Hindawi
Security and Communication Networks
Volume 2021, Article ID 6617944, 17 pages
https://doi.org/10.1155/2021/6617944

Research Article

Robust Secure Color Image Watermarking Using 4D Hyperchaotic System, DWT, HbD, and SVD Based on Improved FOA Algorithm

Hira Nazir, Imran Sarwar Bajwa, Muhammad Samiullah, Waheed Anwar, and Muhammad Moosa

Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

Correspondence should be addressed to Imran Sarwar Bajwa; imran.sarwar@iub.edu.pk

Received 25 December 2020; Revised 1 February 2021; Accepted 12 February 2021; Published 28 February 2021

Academic Editor: Manjit Kaur

In the recent past, a different set of algorithms for watermarking and securing the color images have been developed by using transformation, decomposition, and optimization techniques for watermark embedding and extraction. In this paper, we propose an optimized and robust watermarking algorithm coupled with a 4D hyperchaotic system, and its performance is analyzed by extending and differentiating the existing work. Our contribution in the presented work is watermarking and securing the color images by an optimized algorithm that uses transformation technique such as Discrete Wavelet Transformation (DWT) and decomposition techniques such as Hessenberg decomposition (HbD) and singular value decomposition (SVD) coupled with the 4D hyperchaotic system, while the optimization is carried out by improved evolution fruit fly optimization algorithm (IEFOA). The experimental results based on different types of attacks (filter attacks, noise attacks, cropping attack, JPEG compression, motion blur, sharpening, and rotation), key sensitivity, normalized correlation, peak signal-to-noise ratio, and structural similarity index measure are done for measuring the algorithm's performance regarding invisibility and robustness. The experimental results show that the proposed scheme has excellent invisibility and keeps a good trade-off between invisibility and robustness. The experiment results show that the proposed approach outperforms the previous approaches.

1. Introduction

In the age of cloud computing and for computing, the security of data whether the sensor's data or cloud's data has become the dire need of today's age to secure it from malicious attacks. Similarly, copyright infringement problems and illegal distribution and modification while disseminating information over the Internet may arise quite frequently [1, 2]. Therefore, watermarking coupled with hyperchaotic encryption can cope up with the emerging challenges of copyright infringement, watermarking attacks, and security issues. In watermarking, a logo or secret message is hidden in the host image on the transmitting side while this logo or secret message is extracted at receiving side in order to judge the digital ownership of the received data. With the advancement of computing such as DNA and quantum-based computing, the probability to breach currently highly secured watermarks may also increase. The techniques such as HbD, DWT, and SVD have been widely used by researchers in various watermarking methods to watermark the grayscale and color images. The trade-off between invisibility and robustness has always been a challenging issue in watermarking methods and it needs optimization.

Recently, several algorithms such as the firefly algorithm [3], artificial bee colony (ABC) [4], and particle swarm and fruit fly optimization algorithms [5, 6] are employed to optimize the watermarking technique. The problem of entrusting the watermarking to cloud service provider is addressed in [7], in which the authors made the following contributions: (1) modern public-key cryptosystems are employed to avoid the associated security hazards and implementation costs of key exchange are also considered, (2) reversible watermarking techniques compatible with homomorphic cryptosystems are studied, (3) storage efficiency is studied by encrypting a long sequence of bits, (4) data preprocessing prior to encryption is not required, and (5) both offline and online content-adaptive predictors are developed for various operational requirements. The proposed schemes achieve a remarkable balance between fidelity and reversibility under the given capacity constraints. Moreover, it significantly reduces the size of the encrypted data and improves the space efficiency. Most of the existing watermarking techniques suffer from certain watermarking attacks, are not optimized, and are not coupled with hyperchaotic maps. A few studies have been published on watermarking followed by hyperchaotic encryption [8]. To this end, a novel watermarking technique by exploiting the interblock coefficient correlation for embedding the watermark is proposed by [9], in which chaos and Arnold transform is used for improving security. The modifications are done in such a way that image processing and geometric attacks are resisted. Furthermore, it is testified that watermarking based on DWT has certain advantages such as good compression and imperceptibility; however, DWT-based watermarking schemes are not too much robust against geometric attacks [10]. Therefore, in order to make the scheme more robust against image processing and geometric attacks, matrix decomposition such as SVD and HbD is commonly used. The SVD-based schemes decompose the transformed host image into three vectors called U, S, and V. The digital watermark can be embedded into U or S or V. The S matrix is mostly used for watermark embedding owing to its robust nature against attacks [11]. Additionally, a little change in singular values does not influence the visual quality of the host image. On another note, FPP arises when singular values are used for watermark insertion. The matrices U and V can be replaced by the attacker's desired matrices for the extraction of a new watermark (that has never been inserted) to profess the false ownership. Computer science researchers have proposed the change in singular values with the help of scaling factor to control the strength of digital watermark to be embedded as shown in Sections 4.2 and 4.3 (Eq. (13) and Algorithm 3). The scaling factor can be further optimized by using different algorithms such as particle swarm and improved fruit fly optimization algorithms and bioinspired computing algorithms [5, 6, 12]. The FPP can be solved by encrypting the SVD components by using hyperchaotic systems or by using the one-way hash functions [13, 14]. Hyperchaotic encryption owing to excellent security results is the main source of strong security; i.e., the FPP can be solved. For example, the author in [15] verified the better confusion and diffusion by using the 5D hyperchaotic map to create secret keys for encryption and decryption. The initial parameters for 5D hyperchaotic are tuned by using the dual local search based multiobjective optimization, and the encryption architecture is based on two levels of permutation and diffusion. Similarly, the authors in [16–18] also used the hyperchaotic maps in a novel way for encrypting the images and obtained better results.

Specifically, in this paper, a novel digital watermarking method consisting of DWT, HbD, and SVD based on hyperchaotic encryption, gauging function (GF), and improved evolution fruit fly algorithm (IEFOA) is proposed. Specifically, GF abets IEFOA to find the optimal scaling factor α, for balancing the trade-off between imperceptibility and robustness, while hyperchaotic encryption of watermark before the use of SVD and chaotic encryption of SVD components solves the FPP effectively at a less computational cost. The main contributions of this paper include the following: (1) scheme has shown a good balance of trade-off even with the multiple size watermarks, (2) robustness is improved by coefficient modification through HbD, (3) encryption of color watermark by the 4D hyperchaotic system before SVD procedure and chaotic encryption of SVD components is also applied to make the scheme more secure, and (4) GF and IEFOA are employed to help in finding the optimal scaling factor.

The proposed work is organized as follows. Section 2 gives the related work, Section 3 highlights the preliminaries, Section 4 presents the proposed scheme, and Section 5 contains experimental results and analysis. Concluding remarks with future directions are given in Section 6.

2. Related Work

This section deals with the earlier research work done in designing color watermark embedding and extracting schemes. The list of abbreviations used in this study is shown in Table 1. Imperceptible and robust digital watermarking schemes can be a potential solution for the privacy and security of sensitive information such as Electronic Patient Records (EPRs). To this end, a combination of fast curvelet transform and SVD embeds watermark (EPR) after encoding into patient's healthy and diseased optical coherence tomography (OCT) scans [19]; this scheme has shown a high level of imperceptibility, robustness, and security of EPRs as compared to existing watermarking schemes. A digital watermark protocol proposed by [20] solves the false-positive problem by using a chaotic Kbest gravitational search algorithm in two domains, i.e., SVD and DCT. An efficient watermarking scheme in terms of imperceptibility, security, and robustness proposed by [21] embeds the watermark by Fractional Moments of Charlier–Meixner. The proposed method by [22] achieves robustness against geometric and filtering attacks and shows a better trade-off among robustness and distortion than the state-of-the-art methods. The proposed watermarking scheme in [10] uses a double encryption method based on fractional Fourier transform and DCT in the hybrid wavelet domain. The author in this scheme used multiparameter particle swarm optimization (MP-PSO) for obtaining the optimized embedding factors and reveals high security and invisibility and is robust against geometrical attacks. A robust and secure watermarking scheme to improve the management of medical images is presented in [23]. In this scheme, the techniques of invisible and zero watermarking avoid the detachment between medical images and EPRs and provide authenticity for the identification of patients. Another digital watermarking scheme comprises six modules (level shifting, mixed modulation, sign correlation, orthonormal restoration, distortion compensation, and iterative regulation) that overwhelm the inadequacies of existing SVD-based watermarking schemes while improving

Table 1: List of abbreviations.

Abbreviation	Full form
DCT	Discrete cosine transform
DFT	Discrete fractional angular transform
SVD	Singular value decomposition
LWT	Lifting wavelet transform
FOA	Fruit fly optimization algorithm
DWT	Discrete wavelet transform
HbD	Hessenberg decomposition
IEFOA	Improved evolution fruit fly optimization algorithm
FrMT	Fractional Mellin transform
WL	Wang–Landau
DE	Differential evolution
DNA	Deoxyribonucleic acid

robustness and imperceptibility [24]. In order to provide the copyright protection and ownership of digital data, the authors in [25] present an adaptive and robust watermarking scheme in which the color host and watermark images of the same size are scrambled through Arnold's chaotic map. Then, the approximate subband generated from redundant-DWT goes through SVD to produce the principal component. The principal component of scrambled host image is then embedded with scrambled watermark by using optimized Artificial Bee Colony (ABC) adaptive multiscaling factor. The use of redundant-DWT gives higher embedding capacity while adaptive multiscaling factor improves robustness, security, and visual transparency. Another scheme based on wavelet transformation followed by best-fit equation and Cuckoo Search (CS) algorithm is robust to common attacks, and the watermark is imperceptible to human eyes [26]. On the other hand, the fusion of multiple watermarking techniques such as DCT, DFT, SVD, and LWT improved the security, robustness, imperceptibility, and false-positive problem to a great extent but the authors did not perform scaling factor optimization [27]. SVD and three-level wavelet transform with global optimization scheme based on WL method in [28] keep a better trade-off between robustness and imperceptibility and obtained a better embedding coefficient. A color watermarking scheme presented in [29] converts RGB to YIQ space, separates the luminance component Y, and uses SVD, Arnold Transform, and DWT with DE algorithm for embedding, extraction, and optimization of scalar factors. The reason to choose luminance component Y is that the human eye is not sensitive to this component; thus, embedding watermark information into this component will give strength to invisibility. Watermark encryption and then embedding it in the host image proposed by [8] make use of FrMT, DPMs, and SVD, provide enhanced security due to the nonlinear transformation, and keep a balance between invisibility and robustness to some extent. Combining IWT, DWT, contourlet transform, and 3D Henon Map in embedding and extracting watermark has good imperceptibility and acceptable robustness [30]. The authors in this scheme suggested that the chaotic sequence produced by Henon Map can be used as a pseudorandom number generator after testing it on NIST, DIEHARD, and ENT test suites. To perform the watermarking, the authors in [31] divided the algorithm into four phases called image scaling, block separation by DCT, feature vector computation, watermark spotting regions, message transformation, watermark embedding, IDCT, and message restoration followed by an optimized FCM clustering with Least Favorable Whale Optimization Algorithm based watermarking scheme and obtained the effective results in terms of robustness and invisibility. A substitution scheme for RGB images watermarking based on Fourier transform is proposed in [32]. In this approach, several variants of Fourier transforms are applied to R, G, and B components of an image separately, the watermark is embedded in medium frequency band based on the combined parity of coefficients, and the obtained results are satisfactory in terms of average PSNR greater than 40 decibels for integration into a variant of Fourier transform coefficients. Another blind image watermarking scheme in the transform domain, where there is no need for a watermark and host image for extracting the watermark, gives good imperceptibility and robustness with less computational cost [33]. In this scheme, the host image is split into nonoverlapping blocks each of size 8×8, and DCT coefficients of each block are computed; then, two datasets (d1 and d2) are created from the selected blocks, and DCT coefficients of d1 and d2 are compared with the prefixed threshold values (k1 and k2) as follows: if the watermark bit value is 1, then corresponding d1 and d2 coefficient values are modified with set α value; else, the corresponding d1 and d2 coefficient values are set to zero.

3. Preliminaries

Hessenberg decomposition (HbD) is a transformation of the square matrix A into the unitary matrix Q and Hessenberg matrix H such that $A = QHQ^T$, computed by household matrices, and aids in improving the watermark invisibility [34]. To this end, watermarking based on R level DWT, HbD, SVD, logistic map, and optimization based on FOA through objective evaluation function showed a good trade-off between robustness and invisibility [13]. This scheme can further be improved by using improved FOAs.

Although basic FOA [6] has advantages including fewer parameters and simple principles but has shortcomings such as local optimization, lack of robustness, and slow convergence that can be overcome by IEFOA [35]. The inclusion of two parameters called step control denoted by λ and evolution/elimination control (ec) in IEFOA makes it different and provides an advantage over basic FOA. In basic FOA, the number of iterations in which the algorithm needs to find an optimal solution is the main drawback. In the early stage of iterations with the vast domain, a small search radius (search step) makes basic FOA weak to approach the optimal solution. In the final stage of iterations when the swarm location is close to an optimal solution, a very small scope is a better option for fine-tuning solution vectors. Therefore, a search radius with the big to small (BS) feature may overcome this drawback. The (BS) feature means that a big search step in the early stage can refine the global search ability and a small search step in end stage can refine the local search ability by determining the scale of step for each fruit fly flexibly. Step control parameter (λ) provides the (BS) feature and can be expressed as

$$\lambda = \lambda_{\max} \times \exp\left[\log\left(\frac{\lambda_{\min}}{\lambda_{\max}}\right)\frac{\text{Iter}}{\text{Iter}_{\max}}\right], \tag{1}$$

where λ is the search radius in each iteration, while $\lambda_{\min}$, $\lambda_{\max}$, and Iter are the minimum radius, maximum radius, and iteration number, respectively. The fruit fly gets a bigger search step and hence eludes falling in local optimum value, while in the later iterations, λ decreases slower than linear decreasing.

The second parameter is called elimination parameter ev, in which the inferior swarm is eliminated and the dominant swarm is saved. The *ec* can be expressed as

$$\text{ec} = 1 - \text{elc}, \tag{2}$$

where elc is the elimination coefficient and can be defined as

$$\text{elc} = \text{elc}_{\max} \times \exp\left[\log\left(\frac{\text{elc}_{\min}}{\text{elc}_{\max}}\right)\frac{\text{Iter}}{\text{Iter}_{\max}}\right], \tag{3}$$

where $\text{elc}_{\min}$, $\text{elc}_{\max}$, Iter, and $\text{Iter}_{\max}$ are the minimum elimination coefficient, maximum elimination coefficient, iteration number, and maximum iteration number. Many bad performance swarms are removed as the search starts and the remaining advanced fly swarms will produce a new population. The repetitive process of swarm elimination will lead to the preservation of only a few swarms. The elimination procedure offers the advantage of letting IEFOA jump out of the local extremum (an extreme point having maximum or minimum value) to find a better global optimum. The beauty of IEFOA is the fact that it not only adopts λ but also segregates the inferior swarms by using ec.

The main process of IEFOA can be illustrated as follows:

Step 1. Randomly generate multiple swarms' center locations.

Step 2. Generate N new swarms; PSF in each swarm represents the population size according to the update rule of the Osphresis foraging stage.

Step 3. The optimal fruit fly is selected in each swarm as a new center location by vision foraging phase according to the fitness function value (fval).

Step 4. Center locations of all the new swarms are sorted in ascending order according to their fval.

Step 5. A certain number of inferior swarms are eliminated; the remaining dominant swarms become the next iteration swarm center locations according to the coefficient of elc and the number of swarm locations at present.

Step 6. Repeat Steps 2 to 5 till the satisfaction of termination condition. The global optimum is only obtained when the optimized process is terminated.

4. Proposed Scheme

The watermark encryption algorithm is introduced in Section 4.1 and the embedding algorithm is introduced in Section 4.2, while the extraction and decryption algorithm is introduced in Sections 4.3 and 4.4. Optimization of the proposed watermarking method to achieve the trade-off between invisibility and robustness is given in Section 4.5. The flowchart of the proposed scheme is given in Figure 1.

4.1. Watermark Encryption. A color watermark of multiple sizes ($N \times N$), where $N = 2$, 4, 8, 16, 32, 128, 256, 512 is input to the watermark encryption algorithm. Initial conditions based on the DNA sequence taken from the NCBI dataset are calculated. External key xK is extracted from the DNA sequence taken from the NCBI dataset. For example, we downloaded a DNA sequence of some animals having a length of 183015. The mean intensity value of the watermark image is used as a starting index to cut the DNA sequence from this location having a length of 128. After cutting the DNA sequence of length 128, each nucleotide base is converted into a two-bit binary equivalent according to the DNA mapping rules [36], shown in Table 2, which meet the Watson–Crick complement rule. In this way, a 256-bit binary key binK is obtained. In order to create the initial conditions $x(0), y(0), z(0), u(0)$ for the 4D hyperchaotic system, we divide binK into 32 subgroups where each subgroup g is comprised of 8 bits and is expressed as follows:

$$\text{binK} = \{g1, g2, \ldots, g32\}. \tag{4}$$

Now, the initial conditions using binK are computed as follows:

$$\begin{cases} x(0) = \dfrac{(g1 \oplus g2 \oplus g3 \oplus g4 \oplus g5 \oplus g6 \oplus g7 \oplus g8)}{256} \\[2ex] y(0) = \dfrac{(g9 \oplus g10 \oplus g11 \oplus g12 \oplus g13 \oplus g14 \oplus g15 \oplus g16)}{256} \\[2ex] z(0) = \dfrac{(g17 \oplus g18 \oplus g19 \oplus g20 \oplus g21 \oplus g22 \oplus g23 \oplus g24)}{256} \\[2ex] u(0) = 256 \dfrac{}{(g25 \oplus g26 \oplus g27 \oplus g28 \oplus g29 \oplus g30 \oplus g31 \oplus g32)} \end{cases}. \tag{5}$$

Initial conditions with control parameters (a, b, c, d, e) are input to the 4D hyperchaotic system (Equation (1)). The 4D hyperchaotic at any given initial conditions with control parameters ($a = 27.5$, $b = 3$, $c = 19.3$, $d = 2.9$, $e = 3$)

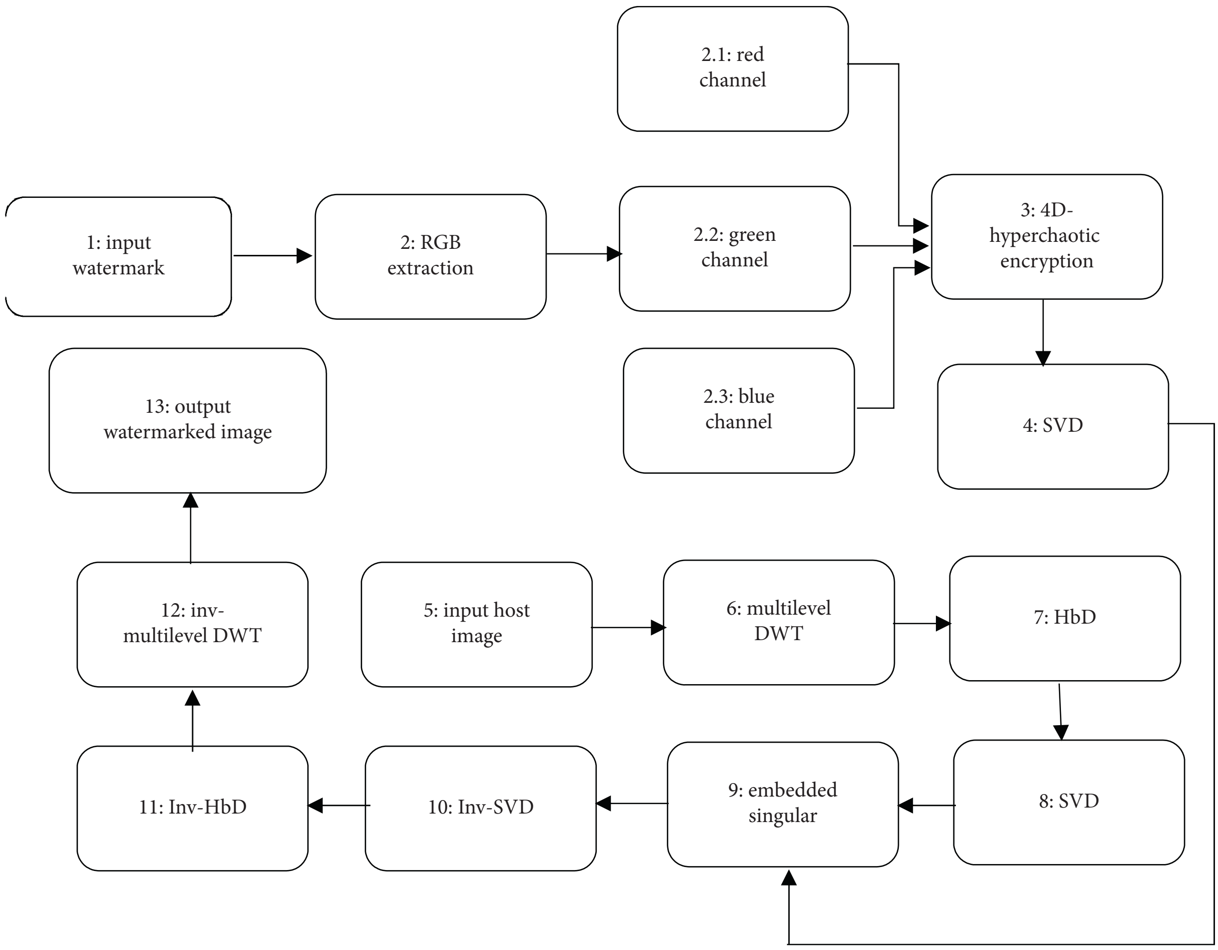


Figure 1: Watermark embedding procedure.

Table 2: DNA mapping rules.

	R1	R2	R3	R4	R5	R6	R7	R8
00	A	A	T	T	C	C	G	G
01	C	G	C	G	A	T	A	T
10	G	C	G	C	T	A	T	A
11	T	T	A	A	G	G	C	C

behaves hyperchaotic and generates a hyperchaotic key called hyp-K which is used to encrypt the watermark.

$$\begin{cases} \dot{x}_1 = a\left(x_2 - \dot{x}_1\right) \\ \dot{x}_2 = bx_1 + cx_2 - x_1 x_3 + x_4 \\ \dot{x}_3 = x_2^2 - dx_3 \\ \dot{x}_4 = -ex_1 \end{cases} \tag{6}$$

Encryption steps based on hyp-K to encrypt the watermark image are as follows (Algorithm 1).

In Algorithm 1, C_o is a constant number ranging from 0 to 255 and mEW is the mean intensity value of EW produced in Step 3.

4.2. Watermark Embedding. The inputs to the watermarking embedding algorithm are the EW of size $(N \times N)$ and the host image HI of size $(M \times N)$. And the output is watermarked host image WHI of size $(M \times N)$. The embedding steps are as follows (Algorithm 2).

4.3. Watermark Extraction. Watermark extraction takes WHI as input and the output is XW, similar to the original color watermark. The size of WHI is $M \times N$ and the size of XW is $N \times N$. The extraction steps are as follows (Algorithm 3).

4.4. Watermark Decryption. Watermark decryption is shown in Algorithm 4.

4.5. Algorithm Optimization Using IEFOA. In this section, an improved evolution fruit fly optimization algorithm (IEFOA) discussed in Section 3 is used to find the optimal scaling factor to solve the trade-off problem between invisibility and robustness. The flowchart to find the optimal scaling factor is shown in Figure 2. Invisibility is measured by PSNR and SSIM while robustness is measured by Normalized Correlation (NC). The steps to find the optimal scaling factor are given as follows.

Step 1. Initialize the parameters $S1 = \beta, \omega_i$ and $S2 = \mathrm{NS}, \mathrm{PS}, \lambda_{\max}, \lambda_{\min}, \mathrm{Iter}_{\max}, \mathrm{elc}_{\max}, \mathrm{elc}_{\min}$. The parameters in S1 such as β are the weight factor and ω_i ($i = 1, 2, 3$) are the quantization coefficients that directly reflect the proportion of invisibility or robustness. The parameters in S2 such as $\mathrm{NS}, \mathrm{PS}, \lambda_{\max}, \lambda_{\min}, \mathrm{Iter}_{\max}, \mathrm{elc}_{\max}, \mathrm{elc}_{\min}$ represent the number of swarms, the population size of the fruit fly, maximum search radius, minimum search radius, maximum iteration number, maximum elimination coefficient, and minimum elimination coefficient, respectively. The set S1 with different scaling factors will be used in the gauging function (GF) which is based on the objective evaluation function (OEF) [13] and is given by

$$\mathrm{GF}(\beta, \omega_i) = \omega_1 \frac{1}{\beta} \mathrm{PSNR}(\mathrm{HI}, \mathrm{WHI}) + \omega_2 \mathrm{SSIM}(\mathrm{HI}, \mathrm{WHI}) + \omega_3 \frac{\left(\sum_{i=1}^{K} \mathrm{NC}(W, \mathrm{DW}_i)\right)}{K}, \tag{7}$$

where DW_i is the *decrypted* watermark, i.e., decrypted from extracted watermark EW_i under i_{th} attack.

The scaling factor *array* is denoted by α_i ($i = 1, 2, \ldots, n$), where n is the max number index. The scaling factors are used in computing PSNR, SSIM, and NC. For example, the scaling factor array α_i is used to embed the watermark to produce the watermarked image, and i_{th} attack is applied on the watermarked image to produce the attacked watermarked image. After that, the PSNR and SSIM between the cover and attacked watermarked images is calculated. Similarly, NC between original and decrypted watermarks is computed. S2 will be used in IEFOA mentioned in the related work section.

Step 2. The GF values of each location for smell judgment are calculated according to Equation (7).

Step 3. In order to get the optimal scaling factor, apply IEFOA discussed in Section 3. The only modification that will be in the IEFOA is to use GF in Step 3 of IEFOA, and repeat Steps 2 to 5 of IEFOA for updating the fruit fly population location when the iterative smell concentration is superior to the previous smell concentration.

5. Experimental Results and Analysis

The invisibility and robustness of the proposed scheme are analyzed in this section. The optimal scaling factor is computed in Section 5.1, invisibility and robustness analysis is carried out in Section 5.2, false-positive problem is done in Section 5.3, and comparison with related works whenever the data is available is done in Section 5.4. Intel(R) core i3 4010 CPU@1.7 GHz with 4.0 GB RAM and MATLAB version R2015a installed on Windows 7, a 64-bit operating system, is used for experimental purposes. Except for the other images, the standard color host images Lena and Pepper each of size 512×512 and color watermark images with sizes of 256×256, 128×128, and 64×64 shown in Figure 3 are used in the experiments. The initial population size of 50 and the maximum number of iterations of 200 are empirically selected in the experiments. Aside from the above parameters, the other parameters are set according to the improved fruit fly optimization algorithm (IFFO) [35, 37]; i.e., $\lambda_{\max} = (UB - LB)/2$, $\lambda_{\min} = 10^{-14}$ $\mathrm{elc}_{\max} = 0.1$, and $\mathrm{elc}_{\min} = 0.05$.

5.1. Finding Optimal Scaling Factor. Optimal state performance is characterized by an optimal scaling factor. According to Section 4.5, an optimal n is decided and is input to gauging function (Equation (7)) to find the optimal scaling factor. The Normalized Correlation (NC) is normally used to evaluate the robustness of the watermarking algorithm and is defined by [13]

$$\mathrm{NC} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} W_{i,j} \mathrm{DW}_{i,j}}{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} W_{i,j}^2} \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{DW}_{i,j}^2}}. \tag{8}$$

The NCs between original watermark (W) and extracted-decrypted DW watermark under various attacks and scaling factors are shown in Figure 4. The attacks used in the simulations are shown in Table 3. NC values vary in the range of $[0:0.06]$ and get stabilized to large extent in the range of $[0.09:0.2]$; therefore, the starting value can be set as $n_1 = 0.09$. Similarly, the curves for PSNR and SSIM are also shown in Figures 5 and 6. Similarly, the starting value for PSNR can be set as $n_2 = [0{:}\ 0.02]$ as values of PSNR have negative correlations with α_i within the range of $[0.009{:}\ 0.2]$, and for SSIM, it can be set as $n_3 = [0{:}\ 0.2]$ as SSIM values are almost constant within this range. And n can be calculated as $n = (n_{\max} - n_s)/M_i$, where $n_{\max} = 0.2$, n_s is a set containing all elements of n_1 that also belong to n_2 and n_3, and M_i is the minimum interval. The value of nn is then used in GF for obtaining the optimal scaling factor. Table 4 shows the better NCs under certain attacks at the scaling factor $\alpha = 0.115$.

5.2. Invisibility and Robustness Analysis. For invisibility performance, we used color images of lena and peppers as host images and colorful logos of the Islamia University of Bahawalpur, Pakistan, as watermarks with different dimensions. Except for visual representation, we also used three metrics, PSNR, SSIM, and NC, to quantify the invisibility. The invisibility performance of the proposed algorithm under no attacks, shown in Figure 7, reflects excellent invisibility. Robustness needs to be assessed when the invisibility is acceptable. In robustness, the quality of extracted watermarks is checked under certain attacks. Several cases of attacks on lena color image (512×512) embedded with watermark (128×128) are shown in Figure 8. Watermarks are extracted from attacked images by the extraction algorithm and are decrypted by the decryption algorithm. The corresponding NC values of extracted-decrypted watermarks are shown in Figure 9. The NC values

Input: color watermark image (W), initial conditions, control parameters.
Output: encrypted watermark image EW.
Step 1. Solve the 4D hyperchaotic system by using initial conditions and control parameters to produce **hyp-K**.
Step 2. $\text{key}(i) = \text{mod}(C_o + \text{hyp} - \text{K}(i), 256)$.
Step 3. $\text{EW}(i) = \text{XOR}(W(i), \text{Key}(i))$.
Step 4. $\text{key}(i) = \text{mod}(\text{mEW} + \text{hyp} - \text{K}(i), 256)$.
Step 5. $\text{EW}(i) = \text{XOR}(\text{EW}(i), \text{Key}(i))$.

ALGORITHM 1: Watermark encryption.

Input: **EW**, color host image (**HI**).
Output: **WHI**.
Step 1. Obtain a low-frequency subband SB1_{HI} of RGB components of **HI** using **HW**.
$\text{SB1}_{\text{HI}} = \text{DWT}(\text{HI})$.
Step 2. Perform Hessenberg decomposition (HbD) on RGB components.
$\text{HQ} = (\text{Id}^n - 2\mu\mu^T)/\mu\mu^T$.
Here, HQ, Id^n, μ, μ^T are household orthogonal matrix, identity matrix, nonzero vector, and the transpose of μ, respectively. For example, HbD on SB1_{HI} is given as:
$\text{P} = (\text{HQ1}, \text{HQ2}, \text{HQ3}, \ldots, \text{HQ}_{n-2})^T \text{SB1}_{HI} (\text{HQ1}, \text{HQ2}, \text{HQ3}, \ldots, \text{HQ}_{n-2})$,
$\Rightarrow H = (P^T)\text{SB1}_{\text{HI}}(P)$,
$\Rightarrow \text{SB1}_{\text{HI}} = \text{PHP}^T$.
Step 3. Perform SVD on H and EW as shown in the following equations. Only the singular value S from $\text{P} = (\text{HQ1}, \text{HQ2}, \text{HQ3}, \ldots, \text{HQ}_{n-2})^T \text{SB1}_{HI} (\text{HQ1}, \text{HQ2}, \text{HQ3}, \ldots, \text{HQ}_{n-2})$ is used here. The other components such as U and V^T are used as a source of information in the extraction process. Similarly, SVD is also applied to the RGB components of EW. Note that components are also encrypted by a logistic map in order to avoid the false-positive problem.
$S = \text{SVD}(H)$,
$U_{\text{ew}}, S_{\text{ew}}, V_{\text{ew}}^T = \text{SVD}(\text{EW})$.
Step 4. Calculate the modified singular values by using the scaling factor α as follows:
$S^* = \alpha S_{\text{ew}}$,
$S^{**} = S + S^*$.
Step 5. Perform an inverse SVD to get H^*.
$H^* = \text{inverseSVD}(U_{\text{ew}}, S^{**}, V_{\text{ew}}^T)$.
Step 6. Perform an inverse HD to get SB1_{HI}^*.
$\text{SB1}_{\text{HI}}^* = \text{inverseHD}(P, H^*, P^T)$.
Step 7. Perform inverse DWT to get watermarked host image **WHI**.

ALGORITHM 2: Watermark embedding.

Input: **WHI**.
Output: extracted watermark **XW**.
Step 1. WHI is decomposed into 4 subbands: $\text{SB1}_{\text{WHI}}, \text{SB2}_{\text{WHI}}, \text{SB3}_{\text{WHI}}, \text{SB4}_{\text{WHI}}$ by using DWT.
Step 2. Perform HbD on SB1_{WHI} and get $P_{\text{WHI}}, H_{\text{WHI}}, P_{\text{WHI}}^T$.
Step 3. Apply SVD on H_{WHI} and obtain $U_{\text{WHI}}, S_{\text{WHI}}, V_{\text{WHI}}^T$.
Step 4. The extracted singular value S^{***} is obtained as follows:
$S^{***} = (S_{WHI} - S^{**})/\alpha$.
Here, $S^{***}$is taken from $S^{**} = S + S^*$.
Step 5. Apply inverse SVD on $U_{\text{ew}}, S^{***}, V_{\text{ew}}^T$ and get **XW**.

ALGORITHM 3: Watermark extraction.

(Figure 9) are acceptable for the median, Gaussian noise, salt and pepper, speckle noise, and JPEG compression. Moreover, NC values of extracted-decrypted watermarks under different parameters suffering from numerous attacks are also shown in Figure 10.

5.3. False-Positive Problem Analysis. Digital watermark ownership protection and authentication is a vital application of watermarking schemes; i.e., only the actual owner should be able to extract the embedded digital watermark from the images correctly. FPP problems are very common

Input: **XW**.
Output: decrypted watermark **DW**.
Steps: inverse steps of Algorithm 1 are carried out in the reverse order.

ALGORITHM 4: Watermark decryption.

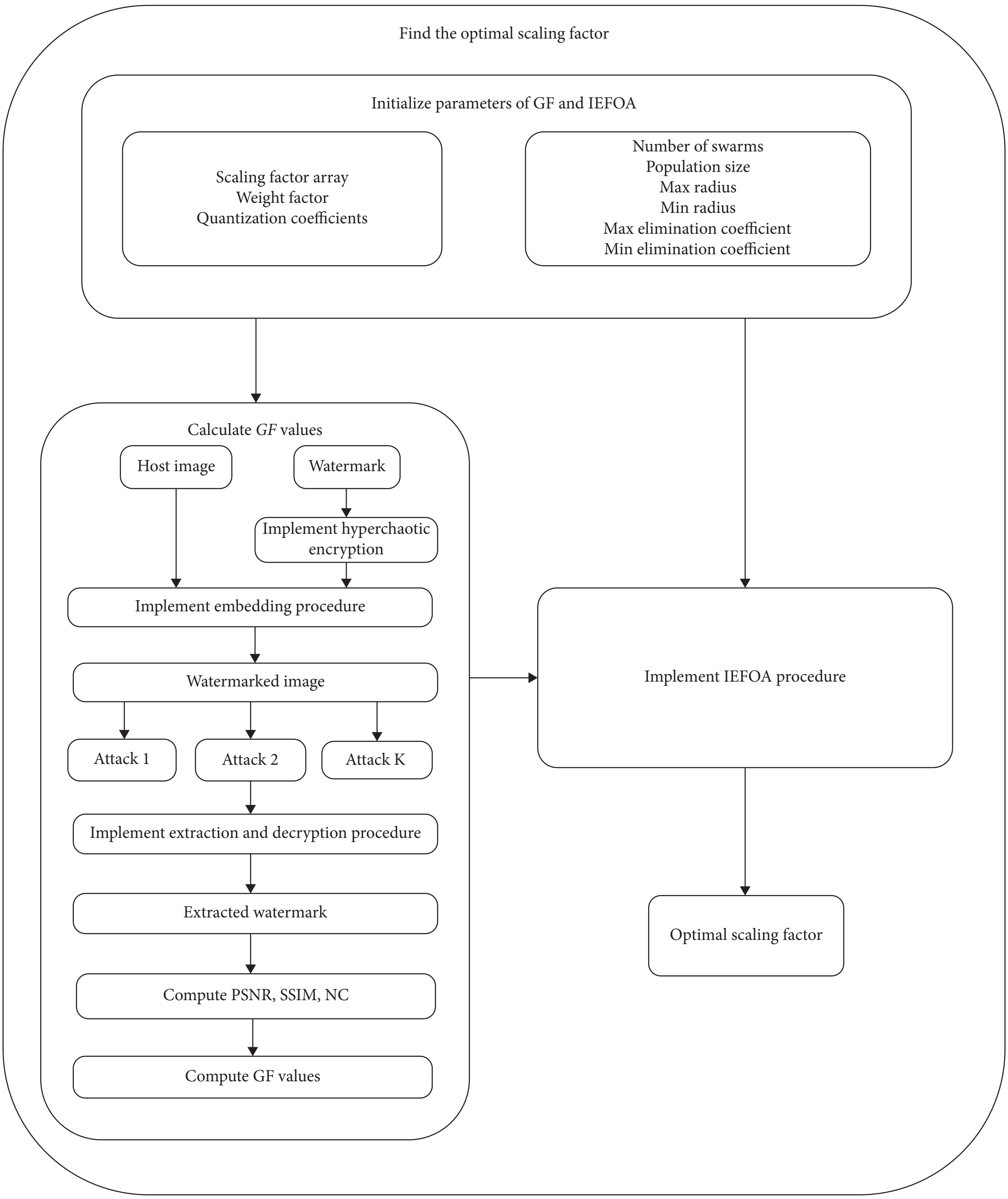


FIGURE 2: Scaling factor optimization.

Figure 3: (a-b) Host images of size 512 × 512. (c–e) Watermarks of size 256 × 256, 128 × 128, and 64 × 64, respectively.

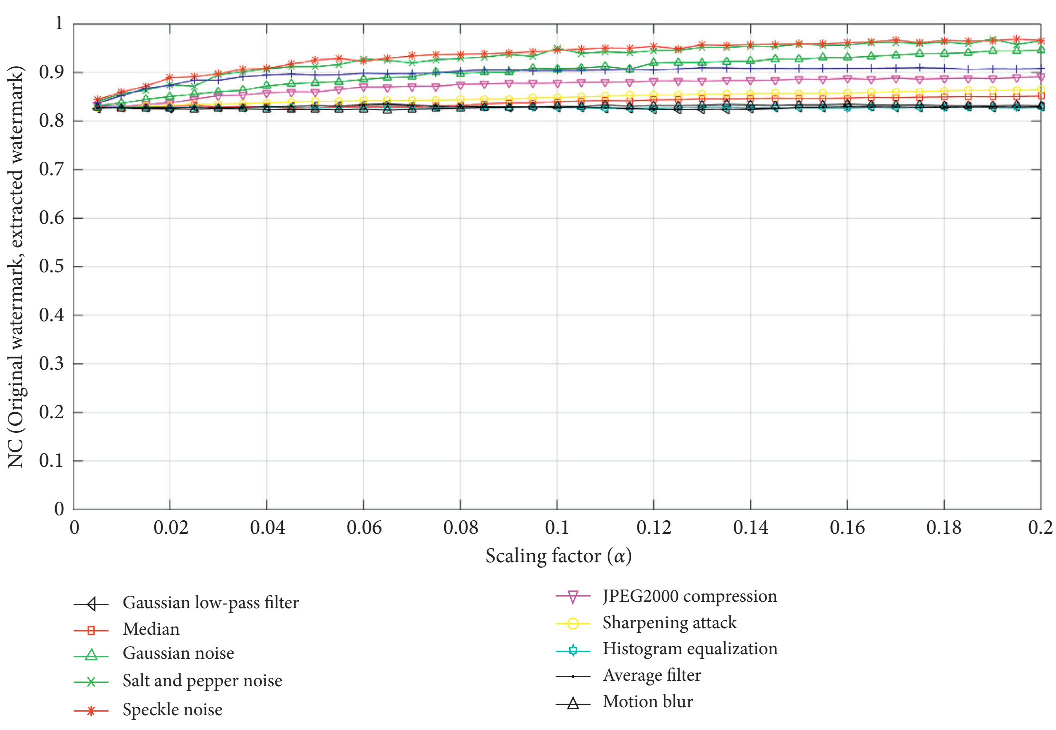


Figure 4: NC values under various scaling factors and attacks.

Table 3: Attacks used for experimental purpose.

Attack	Specification
Filter attack	Wiener filter (3 × 3) Median filter (3 × 3) GLP filter (3 × 3) Average filter (3 × 3)
Noise attack	Salt and pepper noise (0.001) Speckle noise (0.001) Gaussian noise (0.001)
Cropping attack	Percentage 2%
JPEG compression	QF = 50
Motion blur	Theta = 4, len = 7
Sharpening	0.8
Rotation	2 degrees

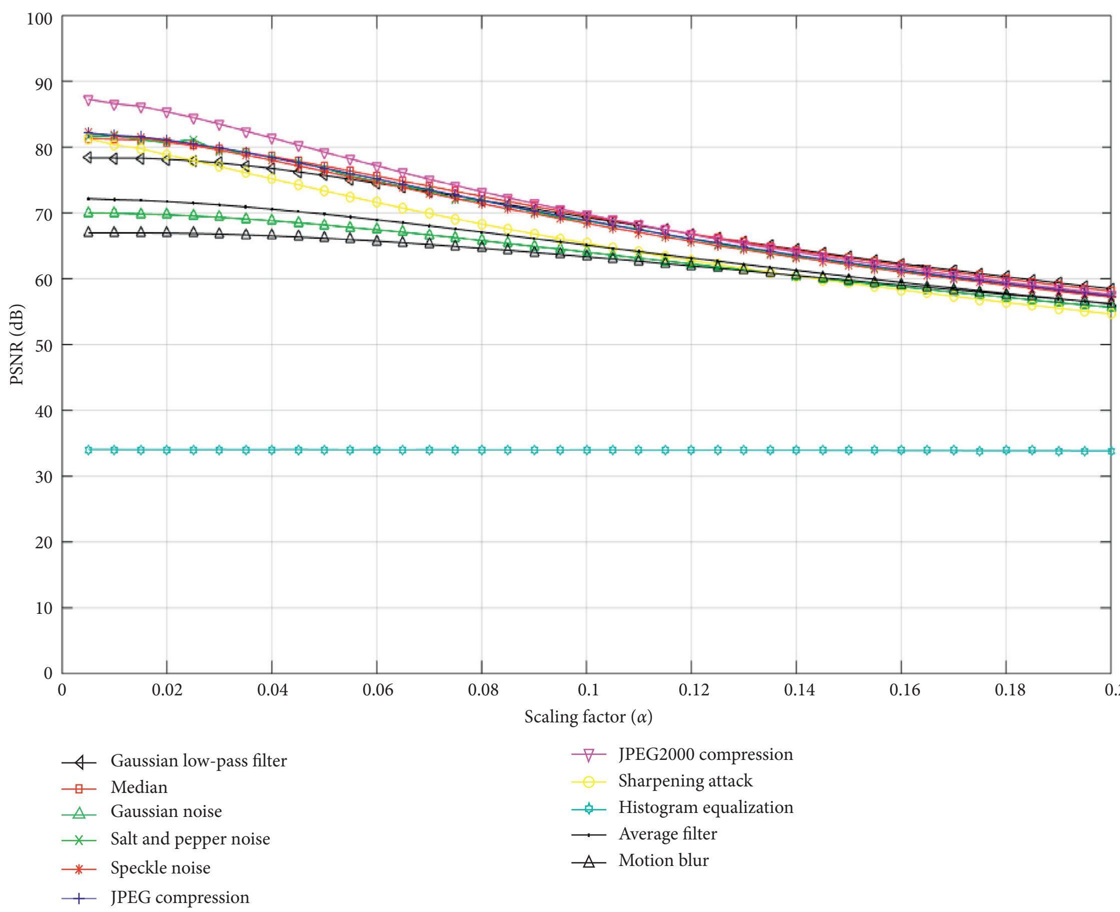


Figure 5: PSNR values under various scaling factors and attacks.

and become a challenging issue in digital watermarking schemes, where an attacker claims false ownership of the watermark by embedding and extracting the forged watermarks. This state is a serious security matter that creates a barrier in confirming the real ownership of digital media [25]. There are two approaches to embed the watermark in the SVD domain: (i) computing the singular values of watermark and cover images and then embedding the singular values of the watermark into the singular values of the cover image or (ii) by directly embedding the watermark bits into the singular values of the cover image. Generally, SVD-based watermarking schemes satisfy the criteria of invisibility and robustness but may be exposed to the increased probability of FPP.

To solve the FPP problem, we have implemented two solutions in our study. First, we have performed encryption on U and V^T components by using the logistic map. Secondly, a 4D hyperchaotic system is used to encrypt the

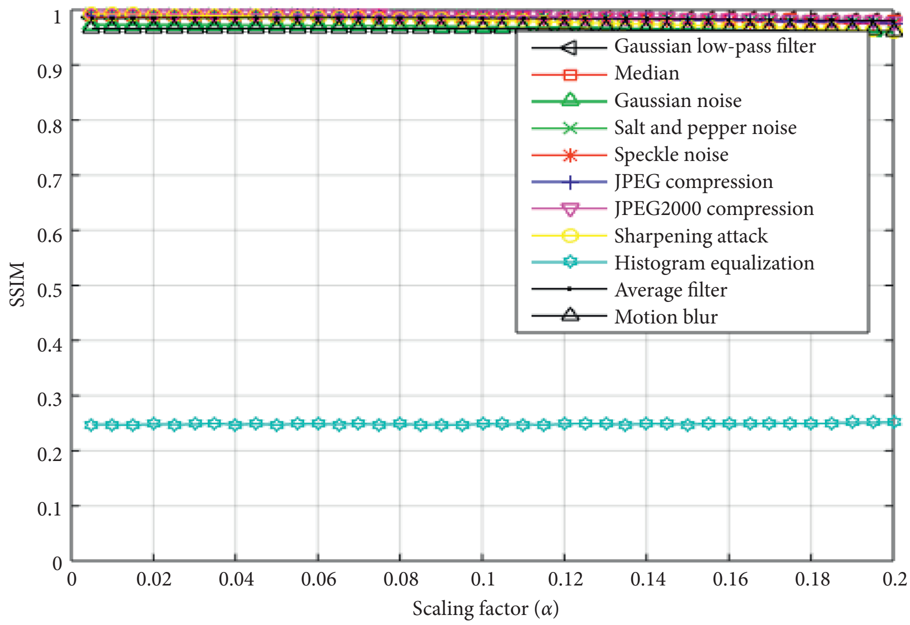


Figure 6: SSIM values under various scaling factors and attacks.

Table 4: NCs for the watermark images extracted from the attacked color watermarked images. Images shown in Figures 3(a) and 3(c) are used to compute the values of NCs under certain attacks.

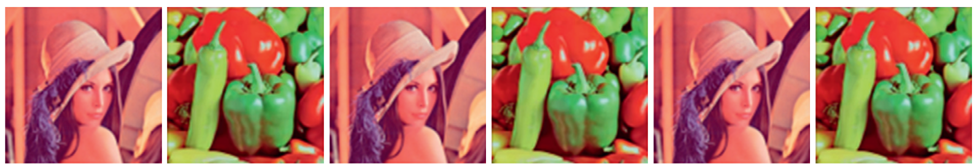
Attacks	NCs at $\alpha = 0.115$
No attack	1.0
Gaussian low-pass filter	0.827244
Median filtering (5, 1)	0.887447
Gaussian noise	0.908452
Salt and pepper noise	0.944562
Speckle noise	0.951758
JPEG compression	0.900064
JPEG2000 compression	0.880474
Sharpening attack	0.99612
Histogram equalization	0.826685
Average filter	0.826939
Motion blur	0.830194

watermark before embedding it into the cover image. This gives an additional layer of security against FPP. Therefore, it will be mandatory to decrypt again the watermark after extraction. In the experimental setup of FPP, a watermark (64×64) is chosen as shown in Figure 11(a). A decrypted watermark with correct parameters having $NC = 1.0000$ is shown in Figure 11(b), while Figure 11(c) is the extracted watermark ($NC = 0.62$) with incorrect parameters which is not recognizable.

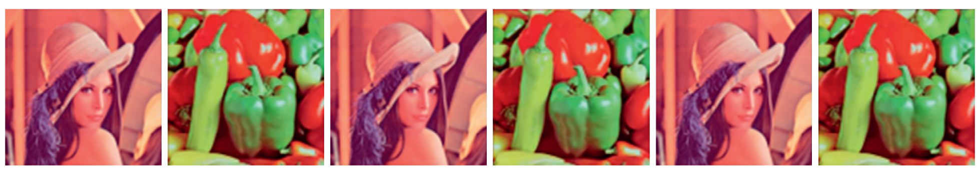
5.4. Performance Comparison. In this section, the proposed watermarking scheme is compared with some recently published schemes. The robustness comparison based on NC values after applying some attacks is shown in Table 5. It is obvious that, under some attacks, our results are better when compared with the recently published schemes. The improved results are written in bold format. The imperceptibility comparisons listed in Table 6 are based on the average NC, PSNR, and SSIM between the cover and watermarked images. It is clear that imperceptibility results are better than some recently published works when compared in most cases. Computational time consisting of watermark embedding time, watermark extraction time, watermark encryption, and decryption time is given in Table 7. The computational time is verified by using five test host images having a dimension of 512×512 taken from the USC-SIPI image database while the three RGB images (Figures 2(c)–2(e)) having dimensions of 256×256, 128×128, and 64×64 are used as watermarks. The improved results such as watermark embedding and extraction time are written in bold format.

256 × 256 128 × 128 64 × 64

(a)



(b)



(c)

35.5933 35.5873 41.5762 41.5509 47.4446 47.3734

(d)

0.9986 0.9980 0.9997 0.9995 0.9994 0.9999

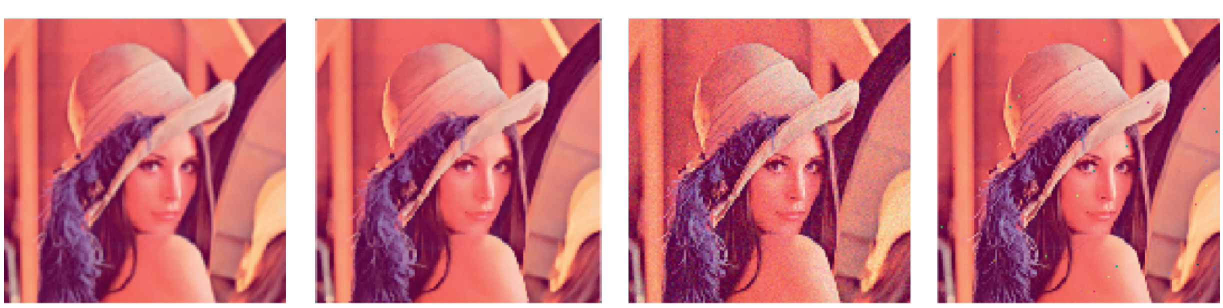
(e)

(f)

1.000 1.000 1.000 1.000 1.000 1.000

(g)

FIGURE 7: Invisibility test results at the scaling factor of 0.115. (a) Watermark. (b) Host image 1024 × 1024. (c) Watermarked images. (d) PSNR (db). (e) SSIM. (f) Extracted watermark. (g) NC (without attack).



(a) (b) (c) (d)

FIGURE 8: Continued.

FIGURE 8: Various attacks on watermarked images. (a) Gaussian low-pass filter, (b) median, (c) Gaussian noise, (d) salt and pepper noise, (e) speckle noise, (f) JPEG compression, (g) JPEG2000 compression, (h) sharpening attack, (i) histogram equalization, (j) average filter, (k) motion blur.

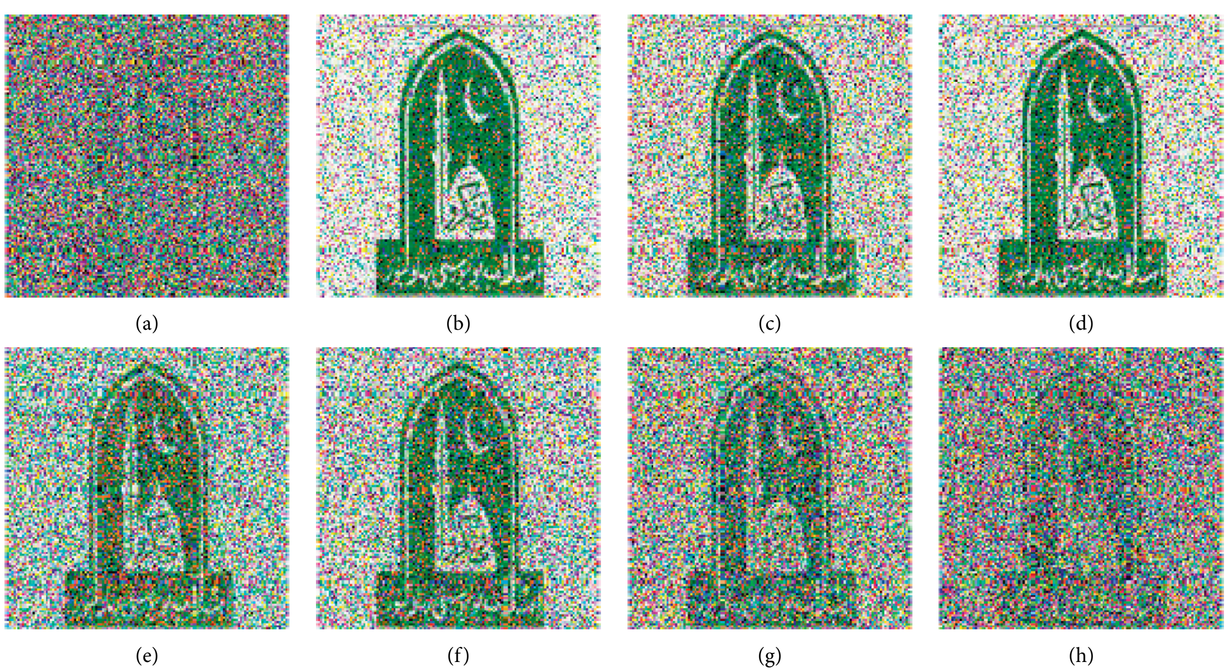
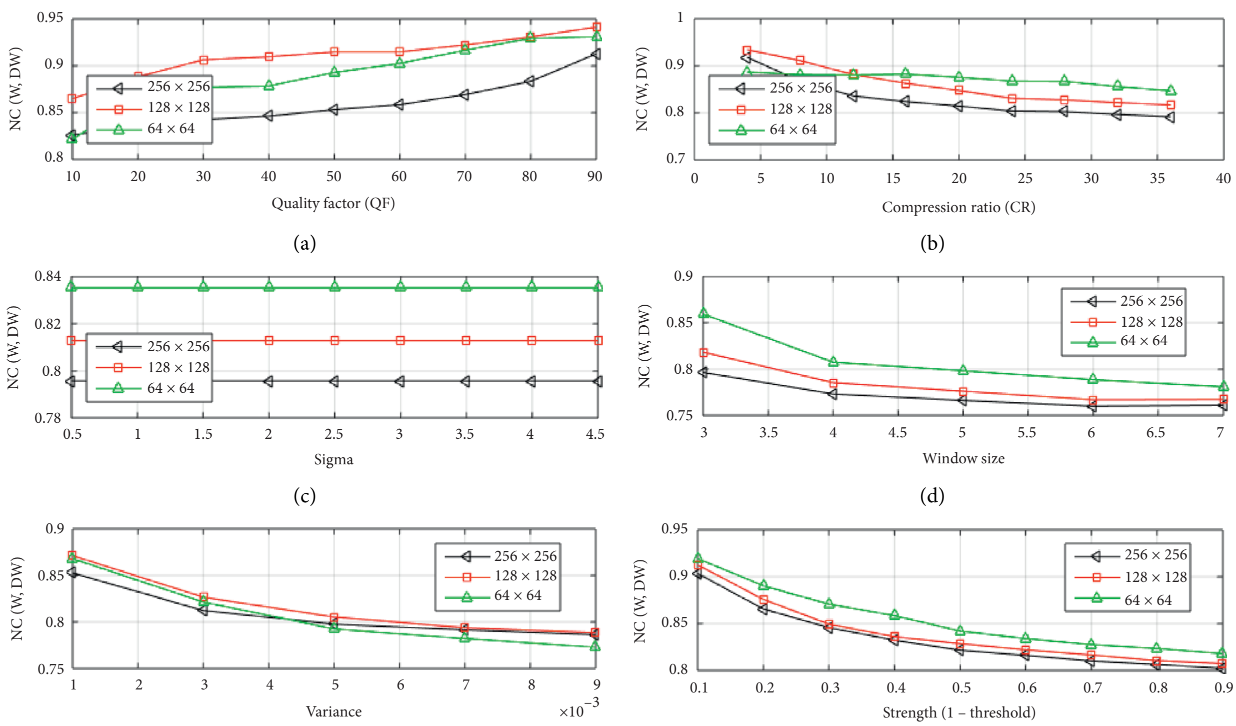


FIGURE 9: Continued.

(i) (j) (k)

Figure 9: Watermarks extracted from attacked watermarked images given in Figure 8. (a) Gaussian low-pass filter: NC = 0.79286. (b) Median: NC = 0.94175. (c) Gaussian noise: NC = 0.9124. (d) Salt and pepper noise: NC = 0.93241. (e) Speckle noise: NC = 0.85761. (f) JPEG compression: NC = 0.88661. (g) JPEG2000 compression: NC = 0.85736. (h) Sharpening attack: NC = 0.82034. (i) Histogram equalization: NC = 0.76479. (j) Average filter: NC =0.79288. (k) Motion blur: NC = 0.76144.



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

Figure 10: NC results under different attacks and parameters. (a) JPEG compression, (b) JPEG2000 compression, (c) Gaussian low-pass filter, (d) median filter, (e) Gaussian noise, and (f) sharpening attack.

(a) (b) (c)

Figure 11: FPP results with correct and incorrect parameters. (a) Original watermark, (b) the decrypted watermark with correct parameters, and (c) the decrypted watermark with incorrect parameters.

Table 5: Robustness comparison on the bases of NCs under certain attacks, whenever the data are available.

Attacks	Our NC (ref NC)
No attack	1.0 (1.0 [25], 1.0 [38])
Gaussian low-pass filter	0.827244
Median filtering (5, 1)	0.8874 (0.9968 [25], 0.5743 [39], 0.9356 [40], 0.8814 [38], 0.8370 [41])
Median filtering (3, 3)	0.847447 (0.7897 [42], 0.9188 [30], 0.9258 [9])
Average filter' (3, 3)	**0.826939** (0.7569 [42])
Gaussian noise ($M = 0$, $V = 0.0001$)	0.94806 (0.9706 [25], 0.9256 [39], 0.9387 [40], 0.9131 [38])
Salt and pepper noise (0.001)	0.984562 (0.9952 [25], 0.9287 [39], 0.9122 [40], 0.9902 [38], 0.9421 [43])
Speckle noise	0.951758
JPEG compression (Qf = 30)	0.87903 (0.9968 [25], 0.8594 [39], 0.9789 [40], 0.8469 [38])
JPEG compression (Qf = 90)	0.92956 (0.9862 [30], 0.9061 [43], 0.89109 [32])
Sharpening attack (0.2)	**0.96612** (0.9138 [42], 0.9579 [9])
Sharpening attack (0.8)	0.856457
Sharpening attack (1.0)	0.8052 (0.9638 [25], 0.9877 [39], 0.9366 [40], 0.9999 [38])
Histogram equalization	0.8785 (0.8805 [11])
Motion blur	0.830194
Flip (horizontal/Vertical)	0.9912 (1.0 [19])

The optimized scaling factor is $\alpha = 0.115$.

Table 6: Average NC, PSNR, and SSIM comparison between original and watermarked images whenever the data are available.

Image dimensions, watermark dimensions	Our NC (ref NC)	Our PSNR (ref PSNR)	Our SSIM (ref SSIM)
512×512, 256×256	0.999830	29.2385	0.9941
512×512, 128×128	0.999959	35.2342	0.9987
512×512, 64×64	0.999990	**41.1389** (40.77 [32], 35.97 [43], 38.95 [44])	**0.9997** (0.9885 [43])

The scaling factor is $\alpha = 0.115$.

Table 7: Computational time comparison whenever the data are available.

Computational time (s)	Our value (ref value)
Watermark encryption time	0.127723
Watermark embedding time	**0.350738** (0.8509 [42], 0.611810 [30], 0.7901 [41])
Watermark extraction time	**0.140354** (0.2295 [42], 0.2015 [41])
Watermark decryption time	0.127649

6. Conclusions and Future Directions

This paper is an attempt toward developing an imperceptible, secure, and robust watermarking framework with the procedure of scaling factor optimization based on IEFOA to solve the issues of authentication, integrity, and FPP. Host images can be embedded with color watermarks of multiple dimensions efficiently. Prior to the embedding procedure, the color watermark is encrypted by using a hyperchaotic system whose initial parameters are found from a DNA sequence taken from the NCBI dataset. After encrypting the RGB components of the watermark image, the embedding procedure consisting of logarithmic-based DWT, HbD, and SVD is utilized to obtain the watermarked image. Host images embedded with watermarks have shown an average PSNR greater than 35 which is considered acceptable and makes watermark invisible to the human visual system. This scheme also accomplishes excellent imperceptibility but with comparable robustness results. Moreover, the double encryption (before SVD and after SVD) makes it more secure to cope up with the security issues. A slight modification in the SVD parameters or hyperchaotic key makes the extracted watermark completely unrecognizable.

In the future, we intend to extend the proposed scheme to DICOM imaging such as ultrasound, X-rays, and magnetic resonance imaging. We also intend to make it more robust against attacks in which it is not robust. Moreover, we intend to adapt this scheme with other frequency transforms by combining it with higher-dimensional hyperchaotic systems to achieve high-efficiency batch processing.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Li, S.-T. Kim, and I.-K. Lee, "Robustness enhancement for image hiding algorithm in cellular automata domain," *Optics Communications*, vol. 356, no. 1, pp. 186–194, 2015.

[2] Q. Su, Y. Niu, H. Zou, Y. Zhao, and T. Yao, "A blind double color image watermarking algorithm based on QR decomposition," *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 987–1009, 2014.

[3] A. Mishra, C. Agarwal, A. Sharma, and P. Bedi, "Optimized gray-scale image watermarking using DWT-SVD and firefly algorithm," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7858–7867, 2014.

[4] I. A. Ansari, M. Pant, and C. W. Ahn, "Artificial bee colony optimized robust-reversible image watermarking," *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 18001–18025, 2017.

[5] V. Aslantas, A. Dogan, and S. Ozturk, "DWT-SVD based image watermarking using particle swarm optimizer," in *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo*, pp. 241–244, Hannover, Germany, June 2008.

[6] W.-T. Pan, "A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[7] C. Chang, S. Member, C. Li, and S. Member, "Privacy-aware reversible watermarking in cloud computing environments," *IEEE Access*, vol. 6, pp. 70720–70733, 2020.

[8] H. Singh, "Watermarking image encryption using deterministic phase mask and singular value decomposition in fractional Mellin transform domain," *IET Image Processing*, vol. 12, no. 11, pp. 1994–2001, 2018.

[9] N. A. Loan, S. Member, N. N. Hurrah, and S. Member, "Secure and robust digital image watermarking using coefficient differencing and chaotic encryption," *IEEE Access*, vol. 6, pp. 19876–19897, 2018.

[10] Y.-M. Li, D. Wei, and L. 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.

[11] T. K. Araghi, A. A. Manaf, and S. K. Araghi, "A secure blind discrete wavelet transform based watermarking scheme using two-level singular value decomposition," *Expert Systems with Applications*, vol. 112, pp. 208–228, 2018.

[12] A. Shaik and V. Masilamani, "A novel digital watermarking scheme using dragonfly optimizer in transform domain," *Computers & Electrical Engineering*, vol. 90, Article ID 106923, 2021.

[13] S. U. Yang, D. Wei, and R. Zhou, "An optimized image watermarking method based on HD and SVD in DWT domain," *IEEE Access*, vol. 7, pp. 80849–80860, 2019.

[14] M. Begum and M. S. Uddin, "Analysis of digital image watermarking techniques through hybrid methods," *Advances in Multimedia*, vol. 2020, no. 2, pp. 1–12, 2020.

[15] M. Kaur and D. Singh, "Multiobjective evolutionary optimization techniques based hyperchaotic map and their applications in image encryption," *Multidimensional Systems and Signal Processing*, vol. 32, pp. 281–301, 2020.

[16] A. Gupta, D. Singh, and M. Kaur, "An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1309–1324, 2020.

[17] M. Kaur, D. Singh, and R. Singh Uppal, "Parallel strength Pareto evolutionary algorithm-II based image encryption," *IET Image Processing*, vol. 14, no. 6, pp. 1015–1026, 2020.

[18] M. Kaur, D. Singh, and V. Kumar, "Color image encryption using minimax differential evolution-based 7D hyper-chaotic map," *Applied Physics B: Lasers and Optics*, vol. 126, no. 9, 2020.

[19] B. Hassan, R. Ahmed, B. Li, and O. Hassan, "An imperceptible medical image watermarking framework for automated diagnosis of retinal pathologies in an eHealth arrangement," *IEEE Access*, vol. 7, pp. 69758–69775, 2019.

[20] R. Singh and A. Ashok, "An optimized robust watermarking technique using CKGSA in frequency domain," *Journal of Information Security and Applications*, vol. 58, Article ID 102734, 2021.

[21] M. Yamni, H. Karmouni, M. Sayyouri, and H. Qjidaa, "Image watermarking using separable fractional moments of Charlier–Meixner," *Journal of the Franklin Institute*, 2021.

[22] M. Sadeghi, R. Toosi, and M. A. Akhaee, "Blind gain invariant image watermarking using random projection approach," *Signal Processing*, vol. 163, pp. 213–224, 2019.

[23] M. Cedillo-Hernandez, A. Cedillo-Hernandez, M. Nakano-Miyatake, and H. Perez-Meana, "Improving the management of medical imaging by using robust and secure dual watermarking," *Biomedical Signal Processing and Control*, vol. 56, p. 101695, 2020.

[24] H.-T. Hu, L.-Y. Hsu, and H.-H. Chou, "An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated," *Information Sciences*, vol. 519, pp. 161–182, 2020.

[25] S. Sharma, H. Sharma, and J. B. Sharma, "An adaptive color image watermarking using RDWT-SVD and artificial bee colony based quality metric strength factor optimization," *Applied Soft Computing Journal*, vol. 84, pp. 1–30, 2019.

[26] M. Sundararajan and G. Yamuna, "Optimization of colour image watermarking using area of best fit equation and Cuckoo search algorithm," *Materials Today: Proceedings*, vol. 5, no. 1, pp. 1138–1146, 2018.

[27] N. R. Zhou, A. W. Luo, and W. P. Zou, "Secure and robust watermark scheme based on multiple transforms and particle swarm optimization algorithm," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2507–2523, 2019.

[28] B. Wang, "An adaptive image watermarking method combining SVD and wang-landau sampling in DWT domain," *Mathematics*, vol. 8, pp. 1–20, 2020.

[29] X. Cui, Y. Niu, X. Zheng, and Y. Han, "An optimized digital watermarking algorithm in wavelet domain based on differential evolution for color image," *PLoS ONE*, vol. 13, no. 5, pp. 1–15, 2018.

[30] M. Yousefi Valandar, M. Jafari Barani, and P. Ayubi, "A blind and robust color images watermarking method based on block transform and secured by modified 3-dimensional Hénon map," *Soft Computing*, vol. 24, no. 2, pp. 771–794, 2020.

[31] K. Soppari and N. S. Chandra, "Development of improved whale optimization-based FCM clustering for image watermarking," *Computer Science Review*, vol. 37, Article ID 100287, 2020.

[32] K. Fares, K. Amine, and E. Salah, "A robust blind color image watermarking based on Fourier transform domain," *Optik*, vol. 208, pp. 1–9, 2020.

[33] Sunesh and R. R. Kishore, "A novel and efficient blind image watermarking in transform domain," *Procedia Computer Science*, vol. 167, no. 2019, pp. 1505–1514, 2020.

[34] Q. Su, "Novel blind colour image watermarking technique using Hessenberg decomposition," *IET Image Processing*, vol. 10, no. 11, pp. 817–829, 2016.

[35] X. Yang, W. Li, L. Su, Y. Wang, and A. Yang, "An improved evolution fruit fly optimization algorithm and its application," *Neural Computing and Applications*, vol. 32, no. 14, pp. 9897–9914, 2019.

[36] J. Sun, M. Peng, F. Liu, and C. Tang, "Protecting compressive ghost imaging with hyperchaotic system and DNA encoding," *Complexity*, vol. 2020, Article ID 8815315, 13 pages, 2020.
[37] Q.-K. Pan, H.-Y. Sang, J.-H. Duan, and L. Gao, "An improved fruit fly optimization algorithm for continuous function optimization problems," *Knowledge-Based Systems*, vol. 62, pp. 69–83, 2014.
[38] Q. Su, G. Wang, and X. Zhang, "A new algorithm of blind color image water- marking based on LU decomposition," *Multidimensional Systems and Signal Processing*, vol. 29, no. 2018, pp. 1055–1074, 2018.
[39] Q. Su and B. Chen, "Robust color image watermarking technique in the spatial domain," *Soft Computing*, vol. 22, no. 1, pp. 91–106, 2017.
[40] S. Roy and A. K. Pal, "An SVD based location specific robust color image watermarking scheme using RDWT and arnold scrambling," *Wireless Personal Communications*, vol. 98, no. 2, pp. 2223–2250, 2018.
[41] K. Prabha and I. Shatheesh Sam, "An effective robust and imperceptible blind color image watermarking using WHT," *Journal of King Saud University-Computer and Information Sciences*, 2020.
[42] H. Zhang and C. Wang, "A robust image watermarking scheme based on SVD in the spatial domain," *Future Internet*, vol. 9, no. 45, pp. 1–16, 2017.
[43] Y. Cao, F. Yu, and Y. Tang, "A digital watermarking encryption technique based on FPGA cloud accelerator," *IEEE Access*, vol. 8, no. 1, pp. 1–15, 2020.
[44] H. Xu, X. Kang, Y. Chen, and Y. Wang, "Rotation and scale invariant image watermarking based on polar harmonic transforms," *Optik*, vol. 183, pp. 401–414, 2019.