 March 2023; Revised 22 November 2023; Accepted 23 March 2024; Published 15 May 2024

Academic Editor: Jesus M. Munoz-Pacheco

The existing watermarking algorithms make it difficult to balance the invisibility and robustness of the watermark. This paper proposes a robust image watermarking method based on discrete wavelet transform (DWT), singular value decomposition (SVD), and chaotic maps. This method is a semiblind watermarking method. First, a chaotic logistic-tent map is introduced, employing an extensive chaotic parameter domain. This map is amalgamated with Arnold's transformation to encrypt the watermark image, thereby bolstering the security of the watermark information. Subsequently, the frequency domain is obtained by applying DWT to the carrier image. Embedding watermarks in the frequency domain ensures the invisibility of the watermark, with a preference for a high-frequency subband after the DWT of the carrier image for enhanced watermark robustness. SVD is then applied to both the high-frequency subband of the carrier image after DWT and the encrypted watermark image. The final step involves embedding the singular values of the encrypted watermark image into the carrier image's singular values, thereby completing the watermark information embedding process. In simulation experiments, an invisibility test was conducted on various carrier images, yielding peak signal-to-noise ratio (PSNR) values consistently exceeding 43, and structural similarity (SSIM) close to 1. Robustness testing against various types of attacks resulted in normalized correlation (NC) values consistently surpassing 0.9, with bit error rate (BER) values approaching 0. In conclusion, the proposed algorithm satisfies imperceptibility requirements while demonstrating formidable robustness.

1. Introduction

With the development of digital image technology, our lives are becoming increasingly dependent on digital images. However, with the growing use of digital images, to transmit and share personal and confidential information, privacy and security concerns have become more pronounced [1–3]. Digital images are vulnerable to malicious tampering, theft, and misuse, causing significant harm to individuals, businesses, and entire societies. Therefore, the development of image watermarking technology is crucial for protecting the privacy and security of digital images. By using image watermarking, digital image copyrights can be verified and traced without compromising their perceptual quality [4, 5]. This technology is essential for achieving image authentication, ownership protection, and content tracing, and has significant applications in various fields, including military, data hiding, and multimedia [6–8].

The motivation for this research stems from the pressing need to address privacy and security concerns surrounding digital images. Existing solutions may either compromise image quality or fail to provide adequate security. Therefore, our study aims to develop an advanced watermarking technology that ensures the perceptual quality of digital images while effectively safeguarding their privacy and security.

Existing watermarking algorithms are mainly based on the spatial domain and the transform domain [9], among which the spatial domain-based algorithm is mainly implemented by modifying the pixels of the image directly. The least significant bit (LSB) algorithm is the most important spatial domain algorithm, whose principle is to quantize the image pixel bit, and then embed the watermark information bit to be hidden in the least significant bit. This algorithm is fast, easy to implement, and can also achieve a good watermark invisibility, but cannot resist various attacks, exhibiting poor robustness [10, 11].

Another algorithm is based on the transform domain, which transforms the image into the frequency domain using various methods, and then embeds the watermark information into the frequencies. Common transformation methods include discrete cosine transform (DCT), discrete Fourier transform (DFT), and discrete wavelet transform (DWT).

The frequency domain algorithm is more suitable for the human visual system (HVS) [12, 13], which embeds the watermark in the visually insensitive region, which not only increases the robustness of the watermark but also does not degrade the quality of the image. Compared with DCT and DFT, DWT describes HVS more accurately and has better robustness to some attacks such as additional noise and resizing [14, 15]. Therefore, the watermark embedding algorithm based on DWT has attracted extensive attention in the research field.

The field of digital watermarking has realized significant achievements in recent years, with many researchers proposing various techniques for protecting digital data. Ernawan et al. proposed an improved image watermarking method by modifying selected discrete wavelet transform and discrete cosine transform coefficients, and the method achieved better imperceptibility and robustness compared to the existing methods [16]. Zermi et al. proposed a blind watermarking approach for protecting medical images based on a DWT-SVD combination, maintaining high-quality watermarked images and demonstrating high robustness against several conventional attacks [17]. Li et al. presented a DWT digital watermarking algorithm based on 2D-LICM hyperchaotic mapping, and the proposed method can resist attacks, including geometric distortion, filtering, and noise, and maintains high watermark imperceptibility [18]. Abdel-Wahab et al. introduced an efficient combination of RSA cryptography, lossy, and lossless compression steganography techniques to conceal data, and the achieved high-security level encompasses both data confidentiality and integrity [19]. Nawaz et al. introduced a sophisticated medical image processing approach, amalgamating deep feature extraction with encrypted watermarking techniques alongside discrete wavelet transform and discrete cosine transform, and through this method, they not only successfully extracted essential features and encrypted watermarks from medical images but also ensured dependable retrieval of ownership and watermark details, showcasing formidable resilience against both conventional and geometric attacks [20]. Kant and Chaudhary proposed a watermarking-based approach for protecting templates in a multimodal biometric system, and the method ensures high security by embedding the watermark within the template and validating it by using a secret key [21]. Ming and Fuken presented a robust and secure watermarking algorithm based on DWT and SVD in the fractional order Fourier transform domain, and the proposed method achieved a high robustness against various attacks, including filtering, noise, and geometric distortion [22]. Singha and Ullah proposed an audio watermarking method to decentralize the watermarks, and the proposed method achieved a high robustness and security by distributing the watermark information among several audio segments [23].

Besides, the rapid advancement of deep learning has revolutionized numerous fields, including computer vision and pattern recognition. Deep learning models, such as convolutional neural networks (CNNs) and graph convolutional networks (GCNs), have demonstrated remarkable performance in tasks such as image classification, object detection, and natural language processing [24]. Deep learning models can provide enhanced features for watermark embedding and extraction, enabling more robust and imperceptible watermarking schemes.

In this paper, a robust watermarking algorithm is proposed, utilizing a combination of DWT and SVD. To heighten the security of the watermark information, the logistic-tent map is introduced for encrypting the watermark image in conjunction with the Arnold transform. In addition, the algorithm is implemented in the frequency domain and is suitable for grayscale watermark images. Considering the robust resistance of high-frequency components to geometric attacks such as shearing and rotation, the high-frequency subband is selected as the embedding area. This portion is chosen for SVD to amplify the numerical values. Subsequently, SVD is applied to the encrypted watermark image, and the watermark information is finally embedded into the S-domain of the high-frequency subband. A proficient watermarking algorithm ensures not only the invisibility of the watermark but also its robustness. The algorithm performs admirably in both aspects.

The rest of the paper is organized as follows: Section 2 introduces the relevant theoretical background. Section 3 provides a performance analysis of the proposed logistic-tent map. Section 4 describes the watermark embedding and extraction processes of the algorithm. Section 5 presents the simulation tests for invisibility, robustness, sensitivity, complexity, and encryption watermark performance. Finally, Section 6 summarizes the paper.

2. Preliminary Knowledge

2.1. Logistic-Tent Map. The logistic map is a classic model for studying chaotic systems, with simple structure and complex chaotic dynamics, so it is often used in the field of encryption [25]. The logistic map is described as follows:

$$x_{n+1} = \mu x_n (1 - x_n), \tag{1}$$

where the control parameter $\mu \in (0, 4]$ and the variable $x \in (0, 1)$. When $\mu \in [3.57, 4.00]$, the system exhibits chaos, and it generates a set of one-dimensional nonperiodic and nonconvergent chaotic sequences.

The tent map is a piecewise linear map with a simple mathematical structure, uniform distribution function, and good correlation, which is widely used in chaotic encryption systems such as image encryption. The tent map is described as [26] follows:

$$x_{n+1} = \begin{cases} \mu x_n, & 0 < x_n < 0.5, \\ \mu (1 - x_n), & 0.5 \le x_n \le 1, \end{cases} \tag{2}$$

where the control parameter $\mu \in (0, 2]$ and the variable $x \in [0, 1]$. When $\mu \in [1, 2]$, the system is chaotic.

Regardless of whether it is the logistic map or the tent map, they are not surjective maps and the distribution of chaos in the system is uneven. The range of the chaotic parameter μ is small and the system parameters are few, which results in a small key space. To address these issues, this paper proposes a new map called the logistic-tent map by combining the structural forms of the logistic and the tent maps. Its mathematical expression is as follows:

$$x_{n+1} = \begin{cases} (\mu + 2)x_n, & \mod 1 \quad 0 < x_n < 0.5, \\ (\mu + 2)x_n(1 - x_n) + (1 - x_n), & \mod 1 \quad 0.5 \le x_n \le 1, \end{cases} \tag{3}$$

where "mod" represents the remainder operation, the control parameter $\mu \in (0, \infty]$, and the variable $x \in [0, 1]$. When μ is greater than 0, the system shows chaotic states.

When an image with the size of $M \times N$ is encrypted, the logistic-tent map needs to iterate $M \times N$ times to obtain a one-dimensional sequence with the length of $M \times N$. Then, the one-dimensional chaotic sequence is transformed into a two-dimensional matrix of $M \times N$, and XOR operation is performed with the original image, where x_0 and μ are the keys. Due to the high sensitivity of chaotic sequences to keys, the sequences generated by the map will be very different even if the key values are extremely close. Therefore, it is difficult for attackers to derive the key value from a finite-length sequence [27, 28].

2.2. Arnold Transform. Arnold transform is to permute the pixel position in an image, which can be defined by the following equation [29]:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b \\ a & ab + 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod N, \tag{4}$$

where a and b are scrambling parameters, (x, y) is the pixel coordinate of the original image, and (x', y') is the pixel coordinate of the new image after transformation, N is the size of the image, and mod represents the remainder operation.

The reverse Arnold transformation is used to restore the image, which is described as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ab + 1 & -b \\ -a & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \bmod N. \tag{5}$$

When a transformed image is attacked, even if the attacker obtains an encrypted image, the original image cannot be restored without knowing the number of the Arnold transforms. So the Arnold transform can further ensure the security of the image.

2.3. Discrete Wavelet Transform (DWT). Image information includes low-frequency information that reflects the main information of the image and high-frequency information that reflects the details of the image. After discrete wavelet transformation, an image is decomposed into four subbands with different resolutions, namely, one low-frequency band LL with the main information of the image and three high-frequency bands LH, HL, and HH with detailed information, as shown in Figure 1.

It can be seen from Figure 1 that the image in the LL subband is closest to the original image, indicating that the LL subband concentrates the main energy of the image, so selecting this area to embed the watermark can greatly ensure the invisibility of the watermark. Still, it has poor robustness when facing attacks such as rotation and noise.

The HH subband exhibits a preponderance of dark pixels, whose value is close to zero. This characteristic provides a favorable foundation for watermark protection since the HH subband has minimal susceptibility to image attacks. Embedding the watermark information within this region effectively shields the attacks and augments its robustness [30]. However, embedding watermarks directly within the HH subband results in great changes to the pixel values, leading to visible watermarks that compromise their intended invisibility.

2.4. Singular Value Decomposition (SVD). A non-negative matrix can represent any image. If the image is represented by A, SVD can be expressed as follows [31]:

$$A = U \times S \times V^T, \tag{6}$$

where U and V are orthogonal matrices, and S is a matrix whose nondiagonal terms are all 0, as follows:

$$S = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \tag{7}$$

where $\Sigma_r = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r)$ is the diagonal matrix, λ_i satisfies $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_r$, and r is the rank of S.

When the image is attacked by geometric attacks, such as rotation, scaling, and compression, the S value of the image matrix changes little. Therefore, the S area is often chosen to embed the watermark, to improve the ability of the watermark to resist geometric attacks. In the watermark embedding algorithm, SVD is usually used together with the frequency domain algorithm to achieve a better embedding effect [32].

To ensure the invisibility of the watermark, an SVD processing step is introduced for the HH subband before embedding the watermark information. This technique allows for the preservation of imperceptibility, and the integrity of the original image structure by modifying the singular values. By selecting appropriate singular values to embed the watermark, both the visual quality of the image and the invisibility of the watermark can be maintained. In

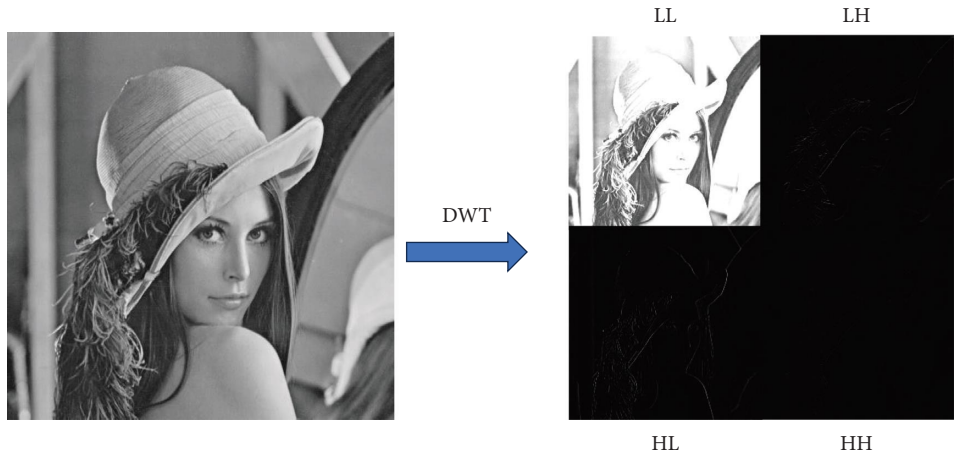

Figure 1: DWT result.

addition, the robustness of the watermark is enhanced via the inherent characteristics of the HH subband, such as their resistance to image processing [33].

Figure 2 shows the values of the HH subband in Figure 1, as well as the values of the S field after SVD.

In Figure 2, the S field post SVD transformation exhibits a diagonal matrix pattern, where the diagonal sections possess non-zero values while the remaining areas are zero. Moreover, the values on the diagonal are amplified, prompting the embedding of the watermark into the diagonal matrix.

3. Performance Analysis for Logistic-Tent Map

3.1. Bifurcation Diagram. The bifurcation diagram reflects the law that the iterative value of a system changes with its parameters, so it vividly describes the chaotic behaviors of the system [34]. The bifurcation diagrams of the logistic map, tent map, and logistic-tent map are shown in Figures 3(a)–3(c), respectively.

Compared with the logistic map and tent map, the logistic-tent map is a map with uniform distribution and a large chaotic parameter area, which indicates that the logistic-tent map has better chaotic properties.

3.2. Lyapunov Exponent. The Lyapunov exponent can be used to describe the sensitivity of chaotic systems to initial conditions. If a Lyapunov exponent of a chaotic system is positive, it indicates it is chaotic. The larger the range of Lyapunov exponent values, greater than 0, the better the chaotic characteristics of the system, besides, the larger the value, the greater the sensitivity to initial conditions [35]. The Lyapunov exponents of the logistic map, tent map, and logistic-tent map are shown in Figures 4(a)–4(c), respectively, where the yellow shaded regions represent the parameter domain with the chaotic property.

We observe from Figure 4 that the logistic-tent map has a greater positive Lyapunov exponent over the range of $\mu \in (0, 2]$, that is, the logistic-tent map has stronger chaotic characteristics and higher sensitivity to the initial value. This suggests that it may be a more appropriate choice for secure image encryption applications.

3.3. NIST SP800-22 Test. To assess the randomness of the chaotic sequence generated by the logistic-tent map, this paper utilized the National Institute of Standards and Technology SP800-22 Standard (NIST SP800-22) for testing. The NIST SP800-22 standard comprises 15 methods for detecting randomness, with each test generating a corresponding P value. P value greater than 0.01 indicates that the sequence passes the test [36]. In this study, 1000 sets of random sequences were generated using the logistic-tent map, each sequence comprising a length exceeding 10^6. The test results are presented in Table 1.

It follows from Table 1 that all P values of all test items are greater than 0.01, indicating that the sequence generated by the logistic-tent map has passed the NIST randomness test. Thus, the sequence has good randomness and is suitable for application in encryption algorithms.

4. Watermark Algorithm

4.1. Watermark Embedding. Let the size of the carrier image A and the watermark image W be $M \times N$ and $M/2 \times N/2$, respectively. The watermark embedding process is mainly divided into chaotic encryption (steps 1 to 3) and encrypted watermark embedding (steps 4 to 8), as shown in Figure 5.

Step 1: the logistic chaotic sequence E with length $M/2 \times N/2$ is generated by using the keys x and μ, and the sequence is converted to the interval [0, 255], obtaining the sequence C:

$$C = \text{floor}\left(E \times 10^8\right) \bmod 256, \tag{8}$$

where "floor" represents a downward integer operation.

Step 2: we divided the chaotic sequence C into $M/2$ parts and arranged them into $N/2$ rows, forming a chaotic image C' with a length of $M/2$ and a width of $N/2$.

Step 3: the XOR operation is then performed between the chaotic image C' and the watermark image W, and then a and b are inputted as scrambling parameters for Arnold transform to obtain the encrypted watermark image W_m.

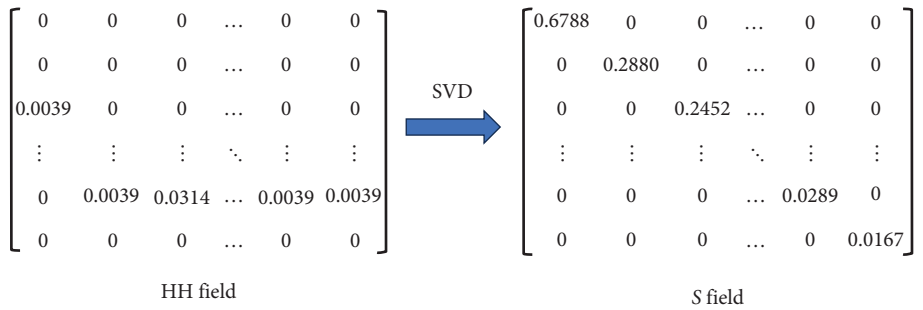

Figure 2: SVD result.

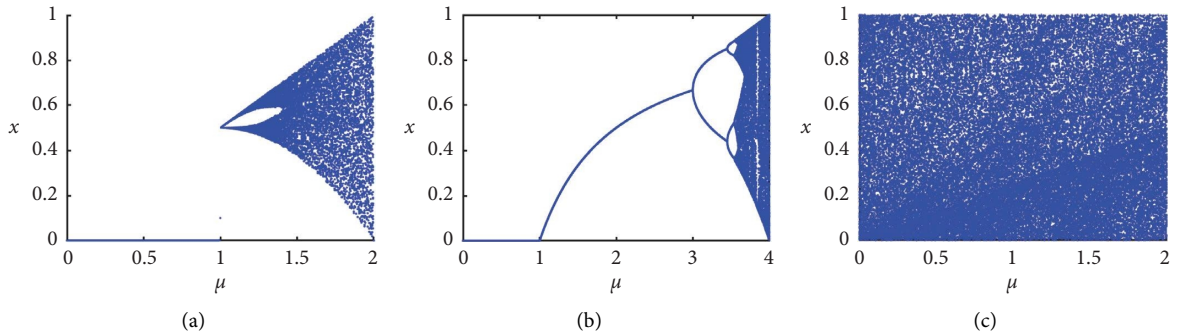

Figure 3: Bifurcation diagrams of different maps. (a) Logistic map. (b) Tent map. (c) Logistic-tent map.

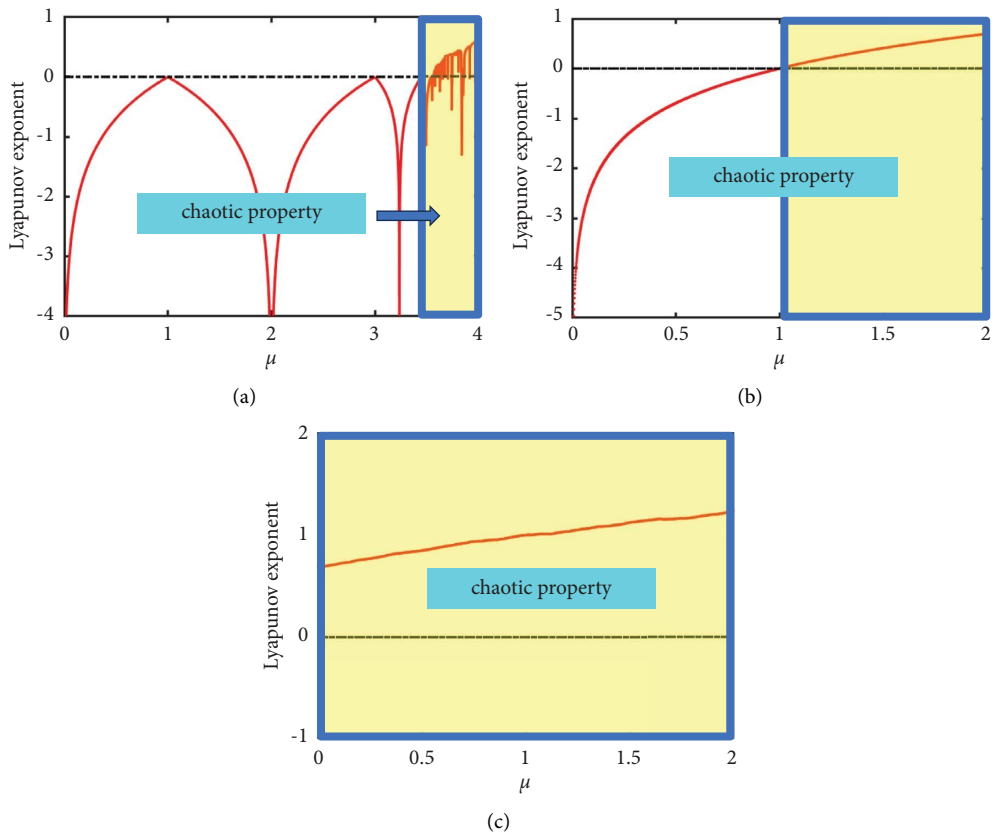

Figure 4: Lyapunov exponents of different maps. (a) Logistic map. (b) Tent map. (c) Logistic-tent map.

Table 1: NIST SP800-22 test.

Test name	P value	Result
Frequency	0.924076	Success
Block frequency	0.350485	Success
Cumulative sums	0.574903	Success
Runs	0.383827	Success
Longest run	0.040108	Success
Rank	0.759756	Success
Fast Fourier transform	0.419021	Success
Nonoverlapping template	0.616305	Success
Overlapping template	0.918793	Success
Universal	0.171867	Success
Approximate entropy	0.228189	Success
Random excursions	0.534146	Success
Random excursions variant	0.739918	Success
Serial	0.262249	Success
Linear complexity	0.595549	Success

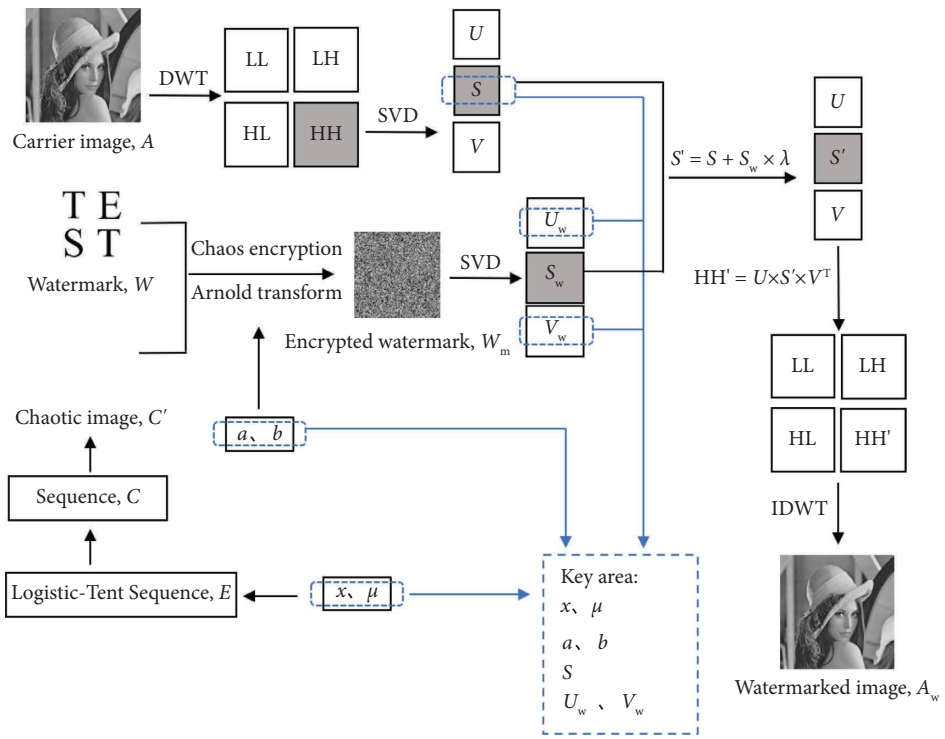

Figure 5: Watermark embedding algorithm flowchart.

Step 4: the DWT operation is performed on the carrier image A, and the HH subband is chosen for SVD to acquire the singular value diagonal matrix S, which is then stored in the key area.

Step 5: the SVD transform is applied to the watermark image W_m resulting in the singular value matrix S_w, as well as the ordinary matrices U_w and V_w. Both U_w and V_w are subsequently stored in the key area.

Step 6: S_w is then embedded into S to obtain S' with watermark information. The specific operations are as follows:

$$S' = S + S_w \times \lambda, \tag{9}$$

where λ is the embedded strength coefficient. Since the energy of the HH domain is low, the corresponding singular value is also low. If the singular value is modified too much, it will lead to great changes in the HH domain of the carrier image and affect the image quality. Therefore, the maximum element in the singular value matrix of the carrier image and the watermark image is taken to calculate the calculation formula as follows:

$$\lambda = \frac{S_1}{(2 \times S_{W1})}. \tag{10}$$

where S_1 is the first element in the singular value matrix of the carrier image in the HH domain and S_{W1} is the first element in the singular value matrix of the watermark image.

Step 7: the high-frequency subband HH′ with watermark information is obtained through inverse SVD transformation, and the calculation formula is as follows:

$$\mathrm{HH}' = U \times S' \times V^T. \tag{11}$$

Step 8: the final watermarked image A_w is obtained by IDWT transform.

4.2. Watermark Extraction. The watermark extraction process is the reverse process of the encryption process, as shown in Figure 6.

Step 1: a DWT is performed on image A_w, which embeds the watermark, to generate subbands LL, LH, HL, and the subband HH′ with watermark information.

Step 2: the subband HH′ is selected for SVD to generate U and V without watermark information and S' with watermark information. Since the watermark information exists in the S' domain, we only operate on the S' domain.

Step 3: the watermark image S_w' value is then calculated by extracting S from the key area and performing calculations with S'. The specific operations are as follows:

$$S_w = \frac{(S' - S)}{\lambda}. \tag{12}$$

Step 4: By extracting U_w and V_w in the key area and combining S_w, the encrypted watermark image W_m is restored. The specific operations are as follows:

$$W_m = U_w \times S_w \times {V_w}^T. \tag{13}$$

Step 5: a length $M/2 \times N/2$ chaotic sequence E is then generated using x and μ extracted from the key area.

Step 6: the chaotic sequence E is then converted into a chaotic sequence C following the calculation formula (8).

Step 7: the sequence C is arranged in rows of $M/2$ to create a chaotic image C' of size $M/2 \times N/2$.

Step 8: Using the scrambling parameters a and b, the re-Arnold transform is performed on W_m, and then the XOR operation on C' is performed to obtain the watermark image W.

5. Experimental Results and Discussion

In this section, we test the invisibility, robustness, sensitivity, and complexity of the algorithm and the performance of the encrypted watermark image. All of the experiments were carried out on a Workstation with an Intel i7 CPU and 16 GB RAM, using the MATLAB 2020b version. The images used in the experiment include (a–g) from the USC-SIPI Image Database and landscape photos (h–j). The grayscale size of each image is 512×512. In addition, a binary watermark image of size 256×256 was used in Figure 7(k).

The peak signal noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate the invisibility of the watermark. The normalized cross-correlation (NC) and bit error rate (BER) are used to evaluate the robustness of the watermark. The number of pixels change rate (NPCR), unified average changing intensity (UACI), pixel correlation, and information entropy are used to evaluate the performance of the encrypted watermark image.

PSNR is the ratio between the maximum value of the measured signal and the amount of noise affecting the signal, which is used to compare the quality of the carrier image before and after embedding the watermark image in decibels (dB). PSNR value lower than 30 dB indicates a low image quality and higher than 40 dB indicates a high image quality [37]. The calculation formula of PSNR is as follows:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{M \times N \times 255^2}{\sum_{i=1}^{M}\sum_{i=1}^{N}\left[A(i,j) - A_w(i,j)\right]^2}\right), \tag{14}$$

where $M \times N$ represents the size of the image, A represents the carrier image, and A_w represents the watermarked image.

SSIM is an index used to measure the similarity between the carrier image and the watermarked image. The closer the SSIM value is to 1, the smaller the difference between the two images, and vice versa, and the greater the difference [38]. The calculation formula of SSIM is as follows:

$$\mathrm{SSIM} = \frac{\left(2\mu_A\mu_{Aw} + c_1\right)\left(2\sigma_{AA_w} + c_2\right)}{\left(\mu_A^2 + \mu_{A_w}^2 + c_1\right)\left(\sigma_A^2 + \sigma_{A_w}^2 + c_2\right)}, \tag{15}$$

where μ_A and μ_{Aw} represent the average value of A and A_w, σ_A and σ_{Aw} represent the variance of A and A_w, σ_{AAw} represents the covariance of A and A_w, and c_1 and c_2 are constants.

NC is used to evaluate the similarity between the extracted watermark and the original watermark, which can be calculated by [39]

$$\mathrm{NC} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}A(i,j)A_w(i,j)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}A(i,j)^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}A_w(i,j)^2}}. \tag{16}$$

BER is used to calculate the error bit ratio between the extracted watermark and the original watermark. The lower the BER, the stronger the robustness of the watermark [40]. The calculation formula of BER is as follows:

$$\mathrm{BER} = \frac{\mathrm{sum}\left(A \oplus A_{\mathrm{w}}\right)}{M \times N} \times 100\%, \tag{17}$$

where sum represents the sum operation and $\oplus$ represents the XOR operation.

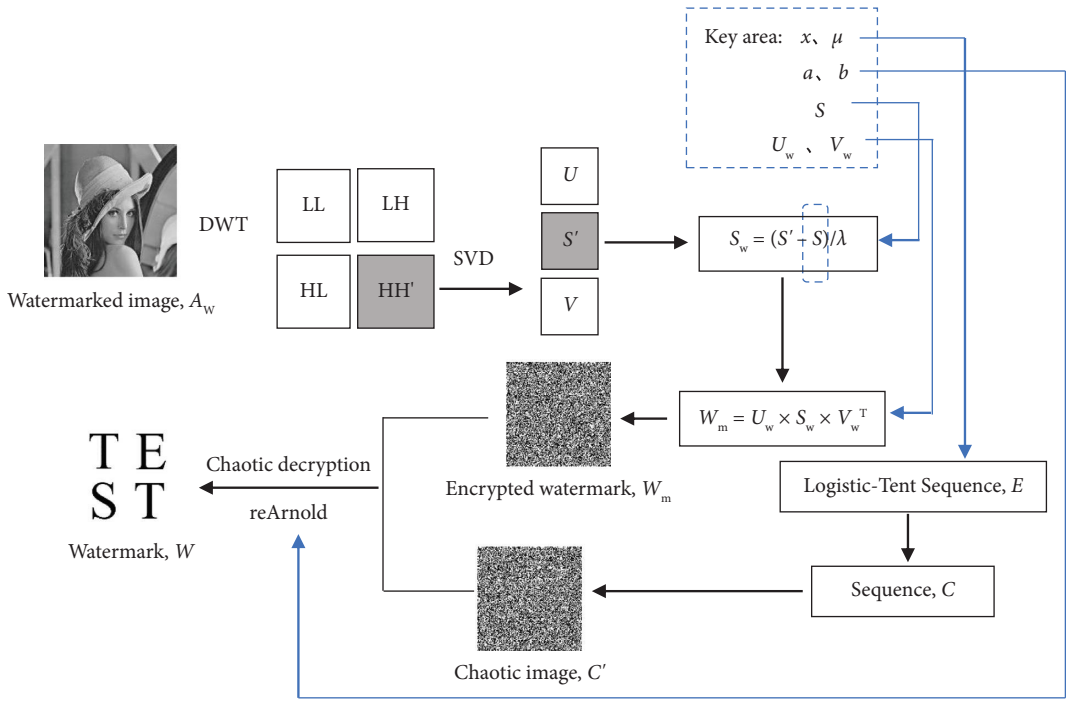

Figure 6: Watermark extraction algorithm flowchart.

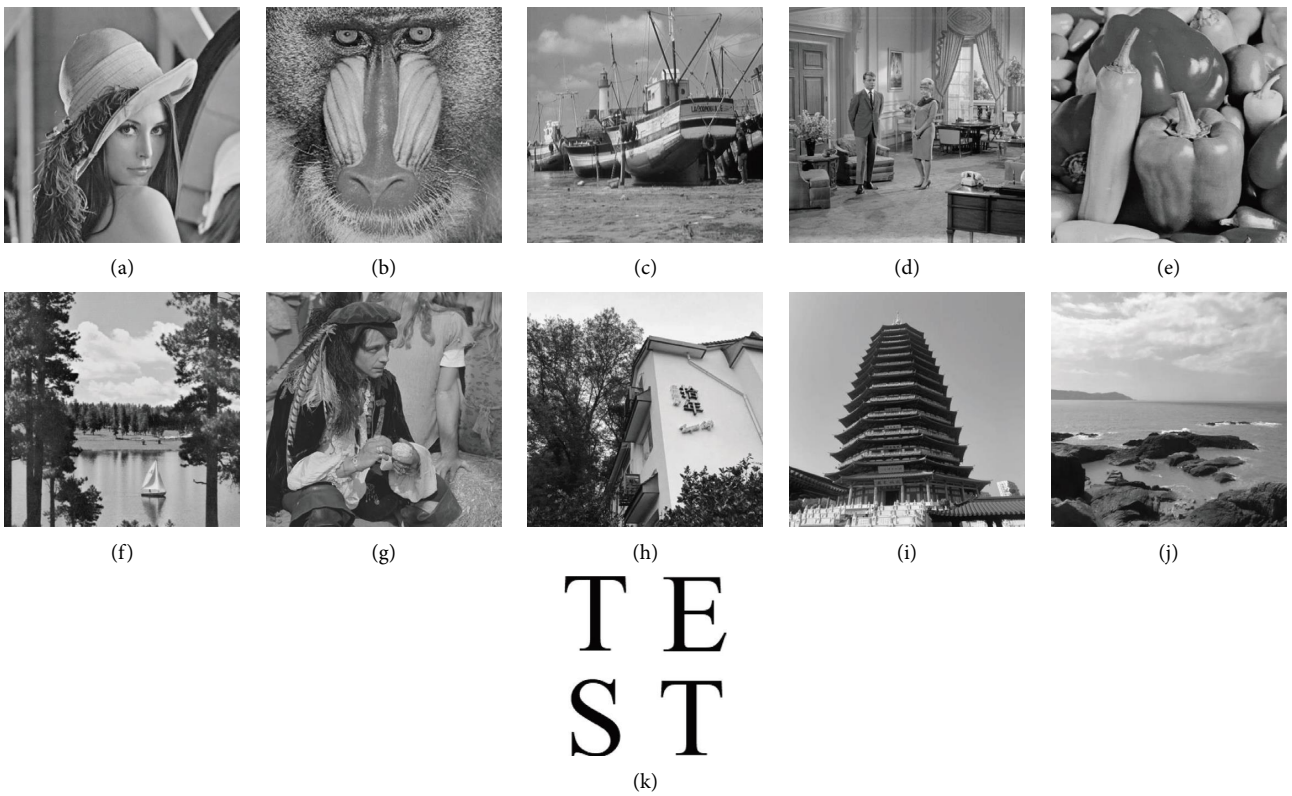

Figure 7: (a–j) Carrier images. (k) Watermark image. (a) Lena. (b) Baboon. (c) Boat. (d) Couple. (e) Pepper. (f) Lake. (g) Man. (h) House. (i) Tower. (j) Sea. (k) Test.

NPCR refers to the proportion of the number of pixels whose pixel values change between two images in the total number of pixels. It is usually used to describe the degree of difference between images. UACI refers to the average change range of pixels with pixel value changes between two images. It is used to describe the average degree of pixel value

change. The calculation formulas of NPCR and UACI are as follows [41]:

$$\mathrm{NPCR}(W_1, W_2) = \frac{\sum_{i,j} G(i,j)}{M \times N} \times 100\%, \tag{18}$$

$$\mathrm{UACI}(W_1, W_2) = \frac{1}{M \times N} \sum_{i,j} \frac{|W_1(i,j) - W_2(i,j)|}{255} \times 100\%, \tag{19}$$

where $W_1(i, j)$ and $W_2(i, j)$ are the pixel values of the pixels in row i and column j of the W_1 image and W_2 image, respectively. When $W_1(i, j) = W_2(i, j)$, $G(i, j) = 0$, otherwise $G(i, j) = 1$. The expected values of NPCR and UACI for grayscale images are 99.6094% and 33.4635%, respectively [28].

The correlation between adjacent pixels refers to the degree of correlation or correlation between adjacent pixels in the image. The range of correlation coefficient r is $[-1, 1]$. The closer the absolute value of correlation is to 0, the smaller the correlation between pixels is. The correlation calculation formula is as follows [42]:

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{20}$$

$$\mathrm{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), \tag{21}$$

$$E(x) = \frac{1}{N} \sum_{i=0}^{N} x_i, \tag{22}$$

$$D(x) = \frac{1}{N} \sum_{i=0}^{N} (x_i - E(x))^2, \tag{23}$$

where x and y are the gray values of adjacent pixels, $\mathrm{cov}(x, y)$ is the covariance, D is the variance, and E is the expectation.

Image information entropy is an index to measure the complexity of image information content. The maximum entropy of a gray image is 8. The better the encryption effect is, the closer the information entropy is to 8. The calculation is as follows [43]:

$$H(x) = -\sum_{i=1}^{2N-1} P_i \log_2 P_i, \tag{24}$$

where P_i is the probability of gray value i.

5.1. Invisibility Test. In the watermark invisibility analysis experiment, the watermark embedding algorithm proposed in this paper is used to embed Figure 7(k) into Figures 7(a)–7(j), respectively, to obtain the watermarked images.

To visually demonstrate the invisibility of the watermark, the analysis focuses on the grayscale distribution of the images through the use of image histograms. An image histogram, acting as a statistical table, illustrates the distribution of grayscale values across the image, offering insights into the overall grayscale distribution. In Figure 8, a comparison is made between the histograms of the original images and the watermarked images from Figures 7(a)–7(g). This comparison facilitates an assessment of how the watermark influences the overall distribution of grayscale values in the images.

To further assess the quality of the watermarked images, a comparative analysis is performed by comparing them with the original carrier images. The results of this evaluation are presented in Table 2, which includes important metrics such as PSNR and SSIM.

Besides, the watermark invisibility of our watermark embedding algorithm is compared with that of other watermark embedding algorithms mentioned in prior literature. The results of the comparison are presented in Table 3.

5.2. Robustness Test. In the robustness test and analysis experiment, Figures 7(a)–7(j) are selected as carrier images, and Figure 7(k) is chosen as the watermark image. The algorithm described in this paper is employed for embedding the watermark in carrier images. Subsequently, salt and pepper noise (density 0.01, 0.1, and 0.3), Gaussian noise (variance 0.01, 0.1, and 0.3), speckle noise (variance 0.05, 0.1, and 0.3), cropping (25% rows and 50% rows), rotation (counterclockwise rotation 45° and 60°), JPEG compression (compression factors 80 and 60, respectively), median filtering (window size 3×3 and 5×5), average filtering (window size 3×3 and 5×5), sharpening, brightening (30%), darkening (30%), and histogram equalization are applied to attack these watermarked images. Finally, the watermark extraction algorithm introduced in this paper is utilized to extract the watermark images.

The attack types and descriptions are shown in Table 4, the extracted watermark images are presented in Table 5, the corresponding NC values are shown in Table 6, and the BER values are reported in Table 7.

Another experiment was conducted using lena, pepper, and boat as the carrier images to evaluate the NC values of the extracted watermark under different attacks, including Gaussian noise (variance 0.001), salt and pepper noise (density 0.001), cropping (25% center, 25% rows, and 50% rows), and JPEG compression (compression factor 70 and 60). The results were compared with those of references [46, 47], as shown in Table 8.

Furthermore, the BER values of the extracted watermark were evaluated under attacks such as Gaussian noise (variance of 0.001 and 0.1), salt and pepper noise (density of 0.01), cropping (25% rows and 50% rows), JPEG compression (compression factors of 80), median filtering (window size 3×3), and average filtering (window size 3×3). The results were compared with those of references [46, 48], as presented in Table 9.

5.3. Sensitivity Test. Using the logistic-tent map for chaotic encryption of the watermark can further ensure the security of watermark information. In the process of chaotic mapping encryption of the watermark, x_0 and μ are used to encrypt and decrypt the watermark. Only users who possess

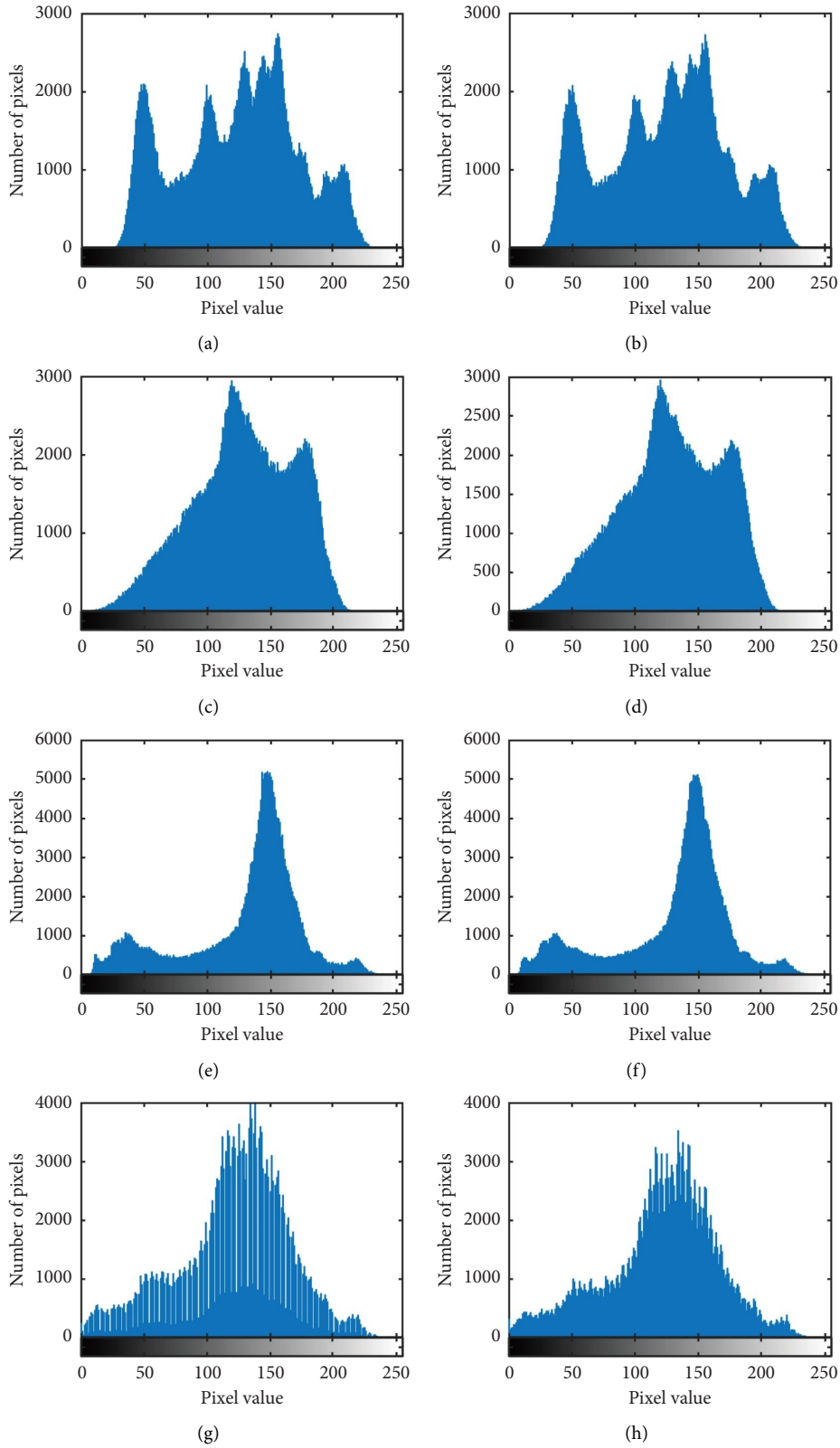

Figure 8: Continued.

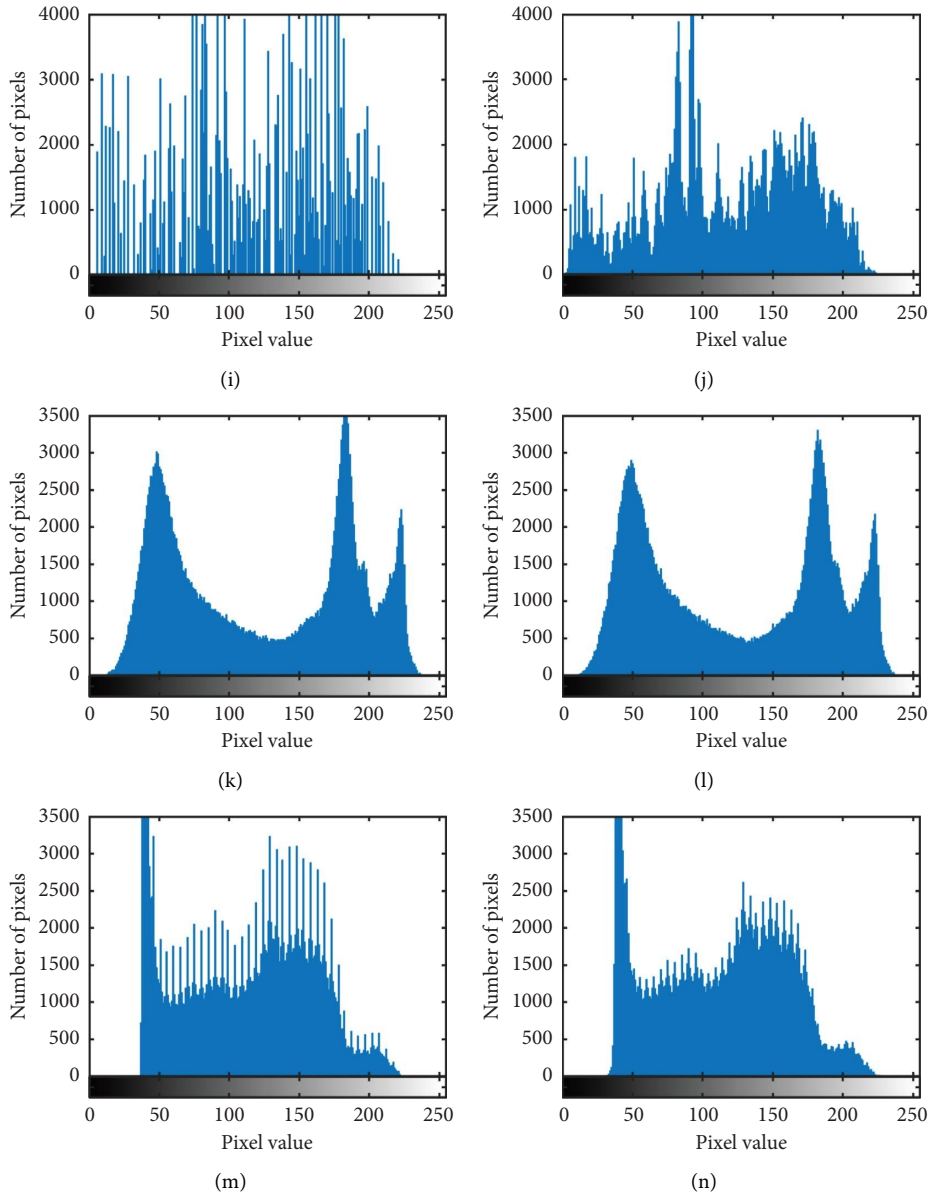

Figure 8: Histogram comparison. (a) Lena. (b) Watermarked lena. (c) Baboon. (d) Watermarked baboon. (e) Boat. (f) Watermarked boat. (g) Couple. (h) Watermarked couple. (i) Pepper. (j) Watermarked pepper. (k) Lake. (l) Watermarked lake. (m) Man. (n) Watermarked man.

Table 2: Invisibility test.

Carrier image	PSNR	SSIM
Lena	43.0618	0.9894
Baboon	43.0897	0.9964
Boat	43.0672	0.9945
Couple	43.0514	0.9856
Pepper	43.0592	0.9875
Lake	43.0617	0.9889
Man	43.0723	0.9902
House	43.0894	0.9961
Tower	43.0903	0.9965
Sea	43.0642	0.9893

Table 3: Invisibility test comparison with related work.

Carrier image	Ref. [40]		Ref. [44]		Ref. [45]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	41.16	0.9878	42.1048	0.9743	35.7755	0.9150	43.0618	0.9894
Pepper	41.27	0.9884	42.2843	0.9775	35.7955	0.9169	43.0592	0.9875
Boat	41.29	0.9915	—	—	—	—	43.0672	0.9945
Lake	—	—	42.1115	0.9808	—	—	43.0617	0.9889
Baboon	—	—	42.2743	0.9920	34.9920	0.9080	43.0897	0.9964

Table 4: Attack types and descriptions.

Index	Description of the attack
1	Attack free
2	Salt and pepper noise with a noise density of 0.01
3	Salt and pepper noise with a noise density of 0.1
4	Salt and pepper noise with a noise density of 0.3
5	Gaussian noise with a variance of 0.01
6	Gaussian noise with a variance of 0.1
7	Gaussian noise with a variance of 0.3
8	Speckle noise with a variance of 0.05
9	Speckle noise with a variance of 0.1
10	Speckle noise with a variance of 0.3
11	Cropping 25% rows
12	Cropping 50% rows
13	Counter-clockwise rotation by 45°
14	Counter-clockwise rotation by 60°
15	JPEG compression with a quality factor of 80
16	JPEG compression with a quality factor of 60
17	Median filtering with a window size of 3×3
18	Median filtering with a window size of 5×5
19	Average filtering with a window size of 3×3
20	Average filtering with a window size of 5×5
21	Sharpening
22	Brightening by 50%
23	Darkening by 30%
24	Histogram equalization

the correct keys can successfully extract the watermark information, thereby protecting the watermark.

By using Figure 7(k) as the watermark image, the watermark was initially encrypted through a chaotic encryption process with the parameters x_0 and μ. Following this, the watermark embedding algorithm proposed in this study was utilized for the embedding process. An extraction algorithm was then applied to recover the encrypted watermark. Throughout the watermark decryption process, one parameter was held constant while making minor adjustments to the other. Subsequently, the decrypted images were extracted, and the results are depicted in Figure 9.

5.4. Complexity Test. An effective watermark algorithm should guarantee minimal computational cost for both embedding and extracting watermark information. Table 10 presents a time comparison between the presented approach and various methods. The time required for the maximum embedding and extraction watermark capacity reported in each reference is selected for comparison. The authors in reference [30] achieved a maximum watermark embedding capacity of 64×64 bits, the authors in reference [49] achieved 128×128 bits, the authors in reference [50] achieved 256×256 bits, and our algorithm achieved a maximum watermark embedding capacity of 256×256 bits.

5.5. Encryption Watermark Performance Test. To evaluate the encrypted watermark image, we conducted NPCR, UACI, entropy, and pixel correlation tests on it.

Figure 7(k) is selected as the plaintext image, and a new plaintext image is obtained by randomly changing the pixels of the plaintext image. Then, the same algorithm is used to encrypt the two plaintext images to obtain the corresponding ciphertext images. Finally, the NPCR and UACI of the two ciphertext images are calculated, respectively. To test the correlation, the plaintext image is first encrypted using the proposed logistic-tent mapping. Then, 5000 pairs of pixels are randomly selected from the original image and its related encrypted images in the horizontal, vertical, and diagonal directions. For the information entropy test, the information entropy of plaintext and ciphertext is calculated, respectively. The results are shown in Table 11.

5.6. Discussion. In this discussion, we will analyze and interpret the findings presented in Figures 8, 9, and Tables 2 to 11 of the paper.

In Figure 8, it is evident that the histogram of the watermarked images closely resembles that of the original carrier images. This observation indicates that the image undergoes minimal alterations upon embedding the watermark. However, when the carrier images exhibit a relatively discrete distribution of pixel values, the pixel value distribution of the watermarked images tends to exhibit an averaging effect.

From Table 2, it is observed that the PSNR values of the carrier images with watermarks are all greater than 43, and the SSIM values are close to 1. This suggests that the images with watermarks have higher quality and show minimal differences from the original carrier images. Hence, the proposed watermarking algorithm demonstrates good invisibility, indicating that the watermark is imperceptible to the human eye.

Table 3 shows that references [40, 44], and [45] achieve remarkable performance in terms of watermark invisibility. However, when compared to these references, the proposed algorithm achieves slightly higher PSNR and SSIM values. These results highlight the superior performance of the proposed algorithm in terms of watermark invisibility compared to the other references.

Table 5: Extracted watermarks for multiple attacks on watermarked images.

Attacks	Lena	Baboon	Boat	Couple	Pepper	Lake	Man	House	Tower	Sea
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										

Table 6: NC values for attacks on watermarked images.

Attacks	Lena	Baboon	Boat	Couple	Pepper	Lake	Man	House	Tower	Sea
1	1	1	1	1	1	1	1	1	1	1
2	0.9996	0.9996	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9995
3	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994
4	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994
5	0.9995	0.9995	0.9995	0.9995	0.9995	0.9995	0.9995	0.9995	0.9995	0.9995
6	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994
7	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994
8	0.9997	0.9997	0.9997	0.9997	0.9998	0.9997	0.9998	0.9996	0.9997	0.9996
9	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9995	0.9995
10	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994
11	0.9877	0.9285	0.9961	0.9933	0.9750	0.9769	0.9030	0.9700	0.9908	0.9926
12	0.8885	0.9436	0.8576	0.8340	0.9224	0.8787	0.9412	0.8864	0.9800	0.9858
13	0.9911	0.9846	0.9986	0.9989	0.9767	0.9804	0.9936	0.9934	0.9946	0.9867
14	0.9882	0.9803	0.9988	0.9993	0.9743	0.9669	0.9877	0.9929	0.9954	0.9941
15	0.9245	0.9707	0.9359	0.9242	0.9037	0.8382	0.9763	0.9945	0.9861	0.9587
16	0.7964	0.9293	0.8536	0.7231	0.7948	0.8557	0.8298	0.9867	0.9707	0.9278
17	0.9418	0.9054	0.9622	0.9080	0.8629	0.9123	0.8756	0.9604	0.9146	0.9540
18	0.8508	0.8217	0.8758	0.8261	0.8040	0.8277	0.8093	0.8594	0.8423	0.8699
29	0.9825	0.9529	0.9880	0.9641	0.9245	0.9293	0.9439	0.9874	0.9461	0.9865
20	0.9714	0.9292	0.8142	0.9450	0.8933	0.8769	0.9097	0.8158	0.9121	0.9796
21	0.9992	0.9989	0.9991	0.9993	0.9993	0.9898	0.9993	0.9993	0.9993	0.9991
22	0.9997	0.9996	0.9997	0.9997	0.9997	0.9997	0.9997	0.9996	0.9994	0.9946
23	0.9788	0.9646	0.9825	0.9731	0.9707	0.9597	0.9691	0.9822	0.9654	0.9811
24	0.9996	0.9994	0.9995	0.9995	0.9996	0.9991	0.9996	0.9992	0.9997	0.9977

Table 7: BER values for attacks on watermarked images.

Attacks	Lena	Baboon	Boat	Couple	Pepper	Lake	Man	House	Tower	Sea
1	0	0	0	0	0	0	0	0	0	0
2	0.0110	0.0108	0.0110	0.0108	0.0109	0.0109	0.0109	0.0107	0.0109	0.0111
3	0.0118	0.0118	0.0118	0.0117	0.0118	0.0118	0.0118	0.0117	0.0118	0.0118
4	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119
5	0.0116	0.0115	0.0116	0.0115	0.0115	0.0115	0.0115	0.0114	0.0115	0.0116
6	0.0119	0.0119	0.0119	0.0119	0.0118	0.0119	0.0119	0.0119	0.0119	0.0119
7	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119	0.0119
8	0.0097	0.0091	0.0097	0.0091	0.0091	0.0094	0.0089	0.0096	0.0094	0.0101
9	0.0109	0.0107	0.0110	0.0109	0.0108	0.0109	0.0108	0.0107	0.0110	0.0111
10	0.0117	0.0117	0.0118	0.0118	0.0118	0.0118	0.0118	0.0118	0.0118	0.0118
11	0.0702	0.0750	0.0650	0.0674	0.0653	0.0732	0.0579	0.0673	0.0720	0.0725
12	0.0894	0.0697	0.0959	0.1253	0.0772	0.0889	0.0421	0.0912	0.0697	0.0725
13	0.0209	0.0240	0.0120	0.0112	0.0306	0.0277	0.0203	0.0167	0.0155	0.0219
14	0.0305	0.0275	0.0127	0.0109	0.0363	0.0378	0.0315	0.0192	0.0160	0.0179
15	0.0764	0.0736	0.0735	0.0749	0.0702	0.0576	0.0648	0.0413	0.0675	0.0656
16	0.1205	0.0737	0.0779	0.2097	0.1897	0.1408	0.1054	0.0515	0.0731	0.0727
17	0.0671	0.0949	0.0496	0.0949	0.1330	0.0790	0.1210	0.0500	0.0836	0.0546
18	0.0744	0.1376	0.0812	0.1359	0.1650	0.1307	0.1570	0.0896	0.1211	0.1179
19	0.0304	0.0520	0.0240	0.0488	0.0809	0.0624	0.0643	0.0253	0.0569	0.0259
20	0.0414	0.0670	0.1627	0.0650	0.1006	0.0810	0.0846	0.1994	0.0771	0.0342
21	0.0116	0.0116	0.0114	0.0117	0.0117	0.0243	0.0117	0.0118	0.0117	0.0115
22	0.0090	0.0096	0.0087	0.0095	0.0092	0.0098	0.0094	0.0098	0.0119	0.0200
23	0.0759	0.0756	0.0759	0.0756	0.0759	0.0759	0.0759	0.0759	0.0758	0.0746
24	0.0098	0.0107	0.0109	0.0106	0.0009	0.0112	0.0101	0.0253	0.0091	0.0309

From Table 5, it can be observed that the watermark remains identifiable for the majority of attacks.

The results from Tables 6 and 7 indicate that the proposed watermark algorithm withstands all the attacks, maintaining NC values close to 1 and BER values close to 0. These results demonstrate that the proposed algorithm exhibits excellent robustness when facing various image attacks.

From Tables 8 and 9, it can be seen that the proposed algorithm performs better than the references in terms of Gaussian noise, salt and pepper noise, and cropping attacks.

Table 8: NC values compared with that of references [46, 47].

Attacks	Lena			Pepper			Boat		
	Ref. [46]	Ref. [47]	Ours	Ref. [46]	Ref. [47]	Ours	Ref. [46]	Ref. [47]	Ours
Gaussian noise (0.001)	0.9270	0.9932	0.9998	0.9702	0.9591	0.9997	0.9727	0.9258	0.9998
Salt and pepper noise (0.001)	0.9728	0.9883	0.9986	0.9865	0.9863	0.9998	0.9728	0.9260	0.9922
Center cropping 25%	0.9777	0.8375	0.9977	0.9610	0.9912	0.9962	0.9604	0.9120	0.9985
Cropping 25% rows	0.8681	0.8429	0.9877	0.8908	0.8392	0.9750	0.8595	0.6119	0.9961
Cropping 50% rows	0.7543	0.7144	0.8885	0.6649	0.7161	0.9224	0.7663	0.4812	0.8576
JPEG (QF = 70)	1	0.9990	0.8252	1	0.9592	0.8231	1	0.9271	0.8763
JPEG (QF = 60)	1	0.9990	0.7964	1	0.9523	0.7948	1	0.9271	0.8536

Table 9: BER values compared with that of references [46, 48].

Attacks	Lena			Pepper			Boat		
	Ref. [46]	Ref. [48]	Ours	Ref. [46]	Ref. [48]	Ours	Ref. [46]	Ref. [48]	Ours
Gaussian noise (0.001)	0.0224	—	0.0090	0.0107	—	0.0097	0.0176	—	0.0086
Gaussian noise (0.01)	—	0.1943	0.0119	—	0.1608	0.0118	—	0.1445	0.0119
Salt and pepper noise (0.01)	0.0439	0.1445	0.0110	0.0566	0.1025	0.0109	0.0551	0.0986	0.0110
Cropping 25% rows	—	0.1221	0.0702	—	0.1143	0.0653	—	0.1143	0.0650
Cropping 50% rows	—	0.2441	0.0894	—	0.2480	0.0772	—	0.2402	0.0959
JPEG (QF = 80)	0	0.0068	0.0764	0	0	0.0702	0	0.0020	0.0735
Median filtering (3 × 3)	0	0.0107	0.0671	0	0.0029	0.1330	0	0.0039	0.0496
Average filtering (3 × 3)	0	0.0039	0.0304	0	0.0078	0.0809	0.0049	0.0088	0.0240

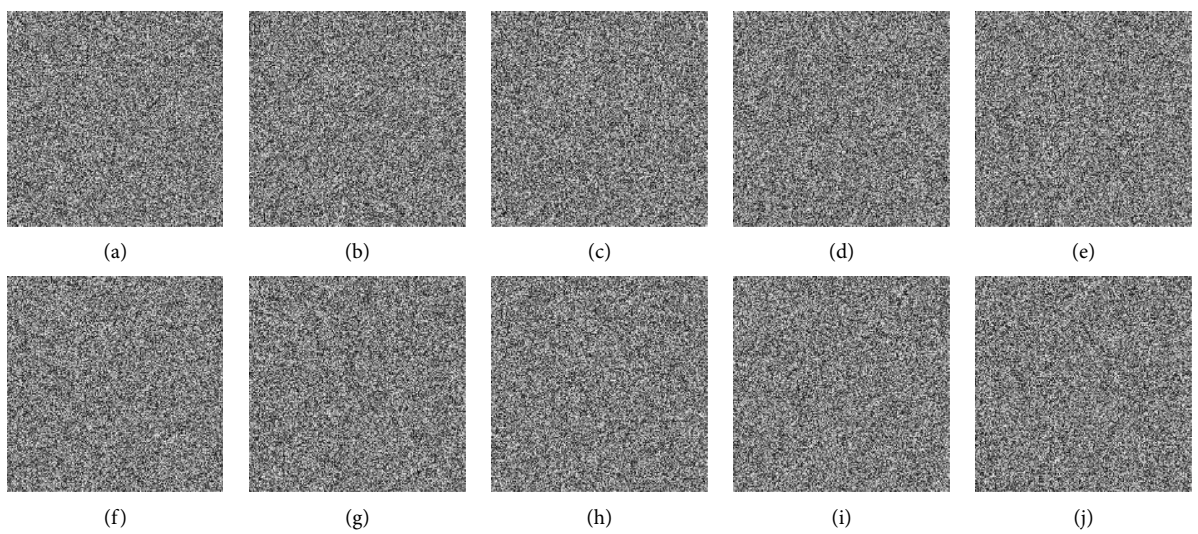

Figure 9: Decrypted watermark images. (a) $x_0 + 1 \times 10^{-8}$. (b) $x_0 - 1 \times 10^{-8}$. (c) $x_0 + 2 \times 10^{-8}$. (d) $x_0 - 2 \times 10^{-8}$. (e) $x_0 + 1 \times 10^{-9}$. (f) $\mu + 1 \times 10^{-8}$. (g) $\mu - 1 \times 10^{-8}$. (h) $\mu + 2 \times 10^{-8}$. (i) $\mu - 2 \times 10^{-8}$. (j) $\mu + 1 \times 10^{-9}$.

Table 10: The time of embedding and extraction (second).

Methods	Embedding process	Extraction process	Total time
Ref. [30]	0.3035	0.2184	0.5219
Ref. [49]	3.8826	2.7631	6.6457
Ref. [50]	1.3675	2.6803	4.0478
Ours	1.4177	1.2847	2.7024

Table 11: Encryption performance indicators of watermarked image

Image	NPCR (%)	UACI (%)	Entropy	Correlation coefficient		
				Horizontal	Vertical	Diagonal
Test (plain)	—	—	0.4309	0.90883	0.91706	0.85716
Test (cipher)	99.6089	33.5712	7.9944	0.00502	0.01194	0.00318

However, it is less effective than the reference [46] in countering compression attacks and filtering attacks.

Figure 9 demonstrates that even a slight change in the key leads to a completely incorrect decrypted image. This finding emphasizes the strong sensitivity of the watermark, ensuring effective protection of the watermark's security.

From Table 10, it can be observed that our algorithm exhibits superior computational efficiency compared to references [49, 50]. Although our algorithm slightly lags behind reference [30] in terms of computational efficiency, the main reason is the inclusion of encryption and decryption of the watermark image during the embedding and extraction processes, respectively. Overall, our algorithm demonstrates satisfactory performance concerning computational efficiency.

It can be seen from Table 11 that the NPCR value, UACI value, and entropy value are close to the theoretical value, and the correlation of the encrypted watermark image in the horizontal, vertical, and diagonal directions is close to 0, indicating that the logistic-tent has a good encryption effect.

In conclusion, the results presented in the tables and figures strongly support the claims made in the paper regarding the effectiveness of the proposed watermarking algorithm. The algorithm achieves a high level of invisibility, demonstrating robustness against various attacks and ensuring the security of the watermark. These findings underscore the significant potential of the proposed algorithm for practical applications in the field of digital watermarking.

6. Conclusions

This paper introduces a robust image watermarking method based on DWT-SVD and a chaotic map. The logistic-tent map is first proposed, and its bifurcation diagram, Lyapunov exponent, and NIST SP800-22 test are tested to demonstrate its robust chaotic properties. By utilizing this map, along with the Arnold transform, the watermark is encrypted. Regarding invisibility, DWT is employed to embed the watermark in the frequency domain, with the high-frequency subband chosen as the embedding region after frequency decomposition. The algorithm undergoes watermark invisibility and robustness testing, revealing that the PSNR values of watermarked images using this algorithm are consistently above 43, and SSIM values are close to 1. The algorithm also exhibits strong resistance to various noise and filtering attacks, as well as different attack parameters for shear, rotation, and JPEG compression attacks. It demonstrates NC values above 0.9 and BER values below 0.1. The algorithm showcases robustness, ensuring watermark invisibility and security, making it suitable for applications in copyright protection and security verification.

However, the algorithm proposed in this paper has limitations. First, this algorithm is a semiblind watermarking algorithm, requiring the original carrier image and watermark image to participate in the watermark extraction process. Second, the image watermarking algorithm is only applicable to embedding grayscale images. Then, due to the length and width of the subband images obtained after DWT being half of the original image, the watermark image is limited by the carrier image, with a maximum size of 1/4 of the carrier image. In future work, we will explore blind watermarking technology, and color watermarking embedding, and adjust the embedding position, capacity, and strength of the color watermark based on the different characteristics of the embedding region, aiming to achieve a good balance between invisibility and robustness. Furthermore, we will explore new methods to enable watermark image embedding without being constrained by the size of the carrier image. Lastly, traditional watermarking algorithms often rely on manually designed features and rules, while deep learning can automatically extract and embed watermarks by learning features and patterns from data. In subsequent research, we will delve into this area.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there have no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to express gratitude to his supervisor, Guangyi Wang, for his professional knowledge, rigorous academic attitude, and patient guidance, which have greatly benefited and allowed to gain a deeper understanding of the academic field. At the same time, the authors would also like to thank his senior colleague, Yujiao Dong, for providing with valuable assistance and suggestions for completing this research. This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China (Grant no. LQ23F010018) and in part by the National Natural Science Foundation of China (Grant no. 61771176).

References

[1] T. F. Li, J. B. Li, J. Liu, M. X. Huang, Y. W. Chen, and U. A. Bhatti, "Robust watermarking algorithm for medical images based on log-polar transform," *EURASIP Journal on Wireless Communications and Networking*, vol. 24, 2022.

[2] K. Jyothsna Devi, P. Singh, H. K. Thakkar, and N. Kumar, "Robust and secured watermarking using Ja-Fi optimization for digital image transmission in social media," *Applied Soft Computing*, vol. 131, Article ID 109781, 2022.

[3] M. Swain and D. Swain, "An effective watermarking technique using BTC and SVD for image authentication and quality recovery," *Integration*, vol. 83, pp. 12–23, 2022.

[4] H. Zhang, Z. Li, X. Liu, C. Wang, and X. Wang, "Robust image watermarking algorithm based on QWT and QSVD using 2D Chebyshev-Logistic map," *Journal of the Franklin Institute*, vol. 359, no. 2, pp. 1755–1781, 2022.

[5] G. Costa, P. Degano, L. Galletta, and S. Soderi, "Formally verifying security protocols built on watermarking and jamming," *Computers & Security*, vol. 128, Article ID 103133, 2023.

[6] U. A. Bhatti, Z. Yu, J. Chanussot et al., "Local similarity-based spatial–spectral fusion hyperspectral image classification with deep CNN and gabor filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[7] U. A. Bhatti, Z. Ming-Quan, Q. S. Huo et al., "Advanced color edge detection using clifford algebra in satellite images," *IEEE Photonics Journal*, vol. 13, no. 2, pp. 1–20, 2021.

[8] U. A. Bhatti, Z. Zeeshan, M. M. Nizamani, S. Bazai, Z. Yu, and L. Yuan, "Assessing the change of ambient air quality patterns in Jiangsu Province of China pre-to post-COVID-19," *Chemosphere*, vol. 288, Article ID 132569, 2022.

[9] G. D. Liu, H. Y. Wang, and C. B. Miao, "A three-dimensional text image watermarking model based on multilayer overlapping of extracted two-dimensional information," *Information Processing & Management*, vol. 60, no. 1, Article ID 103122, 2023.

[10] S. Gupta, K. Saluja, V. Solanki, K. Kaur, P. Singla, and M. Shahid, "Efficient methods for digital image watermarking and information embedding," *Measurement: Sensors*, vol. 24, Article ID 100520, 2022.

[11] S. Kumar and B. K. Singh, "Entropy based spatial domain image watermarking and its performance analysis," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 9315–9331, 2021.

[12] W. Wan, J. Wu, X. Xie, and G. Shi, "A novel just noticeable difference model via orientation regularity in DCT domain," *IEEE Access*, vol. 5, pp. 22953–22964, 2017.

[13] J. Wu, L. Li, W. Dong, G. Shi, W. Lin, and C. Kuo, "Enhanced just noticeable difference model for images with pattern complexity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2682–2693, 2017.

[14] W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun, "A comprehensive survey on robust image watermarking," *Neurocomputing*, vol. 488, pp. 226–247, 2022.

[15] 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.

[16] F. Ernawan, D. Ariatmanto, and A. Firdaus, "An improved image watermarking by modifying selected DWT-DCT coefficients," *IEEE Access*, vol. 9, pp. 45474–45485, 2021.

[17] N. Zermi, A. Khaldi, R. Kafi, F. Kahlessenane, and S. Euschi, "A DWT-SVD based robust digital watermarking for medical image security," *Forensic Science International*, vol. 320, Article ID 110691, 2021.

[18] S. Li, Z. Chen, Y. Xie et al., "A DWT digital watermarking algorithm based on 2D-LICM hyperchaotic mapping," in *Proceedings of the IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 544–547, Dalian, China, September 2022.

[19] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, and A. A. Khalaf, "Efficient combination of RSA cryptography, lossy, and lossless compression steganography techniques to hide data," *Procedia Computer Science*, vol. 182, pp. 5–12, 2021.

[20] S. A. Nawaz, J. Li, M. U. Shoukat, U. A. Bhatti, and M. A. Raza, "Hybrid medical image zero watermarking via discrete wavelet transform-ResNet101 and discrete cosine transform," *Computers & Electrical Engineering*, vol. 112, Article ID 108985, 2023.

[21] C. Kant and S. Chaudhary, "A watermarking based approach for protection of templates in multimodal biometric system," *Procedia Computer Science*, vol. 167, pp. 932–941, 2020.

[22] M. Tang and F. Zhou, "A robust and secure watermarking algorithm based on DWT and SVD in the fractional order fourier transform domain," *Array*, vol. 15, Article ID 100230, 2022.

[23] A. Singha and M. A. Ullah, "Development of an audio watermarking with decentralization of the watermarks," *Journal of King Saud University- Computer and Information Sciences*, vol. 34, no. 6, pp. 3055–3061, 2022.

[24] U. A. Bhatti, M. Huang, H. Neira-Molina et al., "Mffcg – multi feature fusion for hyperspectral image classification using graph attention network," *Expert Systems with Applications*, vol. 229, Article ID 120496, 2023.

[25] A. Kumar and M. Dua, "Audio encryption using two chaotic map based dynamic diffusion and double DNA encoding," *Applied Acoustics*, vol. 203, Article ID 109196, 2023.

[26] T. Umar, M. Nadeem, and F. Anwer, "A new modified Skew Tent Map and its application in pseudo-random number generator," *Computer Standards & Interfaces*, vol. 89, Article ID 103826, 2024.

[27] D. Wei, M. J. Jiang, and Y. Deng, "A secure image encryption algorithm based on hyper-chaotic and bit-level permutation," *Expert Systems with Applications*, vol. 213, Article ID 119074, 2023.

[28] L. F. Liu and J. Wang, "A cluster of 1D quadratic chaotic map and its applications in image encryption," *Mathematics and Computers in Simulation*, vol. 204, pp. 89–114, 2023.

[29] R. Keshavarzian and A. Aghagolzadeh, "ROI based robust and secure image watermarking using DWT and Arnold map," *AEU- International Journal of Electronics and Communications*, vol. 70, no. 3, pp. 278–288, 2016.

[30] Y. M. Li, D. Y. Wei, and L. N. Zhang, "Double-encrypted watermarking algorithm based on cosine transform and fractional Fourier transform in invariant wavelet domain," *Information Sciences*, vol. 551, pp. 205–227, 2021.

[31] W. H. Alshoura, Z. Zainol, J. S. Teh, and M. Alawida, "An FPP-resistant SVD-based image watermarking scheme based on chaotic control," *Alexandria Engineering Journal*, vol. 61, no. 7, pp. 5713–5734, 2022.

[32] O. Evsutin and K. Dzhanashia, "Watermarking schemes for digital images: robustness overview," *Signal Processing: Image Communication*, vol. 100, Article ID 116523, 2022.

[33] N. Zermi, A. Khaldi, M. R. Kafi, F. Kahlessenane, and S. Euschi, "Robust SVD-based schemes for medical image watermarking," *Microprocessors and Microsystems*, vol. 84, 2021.

[34] H. X. Zhao, S. Xie, J. Zhang, and T. Wu, "A dynamic block image encryption using variable-length secret key and modified Henon map," *Optik*, vol. 230, no. 1, Article ID 166307, 2021.

[35] X. Y. Wang and X. Chen, "An image encryption algorithm based on dynamic row scrambling and Zigzag transformation," *Chaos, Solitons & Fractals*, vol. 147, Article ID 110962, 2021.
[36] Z. Yang, Y. Z. Liu, Y. Q. Wu, Y. Qi, F. Ren, and S. Li, "A high speed pseudo-random bit generator driven by 2D-discrete hyperchaos," *Chaos, Solitons & Fractals*, vol. 167, Article ID 113039, 2023.
[37] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.
[38] K. Sehra, S. Raut, A. Mishra et al., "Robust and secure digital image watermarking technique using arnold transform and memristive chaotic oscillators," *IEEE Access*, vol. 9, pp. 72465–72483, 2021.
[39] K. Fares, A. Khaldi, K. Redouane, and E. Salah, "DCT & DWT based watermarking scheme for medical information security," *Biomedical Signal Processing and Control*, vol. 66, Article ID 102403, 2021.
[40] Z. Li, H. Zhang, X. Liu, C. Wang, and X. Wang, "Blind and safety-enhanced dual watermarking algorithm with chaotic system encryption based on RHFM and DWT-DCT," *Digital Signal Processing*, vol. 115, Article ID 103062, 2021.
[41] Q. Lai and H. Zhang, "A new image encryption method based on memristive hyperchaos," *Optics & Laser Technology*, vol. 166, Article ID 109626, 2023.
[42] A. A. Arab, M. J. B. Rostami, B. Ghavami, and B. Ghavami, "An image encryption algorithm using the combination of chaotic maps," *Optik*, vol. 261, Article ID 169122, 2022.
[43] Y. P. Sang, J. Sang, and M. S. Alam, "Image encryption based on logistic chaotic systems and deep autoencoder," *Pattern Recognition Letters*, vol. 153, pp. 59–66, 2022.
[44] D. C. Liu, Q. T. Su, Z. H. Yuan, and X. T. Zhang, "A blind color digital image watermarking method based on image correction and eigenvalue decomposition," *Signal Processing: Image Communication*, vol. 95, Article ID 116292, 2021.
[45] X. Wang, X. Yuan, M. Li et al., "Parallel multiple watermarking using adaptive Inter-Block correlation," *Expert Systems with Applications*, vol. 213, Article ID 119011, 2023.
[46] V. Sisaudia and V. P. Vishwakarma, "A secure gray-scale image watermarking technique in fractional DCT domain using zig-zag scrambling," *Journal of Information Security and Applications*, vol. 69, Article ID 103296, 2022.
[47] D. Ariatmanto and F. Ernawan, "Adaptive scaling factors based on the impact of selected DCT coefficients for image watermarking," *Journal of King Saud University- Computer and Information Sciences*, vol. 34, no. 3, pp. 605–614, 2022.
[48] X. B. Kang, Y. J. Chen, F. Zhao, and G. F. Lin, "Multi-dimensional particle swarm optimization for robust blind image watermarking using intertwining logistic map and hybrid domain," *Soft Computing*, vol. 24, no. 14, pp. 10561–10584, 2020.
[49] X. Y. Wang, J. Tian, J. L. Tian, P. P. Niu, and H. Y. Yang, "Statistical image watermarking using local RHFMs magnitudes and beta exponential distribution," *Journal of Visual Communication and Image Representation*, vol. 77, Article ID 103123, 2021.
[50] X. Y. Wang, F. C. Peng, P. P. Niu, and H. Y. Yang, "Statistical image watermark decoder using NSM-HMT in NSCT-FGPCET magnitude domain," *Journal of Information Security and Applications*, vol. 69, Article ID 103312, 2022.