Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09250-5

A hybrid matrix factorization technique to free the watermarking scheme from false positive and negative problems

K. Meenakshi[1] · K. Swaraja[1] · Padmavathi Kora[1]

Received: 26 April 2019 / Revised: 13 June 2020 / Accepted: 24 June 2020 /
Published online: 12 August 2020

Abstract
Optimization techniques are used in watermarking algorithms to balance the trade-off between the two mutually conflicting parameters-imperceptibility and robustness. Differential Evolution (DE) with Rotational Cross over (RCO), namely DE-RCO, is employed in the proposed algorithm to improve the convergence of optimization. The DE-RCO provides a watermarking scheme with the highest robustness preserving the transparency. The watermark encoding and decoding carried in the transform domain with a hybrid combination of Schur Factorization (SF) and Singular Value Decomposition (SVD). Two-fold security provided with the Digital Signature (DS) generation and B-test. The DS is concealed in the watermarked image for authentication purpose of freeing the algorithm from the false-positive problem. When the signature passed through the insecure channel, it may corrupt and lead to the failure of the A-test, even the correct watermark given at the extraction leading to the false-negative problem. B-test is conducted to free the algorithm from a false negative problem. An enhanced authentication framework designed and implemented to make the algorithm free from the false-positive and false-negative issues using DS and the hash generated by Gaussian Hermite Moments (GHMs). The experimental results compared with the related existing watermarking schemes demonstrate that the proposed scheme exhibit strong robustness to attacks, invisible and highly secure against false-positive and false-negative problems.

Keywords Rotation crossover · Gaussian Hermite Moments · Differential Evolution · Schur Factorization · Singular Value Decomposition

1 Introduction

The widespread usage of multimedia accompanied by high-speed Internet and mobile technology has introduced new problems to content owners of multimedia documents [13, 14, 34]. Digital watermarking protects the interests of content owners by concealing the

✉ K. Meenakshi
mkollati@gmail.com

[1] Department of ECE, GRIET, Hyderabad, India

owner's information as watermark [43]. The watermark is embedded in the host media with a slight modification to obtain the watermarked image. The watermark must be decrypted at the extraction side even when different attacks distort the watermarked image. Because of these characteristics, the watermarking techniques are widely used in copyright protection, broadcast monitoring, and multimedia anti-piracy.

In the literature, watermarking is performed in three types of domains, namely spatial, frequency, and hybrid domains. In the spatial domain [5, 28, 29, 44], watermark is embedded directly into the pixels of image. Examples of watermarking in pixel domain are embedding of the watermark in the Least Significant Bit (LSB) plane or Three–Dimensional (3-d) meshes [5]. The advantage of the spatial domain technique lies in simple implementation and requires less processing time for the execution. Su et al. [44] computed approximated Eigen Values (EVs) of Schur Factorization (SF), and later these coefficients are used to conceal the RGB watermark in the color host image. The algorithm has a real-time advantage and resistant to attacks such as rotation, 50% cropping and Salt and Pepper noise, etc. Spatial domain watermarking schemes are seldom used because the watermark is erased easily by hackers using simple cropping and rotation. On the other hand, in the transform domain techniques, the watermark is concealed in the transform coefficients. The image in the spatial domain has been transformed into a frequency domain with image transforms such as Walsh-Hadamard transform(WHT) [11, 32], Discrete Cosine transform(DCT) [39, 40, 47], Discrete Wavelet transform (DWT) [45], Curvelet Transform [36], Contourlet Transform (CT) [26], Redundant Discrete Wavelet Transform (RDWT) [49], Sequency Complex Hadamard Transform (SCHT) [10] and Slant Transform [33, 35] etc. Transform domain techniques have several advantages over spatial domain methods such as the user has freedom to select the middle frequency band for embedding the watermark to protect it from the third party. The watermarks in the transform domain techniques are more robust to geometrical and signal processing attacks. Further, image transforms like Radon and Discrete Fourier Transform (DFT) are robust to affine transformation. Finally, the watermark is dispersed irregularly, which makes the third party hard to erase the watermark. The disadvantage of this technique is, it requires more processing time when compared with spatial domain techniques.

The efficiency of the watermarking scheme is further enhanced by combining a cascade of image transforms called hybrid watermarking. The attributes extracted from the hybrid domain remarkably are more stable against attacks compared to a single transform. Reference [26] proposed a hybridized watermarking technique based on Non-sub sampled CT and SVD. The technique exploits multi-resolution and multi-directional features of CT and the intrinsic algebraic properties of SVD to improve the robustness. Reference [7] developed a hybrid watermarking using DWT and two-level SVD. The host image is segmented into LL, LH, HL, and HH band. They experimentally found that the watermark embedded in the HH band is more robust to attacks. Therefore, the authors inserted a watermark in the HH band. The false-positive and false-negative problems are resolved in work [7] by conducting two authentication tests, namely A-test and B-test. In A-test, a Digital Signature (DS) is generated based on eight coefficients randomly taken from DWT. The B test utilized hash generated by SVD-SVD hashing [27] using matrix invariants of SVD.

The drawback of watermarking in pixel, transform, or hybrid domain is that the two watermarking performance metrics imperceptibility and robustness are mutually conflicted with each other [8]. If imperceptibility increases, the robustness of watermarking scheme degrades and vice versa. Therefore, in recent years to optimize transparency and robustness simultaneously watermarking techniques are combined with optimization methods such as Fuzzy Logic [31], Genetic Algorithm (GA) [8, 41], Particle Swarm Optimization (PSO)

[9] and Cuckoo Search (CS) algorithm [4]. The The fuzzy-based watermarking scheme may not yield optimal results because it requires accurate and complete information about the parameters of the watermarking scheme. GA based watermarking scheme in SVD domain developed by [8] achieves the highest robustness against attacks preserving the transparency. The optimization algorithm GA considers 400 generations and a population size of 150 to obtain optimum scaling factors. The disadvantage of GA based algorithm is that it suffers from high computational time. Further, as this scheme employed SVD in the final stage of watermarking, it suffers from the false-positive problem. Reference [41] employed second and third-level horizontal detail sub-band (LH2 and LH3) coefficients of DWT in the cover image which grouped as blocks of five for watermark insertion. The coefficients grouped in such a way that each block should contain one coefficient from the LH3 sub-band and the remaining four coefficients from the LH2 sub-band. In each block, the first and second minimum is identified and modified according to the watermark bit. After watermark insertion, inverse DWT is applied to the sub-bands with modified coefficients to obtain the watermarked image. A threshold-based decoder is designed in the algorithm for watermark extraction. This watermarking is controlled by four scaling parameters derived from GA. The main limitation of this work is that as the number of scaling factors reduces, the optimization algorithm may only give moderate results for the chosen attacks. A watermarking scheme using dynamic PSO is proposed by [42], for color images. The authors employed dynamic PSO to mitigate the stagnation of local and global best p_{best} and g_{best} occurred in classical PSO. The main advantage is that it allows the algorithm to reach the global minima without trapping in the local minima. PSO based watermarking schemes provide optimum results, yet they have suffered from the curse of high dimensionality. Reference [4] proposed a robust watermarking in the DWT domain using a CS algorithm. The main advantage of the CS algorithm is that it uses Levy flight as a global search strategy. The LL and HH bands of DWT coefficients are changed with the watermark using two scaling factors derived from these respective bands using CS. The main limitation of this work is that the robustness is poor if the scaling factors are less. Reference [2] employed a DCT-SVD based gray image watermarking using Differential Evolution (DE). In DE, the authors utilized Binomial Crossover, which may not always guarantee the algorithm to reach global minima as its convergence is unstable and traps itself to local minima. Moreover, the DE proposed by this algorithm has limited diversity and slow convergence. A highly convergent optimization using DE-RCO with a multi-angle strategy propounded by [15]. It is utilized in the present work to mitigate the above problems. The watermark is concealed in transform coefficients obtained with two matrix factorization techniques, namely Schur Factorization (SF) and Singular Value Decomposition (SVD). The cost function of DE-RCO devised in the proposed work is in such a fashion that the algorithm sustains to geometrical and signal processing attacks. Above all, an efficient authentication framework is designed and implemented to free the proposed watermarking scheme from false-positive and false-negative problems.

The remainder of the paper is organized as follows: In Section 2, background material, and false-positive problem in SVD described. The proposed watermark embedding and extraction using SF-SVD with DE-RCO presented in Section 3. An authentication framework is designed and incorporated in watermark embedding and extraction to withstand the proposed scheme against false-positive and false-negative problems. In Section 4, the proposed watermarking scheme is compared with other schemes in the literature in terms of imperceptibility, robustness, capacity, security, and computational time to demonstrate the effectiveness of the proposed scheme. Finally, conclusions presented in Section 5.

2 Background material

A brief introduction of transforms SF and SVD used in the algorithm are discussed in Sections 2.1 and 2.2. The classical DE and motivation for using DE plus RCO are provided in the following Sections 2.3 and 2.4. The false-positive problem and its rectification in literature is presented in Section 2.5.

2.1 Schur Factorization

Schur Factorization (SF) [24] is an intermediary step in SVD decomposition. It has advantages such as in comparison with SVD, less number of flip flops is required for real time implementation. This transform is imperceptible and robust to attacks such as salt and pepper noise, rotation etc [24]. SF factorizes the image I into unitary matrix A and Quasi-upper triangular matrix T as given in (1).

$$[\mathrm{A}, \mathrm{T}] = schur(\mathrm{I}) \tag{1}$$

Where, $AA' =$ Identity matrix I_n and inverse SF is given by (2).

$$\mathrm{I}_{rec} = \mathrm{ATA}' \tag{2}$$

Where I_{rec} is the reconstructed image and the diagonal elements of T contain Eigen Values (EVs) of I.

2.2 Singular Value Decomposition

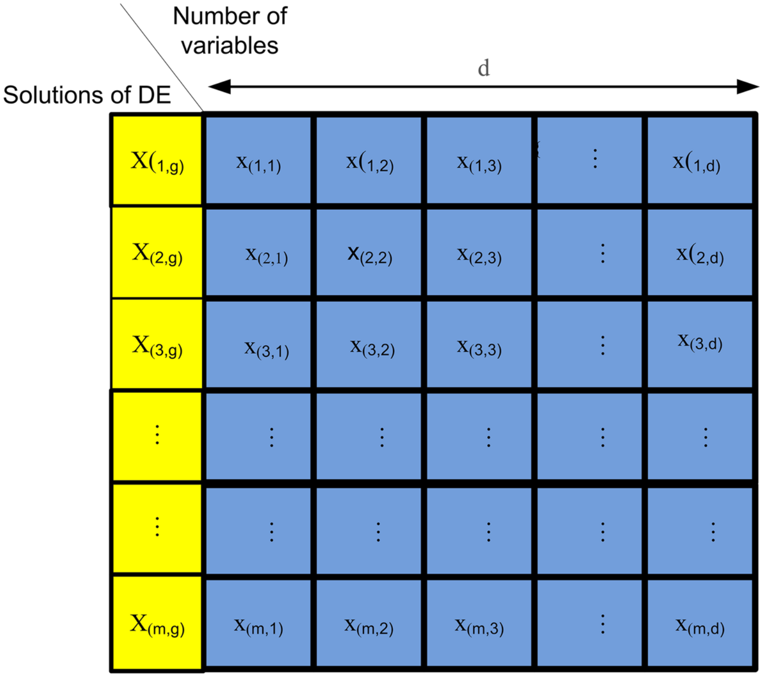
The data in SVD is decorrelated so that most of the information in the pixel domain is packed in few coefficients after the transformation [3]. SVD decomposes the image into three individual matrices U, V and S.

$$I = \sum_{j=1}^{r} U_j S_j V_j^T \tag{3}$$

Where, U and V are left and right singular vectors which are orthogonal matrices satisfying UU' and VV' equal to identity matrix I_n, U' and V' are the transpose of U and V. S is a diagonal matrix with Singular Values (SVs) as diagonal elements. The SVs are in descending order and the elements above rank r are zero, satisfying (3).

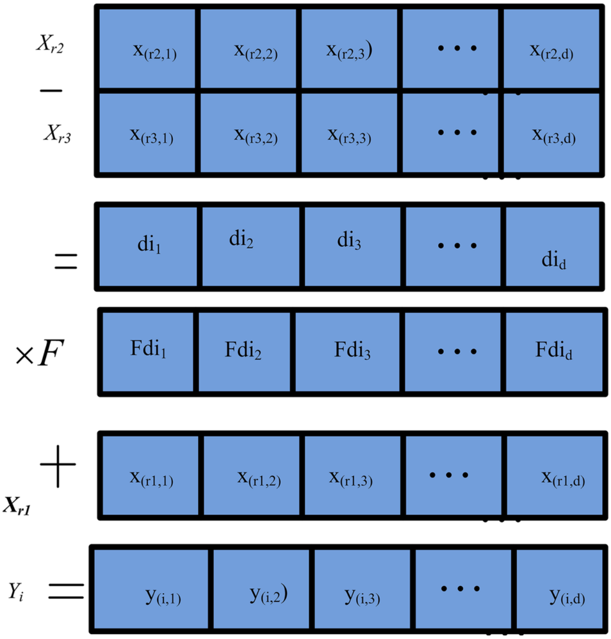
2.3 Classical Differential Evolution

DE is the most competent and flexible evolutionary computing algorithm used in various domains such as image retrieval [18, 51], watermarking [43], image registration [20], forensic image processing [12] and thermodynamics, process control and industrial automation [19]. In this work, the classical DE is modified to balance the trade-off between exploration and exploitation in search space of the optimization algorithm. The main steps in DE are mutation, crossover and selection. The step mutation provides the needed diversity in the population and the other step crossover assures the capacities of the population to survive in the next generation. The DE operates with a population of random initial solutions of dimension d, $X_{i,g} = \left[x_{i,1}, x_{i,2}, \cdots x_{i,d}\right]$, $i = 1, 2, \cdots m$ where index i denotes the solution of the population at generation g as shown in the Fig. 1. In mutation, three distinct vectors$(i.e., r_1 \neq r_2 \neq r_3 \neq i)$ out of m are selected for obtaining mutant vector $Y_{i,g}$ as shown in Fig. 2. The scaled difference of two vectors $\mathbf{X}_{r_2} = X_{(r_2,1)}, X_{(r_2,2)} \cdots$,



$X_{(1,g)}$	$x_{(1,1)}$	$x_{(1,2)}$	$x_{(1,3)}$	⋮	$x_{(1,d)}$
$X_{(2,g)}$	$x_{(2,1)}$	$x_{(2,2)}$	$x_{(2,3)}$	⋮	$x_{(2,d)}$
$X_{(3,g)}$	$x_{(3,1)}$	$x_{(3,2)}$	$x_{(3,3)}$	⋮	$x_{(3,d)}$
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
$X_{(m,g)}$	$x_{(m,1)}$	$x_{(m,2)}$	$x_{(m,3)}$	⋮	$x_{(m,d)}$

Fig. 1 Initial Population of DE



X_{r2}	$x_{(r2,1)}$	$x_{(r2,2)}$	$x_{(r2,3)}$	• • •	$x_{(r2,d)}$
$-$					
X_{r3}	$x_{(r3,1)}$	$x_{(r3,2)}$	$x_{(r3,3)}$	• • •	$x_{(r3,d)}$
$=$	di_1	di_2	di_3	• • •	di_d
$\times F$	Fdi_1	Fdi_2	Fdi_3	• • •	Fdi_d
$+$					
X_{r1}	$x_{(r1,1)}$	$x_{(r1,2)}$	$x_{(r1,3)}$	• • •	$x_{(r1,d)}$
$Y_i =$	$y_{(i,1)}$	$y_{(i,2)}$	$y_{(i,3)}$	• • •	$y_{(i,d)}$

Fig. 2 Mutation operation of DE

$X_{(r_2,d)}$ and $X_{r_3} = X_{(r_3,1)}, X_{(r_3,2)} \cdots, X_{(r_3,d)}$ is given by a difference vector $di_1, di_2, \cdots, di_d$. The obtained difference vector is scaled by a constant scaling factor F and later it is added to third vector $\mathbf{X}_{r_1} = X_{(r1,1)}, X_{(r1,2)} \cdots, X_{(r1,d)}$ to obtain mutant vector $Y_{i,g} = y_{(i,1)}, y_{(i,2)} \cdots, y_{(i,d)}$ as shown in Fig. 2. The equation used to compute mutant vector $Y_{i,g}$ is

$$Y_{i,g} = X_{r1,g} + F \times \left(X_{r2,g} - X_{r3,g}\right) \tag{4}$$

Later, crossover operation is performed on mutant vector $Y_{i,g}$ and target vector $X_{i,g}$ to produce trial vector, $Z_{i,g} = \left[z_{i,1}, z_{i,2} \cdots z_{i,d}\right]$. As graphically shown in Fig. 3 if $Cr \leq r_{ij}$, the elements in the trial vector are replaced with elements of mutant vector. Otherwise, they are replaced with the elements of the target vector.

$$Z_{i,g} = \begin{cases} Y_{i,g} \;\; if \;\; r_{ij} \leq Cr \vee j = k \\ X_{i,g} \qquad\quad otherwise \end{cases} \tag{5}$$

where $j = 1, \cdots d$ and $k \in \{1, \cdots d\}$ is a random parameter chosen for each i and $Cr \in [0, 1]$ is crossover probability and $r_{i,j}$ is a random number generator between 0 and 1. After obtaining the trial vector, in the final step, selection, the cost function of trial and target vectors is computed. If the trial vector's cost function is less than the target vector, then the vector carried in the next-generation is a trial vector. Otherwise, it is the target vector as shown in Fig. 3.

$$X_{i,g+1} = \begin{cases} Z_{i,g} \;\; if \;\; f\left(Z_{i,g}\right) < f\left(X_{i,g}\right) \\ X_{i,g} \qquad\quad otherwise \end{cases} \tag{6}$$

2.4 Motivation to use DE-RCO

The watermarking scheme proposed by [2] is based on DE, employing binomial Crossover, which produces three probable trial vectors in a 2-D search space $X1$ and $X2$, by combining the target vector $X_{i,g}$ with the mutant vector $Y_{i,g}$ in a uniform manner as explained below

(i) Both the components of the trial vector come from the mutant vector, denoted as $Y_{i,g}$.
(ii) The first component of the trial vector (j = 1) comes from $Y_{i,g}$ while the second component (j = 2) comes from $X_{i,g}$, denoted as $Y^{|}_{i,g}$.

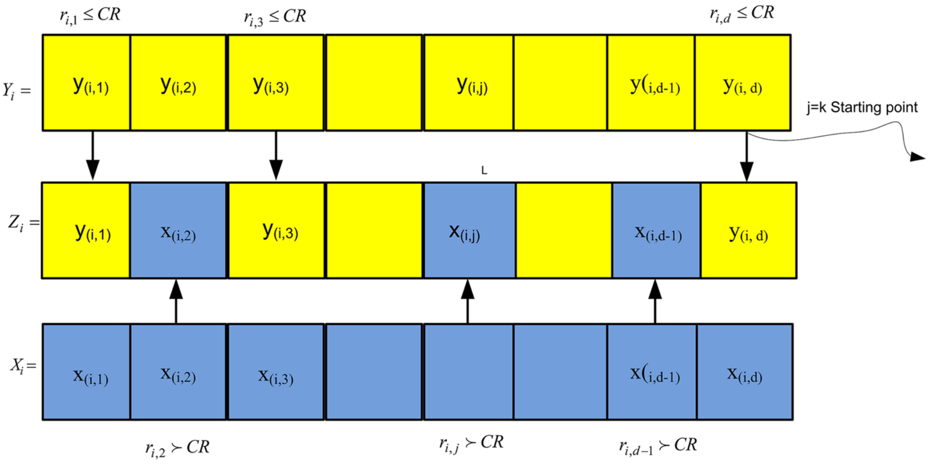


Fig. 3 Crossover step of DE

(iii) The first component of the trial vector (j = 1) is inherited from $X_{i,g}$, and the second component (j = 2) comes from $Y_{i,g}$, denoted as $Y_{i,g}^{||}$.

The possible three trial vectors are generated by the conventional binomial crossover presented in Fig. 4 exchanging components of the target and mutant vector along the axis. In the binomial crossover, the difference vectors of the offspring in the succeeding generation are parallel to the axes. Thus, in that case, most of the difference vectors of target vectors are perpendicular to the same axis. Therefore, standard binomial crossover operators will have problems exploring the search space due to a lack of population diversity. In the binomial crossover, some trial vectors produced near optimum global points. Even then, the exploring competence is too low, and its robustness is not strong enough; thus, DE with binomial crossover is not much reliable. In this work, in place of binomial crossover, Rotation crossover operator (RCO) with DE named as DE-RCO [4] is explored. The modified DE with RCO has the advantage of preventing the population, generating weak individuals, increasing the search space tactically, and guiding the algorithm's evolution to reach the global optimum. Consider, the violet and green regions in the Fig. 5 corresponds to the possible locations for the offspring of DE-RCO. The red regions indicate the possible three trial vectors produced in the binomial crossover. From this, it understood that the search space is radically enhanced from three points to an approximate circular space around the donor and target individuals by taking advantage of both binomial crossover and RCO. The convergence of DE-RCO is further enhanced by introducing two variables-rotation control parameter ϕ and binary control parameter bin. ϕ and bin are used to generate trial vectors from both target and mutant vectors at a certain angle and distance. Binary parameter bin has values ± 1 that control the direction, 1 for forward and -1 for backward. ϕ is used to control the rotation angle and Radii of the trial vector. The advantage of RCO over classical binomial crossover is that the crossover probability is controlled adaptively. Further, all possible trial vectors can be generated by recombining the mutant vector and target vector in 2^d search space. The rotating control vector is given in (7).

$$\overrightarrow{\phi} = \overrightarrow{\phi^*}/\sqrt{g+1} \tag{7}$$

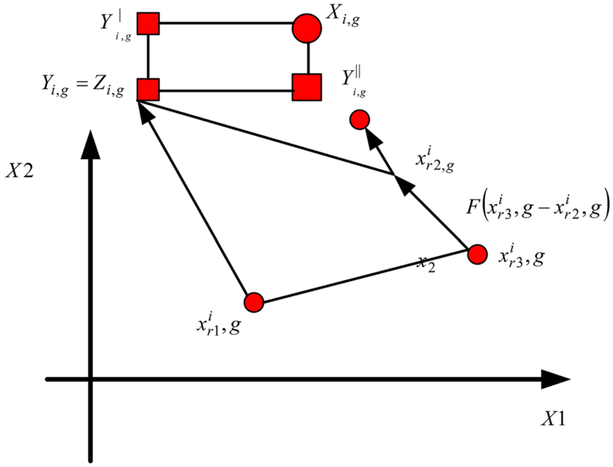


Fig. 4 Different crossover operations of Differential Evolution

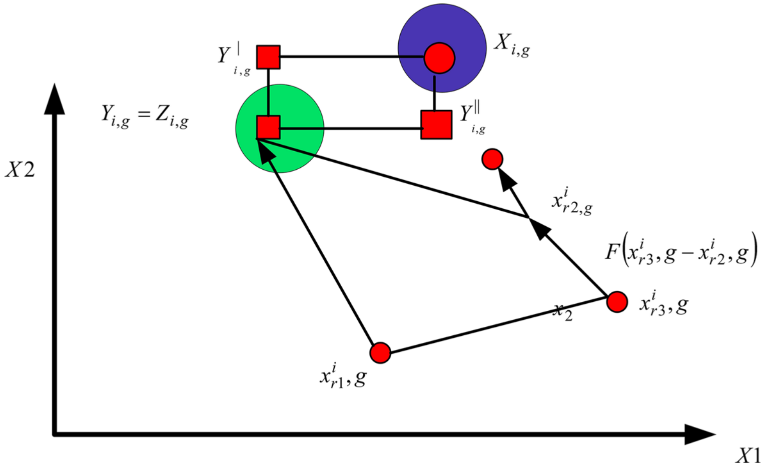


Fig. 5 Distinct trial vectors generated with binomial crossover operator between mutant and target vectors in 2-D search space

Where g is the iteration count of generation. With the increase in number of generations, rotation radius diminishes. This enhances convergence speed of the evolutionary process. The variable $\overrightarrow{\phi}^{\star}$ follows Levy probability distribution [15, 50]. It is a probability distribution function (pdf) defined with parameters m, d, μ and σ.

$$\overrightarrow{\phi}^{\star} = randlevy(m, d, \mu, \sigma) \tag{8}$$

Where m, d, μ and σ denotes the size of the population, dimension, location, and second-order moment, respectively. The probability density function of Levy distribution [50] is given as

$$L_{\mu,\sigma}\left(y\right) = -\frac{1}{\pi}\int_{0}^{\infty} e^{\sigma q^{\mu}} cos\left(qy\right) dq,\ y \in \phi_0 \tag{9}$$

Where ϕ_0 represents the field of real numbers, and the generic expression ϕ is not adopted to avoid confusion with the above symbols. The Cauchy and Gaussian PDF are both the special cases of the Levy distribution, and out of the entire three Levy distribution has a much longer tail. Therefore, the offspring produced in the Levy distribution can be quite distinct from their parents. In DE-RCO, the fundamental steps of DE such as selecting initial random population, mutation and selection process are retained except the trial vector is modified by RCO as given in (10).

$$\overrightarrow{Z}_{i,g} = \begin{cases} \overrightarrow{\phi}_{i,g} * (bin \cdot (\overrightarrow{Y}_{i,g} - \overrightarrow{X}_{i,g})/2) + \overrightarrow{Y}_{i,g}, \\ \qquad \text{if } \text{rand}_{i,j} \leq Cr \text{ or } j = j_{\text{rand}} \\ \overrightarrow{\phi}_{i,g} * (bin \cdot (\overrightarrow{Y}_{i,g} - \overrightarrow{X}_{i,g})/2) + \overrightarrow{X}_{i,g}, \\ \qquad \text{otherwise} \end{cases} \tag{10}$$

In this, the trial vector is formed around the mutant vector $\overrightarrow{Y}_{i,g}$, if the generated random number is less than CR or j is equal to j_{rand}. Otherwise, the trial vector is formed around the target vector $\overrightarrow{X}_{i,g}$. Thus, we make the combination of these two operators and parameter $p(0 < p < 1)$ is introduced to adjust the weight of proposed crossover and binomial crossover operator. p is adopted for selection of crossover operator. If p is less than 0.5, the optimization algorithm takes binomial crossover, otherwise, it takes RCO. The desirable feature of RCO operator is that it can be easily implemented in other DE variants with minor modifications. The main control parameters in DE are population size m, Scaling mutation factor F and Crossover probability CR. In classic DE, the above control parameters are fixed and tuned at generation level. Adaptive and Self adaptive Differential techniques tune the control parameters at individual level in each generation g. In DE with RCO, the Crossover probability CR_i of each individual x_i is generated independently at generation g based on normal distribution with mean μ_{Cg} and standard deviation of 0.1

$$CR_i = randn_i \left(\mu_{Cg}, 0.1\right) \tag{11}$$

Later it is truncated to [0, 1]. The low value of standard deviation used in (11) allows the adaption to function correctly. The mean Crossover μ_{Cg} is initially set to 0.5 and then generation count i updated and its value at generation g as

$$\mu_{Cg} = (1 - C) \cdot \mu_{Cg} + C \cdot mean_L(S_{Cg}) \tag{12}$$

Where C is a positive constant between 0 and 1 and S_{Cg} denotes the set of successful crossover probabilities at generation g. Recording of S_{Cg} in optimization process allows to guide for generating new CR_i that allows the individuals in each generation of DE-RCO to survive better.

$Mean_L$ (.) is the Lehmer Mean

$$Mean_L(S_{Cg}) = \frac{\sum_{CR \in S_{Cg}} CR^2}{\sum_{CR \in S_{Cg}} CR} \tag{13}$$

In the calculation of new crossover probabilities at each generation, the $Mean_L$ is helpful to propagate larger crossover probabilities, which might improve the progress rate of the evolutionary process. Better the control parameter CR_i, there is a higher chance of producing a better and diverse population, which may result in fast convergence of the DE algorithm.

The next subsection deals with the false-positive problem present in the SVD based watermarking schemes and discuss the methods in literature for tackling the problem.

2.5 Description of false-positive problem and prevention techniques in literature

Most of the SVD based watermarking schemes in the literature [8, 9, 32] employ SVs in SVD to conceal the watermark. In SVD, most of the energy is packed into the left and right singular vectors U and V, and therefore these two play a key role in extraction, when the inverse SVD is applied. Hence, the SVD based watermarking schemes [33] suffered from an ambiguous situation called false positive problem and is illustrated as follows: The logo, left singular vectors, SVs and right singular vectors of owner and pirate are shown in Figs. 6a, b, c and d and 7a, b, c and d respectively.

At the extraction phase, the hacker introduces singular vectors of pirated logo U_{wp} and V_{wp} in Fig. 7 and later it is combined with the SVs of owners logo S_{wo}. Therefore, instead of the embedded watermark, the counterfeit logo obtained, as shown in Fig. 8. This is called as false-positive problem. Several alternatives are proposed in the literature to prevent the false-positive problem. One solution for preventing the false positive problem is the insertion of the watermark in either one of singular vectors rather than SVs during watermark insertion. Another solution is instead of SVs, principal components $U \times S$ and $S \times V^T$ of watermark are inserted into the host image [6, 21, 22].The principal components of watermark $U \times S$ and $S \times V^T$ are multiplied by scaling factor and added to SVs of the host image. Later inverse SVD is applied to obtain the watermarked image. However, the main

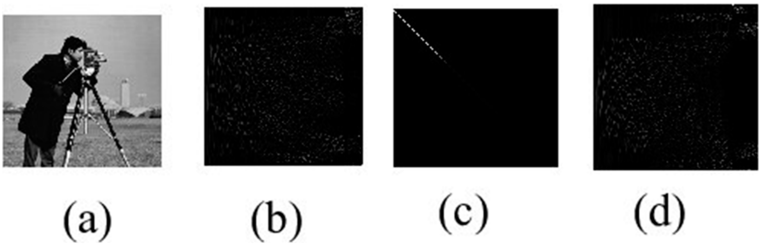


Fig. 6 **a** Owner's logo **b** U_{wo} **c** S_{wo} **d** V_{wo}

drawback is that watermark concealed in singular vectors, or principal components are less robust to attacks than the watermark concealed in SVs. Another solution [30] used to prevent false-positive problem is by using hash or image digest for authentication purpose. When the watermarked image is severely affected by geometric and noise attacks, it may affect the watermark as well as the hash inside the watermarked images. Reference [7] conducted two-fold authentication tests based on digital signature A-test and B-test to resolve the false-positive and false-negative problems. Initially, the image is given to a hash function and generates a fixed-length code (hash) created for the embedded watermark. Hash or message digest is a fixed-length string generated from the contents of the image [48, 52]. Recently image hashes is replacing cryptographic hashes such as MD-5, SHA-1. The main drawback in cryptographic hashes is a minute change in input bits, which causes a large variation in the output. Image hashing has two conflicting requirements of perpetual robustness and discrimination. Perpetual robustness must ensure similar images (Lena image, Lena image with low pass filtering, Lena image with JPEG compression) must have similar hash. In contrast, discrimination must ensure that different images must generate a different hash for each of the images. The 8-bit Digital Signature (DS) generally embedded after watermark concealed in the cover image. Later the DS of the obtained watermark singular vectors (it may belong to either owner or pirate) is compared with a DS from the signed image. If the DS of singular vectors of watermark and DS from the signed image belongs to the owner's logo cameraman, the watermarking scheme authenticated, that is, A-test is passed (A = 1), and the watermark extraction is allowed. If the authentication test is failed then (A = 0) and two situations might arise. The first situation may be the watermark extracted is the counterfeit logo (rose). When this happens, we can assume that the adversary has introduced the singular vectors of the fake watermark. The second situation

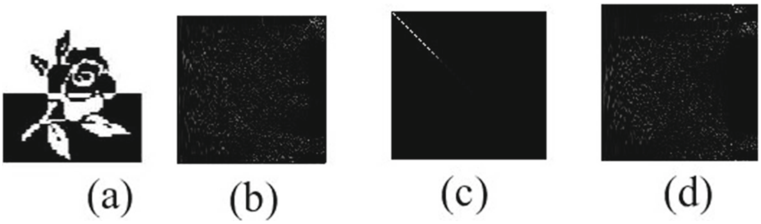


Fig. 7 **a** Pirated logo **b** U_{wp} **c** S_{wp} **d** V_{wp}

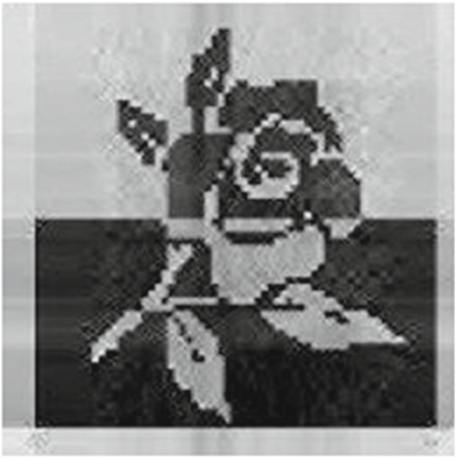


Fig. 8 Counterfeit logo rose

arose even when the correct watermark extracted, the authentication test A failed due to the DS in the watermarked image completely degraded with the severity of attacks. This problem, which is not resolved in the previous works, is addressed in [7] by incorporating an additional test called B-test. So if A = 0 and B = 0, the authentication test is failed. The authentication test is passed if A = 1 and B = 1 or A = 0 and B = 1. In the work of [7], the DWT hashing scheme of [38] is modified to form the DS. Later DWT and SVD are applied, and the second-row first column element of right singular vector $V_{2,1}$ of selected eight 4×4 blocks used for concealing the 8-bit signature.

On the extraction side, the 8 bit DS extracted from the signed image is compared with the 8 bit DS generated by the user entered singular vectors. Suppose user entered singular vectors are U_{WE} and V_{WE}. If the DS obtained from singular vectors at the extraction phase is not the same as signature extracted from the signed image, the extraction of the watermark delayed by conducting another test called B-test.

In B-test, the authors of [7] uses image hashing by using matrix invariants of SVD known as SVD-SVD hashing [27]. In this hashing scheme, images and attacks on them are assumed as a sequence of linear operations. The authors used overlapping blocks whose size and locations are randomly chosen from a suitable distribution to form a secondary image from the input image. The image hashing carried in two stages. In the first stage of image hashing, robust feature vectors derived from pseudo-random semi-global regions via matrix invariants termed “intermediate features.” In the second stage, the obtained features used to construct a pseudo-random secondary image, which then used to extract the final feature vectors, which constitute the hash value of the image. The second stage increases the robustness against attacks (e.g., compression. rotation, cropping, etc.)

On the other hand, the proposed watermarking scheme employed dither quantization for DS embedding and extraction. Hash using Gaussian Hermite Moments (GHMs) is used in B-test. The proposed algorithm compared with [7] and hashing technique used in B-test found to be more robust to content manipulations. Further, the high TPR and low FPR of hashing technique used in the B-test of the proposed algorithm demonstrate that the proposed hashing technique efficient compared to [7] in balancing the robustness and discriminating capability of the hashing scheme.

3 Watermark concealing and extraction of proposed watermarking scheme based on SF and SVD using DE-RCO

In the proposed watermarking scheme, Multiple Scaling Factors (MSFs) are computed from a randomly generated initial population using DE–RCO. The obtained MSFs $(k_1, k_2, \cdots k_d)$ are applied to embedding and extraction blocks of watermarking to obtain optimum results. The three steps which are mainly used in the algorithm are:

- Watermark embedding
- Watermark extraction
- Computation of MSFs

3.1 Watermark embedding process

In order to embed a watermark w of size d × d into a host image I of size N × N, the following steps are performed to obtain the watermarked image. The steps of watermark concealing algorithm are illustrated in Fig. 9.

3.1.1 Generation of watermarked image

In this the watermarked image is generated from the cover image and watermark using hybrid combination of SF and SVD. The optimum MSFs K_d are taken from DE-RCO.

Step 1: Decompose image I into sub images of 8 × 8 block and apply SF. The size of sub image block selection of 8 × 8 depends on the transform coding error and computational complexity [17]. After SF, the image I is segmented into unitary

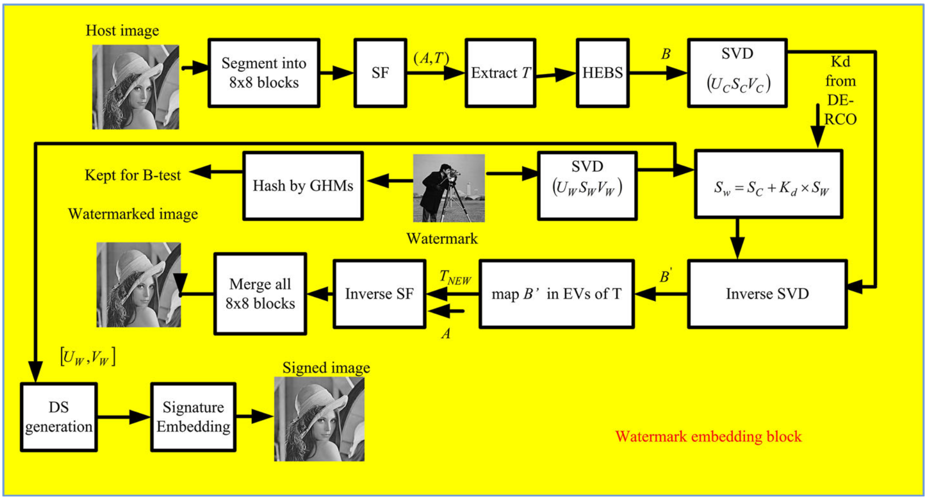


Fig. 9 Watermark embedding process

matrix A and upper triangular matrix T. The sub images size 8 × 8 block of I, A and T are given below:

$$\begin{bmatrix} I_{1,1} & I_{1,2} & \cdots & I_{1,8} \\ I_{2,1} & I_{2,2} & \cdots & I_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ I_{8,1} & I_{8,2} & \cdots & I_{8,8} \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,8} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ A_{8,1} & A_{8,2} & \cdots & A_{8,8} \end{bmatrix} \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & \cdots & \cdots & \cdots T_{1,8} \\ 0 & T_{2,2} & \cdots & \cdots & \cdots & \cdots T_{2,8} \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & T_{7,7} & T_{7,8} \\ 0 & 0 & 0 & 0 & 0 & T_{8,8} \end{bmatrix} \tag{14}$$

Step 2: The EVs in T matrix packs the most of image information. The first element of each 8 × 8 block is the highest Eigen value in T matrix which holds the highest energy of the respective block. The selection of the first element in each 8 × 8 block of the whole image is termed as High Eigen value Block Selection (HEBS). HEBS is performed on the whole host image to gather the largest EVs in each 8 × 8 block to form a matrix array B of size $\frac{\mathrm{N \times N}}{8 \times 8}$ which is equal to d × d.

$$B = \begin{bmatrix} T_{1,1} & T_{1,9} & T_{1,17} & \ldots & T_{1,N-7} \\ T_{9,1} & T_{9,9} & T_{9,17} & \ldots & T_{9,N-7} \\ T_{17,1} & T_{17,9} & T_{17,17} & \ldots & T_{17,N-7} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ T_{N-7,1} & T_{N-7,9} & T_{N-7,17} & \cdots & T_{N-7,N-7} \end{bmatrix} \tag{15}$$

Step 3: Forward SVD is applied to B which segments B into U_C, V_C and S_C of size d × d.

$$[U_C, S_C, V_C] = SVD\,(B) \tag{16}$$

Step 4: Apply forward SVD to watermark w of size d × d. It factorizes the watermark image into U_{WA}, S_{WA} and V_{WA}.

$$[U_{WA}, S_{WA}, V_{WA}] = SVD\,(w) \tag{17}$$

Step 5: The MSFs k obtained from DE-RCO is of the form shown in Fig. 10. The MSFs k_1, k_2, k_3, k_4, $\cdots$ and k_d are arranged along diagonal elements of matrix K_d as given in (19). Obtain S_w of the watermarked image by adding S_C with S_{WA} multiplied by scaling factor K_d as given in (18).

$$S_w = S_C + K_d \times S_{WA} \tag{18}$$

$$K_d = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_d \end{bmatrix} \tag{19}$$

Step 6: Inverse SVD is applied on the host image singular vectors U_C and V_C with modified S_w to obtain B'.

$$B' = U_c S_w V_c' \tag{20}$$

Step 7: Remap B' into largest EVs of T matrix denoted by T_{NEW}.

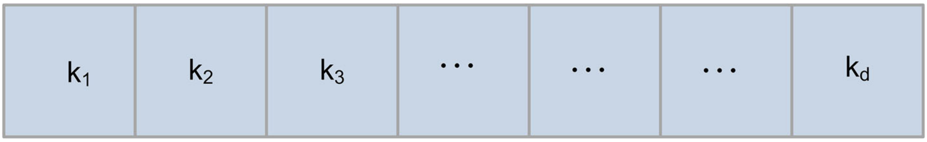


Fig. 10 MSFs from DE-RCO

Step 8: Apply inverse SF on each 8 × 8 blocks to obtain watermarked image I_{new}.

$$I_{new} = AT_{NEW}A' \tag{21}$$

3.1.2 Signature generation and signature embedding

Let a signed image formed with the dither quantization [37] by implanting an 8 bit DS in the watermarked image.

Signature generation Figure 9 shows formation of a signed image after concealing the 8-bit DS into the watermarked image. In A–test, 8-bit DS generated from the contents of singular vectors U_W and V_W of the watermark shown in Fig. 11 using the steps given below.

Step 1: Apply SVD on the watermark to extract U_{WA}, S_{WA}, and V_{WA}.
Step 2: Extract U_W and V_W. The coefficients of U_W and V_W are signed. Take the absolute value of U_W and V_W and later, round it off the values of U_W and V_W to the nearest integer.
Step 3: Reshape U_W and V_W from 2-D array to 1-D array.
Step 4: Apply SHA-1 hashing on an individual 1-D array of U_W and V_W. The size of hashed values obtained for 1-D array U_{WA} and V_{WA} are fixed-length code of 20 bytes or 160 bits.
Step 5: Perform XOR operation on a hashed 1-D array of U_W and V_W to obtain 160 bits.
Step 6: Out of 160 bits, 8 bits are selected using a seeded key.

Signature embedding Dither quantization employed to conceal the 8-bit DS in the watermarked image to obtain the signed image. The procedure for signature embedding is shown in Fig. 12.

Step 1: Segment the watermarked image into 4 × 4 blocks.
Step 2: Apply SVD on 4 × 4 blocks.

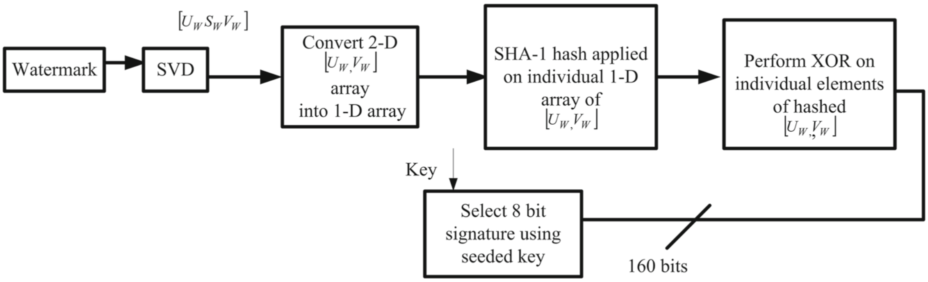

Watermark
SVD
$[U_W S_W V_W]$
Convert 2-D $[U_W, V_W]$ array into 1-D array
SHA-1 hash applied on individual 1-D array of $[U_W, V_W]$
Perform XOR on individual elements of hashed $[U_W, V_W]$
160 bits
Key
Select 8 bit signature using seeded key

Fig. 11 Signature or hash generated from the contents of watermark

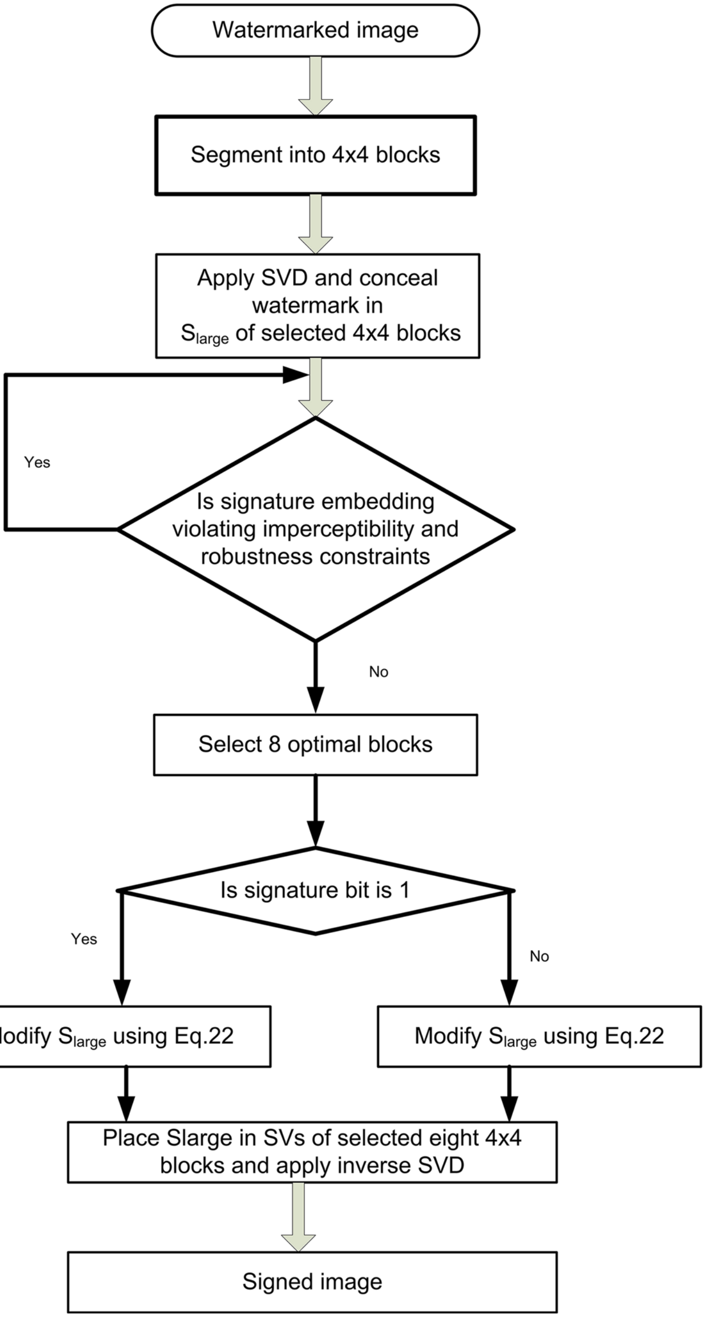

Watermarked image
Segment into 4x4 blocks
Apply SVD and conceal watermark in S_{large} of selected 4x4 blocks
Yes
Is signature embedding violating imperceptibility and robustness constraints
No
Select 8 optimal blocks
Is signature bit is 1
Yes
No
Modify S_{large} using Eq.22
Modify S_{large} using Eq.22
Place Slarge in SVs of selected eight 4x4 blocks and apply inverse SVD
Signed image

Fig. 12 Signature embedding in the proposed watermarking scheme

Step 3: Select the eight 4 × 4 squares based on satisfying the two constraints of imperceptibility and robustness. If the value of quantization step size δ is more, the robustness is more but the imperceptibility suffers. Contrary is true if the value of δ is less. The blocks which are meeting these two requirements of imperceptibility and robustness are selected for the signature insertion.

Step 4: Collect the largest singular value SV_{large} of each block.

Step 5: Compute the minimum and maximum value of SV_{large} and represent the values within the range of S_{min} to S_{max}.The range $[S_{min},\ S_{max}]$ is segregated into uniform intervals $[sl sh]$ of width δ. The intervals are taken as $[S_{min} - \delta : S_{min}]$, $[S_{min} : S_{min} + \delta] \cdots [S_{max} : S_{max} + \delta]$ as shown in Fig. 13.

Step 6: For each chosen block, identify the interval z to which the blocks largest singular value belongs, based on this modify SV_{large} as

$$SV_{large}^{mod} = \begin{cases} \frac{sl+savg}{2}\ if\ \ signature\ bit = 1 \\ \frac{sh+savg}{2}\ if\ \ signature\ bit = 0 \end{cases} \tag{22}$$

Where, $savg = \dfrac{sl + sh}{2}$ and δ is the quantization step size.

Step 7: After the amendment of SV_{large} values, inverse SVD is applied on selected blocks to obtain the signed image.

The signature image has to pass through an insecure channel where it is assaulted by signal, compression, and geometric attacks. A signature is extracted from the signed image prior to the watermark extraction for validating the A-test. The DS extracted from the signed image is compared with the DS from the user entered watermark. If both are the same, A-test is passed. If A-test is failed due to attack severity, another test B-test is conducted to free the algorithm from a false- negative problem. The extraction process is not allowed if the watermarking scheme failed in passing both tests. The extraction of watermarks allowed only if (A = 1 and B = 1) or (A = 0 and B = 1).

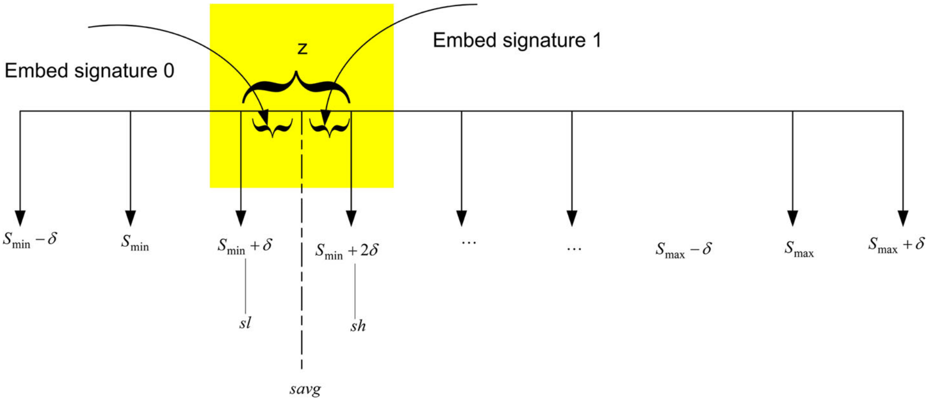


Fig. 13 Segregating S_{min} to S_{max} into quantization intervals and selection of quantization interval for the signature insertion

3.1.3 Need for B-test

The B-test is a necessary step required in the authentication framework to prevent false-negative problem. False-negative problem arises in watermarking because the digital signature is not exactly recovered due to attack severity. Though correct watermark cameraman is given, the signature is not properly recovered. So the permission for watermark extraction is not given and the user is assumed as hacker. To avoid this, hash computed using Gaussian Hermite Moments are used. Moments [16] are scalar quantities used to characterize a function and to capture its prominent features. They are "projections" of a function onto a polynomial basis. There are two types of moments: geometric and orthogonal. These two types differ in terms of stability and computational issues in the discrete domain. Generally, exponential powers are used in computation of geometrical moments. Therefore, computed values of the moment increase rapidly as the order increases for the small and large values of the exponent. On the other hand in orthogonal moments, features are captured with more precision due to the representation of moments in recurrent relation comparison with the standard powers used in geometrical moments. Gaussian Hermite Moments (GHMs) are orthogonal moments. The main advantage of these moments lies in its ability to represent zero crossings in the basis function more evenly than other moments such as Legendre and Krawtchouk moments. This feature makes it robust to rotation, translation and noise attacks [50]. Gaussian Hermite Moments M_{mn} of order (m+n) are as follows

$$M_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(p,q)\hat{H}_m\left(\frac{p}{\sigma}\right)\hat{H}_n\left(\frac{q}{\sigma}\right)dpdq \tag{23}$$

Where $I(p,q)$ is the image gray values at spatial coordinates p, q and σ is the standard deviation used in Gaussian function. GHM′s function $\hat{H}_m\left(\frac{p}{\sigma}\right)$ is obtained by

$$\hat{H}_m\left(\frac{p}{\sigma}\right) = \frac{1}{\sqrt{2^m m!\sqrt{\pi}}}e^{\frac{-p^2}{2\sigma^2}}H_m\left(\frac{p}{\sigma}\right) \tag{24}$$

Where, Hermite function $H_m(p)$ is given by

$$H_m(p) = m!\sum_{i=0}^{\frac{m}{2}}(-1)^i\frac{1}{i!(m-2i)!}(2p)^{m-2i} \tag{25}$$

For generating Hash, GHM of order n = 5 is used. The Hash length is 21 values as given in Table 1. The 21 values are rounded to nearest integer and converted into 8 bits. So 21 × 8 = 168 bits are used for hash [23] generation. The steps used for hash generation are

- Rescale all watermark images to 64 × 64 with bilinear interpolation. Later preprocess with Gaussian low pass filter to smooth noise and high-frequency content.

Table 1 GHMs of different orders

Order	GHMs	Number of moments
0	$M_{0,0}$	1
1	$M_{1,0}$, $M_{0,1}$	3
2	$M_{2,0}$, $M_{1,1}$, $M_{0,2}$	6
3	$M_{3,0}$, $M_{2,1}$, $M_{1,2}$, $M_{0,3}$	10
4	$M_{4,0}$, $M_{3,1}$, $M_{2,2}$, $M_{1,3}$, $M_{0,4}$	15
5	$M_{5,0}$, $M_{4,1}$, $M_{3,2}$, $M_{2,3}$, $M_{1,4}$, $M_{0,5}$	21

- Compute GHM's with $\sigma = 0.2$ for order 5. They yield twenty-one features. Each feature takes 8 bits. So 168 bits are used in hash generation.
- The similarity between the original watermark and the extracted watermark are computed by norm2

$$hd(h_1, h_2) = \sqrt{\sum_{i=1}^{168} h_1(i)^2 - h_2(i)^2} \tag{26}$$

where $h1$ and $h2$ are two image hashes, and 168 bits are the length of a hash vector. If hd is bigger than a pre-defined threshold Th, the corresponding images of the input hash are judged as visually distinct; otherwise, they are treated as similar images. The hash for the embedded watermark is computed in watermark embedding block and kept aside for B-test in authentication verification.

3.2 Watermark extraction process

The flow diagram of the watermark extraction process contains two blocks: An authentication block and a watermark extraction block as shown in Fig. 14. The watermark extraction process is allowed only if A-test and B-test passed. In the authentication framework, it is always not necessary that the signature transmitted along with the image. In some cases, signature can be transmitted as a separate entity [1]. However, in the works of [46, 49], the authentication code or DS is embedded in the watermarked image along with the watermark to avoid the false-positive problem. In many cases, this problem not completely resolved, and another problem arises called the false-negative problem. The false-negative problem arises when the signature is not recovered properly from the signed image due to the attack severity even when the correct watermark given at the extraction. A-test is done based on comparing the signature extracted from the signed image and the signature generated from user-entered watermark U_{WE} and V_{WE}. The procedure for the authentication is as follows.

Step 1: Extract the $DS_{original}$ from the signed image using the following steps as shown in Fig. 15.

(i) Segment the signed image into 4 × 4 blocks.

(ii) The same eight 4 × 4 squares which are used for insertion are selected for signature extraction.

(iii) Apply SVD on the chosen eight 4 × 4 blocks.

(iv) A look up table is formed with entries of $[S_{min} - \delta : S_{min}]$, $[S_{min} : S_{min} + \delta] \cdots [S_{max} : S_{max} + \delta]$.

(v) If SV_{large}^{mod} of each chosen 4 × 4 block is checked against entries in the look up table.

$$signature\ bit = \begin{cases} 1 \quad if \quad sl \leq SV_{large}^{mod} \leq \dfrac{sl + sh}{2} \\ 0 \quad if \quad \dfrac{sl + sh}{2} \leq SV_{large}^{mod} \leq sh \end{cases} \tag{27}$$

(vi) Concatenate all the signature bits to form $DS_{original}$

Step 2: For example, the user entered watermark (may be singular vectors of owner or hacker) is denoted by U_{WE} and V_{WE}. Apply signature generation procedure in Section 3.1.2 to obtain DS_O.

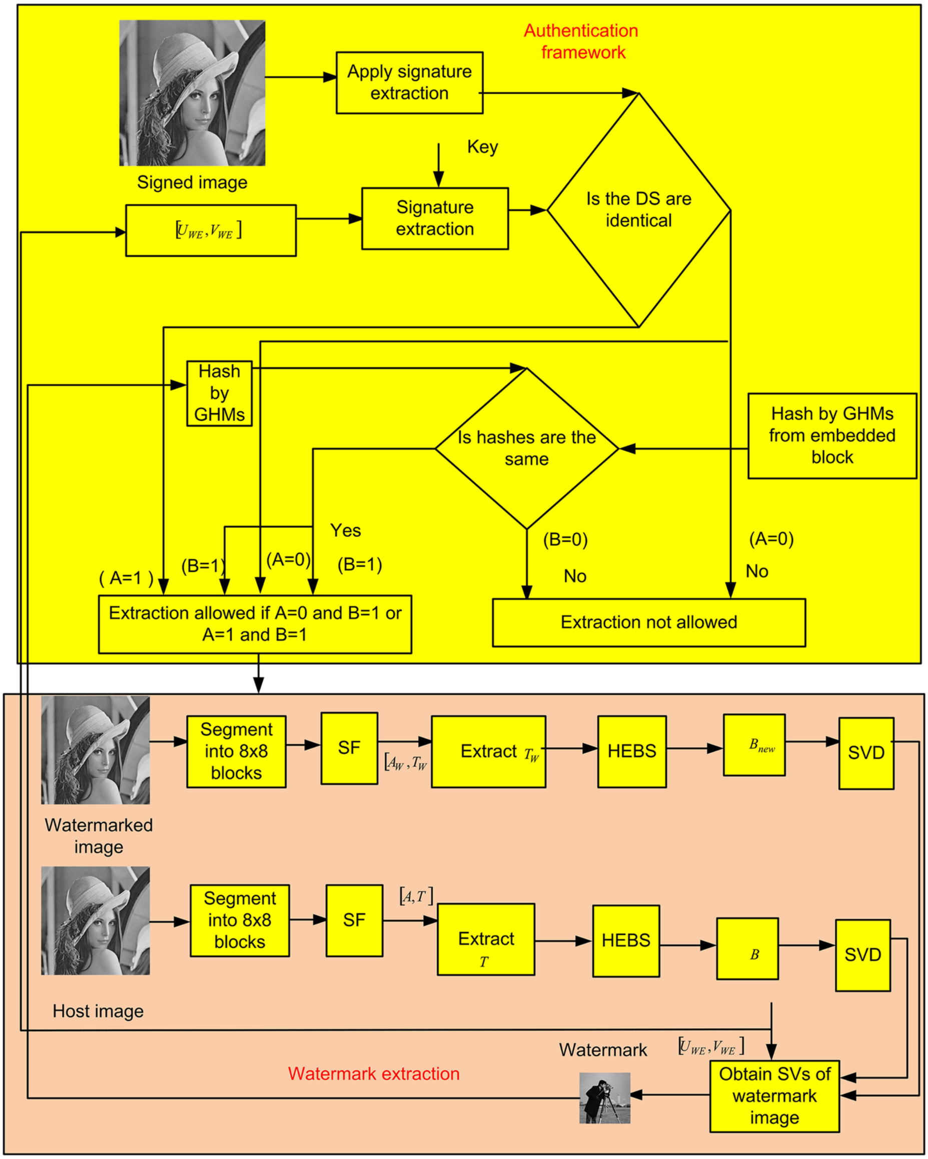


Fig. 14 Watermark extraction process

If both are $DS_{original}$ and DS_O are the same, A-test is passed and the permission for watermark extraction is granted. Otherwise, there is false negative situation which is prevented by conducting another test called the B-test.In the B-test, the hash of original watermark is compared with the hash of extracted watermark. If B = 1, permission is granted to the watermark extraction. It happens in two scenarios when (A = 0 and B = 1) and (A = 1 and B = 1).

Step 3: If the either one of the two conditions met, the watermark extraction is granted and steps used in watermark extraction are given below:

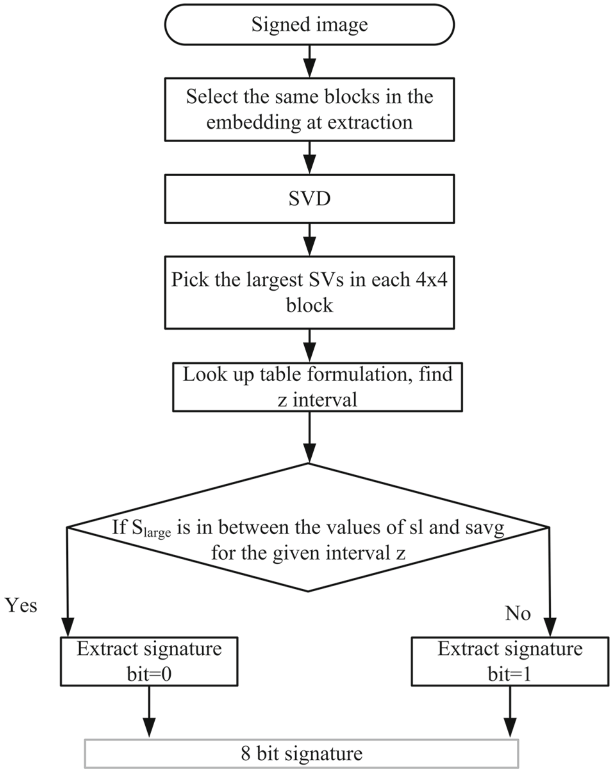

Signed image
Select the same blocks in the embedding at extraction
SVD
Pick the largest SVs in each 4x4 block
Look up table formulation, find z interval
If S$_{large}$ is in between the values of sl and savg for the given interval z
Yes
No
Extract signature bit=0
Extract signature bit=1
8 bit signature

Fig. 15 Signature extraction block

(i) Perform steps 1–2 of the watermark concealing algorithm on watermarked and host image to obtain B_{new} and B respectively.

(ii) Apply SVD on B_{new} and B to obtain SVs, S_{new} and S_C of watermarked and host image.

(iii) Obtain SVs of watermark S_{water} from S_{new} and S_C using (28)

$$S_{water} = \frac{S_{new} - S_C}{K_d} \tag{28}$$

(iv) Apply inverse SVD on U_{WA}, S_{water} and V_{WA} to obtain the extracted watermark w_{ext} using (29).

$$w_{ext} = U_{WA} S_{water} V'_{WA} \tag{29}$$

3.3 Computation of MSFs

Generally in spread spectrum watermarking, the watermarked image I_w is obtained by adding the coefficients of cover image I with watermark w multiplied by a scaling factor kx as given in (30)

$$I_w = I + kx * w \tag{30}$$

Here, kx is a single scaling factor used to control the watermark strength. If the value of kx is more, the watermarking scheme exhibits weaker transparency and, at the same time, shows more robustness to attacks. On the other hand, if kx is small, the contrary is valid, which means the image quality is better and is more vulnerable to attacks. So, when a single scaling factor is used, transparency and robustness mutually conflict with each other. Spectrum coefficients of hybrid SF-SVD exhibit different tolerance to modifications [8] and therefore, single scaling factor kx is not effective in modifying I.Therefore, MSFs are utilized in the place of a single scaling factor to optimize both transparency and robustness. The steps used for calculation of MSFs illustrated below:

3.3.1 Steps used to obtain MSFs

Step 1: Define the $Iter_{max} = 10000$, $\mu_{Cg} = 0.5$, weight control parameter $p = 0.5$ and $g = 0$.

Step 2: Generate the initial population of size m × d, where m is the size of population and d is the number of variables as shown in Fig. 1. In this array, each row is one of the possible solutions to the problem.

Step 3: In the initial population, initial solutions are randomly produced.

Step 4: In the proposed scheme, the K_d in (18) is a diagonal matrix whose elements are non zero along the diagonal, and they represent MSFs.

Step 5: A cost function defined with input taken from the randomly generated MSFs, and the output is the fitness value.

Step 16: The body of the cost function contains both the watermark embedding block, signature embedding block, signature extraction block, and the watermark extraction block.

Step 7: With the obtained value of K_d, apply steps in the watermark concealing algorithm given in Section 3.1 to obtain the watermarked image.

Step 8: Calculate the Structural Similarity Index Measure (SSIM) values between the host image and signed images using (31).

$$SSIM\,(p,q) = \frac{\left(2\mu_p\mu_q + c_1\right) + \left(2\sigma_{pq} + c_2\right)}{\left(\mu_p^2 + \mu_q^2 + c_1\right) + \left(\sigma_p^2 + \sigma_q^2 + c_2\right)} \tag{31}$$

Where,

μ_p-Mean of p.
μ_q-Mean of q.
σ_p^2-Variance of p.
σ_q^2-Variance of q.
σ_{pq}-Covariance of pq.
$c_1 = K_1L^2$- constant to avoid instability when $\mu_p^2 + \mu_q^2$ is close to zero. $K_1 = 0.01$, L = 255.
$c_2 = K_2L^2$- constant to avoid instability

when $\sigma_p^2 + \sigma_q^2$ is close to zero. $K_2 = 0.01$, L = 255.

Step 9: Apply the signal processing and compression attacks upon the watermarked image one by one and extract the watermarks from assaulted watermarked images using the extraction procedure described in Section 3.1.3. and calculate the Normalized Cross Correlation(NCC) values between the embedded and extracted watermark. The average NCC is computed for t number of attacks. When SSIM and average NCC tries to reach the 1, the cost function tries to reach the global minima. The equations required to compute NCC specified in (32).

$$NCC = \frac{\sum_{p=0}^{M-1} \sum_{q=0}^{N-1} w(p,q) w_{ext}(p,q)}{\sum_{p=0}^{M-1} \sum_{q=0}^{N-1} w(p,q)^2} \tag{32}$$

Step 10: The cost function obtained by using (34). taking the reciprocal of f_i.

$$f_i = \frac{1}{\left(\frac{\sum_{i=1}^{t} (NCC(W, W_{ext})}{t} \right) - SSIM(I, I_w)} \tag{33}$$

Where, t used in (33) is the number of attacks applied on watermarked image, w and w_{ext} are the original and extracted watermarks.

$$Costfunction = \frac{1}{f_i} \tag{34}$$

Step 11: The main loop of DE-RCO discussed in Algorithm 1 of [4] is used to perform a mutation, RCO, and selection.

Step 12: In selection, the cost function in (34) is computed for the target vector $X_{i,g}$ and trial vector $Z_{i,g}$. If the cost function of $Z_{i,g}$ is less than the cost function of $X_{i,g}$ then the solution in the subsequent generation $X_{i,g+1}$ is $Z_{i,g}$, otherwise it is $X_{i,g}$.

Step 13: Termination criterion is applied when the algorithm completes a maximum number of generations or when there is no change in the rate of the cost function (convergence reached). The convergence criterion $CV_{\underset{\approx}{\succ},g}$ is taken based on [53] and given by

$$CV_{\underset{\approx}{\succ},g} = \frac{s_g}{\overline{OBJ_g}} \tag{35}$$

$CV_{\underset{\approx}{\succ},g}$ gives convergence property of DE-RCO algorithm. Here s_g is standard deviation for the m cost function values of population and $\overline{OBJ_g}$ is the mean of all cost functions at the generation g. Termination criterion is reached if $CV_{\underset{\approx}{\succ},g}$ is less than τ which is a very low value and must be taken less than 10^{-6}.

Step 14: Steps 3–12 are repeated until the termination criterion not met. Consider, the MSFs, which give the minimum cost function in the final generation, is utilized for watermark concealing and extraction blocks.

The spatial resolution of cover image and watermark used in the proposed algorithm are N × N and d × d . The algorithm starts with segmenting the host image I into 8 × 8 blocks. Later, SF is employed to decompose the host image I into A and T. In T, the largest EVs from each 8×8 block are collected and formed into an array B of size $\frac{\mathrm{N} \times \mathrm{N}}{8 \times 8} = \mathrm{d} \times \mathrm{d}$. When SVD applied to matrix array B, we obtain SVs of cover image as d number of coefficients. The same number of d coefficients obtained when SVD applied to watermark. The SVs of watermark are multiplied by the diagonal of MSFs K_d obtained from DE-RCO. In DE-RCO, MSFs are initially produced randomly and taken as a string where K_d is the diagonal matrix of MSFs. Strings generally represent potential solutions, so DE-RCO searches from a population of potential solutions but not using a single solution. If m is the number of the initial population, m number of strings with size d are produced randomly. These strings are MSFs, and diagonal of MSFs applied one after the other to both watermark embedding and extraction blocks in the current generation. In the watermark concealing algorithm, SSIM computed between the host and signed images. Similarly, in the watermark extraction algorithm for t number of attacks, t number of NCC values are obtained. Later average NCC has computed for t attacks. Finally, with the obtained values of SSIM and average NCC, the fitness value is computed. The algorithm chooses randomly three fit individuals and carries the sequence of mutation, RCO, and selection, as described in Section 2.4 to generate a new population of strings from previous ones for the next generation. The procedure repeated when the maximum number of generations reached or satisfies convergence criteria given in (35). The the fittest string that represents in the final generation is called MSFs.

4 Experimental results

In this work, 200 openly available images on the Internet are used as host images. Because of limited space, we confine our experimentation on a few grayscale and medical images. Some of them used in experimentation are (a) Bridge (b) Coco (c) Einstein (d) Flintstones (e) Gold hill (f) Barbara (g) House (h) Mandrill (i) Axial MR-Flair (j) Axial MR-T2 weighted (k) Sagittal–CT– non contrast (l) CT- contrast (m) Airplane (n) Couple (o) Lena and (p) pirate is as shown in Fig. 16. Cameraman is used as watermark.The host images used in the proposed method are of same size 512 × 512 as shown in Fig. 16. The watermark

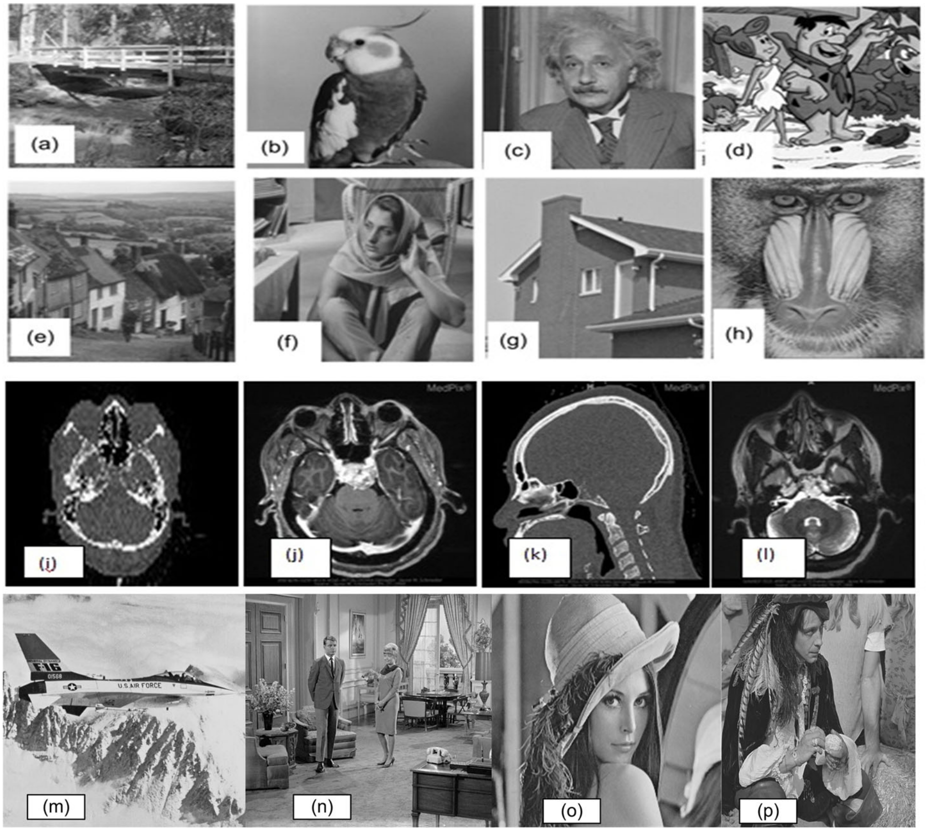


Fig. 16 Host images used in experimentation

size is 64 × 64. So d used in the experimentation is 64. The proposed work is carried in Matlab 2013A on Pentium 5 processor with 4 GB RAM on Windows 11 operating system. The parameter settings of DE are CR = 0.5, F = 0.5, $m = 50$ and $g = 200$.

4.1 Transparency

For evaluation of this parameter in proposed watermarking scheme, subjective assessment based on Double Stimulus Continuous Quality Scale (DSCQE) test and objective assessment based on PSNR is done. In DSCQE test, 200 participants were asked to grade the quality of watermarked image on a continuous scale of five grades as given in [25] by showing both cover and watermarked image concurrently to notice any type of degradation seen in the watermarked image. The scores are given by human observers to obtain the Mean Opinion Score (MOS) for analyzing the quality of signed image. Figure 17 shows the MOS of proposed algorithm compared with the MOS of some other optimization algorithms—[2, 4] and [9] in terms of MSFs of K_1, K_2, K_3 of each size d obtained at different runs of the optimization algorithm. Outcome of MOS plot shows that the proposed method is highly imperceptible and also the degradation is unnoticeable to naked eye. The watermarked images of proposed algorithm are shown in Fig. 18. It is further validated by taking the error image between host and watermarked image Barbara shown in Fig. 19. PSNR between original and watermarked image is computed as given in (36).

$$PSNR = 20log_{10}\frac{255}{MSE} \tag{36}$$

Where, MSE is the Mean Square Error between host and watermarked image. Fig. 20 shows comparison of the PSNR between watermarked and host images of Bridge, Coco, Flintstone, Goldhill, Barbara, Horse, MRI, leg fracture and Mandrill with proposed method and other optimization algorithms—[7, 9, 30] and [2] is carried out and the high PSNR of the proposed algorithm emphasizes that the method is highly imperceptible. Though watermark image is part of the watermarked image, the degradation is not noticeable in signed images

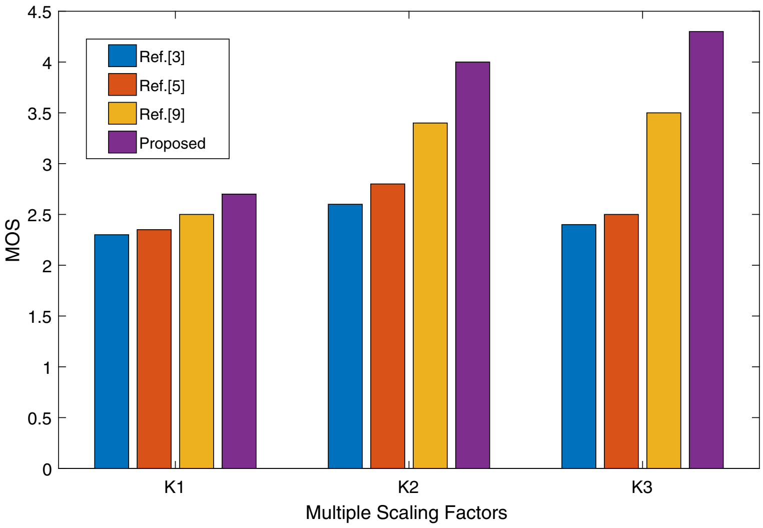


Fig. 17 Comparison of MOS of host images in proposed algorithm with Ref. [2], Ref. [4] and Ref. [9]

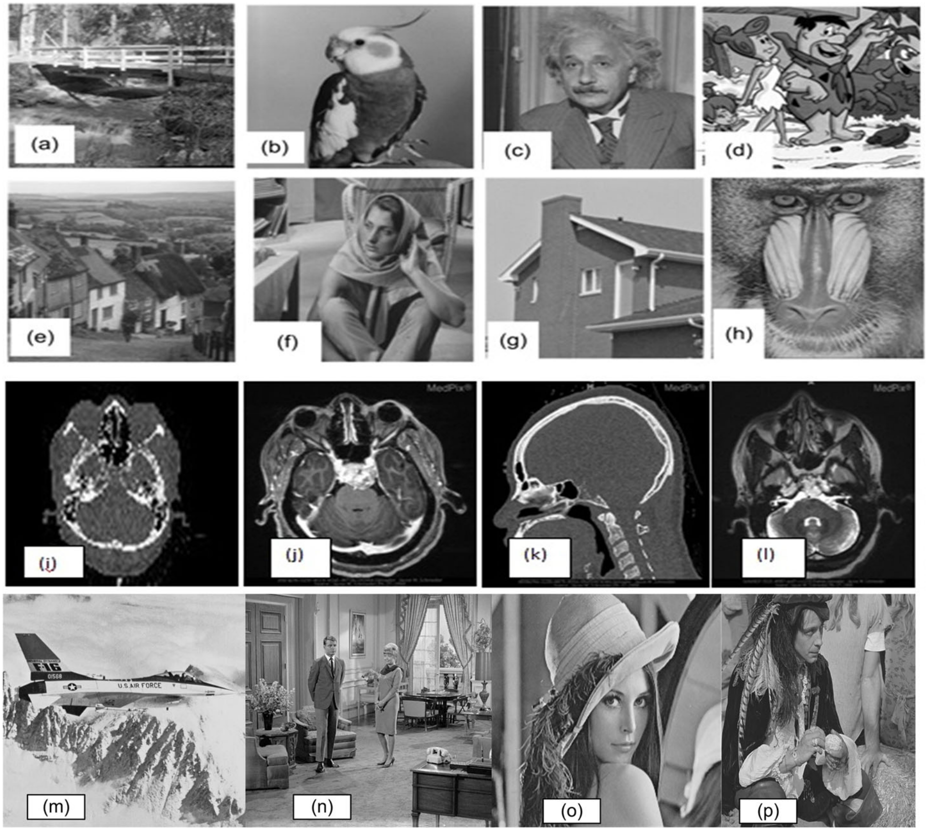


Fig. 18 Watermarked images obtained with DE-RCO

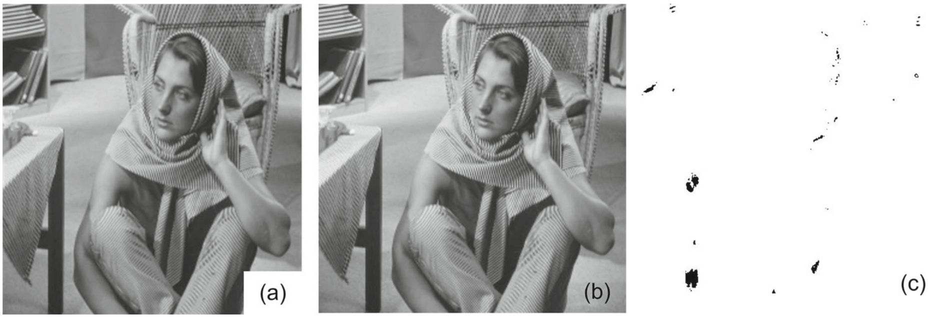


Fig. 19 **a** Host image Barbara **b** Watermarked image Barbara **c** Error image between host and watermarked image

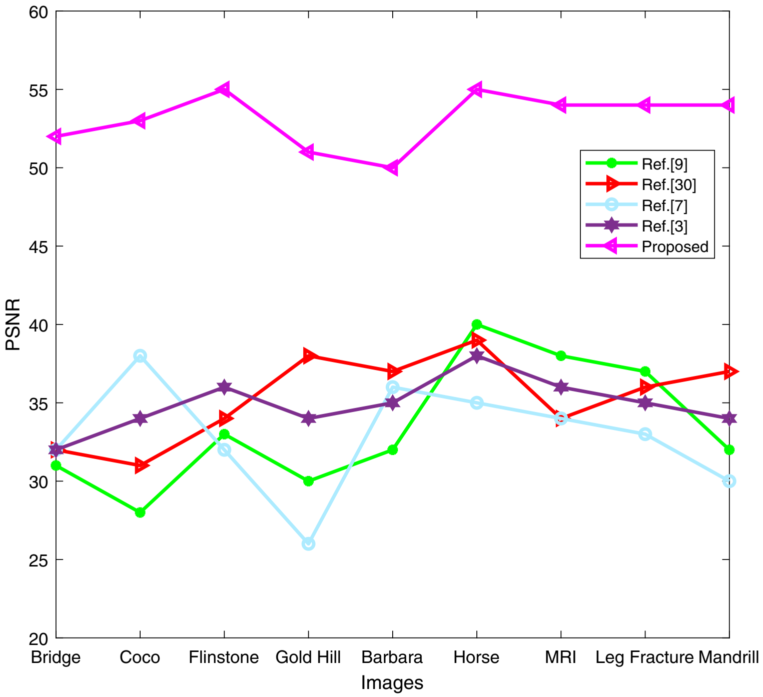


Fig. 20 Comparison of PSNR of host images in proposed algorithm with [7, 30] and [2] in terms of PSNR under no attack

of Fig. 18 which does not contain any visible artifacts. The average PSNR of the proposed watermarking scheme is 56.6 dB under no attack due to single bit insertion of 8 × 8 blocks.

4.2 Robustness assessment

We have employed several experiments to examine the effectiveness of the robustness of the proposed scheme. The attacks we employed in the proposed algorithm is the same attacks that are employed in the [7]. For fair comparison with [7], the same host images Lena and Medical sample which are used in [7] are employed in the proposed algorithm. Further, the same watermark cameraman which is used in [7] employed in the proposed algorithm. The main drawback of [7] is that they do not employed any type of optimization. The main requirement of any invisible watermarking is maintaining the watermark invisibility plus robustness to attacks. In [7], the watermark imperceptibility and robustness are controlled by single scaling factor kx. If $kx = 5$ in [7], the watermark is invisible and no blocking artifacts are seen as shown in Fig. 21. If kx is increased to 10, visible blocking artifacts begin to appear in the watermarked image of [7]. When $kx = 5$, the robustness is very poor and the NCC lies in between 0.32 to 0.399 for all the attacks subjected to the watermarked image Lena. If kx is increased to 3000, the watermarked image is completely distorted and the NCC between the original and extracted watermarks increased from 0.991 to 0.999 as shown in Fig. 21. The watermarked Lena and medical image sample subjected with various attacks are shown in Figs. 22 and 23. The watermarked Lena image subjected with attacks such as (a) Average filtering with neighborhood 3 × 3 (b) Median filtering with

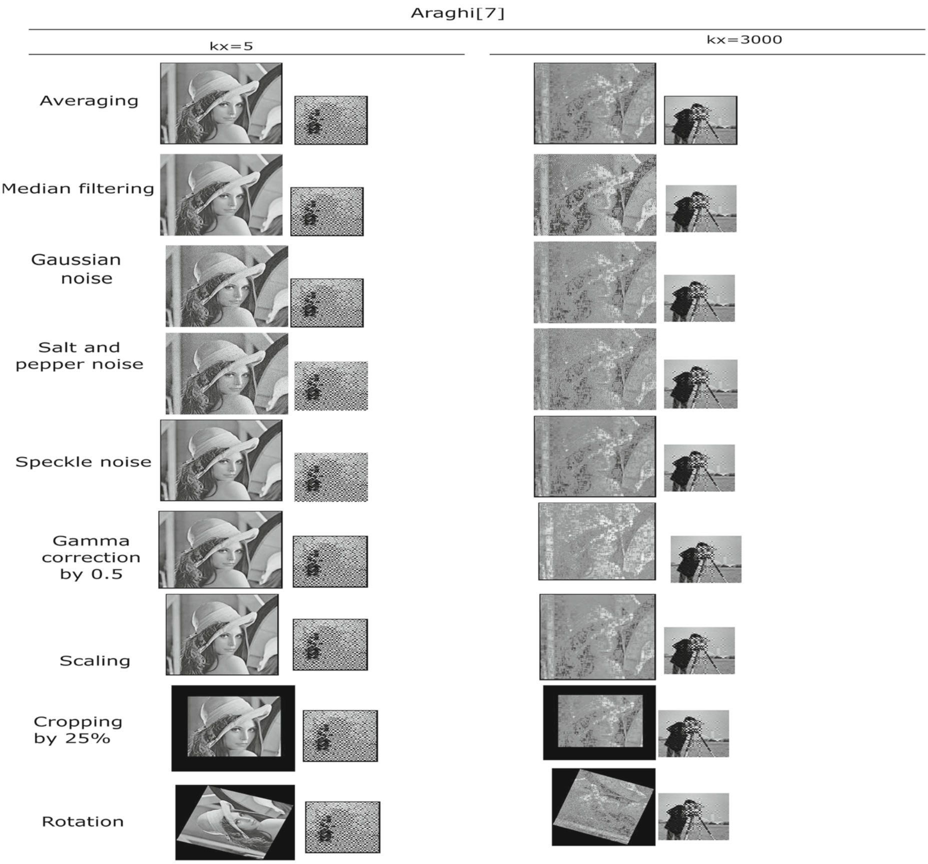


Fig. 21 Effect of single scaling factor kx on imperceptibility and robustness in [7]

neighbor615 hood 3 × 3 (c) Gaussian noise with standard deviation 0.01 (d) Salt and Pepper noise with 6160.01, (e) Speckle noise with density 0.01, (f) Gamma correction by 0.5 (g) Scaling 512 → 4096 → 512 (h) Quarter Cropping (i) Rotation by 20° (j) Rotation by 50 (k) Rotation by 70° (l) Histogram Equalization.

The medical image sample is subjected with more severe attacks such as (a) Average filtering with neighbourhood 5 × 5 (b) Median filtering with neighbourhood 5 × 5 (c) Gaussian noise with standard deviation 0.05 (d) Salt and Pepper noise with 0.05, (e) Speckle 622 noise with density 0.05, (f) Gamma correction by 0.5 (g) Scaling 512 → 4096 → 512 (h) Cropping 110 rows and 383 columns (i) Rotation by 5° (j) Rotation by 70, (k) Rotation by 140, (l) Histogram Equalization. The attacks we employed in the proposed algorithm is the same attacks that are employed in the [7]. For fair comparison with [7], the same . So in order to maintain the imperceptible watermarking, the NCC of extracted watermarks in [7] must be taken in between 0.32 to 0.399. Therefore, in the work of [7], NCC for the extracted watermark is far below 0.7 and it is too low to be accepted. On the other hand, the results of proposed watermarking with DE-RCO are promising as the range of NCC values is above 0.7. The extracted watermarks from watermarked Lena and medical sample, the NCC values between the original and extracted watermarks are shown in Figs. 24 and 25. In most

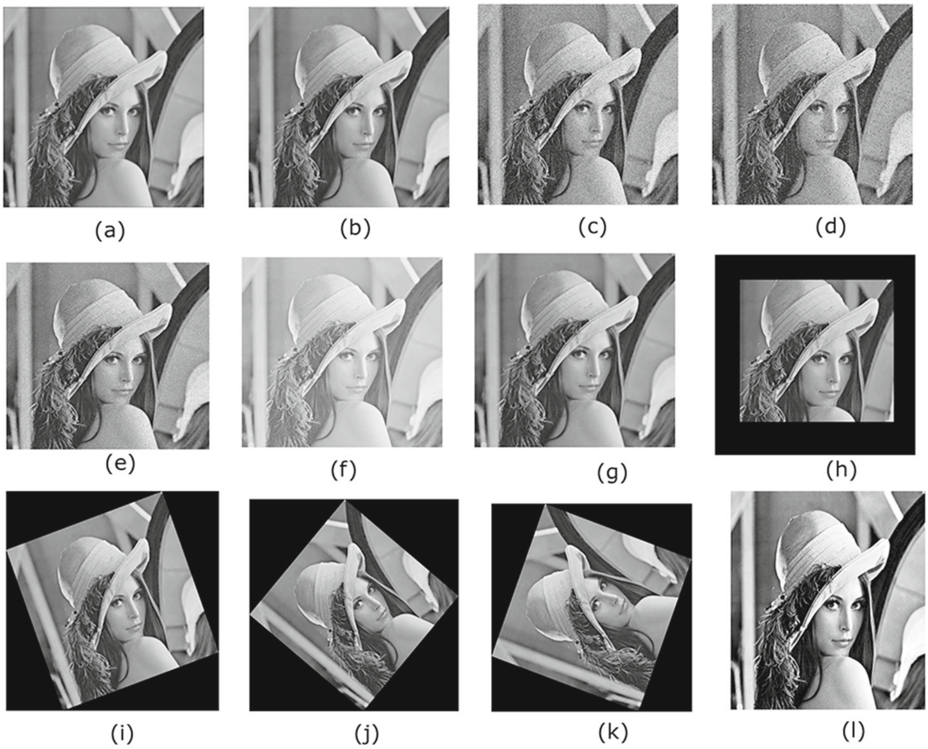


Fig. 22 Attacks applied on watermarked Lena **a** Averaging filtering with neighborhood 3×3 **b** Median filtering with neighborhood 3×3 **c** Gaussian noise with standard deviation 0.01 **d** Salt and Pepper noise with 0.01, **e** Speckle noise with density 0.01, **f** Gamma correction by 0.5 **g** Scaling $512 \rightarrow 4096 \rightarrow 512$ **h** Quarter Cropping **i** Rotation by 20° **j** Rotation by 50°, **k** Rotation by 70°, **l** Histogram Equalization

of the attacks, value of NCC is approaching to 1. It signifies the superiority of proposed algorithm in terms of robustness.

The robustness of the proposed scheme against checkmark attacks such as wiener filter soft thresholding, Template removal, bit plane removal, row and column copying and row column blanking are shown in Table 2. The high NCC value of the proposed algorithm proves that the method is highly robust compared to [2, 4] and [9].

4.3 Authentication tests and its validation

in terms of A-test and B-test under no attack shown in Table 3. The first row in Table 3 uses rose as a counterfeit watermark. Hence, both the A-test and B-test failed. The authentication framework does not allow the extraction process as the watermark is fake and used as a rose. In the second, third, and fourth row of Table 3 , the original watermark cameraman used. Therefore both watermarking schemes passed the authentication tests of A and B. Therefore, extraction of the watermark is allowed. The authentication test A assumes passed when the retrieved bits are greater than or equal to 5. In [7] the A-test failed except for minor attacks such as Averaging, Median filtering and Gamma correction as shown in Table 4. On the other hand as shown in Table 4 the proposed watermarking scheme passes the A-test except for attacks with large rotation angles when the correct watermark given at the detection side.

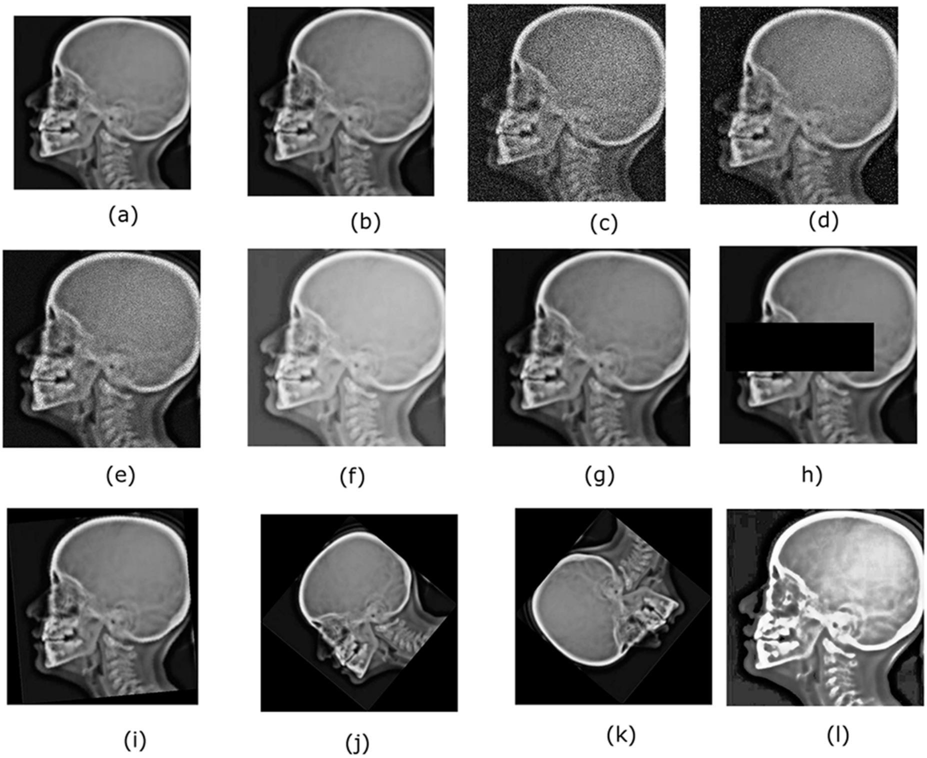


Fig. 23 Attacks applied on watermarked Medical sample **a** Averaging filtering with neighborhood 3 × 3 **b** Median filtering with neighborhood 3 × 3 **c** Gaussian noise with standard deviation 0.05 **d** Salt and Pepper noise with 0.05, **e** Speckle noise with density 0.05, **f** Gamma correction by 0.5 **g** Scaling 512 → 4096 → 512 **h** Cropping 110 rows and 383 columns **i** Rotation by 5° **j** Rotation by 70°, **k** Rotation by 140°, **l** Histogram Equalization

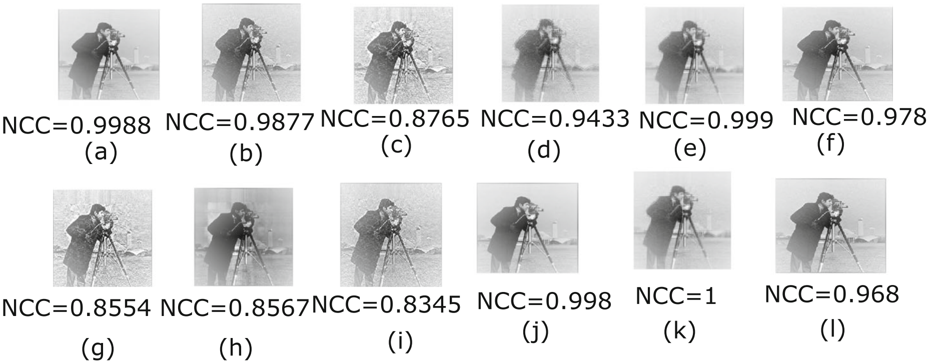


Fig. 24 Extracted watermarks from watermarked lena attacked by **a** Average filtering with neighborhood 3 × 3 **b** Median filtering with neighborhood 3 × 3 **c** Gaussian noise with standard deviation 0.01 **d** Salt and Pepper noise with 0.01 **e** Speckle noise with density 0.01 **f** Gamma correction by 0.5 **g** Scaling 512 → 4096 → 512 **h** Quarter Cropping **i** Rotation by 20° **j** Rotation by 50° **k** Rotation by 70° **l** Histogram Equalization

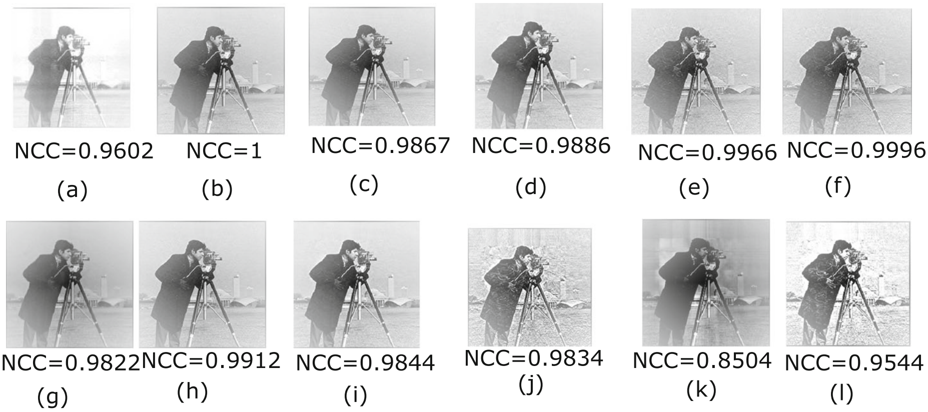


Fig. 25 Extracted watermarks from watermarked medical image sample attacked by a Average filtering with neighborhood 5 × 5 b Median filtering with neighborhood 5 × 5 **c** Gaussian noise with standard deviation 0.05 **d** Salt and Pepper noise with 0.05 **e** Speckle noise with density 0.05 **f** Gamma correction by 0.5 **g** Scaling 512 → 4096 → 512 **h** Cropping 110 rows and 383 columns **i** Rotation by 5° **j** Rotation by 70°, **k** Rotation by 140°, **l** Histogram Equalization

The SVD-SVD hashing used in the B-test [7] is robust to rotation at the cost of a significant decrease in discrimination which means distinct images produces identical hash which is not desirable in the authentication. Therefore, The B-test in [7] failed except for the minor attacks such as average filtering and median filtering as shown in Table 4. In the proposed algorithm, the B-test is passed for all the different attacks as shown in Table 4 .

The parameters used for the evaluation of hashes based on True Positive Rate (TPR) and False Positive Rate (FPR). TPR is the ratio of identical images considered as similar images to the total number of pairs of similar images. FPR is the ratio of distinct images considered as similar images to the total number of pairs of different images. TPR and FPR reflect the perpetual robustness and the discriminating capability of the hashing scheme. If we compare two algorithms with the same FPR, the scheme with a high TPR outperforms the one with a small value. Similarly, if they have the same TPR, the hashing with a low FPR is better than that with high value. We choose thresholds for the proposed hashing and SVD-SVD hashing of [7], respectively, and calculate their TPR and FPR. The used thresholds for the proposed hashing are: 2, 5, 10, 20, 30, 40, 50, and those for the SVD-SVD hashing [7] are: 0.1, 0.2, 0.3, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8 and 0.9. When the FPR is nearer to 0, the TPR

Table 2 Comparison of NCC values of proposed algorithm with Ref. [2], Ref. [4] and Ref. [9] for different checkmark attacks

Checkmark attacks	NCC of proposed method	NCC of Ref. [2]	NCC of Ref. [4]	NCC of Ref. [9]
Wiener Filter	0.8233	0.8678	0.8447	0.7654
Soft threshholding	0.733	0.725	0.6993	0.745
Template removal	0.7954	0.7012	0.7689	0.7213
Bit plane removal	0.965	0.9100	0.9345	1
Row and column copying	0.9876	0.9244	0.9201	0.8466
Row column blanking	0.911	0.8333	0.8567	0.8582
Trimmed Mean Alpha	0.8434	0.7533	0.8234	0.7648

Table 3 Comparative analysis of proposed scheme with Ref. [7] in terms of A-test and B-test

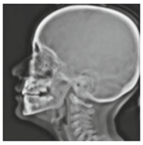
Watermarked image	Watermark	Signature bits	Proposed		Ref. [7]	
			A-Test	B-test	A-test	B-test
		8 bits	×	×	×	×
		8 bits	✓	✓	✓	✓
		8 bits	✓	✓	✓	✓
		8 bits	✓	✓	✓	✓

Table 4 Attack details and authentication results of normal image Lena

Index	Attack	Parameters	Proposed			Ref. [7]		
			Signature bits	A-test	B-test	Signature bits	A-test	B-test
1	Average filter	Filter size 3 × 3	8	A = 1	B = 1	5	A = 1	B = 1
2	Median filter	Filter size 3 × 3	5	A = 1	B = 1	6	A = 1	B = 1
3	Gaussian noise	0.01	4	A = 0	B = 1	4	A = 0	B = 0
4	Salt and pepper noise	0.01	5	A = 1	B = 1	4	A = 0	B = 0
5	Speckle noise	0.01	6	A = 1	B = 1	5	A = 0	B = 0
6	Gamma correction	0.5	6	A = 1	B = 1	5	A = 1	B = 0
7	Scaling	0.5	6	A = 1	B = 1	4	A = 0	B = 0
8	Cropping	25%	5	A = 1	B = 1	4	A = 0	B = 0
9	Rotation	20°	6	A = 1	B = 1	4	A = 0	B = 0
10	Rotation	50°	4	A = 0	B = 1	4	A = 0	B = 0
11	Rotation	70°	4	A = 0	B = 1	4	A = 0	B = 0
12	Histogram Equalization	–	5	A = 1	B = 1	4	A = 0	B = 0

Table 5 Comparison of the proposed method with Ref. [7] in terms of capacity

Method	Capacity
Araghi [7]	32 × 32 + 8 bit DS
Priopposed method	64 × 64 + 8 bit DS

of the proposed hashing is 0.988, while that of the SVD–SVD hashing employed is about 0.11. Similarly, when the TPR reaches 1.0, the optimal FPR of the proposed hashing is about 0.068, and that of the SVD-SVD hashing is approximately 0.95. These results evident that B-test used in [7]. is not discriminating and not suited better for the authentication. As shown in Table 4, the authentication test B failed in [7] to all the attacks except for the minor attacks such as average and median filtering. GHMs used in the proposed algorithm is stable against rotation, translation, and noise attacks. So the proposed algorithm passes the authentication test for all the chosen attacks without losing the discrimination.

4.4 Capacity assessment

Capacity refers to the amount of information that can hide in the cover medium. It depends on the size of watermark and DS.

Table 5 shows the comparison of the proposed watermarking scheme with [7]. The proposed watermarking scheme capacity is 4104 bits, whereas the capacity of [7] is only 1032 bits.

4.5 Computation time assessment

The computational time of the proposed algorithm compared with the recent works in terms of time required for watermark embedding, watermark extraction and computation of MSFs as shown in Table 6. The watermark embedding with proposed method excluding signature generation and embedding block takes a time of 0.844755 s. If signature generation and embedding blocks included for the watermark embedding, the time taken for it increased to 0.9432 s. Similarly, for watermark extraction using the proposed method when signature extraction and B-test excluded is 0.141584 s. If signature extraction and B-test with GHMs considered, it increases to 0.14378 s. On the other hand for the same conditions, DCT and SVD by [2] requires 2.197 s for watermark embedding and 0.657322 s for watermark extraction, whereas RDWT and SVD by [49] requires 2.83538 s for watermark embedding and 0.5877322 s for watermark extraction. GA [8], DE with Binomial

Table 6 Comparison of computational time of proposed method with [2] and [49]

	Embedding time in sec		Extraction time in sec	
	Including signature generation	Excluding signature generation	Including signature generation	Excluding signature generation
Proposed method	0.9432	0.844755	0.14378	0.141584
Ref. [2]	–	2.197	–	0.657322
Ref. [49]	–	2.83538	–	0.5877322

Table 7 Comparison of computational time required for calculating MSFs of proposed algorithm with [2] and [8]

Optimization algorithm	Number of variables	Control parameters	Size of initial population	Number of generations	Convergence time
GA [8]	64	Cr = 0.7, MR = 0.004	150	400	180.22
DE with Binomial Cross over [2]	64	F = 0.5, CR = 0.5	50	200	30.22
Proposed method	64	F = 0.4, CR = adaptive	50	200	36.44

Cross over [2] and proposed DE-RCO requires 180.22, 30.22 and 36.44 minutes respectively for computation of MSFs shown in Table 7. By this, the computational time of GA and DE-RCO is more than DE with binomial crossover. In the case of DE with a binomial crossover as there is no change in the objective function at local minima, the algorithm stops. The watermarking scheme struck in local minima shows less transparency and robustness compared to the watermarking scheme struck in global minima. On the other hand, the proposed watermarking scheme with DE-RCO reaches global minima due to its advantages of employing an adaptive crossover operator. All possible trial vectors can generate by recombining the mutant vector and target vector in 2^d search space. Thus the algorithm has more transparency and robustness than DE with binomial crossover though it requires more computational time than DE with binomial crossover. The other optimization algorithm GA [8] requires high computational time. The time required for watermark embedding and extraction of the proposed algorithm is less than the DCT–SVD [2] and RDWT-SVD [49] based watermarking schemes. Therefore the proposed DE-RCO employing a hybrid combination of SF and SVD algorithm is highly efficient and secure due to the implementation of a two-level authentication framework comprising A-test and B-test and, therefore, can be used as an alternative technique in watermarking.

5 Conclusion and future work

In this work, an improved watermarking scheme based on SF and SVD is designed and simulated. The false-positive problem and false-negative problem prevented by employing a 2 level authentication framework consist of A-test and B-test in such a way that only the authenticated user can extract the watermark. The experimental results demonstrate that the proposed hashing technique used in B-test is robust to content manipulation. It also shows a better balance between robustness and discrimination by having high TPR and low FPR. Further, to test and evaluate imperceptibility of the proposed scheme, 200 samples of medical and standard benchmark images were selected. The proposed watermarking scheme checked in terms of subjective and objective criteria. Experimental results show high transparency with the average PSNR of 56.7850 dB for the proposed work, and robustness increased due to the implementation of optimization algorithm DE-RCO. The time required for the watermark, signature embedding, and the time needed for watermark extraction and the computation of MSFs for the proposed watermarking algorithm is less than DCT-SVD and RDWT-SVD. So the proposed algorithm can be used as an alternative to DCT-SVD and RDWT-SVD. Watermarking has a broad spectrum of applications such as medical watermarking in Tele-Medicine, 3D model watermarking, watermarking in cloud computing, multi-core environment, bio-metric watermarking, watermarking using a mobile device for securing online social networks contents. There are broad challenges in watermarking such as obtaining optimum trade-off between robustness, imperceptibility, capacity, security, and computational complexity simultaneously. The techniques that need to be focussed soon to realize the challenges in watermarking are scalable techniques for visual contents, Multi-layer watermarking methods to reduce computational time, and Quantum watermarking techniques to enhance transparency.

References

1. Ahmed F, Siyal MY, Abbas VU (2010) A secure and robust hash-based scheme for image authentication. Signal Process 90(5):1456–1470
2. Ali M, Ahn CW, Pant M (2014) A robust image watermarking technique using SVD and Differential Evolution in DCT domain. Optik-Int J Light Electron Opt 125(1):428–434
3. Ali M, Ahn CW, Pant M, Siarry P (2015) An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony. Inf Sci 301:44–60
4. Ali M (2018) An optimal image watermarking approach through cuckoo search algorithm in wavelet domain. Int J Syst Assur Eng Manag 9(3):602–611
5. Al-Haj AM (ed.) (2010) Advanced techniques in multimedia watermarking: image, video and audio applications: image, video and audio applications. IGI Global
6. Ansari IA, Pant M (2017) Multipurpose image watermarking in the domain of DWT based on SVD and ABC. Pattern Recognit Lett 94:228–236
7. Araghi TK, Manaf AA, Araghi SK (2018) A secure blind discrete wavelet transform based watermarking scheme using two-level singular value decomposition. Expert Syst Appl 112:208–228
8. Aslantas V (2008) A singular-value decomposition-based image watermarking using genetic algorithm. AEU-Int J Electron Commun 62(5):386–394
9. Aslantas V, Dogan AL, Ozturk S (2008) DWT-SVD based image watermarking using particle swarm optimizer. In: 2008 IEEE International conference on multimedia and expo. IEEE, pp 241–244
10. Aung A, Ng BP, Rahardja S (2011) A robust watermarking scheme using sequency-ordered complex Hadamard transform. J Signal Process Syst 64(3):319–333
11. Bamal R, Kasana SS (2019) Dual hybrid medical watermarking using walsh-slantlet transform. Multimedia Tools Appl 78(13):17899–17927
12. Berger CE, de Koeijer JA, Glas W, Madhuizen HT (2006) Color separation in forensic image processing. J Forensic Sci 51(1):100–102
13. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2007) Digital watermarking and steganography. Morgan Kaufmann, San Mateo
14. Delp EJ III, Wong PW (2003) Security and watermarking of multimedia contents V. In: Security and watermarking of multimedia contents V, vol 5020
15. Deng LB, Wang S, Qiao LY (2017) DE-RCO: rotating crossover operator with multiangle searching strategy for adaptive differential evolution. IEEE Access 6:2970–2983
16. Flusser J, Zitova B, Suk T (2009) Moments and moment invariants in pattern recognition. Wiley, New York
17. Gonzales RC, Woods RE (1992) Digital image processing. Reading, Mass: Addison-Wesley, 518–524
18. Grycuk R, Gabryel M, Nowicki R, Scherer R (2016) Content-based image retrieval optimization by Differential Evolution. In: 2016 IEEE congress on evolutionary computation (CEC). IEEE, pp 86–93
19. Henderson N, Sacco WF, Barufatti NE, Ali MM (2010) Calculation of critical points of thermodynamic mixtures with Differential Evolution algorithms. Ind Eng Chem Res 49(4):1872–1882
20. Hill DL, Batchelor PG, Holden M, Hawkes DJ (2001) Medical image registration. Phys Med Biol 46(3):R1
21. Horng SJ, Rosiyadi D, Fan P, Wang X, Khan MK, Pan Y (2011) An efficient copyright protection scheme for e-government document images. In: IEEE MultiMedia
22. Horng SJ, Rosiyadi D, Fan P, Wang X, Khan MK (2014) An adaptive watermarking scheme for e-government document images. Multimed Tools Appl 72(3):3085–3103
23. Hosny KM, Khedr YM, Khedr WI, Mohamed ER (2018) Robust image hashing using exact Gaussian–Hermite moments. IET Image Process 12(12):2178–2185

24. Hsu LY, Hu HT, Chang YH (2018) An improvement of embedding color watermarks in color images based on schur decomposition. In: 2018 41st international conference on telecommunications and signal processing (TSP). IEEE, pp 1–5
25. ITU R (2002) Methodology for the subjective assessment of the quality of television pictures. Recommendation ITU-R BT. 500-11
26. Jayalakshmi M, Merchant SN, Desai UB (2006) Digital watermarking in contourlet domain. In: 18th International conference on pattern recognition (ICPR’06), vol 3. IEEE, pp 861–864
27. Kozat SS, Venkatesan R, Mihçak MK (2004) Robust perceptual image hashing via matrix invariants. In: 2004 International conference on image processing, 2004. ICIP’04, vol 5. IEEE
28. Lin WH, Horng SJ, Kao TW, Fan P, Lee CL, Pan Y (2008) An efficient watermarking method based on significant difference of wavelet coefficient quantization. IEEE Trans Multimed 10(5):746–757
29. Lin WH, Horng SJ, Kao TW, Chen RJ, Chen YH, Lee CL, Terano T (2009) Image copyright protection with forward error correction. Expert Syst Appl 36(9):11888–11894
30. Loukhaoukha K (2013) Image watermarking algorithm based on multi-objective ant colony optimization and singular value decomposition in wavelet domain. J Optim 2013:1–10
31. Meenakshi K, Bethel GB (2014) Design and simulation of Constant bit rate compressor using fuzzy logic. In: 2014 First international conference on networks & soft computing (ICNSC2014). IEEE, pp 309–313
32. Meenakshi K, Rao CS, Prasad KS (2014) A robust watermarking scheme based Walsh-Hadamard transform and SVD using ZIG ZAG scanning. In: 2014 International conference on information technology. IEEE, pp 167–172
33. Meenakshi K, Srinivasa Rao C, Satya Prasad K (2014) A scene based video watermarking using slant transform. IETE J Res 60(4):276–287
34. Meenakshi K, Prasad KS, Rao CS (2017) Development of low-complexity video watermarking with conjugate symmetric sequency–complex Hadamard transform. IEEE Commun Lett 21(8):1779–1782
35. Meenakshi K, Swaraja K, Kora P, Ch UK (2019) Texture feature based oblivious watermarking with slant transform using fuzzy logic. In: 2019 IEEE 5th international conference for convergence in technology (I2CT). IEEE, pp 1–5
36. Meenakshi K, Padmavathi Kora DK (2019) Video watermarking with curvelet transform. Int J Innov Technol Explor Eng (IJITEE), ISSN 2278–3075
37. Mohan BC, Srinivaskumar S, Chatterji BN (2008) A robust digital image watermarking scheme using singular value decomposition (SVD), dither quantization and edge detection. J ICGST-GVIP 8:17–23
38. Monga V (2005) Perpetually based methods for robust image hashing
39. Meenakshi K, Swaraja K, Kora P (2019) A robust DCT-SVD based video watermarking using zigzag scanning. In: Soft computing and signal processing. Springer, Singapore, pp 477–485
40. Nilchi ARN, Taheri A (2008) A new robust digital image watermarking technique based on the Discrete Cosine Transform and Neural Network. In: 2008 International symposium on biometrics and security technologies. IEEE, pp 1–7
41. Ramanjaneyulu K, Rajarajeswari K (2012) Wavelet-based oblivious image watermarking scheme using genetic algorithm. IET Image Process 6(4):364–373
42. Saxena N, Mishra KK, Tripathi A (2018) DWT-SVD-based color image watermarking using dynamic-PSO. In: Advances in computer and computational sciences. Springer, Singapore, pp 343–351
43. Sequeira A, Kundur D (2001) Communication and information theory in watermarking: a survey. In: Multimedia systems and applications IV, vol 4518. International Society for Optics and Photonics, pp 216–227
44. Su Q, Yuan Z, Liu D (2018) An approximate Schur decomposition-based spatial domain color image watermarking method. IEEE Access 7:4358–4370
45. Swaraja K (2018) Medical image region based watermarking for secured telemedicine. Multimed Tools Appl 77(21):28249–28280
46. Swaraja K, Meenakshi K, Kora P (2020) An optimized blind dual medical image watermarking framework for tamper localization and content authentication in secured telemedicine. Biomed Signal Process Control 55:101665
47. Swaraja K, Madhaveelatha Y, Reddy VSK (2014) A pristine digital video watermarking in H. 264 compressed domain. In: 2014 IEEE International conference on computational intelligence and computing research. IEEE, pp 1–4
48. Tang Z, Huang L, Zhang X, Lao H (2016) Robust image hashing based on color vector angle and canny operator. AEU-Int J Electron Commun 70(6):833–841
49. Vali MH, Aghagolzadeh A, Baleghi Y (2018) Optimized watermarking technique using self-adaptive Differential Evolution based on redundant discrete wavelet transform and singular value decomposition. Expert Syst Appl 114:296–312

50. Wolfe SJ (1980) A characterization of Lévy probability distribution functions on Euclidean spaces. J Multivar Anal 10(3):379–384
51. Yu F, Li Y, Wei B, Kuang L (2016) Interactive differential evolution for user-oriented image retrieval system. Soft Comput 20(2):449–463
52. Zhao Y, Wang S, Feng G, Tang Z (2010) A robust image hashing method based on Zernike moments. J Comput Inf Syst 6(3):717–725
53. Zheng F, Zecchin AC, Simpson AR (2013) Self-adaptive differential evolution algorithm applied to water distribution system optimization. J Comput Civil Eng 27(2):148–158

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