Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19889-z

PBNHWA: NIfTI image watermarking with aid of PSO and BO in wavelet domain with its authentication for telemedicine applications

Divyanshu Awasthi[1] · Priyank Khare[2] · Vinay Kumar Srivastava[1]

Received: 30 August 2023 / Revised: 19 June 2024 / Accepted: 13 July 2024

Abstract
Due to the vast development of medical image technologies the orientation, dimension, and format of the medical images are changing drastically. The volume of these data is also changing drastically in the current scenario due to the unwanted pandemic circumstances. The copyright protection and privacy of these data is also the need of the present era. Image watermarking is one of the best methods to provide copyright protection to the medical images. The proposed watermarking method provides copyright protection to NIfTI (Neuroimaging informatics technology initiative) images by using Redundant discrete wavelet transform (RDWT), Randomized singular value decomposition (RSVD), Hessenberg decomposition (HD) along with discrete cosine transform (DCT). Watermark image (Aadhar card) is splits into two parts, further two hybrid watermarks are generated from these parts to enhance copyright protection. To get the optimum scaling factor two different optimization techniques i.e., Particle swarm optimization (PSO) and Bat optimization (BO) are used. Speeded up robust features (SURF) and Binary robust invariant scalable keypoints (BRISK) are used for critical region of interest (ROI) verification. Secured hash algorithm (SHA) is used to generate the hash values corresponding to the media access control (MAC) and patient number generated by hospital. The average percentage improvement in imperceptibility is 20.34% and in robustness is 11.54%.

Keywords NIfTI · PSO · BO · Telemedicine applications · Confidentiality

✉ Divyanshu Awasthi
divyanshuawasthi83@gmail.com

Priyank Khare
priyank22mnnit@gmail.com

Vinay Kumar Srivastava
vinay@mnnit.ac.in

[1] Department of Electronics and Communication Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India

[2] Department of Electronics and Communication Engineering, Indian Institute of Information Technology Ranchi, Ranchi, Jharkhand, India

Published online: 02 August 2024

1 Introduction

In tele-medical services, security and privacy of medical information are often highlighted as key concerns [1]. So, it was necessary to develop a method to ensure safety and confidentiality for tele-health services when exchanging patient records via networks. The tele-medical services handle, preserve, and transmit medical records using cutting-edge, cost-effective information technology (IT) systems for a variety of purposes [1]. These services are practical and may be helpful in the medical field, but they also pose a risk of identity theft, privacy invasion, and data tampering at the same time [2]. The modern medical era uses various formats of medical images as per the requirement. Neuroimaging informatics technology initiative (NIfTI) is a format for the storage and representation of medical images. The NIfTI format exists in two different variations: NIfTI-1 and NIfTI-2. The supported data types, accuracy, and voxel size of NIfTI-1 are all improved in NIfTI-2 [10]. Figure 1 shows the comparison between NIfTI and Digital imaging and communications in medicine (DICOM) formats. The copyright protection of these images is the major concern due to their use by the doctors in modern disease diagnosis. The total number of security incidents handled by the Indian computer emergency response team (CERT-In) in 2022 was 1391457 (https://www.cert-in.org.in/s2cMainServlet?pageid=PUBANULREPRT). In which phishing is 1714, probing is 324620, vulnerable services are 875892, malicious codes are 161757, website defacements are 19793, malware propagation is 2164, and others are 5517. These attacks are exponentially increasing yearly, which affects the whole digital ecosystem.

The following are the major novelties of the proposed work:

(1) Dual authentication with real-time verification:

The proposed watermarking is highly imperceptible and uses dual authentication. SURF [9] and BRISK [13] feature authentication is used for concerned ROI verification of medical NIfTI images. SURF features are computationally more efficient as well as invariant to blur, scale, rotation, noise, Illumination, and warping [9]. BRISK features are more efficient than SURF [13]. The identity of patient is also verified at the receiver's end using real time cloud data transfer.

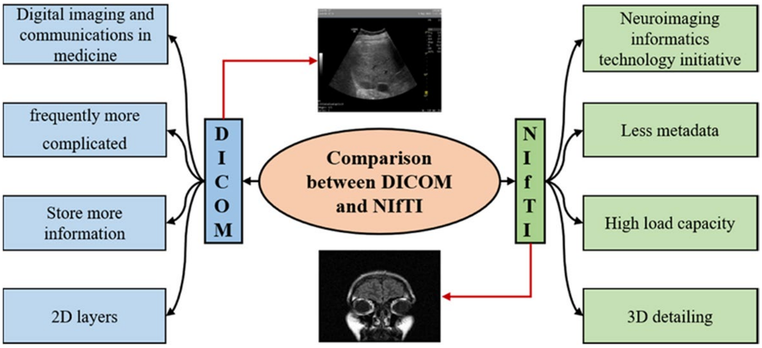

Digital imaging and communications in medicine
frequently more complicated
Store more information
2D layers
D I C O M
Comparison between DICOM and NIfTI
N I f T I
Neuroimaging informatics technology initiative
Less metadata
High load capacity
3D detailing

Fig. 1 Comparison between DICOM and NIfTI

(2) High robustness with better quality extraction:

The combination of RDWT [3], RSVD [12], HD [4], and DCT [5] provides high robustness and the quality of the extracted watermark image is also better. RDWT provides a vital advantage over traditional transforms as it is shift invariant [8]. RSVD is used instead of conventional singular value decomposition (SVD), as RSVD has lesser computational complexity than SVD and also requires less memory storage [8]. DCT converts the image into various frequency ranges which further enhances the imperceptibility by selective embedding [5].

(3) Optimized scaling factor:

Scaling factor plays an important role to balance the crucial properties of a watermarking system. The proposed watermarking method uses two different optimization technique i.e., PSO [5] and BO [21]. The comparison of these two techniques based on performance parameters are also computed.

(4) Enhanced security:

SHA-384 [14] is used to generate the hash value corresponding to MAC address and SHA-512 [14] is used for identity number of patient generated by hospital. Further the Aadhar card of patient is splitted into two parts and previously generated hash values are used to get the hybrid watermark with the help of lifting wavelet transform (LWT) [18, 19] and RSVD.

The rest of the paper is organized as follows: Section 2 contains the literature survey and concerned problems. Preprocessing steps are discussed in Section 3 while Watermark embedding and extraction process is included in Section 4. The simulation results analysis and discussion are mentioned in Section 5. Comparison of results are done in Section 6 whereas conclusion and future works are included in Section 7. Figure 2 shows the organization of paper graphically.

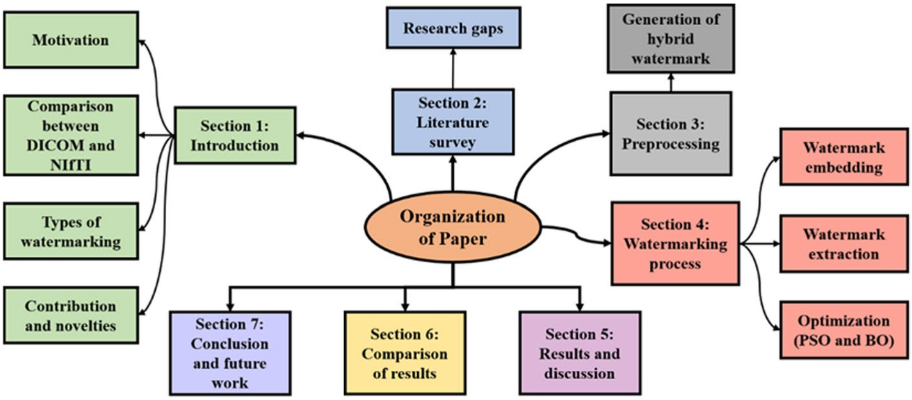

Motivation
Comparison between DICOM and NIfTI
Types of watermarking
Contribution and novelties
Section 1: Introduction
Research gaps
Section 2: Literature survey
Generation of hybrid watermark
Section 3: Preprocessing
Organization of Paper
Section 4: Watermarking process
Watermark embedding
Watermark extraction
Optimization (PSO and BO)
Section 7: Conclusion and future work
Section 6: Comparison of results
Section 5: Results and discussion

Fig. 2 Organization of paper

2 Literature survey

In this section, state-of-the-art literature survey is discussed. A novel watermarking technique using Hamming code for channel noise distortion is proposed in [1]. The combination of encryption and compression is used for embedding. Three different compression techniques and their comparison are also mentioned. The work proposed [1] is highly robust and imperceptible but can further be improved for rotation and cropping type of attacks. Hybrid PSO and firefly optimization-based watermarking is proposed in [2]. RDWT and RSVD is used for embedding. Hybrid watermark is generated is generated by using computational inefficient discrete wavelet transform (DWT) and SVD. Encryption technique is used to enhance the security of the proposed work. The proposed work can further be improved against geometric type of attacks.

RDWT-RSVD based image watermarking techniques with enhanced security is proposed in [3]. Hash value and turbo code are used for generation of intermediate watermark. The proposed method tested for various scaling factor. The scaling factor can further be evaluated using optimization techniques and performance can be improved against histogram equalization.

Robust and imperceptible watermarking with HD along with PSO and JAYA optimization (JO) is proposed in [4]. The comparison of different wavelets and optimization techniques are also evaluated. The comparison of LWT and discrete wavelet transform (DWT) is mentioned in [5]. The comparison of PSO and JO are also calculated for different parameters. The proposed method can further be improved against geometric type of attacks.

The comparison of PSO and JO is mentioned in [6]. The block like structures is appeared when the watermark image is extracted against rotation and dither attacks. LWT-DCT-SVD along with Schur decomposition (SD) based DICOM image watermarking technique is proposed in [7]. Firefly optimization technique is used to get the optimum scaling factors and make a proper balance between the vital properties of watermarking. SURF features are used for feature verification. A novel BacterialWmark technique is proposed in [8]. Bacterial foraging optimization technique is used for optimum scaling factor. Homomorphic transform with DWT based robust watermarking technique is proposed in [11]. The robustness can further be improved against geometric type of attacks. NIfTI image watermarking method is proposed in [14] with the combination of LWT and Arnold map (AM). Robustness of this method can be improved further against various attacks. NIfTI image watermarking with LWT, QR decomposition and Multiresolution SVD is proposed in [15].

Singh et al. [16] proposes a NIfTI image watermarking by the utilization of LWT, QR, and multiresolution SVD. Multiple watermarks are used to enhance the copyright protection. The robustness of the proposed method can be further enhanced against some of the attacks. A multifunctional medical image watermarking technique is presented by Sinhal et al. [17] to provide ownership protection, tamper detection (for ROI and other region of non-interest (RONI) segments), and 100% reversible self-recovery of the ROI. Using the Lempel–Ziv-Welch technique, the recovery information of the host image's ROI is first compressed. After that, a transform domain based embedding approach is used to implant the strong watermark into the host image. Additionally, the SHA-256 technique is used to construct the 256-bit hash keys for the robust watermarked image's ROI and eight RONI regions. Using an LSB replacement based fragile watermarking approach, the compressed recovery data and hash keys are merged and then inserted into the segmented RONI area of the robust watermarked image.

An effective watermarking method is presented by Thakur et al. [20] to improve the effectiveness of the DWT-SVD-based strategy. The methodology reduces channel noise distortion and enhances technique security by utilizing well-known error-correcting code and chaotic encryption, respectively. The sub-bands are chosen for the watermark embedding after DWT transforms the cover image. The chosen sub-bands are then subjected to additional transformation via SVD.

Tan et al. [22] proposes a steganography approach-based on generative adversarial networks. The suggested concept consists of three subnetworks: an extractor takes the payload out of stego images, a generator embeds it in cover images, and a potent steganalysis serves as a discriminator to improve steganographic security. By taking use of channel interdependencies, this work constructs a particular channel attention module that dynamically tunes channel-wise characteristics in the deep representation of images. In order to divide the embedding capacity among RGB channels adaptively, Liao et al. [23] present a novel channel-dependent payload division technique based on amplifying channel modification probability. To enhance empirical steganographic security against the identification of channel co-occurrences, the modification probabilities of three corresponding pixels in RGB channels are simultaneously enhanced. This could lead to a clustering of embedding impacts.

Based on image texture attributes, an adaptive payload distribution in multiple images steganography technique is presented in [24]. It also offers a theoretical security analysis from the perspective of the steganalysis. In this work, two payload distribution algorithms are described, one based on image texture complexity and the other on distortion distribution. The suggested tactics can be utilized in conjunction with these cutting-edge single picture steganographic techniques. Liao et al. [25] describe an order forensics framework based on convolutional neural network (CNN) for visual operator chain detection. The purpose of the two-stream CNN architecture is to record evidence of both local noise residuals and tampering artefacts. The novel CNN-based technique, which may automatically learn manipulation detection features straight from image data, is specifically suggested for forensically detecting a chain consisting of two image operators.

An image encryption technique based on bit replacement, chaotic systems, and deoxyribonucleic acid (DNA) coding is presented by Yousif et al. [26]. Its goal is to safeguard the digital photos' privacy and confidentiality. Initially, using this method, every pixel in the image is translated into its matching binary sequence, which consists of bits that are zeros and ones. Subsequently, the one bit is substituted with (0 and 1) bits, and the zero bit is replaced with (1 and 0) bits. By repeatedly altering all of the image pixel bits, two distinct images are ultimately produced. Second, high dimensional chaotic systems based on permutation and diffusion are used to encrypt the resulting images. Finally, DNA encryption is also utilized. An innovative method of securing images is presented in [27] by combining chaotic systems, the El-Gamal public key cryptosystem, and scanning techniques. To put it briefly, a permuted image is initially constructed using spiral and zigzag scanning. Next, the permuted image is encrypted using the El-Gamal encryption technique. Finally, in the confusion and diffusion stages, the pixel locations are randomly generated using Lorenz and Rössler chaotic sequences. A block-based method is put out by Salim et al. [28] to safeguard the authenticity of images by identifying and pinpointing counterfeit. It uses a watermarking technique based on visual cryptography to give forgery detection and localization features. Features, key shares, and secret shares are generated in this watermarking system. DWT, Walsh transform, and local binary pattern are used to extract features from equal-sized blocks of the image and create the feature share.

Subsequently, a random key share is generated from every image block, and the secret share is created by XORing the feature share, watermark, and key share.

(1) The metaheuristic algorithms help us in computing the optimum scaling factor under certain conditions effectively. If we do not consider these approaches and go for the hit and trial method for finding the scaling factor. This process will not result in a proper trade-off among different characteristics. So, in order to maintain this trade-off, it is required to consider this optimization technique for efficient watermarking.

(2) There are various optimization techniques available in the literature out of which we have to choose that provides us the better optimal results under certain conditions.

(3) Further, the metaheuristic algorithm is selected in such a way that have less elapsed time with a fast convergence rate.

(4) Thus, overview of the existing techniques discussed above leads to the research findings to fill the research gap that motivates us to propose an efficient optimized watermarking technique with optimum gain value using the desired optimization techniques.

3 Preprocessing

In this section hybrid watermark is generated using SHA-384 [14], SHA-512 [14], LWT [18, 19], and RSVD [12].

Following are the steps of hybrid watermark generation:

Step 1: Take the Aadhar card of the patient and split it into two parts as shown in Fig. 3.

Step 2: Apply one level LWT to both the parts of Aadhar card as shown in Eq. (1):

$$\begin{cases} \left[LL_1, LH_1, HL_1, HH_1\right] = lwt2(W1) \\ \left[LL_2, LH_2, HL_2, HH_2\right] = lwt2(W2) \end{cases} \tag{1}$$

Step 3: Apply RSVD to previously generated LH_1, and HL_2 as shown in Eq. (2):

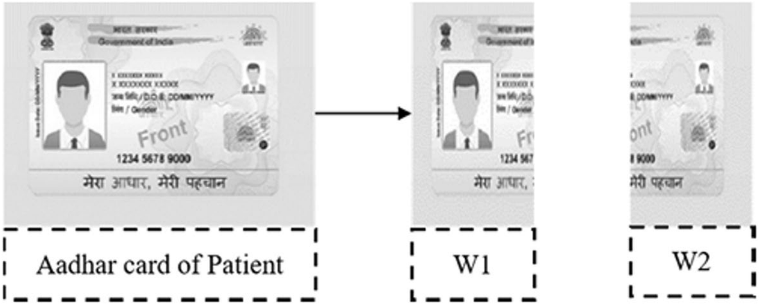


Fig. 3 Two segments of Aadhar card of patient

$$\begin{cases} [U_1, D_1, V_1] = rsvd(LH_1) \\ [U_2, D_2, V_2] = rsvd(HL_2) \end{cases} \tag{2}$$

Step 4: Take MAC address and generate corresponding hash value (SHA-384) similarly take the patient number generated by hospital and generate corresponding hash value (SHA-512) and generate corresponding image of size 128 × 128 (I_{hash1}, I_{hash2}).

MAC address: 84A938D7266C.

Hash values: 206 187 14 18 48 187 45 158 2 117 39 4 161 216 208 15 216 135 106 232 64 59 39 235 178 82 238 202 231 132 214 242 197 170 60 212 87 212 119 100 15 67 156 132 235 129 108 157.

Hospital generated patient number: NAMEAGESEXDISEASECODE.

Hash values: 162 212 97 133 192 8 208 126 206 244 137 170 158 11 255 87 201 98 66 185 97 239 235 82 156 62 24 13 25 242 173 125 67 164 171 168 107 96 209 32 194 158 140 35 34 7 177 88.

Step 5: Apply one level LWT to both the previously generated images corresponding to hash values as shown in Eq. (3):

$$\begin{cases} [LL_{H1}, LH_{H1}, HL_{H1}, HH_{H1}] = lwt2(I_{hash1}) \\ [LL_{H2}, LH_{H2}, HL_{H2}, HH_{H2}] = lwt2(I_{hash2}) \end{cases} \tag{3}$$

Step 6: Apply RSVD to previously generated LH_{H1}, and HL_{H2} as shown in Eq. (4):

$$\begin{cases} [U_{H1}, D_{H1}, V_{H1}] = rsvd(LH_{H1}) \\ [U_{H2}, D_{H2}, V_{H2}] = rsvd(HL_{H2}) \end{cases} \tag{4}$$

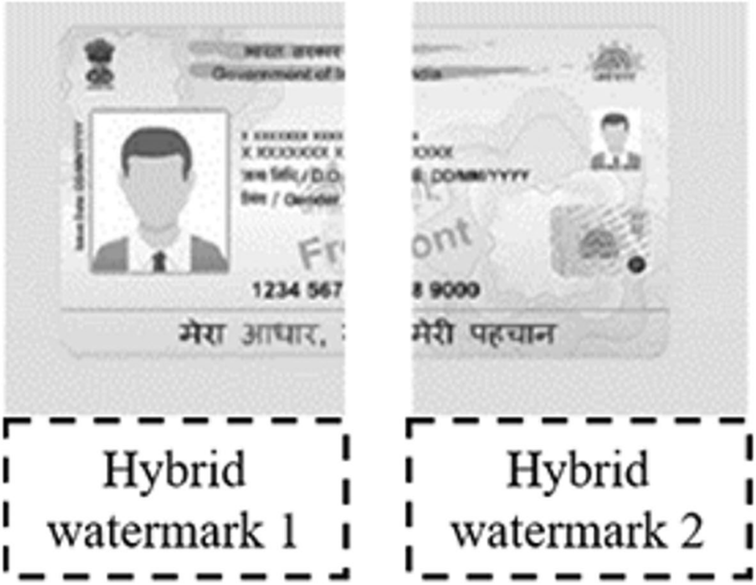


Fig. 4 Hybrid watermarks

Step 7: Add the corresponding dominant components to generate modified dominant component as shown in Eq. (5):

$$\begin{cases} D_{mod1} = D_1 + \beta \times D_{H1} \\ D_{mod2} = D_2 + \beta \times D_{H2} \end{cases} \tag{5}$$

where β is the scaling factor.
Step 8: Apply inverse RSVD and LWT to generate hybrid watermarks (HB_{W1} and HB_{W2}) as shown in Fig. 4. Hybrid watermarks generation process in shown in Fig. 5.

Algorithm 1 Pre-processing

```
Input: Aadhar card of patient, MAC address, β
Output: Hybrid watermarks
I1=imread('Aadhar card of patient')
W1= I1(1:128,128:128)
  W2= I1(1:128,129:256)
$[LL_1, LH_1, HL_1, HH_1] = lwt2(W1)$
  $[LL_2, LH_2, HL_2, HH_2] = lwt2(W2)$
$[U_1, D_1, V_1] = rsvd(LH_1)$
  $[U_2, D_2, V_2] = rsvd(HL_2)$
String1= MAC address (84A938D7266C)
hashObject=System.Security.Cryptography.SHA384Managed
message_Digest1= uint8(hashObject.ComputeHash(uint8(string1)))
String2= Hospital generated patient number (NAMEAGESEXDISEASECODE)
hashObject=System.Security.Cryptography.SHA512Managed
message_Digest2= uint8(hashObject.ComputeHash(uint8(string2)))
Convert both the generated hash values into image form
$[LL_{H1}, LH_{H1}, HL_{H1}, HH_{H1}] = lwt2(I_{hash1})$
  $[LL_{H2}, LH_{H2}, HL_{H2}, HH_{H2}] = lwt2(I_{hash2})$
$[U_{H1}, D_{H1}, V_{H1}] = rsvd(LH_{H1})$
  $[U_{H2}, D_{H2}, V_{H2}] = rsvd(HL_{H2})$
$D_{mod1} = D_1 + \beta \times D_{H1}$
  $D_{mod2} = D_2 + \beta \times D_{H2}$
$HB_{W1}$; $HB_{W2}$= inverse rsvd; inverse lwt
Return $HB_{W1}$ and $HB_{W2}$
```

4 PBNHWA process

The proposed NIfTI image hybrid watermarking technique uses RDWT [3], HD [4], DCT [5], and RSVD [12] for watermark embedding and extraction. The PSO [5], and BO [21] are utilized to get the optimum scaling factor. SURF [9] and BRISK [13] matching points are used for ROI verification. Following are the steps of the watermarking process:

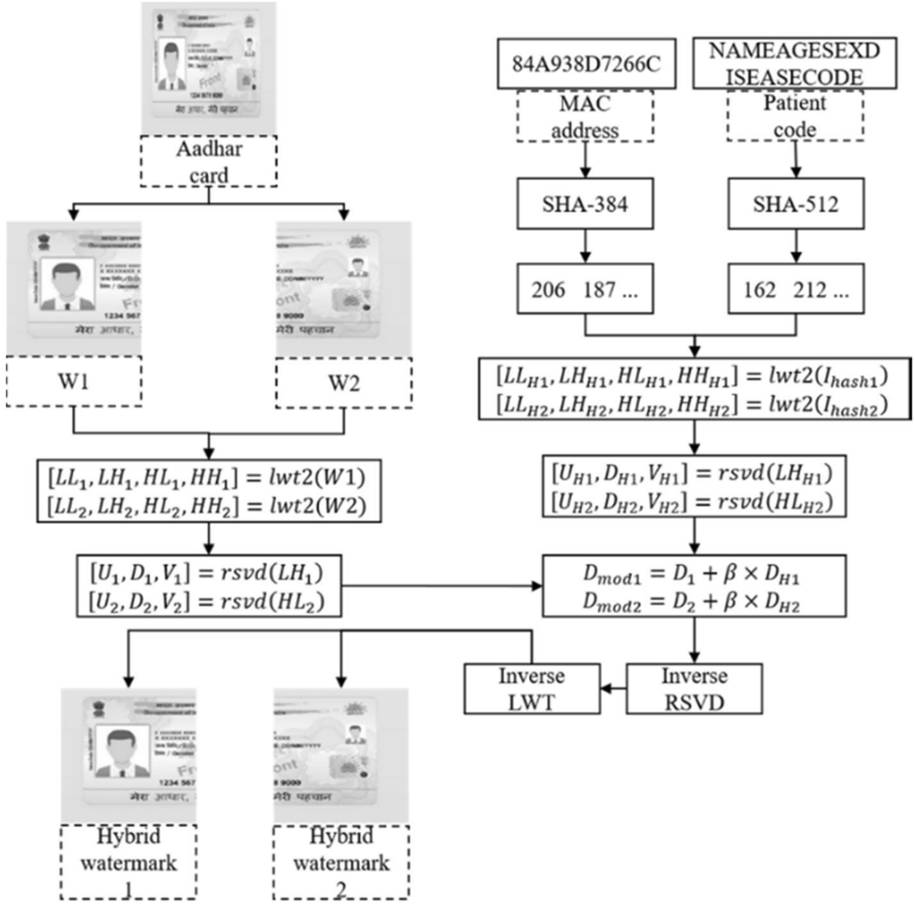

Aadhar card
W1
W2
84A938D7266C
MAC address
NAMEAGESEXD ISEASECODE
Patient code
SHA-384
SHA-512
206 187 ...
162 212 ...
$[LL_{H1}, LH_{H1}, HL_{H1}, HH_{H1}] = lwt2(I_{hash1})$
$[LL_{H2}, LH_{H2}, HL_{H2}, HH_{H2}] = lwt2(I_{hash2})$
$[LL_1, LH_1, HL_1, HH_1] = lwt2(W1)$
$[LL_2, LH_2, HL_2, HH_2] = lwt2(W2)$
$[U_{H1}, D_{H1}, V_{H1}] = rsvd(LH_{H1})$
$[U_{H2}, D_{H2}, V_{H2}] = rsvd(HL_{H2})$
$[U_1, D_1, V_1] = rsvd(LH_1)$
$[U_2, D_2, V_2] = rsvd(HL_2)$
$D_{mod1} = D_1 + \beta \times D_{H1}$
$D_{mod2} = D_2 + \beta \times D_{H2}$
Inverse LWT
Inverse RSVD
Hybrid watermark 1
Hybrid watermark 2

Fig. 5 Hybrid watermarks generation procedure

Step 1: Take the NIfTI image as the input and select a slice as shown in Eq. (6):

$$\begin{cases} path = \varepsilon barin.nii\varepsilon \\ I = niftiread(path) \\ \quad C = I(:, :, s(i)) \end{cases} \tag{6}$$

where $s(i)$ is the slice number and $1 \le i \le 21$.

Step 2: Apply the RDWT to the previously selected slice of NIfTI image as shown in Eq. (7):

$$\left[LL_{I1}, LH_{I1}, HL_{I1}, HH_{I1}\right] = RDWT(C) \tag{7}$$

Step 3: Apply HD to LH_{I1} and HL_{I1} as shown in Eq. (8):

$$\begin{cases} [U_1, T_1] = hess(LH_{I1}) \\ [U_2, T_2] = hess(HL_{I1}) \end{cases} \tag{8}$$

Step 4: Apply DCT to previously obtained Hessenberg matrices as shown in Eq. (9):

$$\begin{cases} D_{c1} = dct2(T_1) \\ D_{c2} = dct2(T_2) \end{cases} \tag{9}$$

Step 5: Select the low and mid frequency components as shown in Eq. (10):

$$\begin{cases} main_1 = D_{c1}(1 : 128,1 : 128) \\ main_2 = D_{c2}(1 : 128,1 : 128) \end{cases} \tag{10}$$

Step 6: Apply RSVD to previously obtained bands of frequency as shown in Eq. (11):

$$\begin{cases} [P_1, Q_1, R_1] = RSVD(main_1) \\ [P_2, Q_2, R_2] = RSVD(main_2) \end{cases} \tag{11}$$

Step 7: Take the hybrid watermark images and apply RDWT as shown in Eq. (12):

$$\begin{cases} [LL_{HB_{W1}}, LH_{HB_{W1}}, HL_{HB_{W1}}, HH_{HB_{W1}}] = RDWT(HB_{W1}) \\ [LL_{HB_{W2}}, LH_{HB_{W2}}, HL_{HB_{W2}}, HH_{HB_{W2}}] = RDWT(HB_{W2}) \end{cases} \tag{12}$$

Step 8: Apply RSVD to previously obtained $HH_{HB_{W1}}$ and $HH_{HB_{W2}}$ as shown in Eq. (13):

$$\begin{cases} [U_{W1}, S_{W1}, V_{W1}] = RSVD(HH_{HB_{W1}}) \\ [U_{W2}, S_{W2}, V_{W2}] = RSVD(HH_{HB_{W2}}) \end{cases} \tag{13}$$

Step 9: Apply embedding procedure by taking optimized scaling factor (α) using PSO and BO as shown in Eq. (14):

$$\begin{cases} M_1 = Q_1 + \alpha \times S_{W1} \\ M_2 = Q_2 + \alpha \times S_{W2} \end{cases} \tag{14}$$

Step 10: Apply inverse RSVD and insert the modified low and mid frequency bands to rest of the bands and take the inverse DCT.
Step 11: Apply inverse HD and RDWT to get the watermarked image (I_{marked}).
Step 12: Take the watermarked image and apply RDWT to get the sub-bands as shown in Eq. (15):

$$[LL_W, LH_W, HL_W, HH_W] = RDWT(I_{marked}) \tag{15}$$

Step 13: Apply HD to previously obtained LH_W and HL_W as shown in Eq. (16):

$$\begin{cases} \left[U_{1marked}, T_{1marked}\right] = hess\left(LH_{W}\right) \\ \left[U_{2marked}, T_{2marked}\right] = hess\left(HL_{W}\right) \end{cases} \tag{16}$$

Step 14: Apply DCT to previously obtained Hessenberg matrices as shown in Eq. (17):

$$\begin{cases} D_{c1marked} = dct2\left(T_{1marked}\right) \\ D_{c2marked} = dct2\left(T_{2marked}\right) \end{cases} \tag{17}$$

Step 15: Select the low and mid frequency components as shown in Eq. (18):

$$\begin{cases} main_{1marked} = D_{c1marked}(1 : 128,1 : 128) \\ main_{2marked} = D_{c2marked}(1 : 128,1 : 128) \end{cases} \tag{18}$$

Step 16: Apply RSVD to previously obtained bands of frequency as shown in Eq. (19):

$$\begin{cases} \left[P_{1M}, Q_{1M}, R_{1M}\right] = RSVD\left(main_{1marked}\right) \\ \left[P_{2M}, Q_{2M}, R_{2M}\right] = RSVD\left(main_{2marked}\right) \end{cases} \tag{19}$$

Step 17: Apply inverse embedding by taking optimum scaling factor as shown in Eq. (20):

$$\begin{cases} M_{1W} = \left(Q_{1M} - Q_{1}\right)/\alpha \\ M_{2W} = \left(Q_{1M} - Q_{2}\right)/\alpha \end{cases} \tag{20}$$

Step 18: Apply inverse RSVD and RDWT to get the extracted watermarks.

Step 19: Combine both the extracted watermark images to get the merged or complete watermark as shown in Fig. 6. Figure 7 shown the watermarking process.

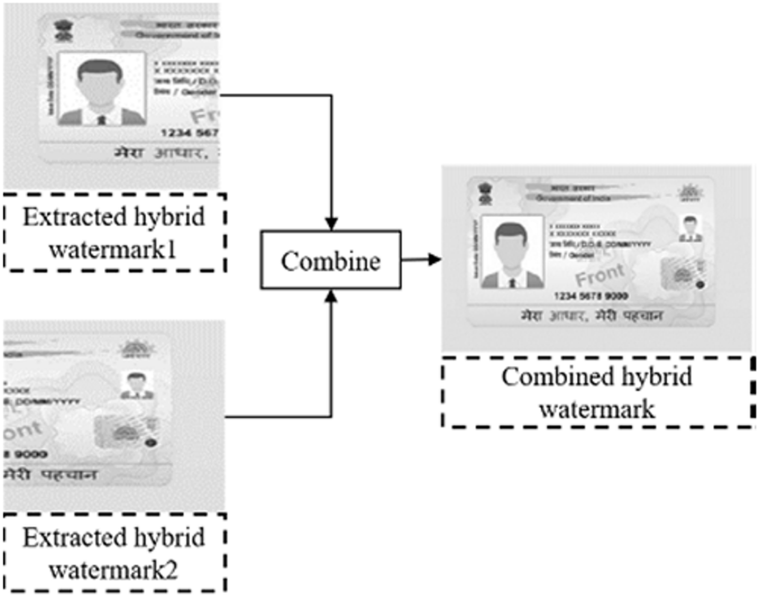

Extracted hybrid watermark1
Combine
Combined hybrid watermark
Extracted hybrid watermark2

Fig. 6 Combined hybrid watermark

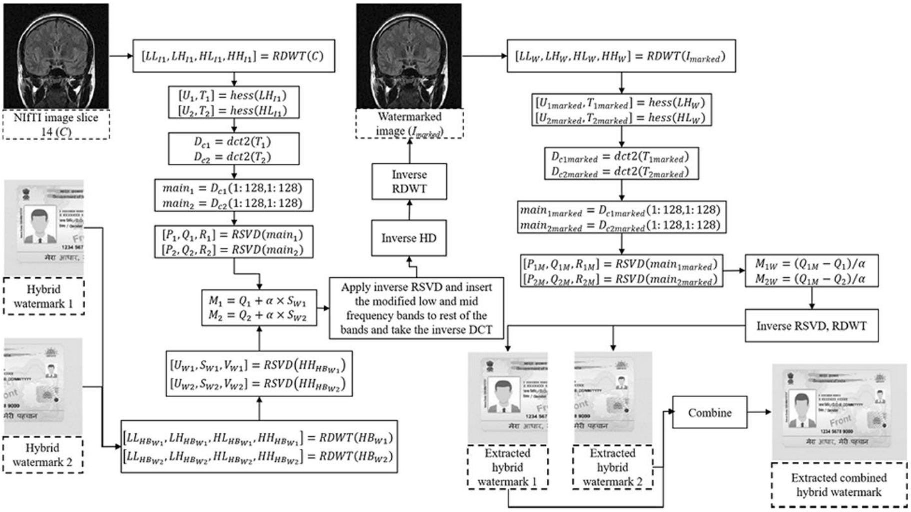

NIfTI image slice 14 (C)
$[LL_{I1}, LH_{I1}, HL_{I1}, HH_{I1}] = RDWT(C)$
$[U_1, T_1] = hess(LH_{I1})$
$[U_2, T_2] = hess(HL_{I1})$
$D_{c1} = dct2(T_1)$
$D_{c2} = dct2(T_2)$
$main_1 = D_{c1}(1{:}128, 1{:}128)$
$main_2 = D_{c2}(1{:}128, 1{:}128)$
$[P_1, Q_1, R_1] = RSVD(main_1)$
$[P_2, Q_2, R_2] = RSVD(main_2)$
Hybrid watermark 1
$M_1 = Q_1 + \alpha \times S_{W1}$
$M_2 = Q_2 + \alpha \times S_{W2}$
Apply inverse RSVD and insert the modified low and mid frequency bands to rest of the bands and take the inverse DCT
$[U_{W1}, S_{W1}, V_{W1}] = RSVD(HH_{HB_{W1}})$
$[U_{W2}, S_{W2}, V_{W2}] = RSVD(HH_{HB_{W2}})$
Hybrid watermark 2
$[LL_{HB_{W1}}, LH_{HB_{W1}}, HL_{HB_{W1}}, HH_{HB_{W1}}] = RDWT(HB_{W1})$
$[LL_{HB_{W2}}, LH_{HB_{W2}}, HL_{HB_{W2}}, HH_{HB_{W2}}] = RDWT(HB_{W2})$
Watermarked image (I_{marked})
Inverse RDWT
Inverse HD
$[LL_W, LH_W, HL_W, HH_W] = RDWT(I_{marked})$
$[U_{1marked}, T_{1marked}] = hess(LH_W)$
$[U_{2marked}, T_{2marked}] = hess(HL_W)$
$D_{c1marked} = dct2(T_{1marked})$
$D_{c2marked} = dct2(T_{2marked})$
$main_{1marked} = D_{c1marked}(1{:}128, 1{:}128)$
$main_{2marked} = D_{c2marked}(1{:}128, 1{:}128)$
$[P_{1M}, Q_{1M}, R_{1M}] = RSVD(main_{1marked})$
$[P_{2M}, Q_{2M}, R_{2M}] = RSVD(main_{2marked})$
$M_{1W} = (Q_{1M} - Q_1)/\alpha$
$M_{2W} = (Q_{1M} - Q_2)/\alpha$
Inverse RSVD, RDWT
Extracted hybrid watermark 1
Extracted hybrid watermark 2
Combine
Extracted combined hybrid watermark

Fig. 7 Embedding and extraction process

Algorithm 2 PBNHWA embedding process

```
Input: NIfTI image, Hybrid watermarks, α
Output: Watermarked image
for i=1:21
    $path = "barin.nii"$
    $I = niftiread(path)$
    $C = I(:, :, s(i))$
    $[LL_{I1}, LH_{I1}, HL_{I1}, HH_{I1}] = RDWT(C)$
    $[U_1, T_1] = hess(LH_{I1})$
    $[U_2, T_2] = hess(HL_{I1})$
    $D_{c1} = dct2(T_1)$
    $D_{c2} = dct2(T_2)$
    $main_1 = D_{c1}(1{:}128, 1{:}128)$
    $main_2 = D_{c2}(1{:}128, 1{:}128)$
    $[P_1, Q_1, R_1] = RSVD(main_1)$
    $[P_2, Q_2, R_2] = RSVD(main_2)$
    $[LL_{HB_{W1}}, LH_{HB_{W1}}, HL_{HB_{W1}}, HH_{HB_{W1}}] = RDWT(HB_{W1})$
    $[LL_{HB_{W2}}, LH_{HB_{W2}}, HL_{HB_{W2}}, HH_{HB_{W2}}] = RDWT(HB_{W2})$
    $[U_{W1}, S_{W1}, V_{W1}] = RSVD(HH_{HB_{W1}})$
    $[U_{W2}, S_{W2}, V_{W2}] = RSVD(HH_{HB_{W2}})$
    $M_1 = Q_1 + \alpha \times S_{W1}$
    $M_2 = Q_2 + \alpha \times S_{W2}$
    Apply inverse RSVD and insert the modified low and mid frequency bands to rest of the bands and take the inverse DCT
    $I_{marked} \leftarrow$ Apply inverse HD and RDWT to get the watermarked image
end
Return $I_{marked}$
```

Algorithm 3 PBNHWA extraction process

```
Input: Watermarked image, α
Output: Combined extracted watermark
1: $[LL_W, LH_W, HL_W, HH_W] = RDWT(I_{marked})$
2: $[U_{1marked}, T_{1marked}] = hess(LH_W)$
3: $[U_{2marked}, T_{2marked}] = hess(HL_W)$
4: $D_{c1marked} = dct2(T_{1marked})$
5: $D_{c2marked} = dct2(T_{2marked})$
6: $main_{1marked} = D_{c1marked}(1{:}128, 1{:}128)$
7: $main_{2marked} = D_{c2marked}(1{:}128, 1{:}128)$
8: $[P_{1M}, Q_{1M}, R_{1M}] = RSVD(main_{1marked})$
9: $[P_{2M}, Q_{2M}, R_{2M}] = RSVD(main_{2marked})$
10: $M_{1W} = (Q_{1M} - Q_1)/\alpha$
11: $M_{2W} = (Q_{1M} - Q_2)/\alpha$
12: Extracted watermarks ⟵ Apply inverse RSVD and RDWT to get the extracted watermarks
13: Combined extracted watermark ⟵ [Extracted1 Extracted2]
Return Extracted watermark (combined)
```

5 Optimization process

PSO is inspired by swarm foraging and social behavior. Simple ideas and basic operators are used to construct PSO. PSO has a low cost of computation in terms of memory and performance. PSO begins by randomly initializing the population. To investigate the search space, solutions are assigned with randomized velocities. In PSO, a particle is a solution. There are three distinct features of PSO: (1) best fitness of each particle, (2) best fitness of swarm, (3) velocity and position update of each particle [29].

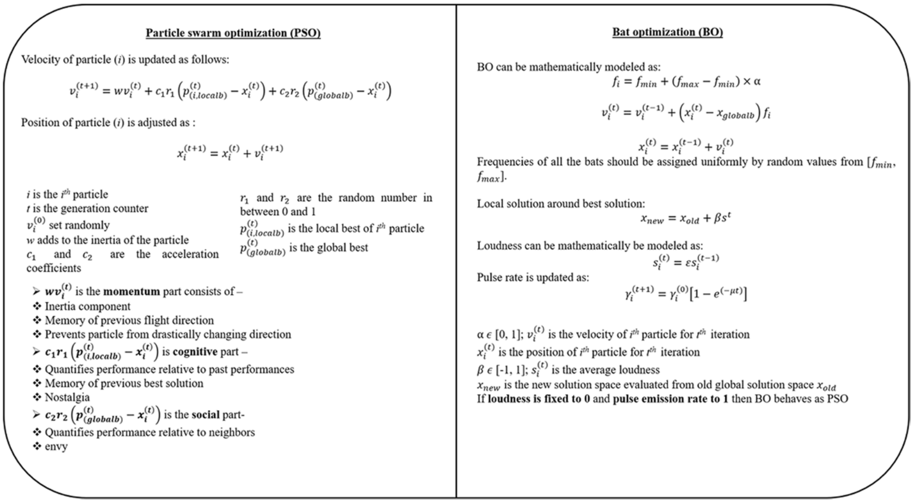
All bats utilize echolocation to gauge distance, and somehow, they also know the difference between background barriers and food/prey. In order to find prey, bats fly erratically at a set frequency, changing wavelength, and loudness. Depending on how close their target is, they can automatically modify the wavelength (or frequency) of their generated pulses and adjust the rate of pulse emission γ in the range [0, 1]. Figure 8 shows the process used in PSO and BO.

Following is the fitness equation used in optimization:

$$fitness = \left[\left(SSIM + \frac{1}{PSNR}\right) + \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{NCC}\right)\right]\frac{\alpha}{S} \tag{21}$$

In Eq. (21), *SSIM* is the structural similarity index measure [5], *PSNR* is peak signal to noise ratio [5], *NCC* is normalized correlation coefficient [5], *S* is the number of slices of NIfTI image, α is the balancing factor taken as 10, and *N* is the number of attacks (16). Flow charts of PSO and BO are shown in Figs. 9 and 10 respectively.

Parameters used in PSO:



Particle swarm optimization (PSO)

Velocity of particle (*i*) is updated as follows:

$$v_i^{(t+1)} = wv_i^{(t)} + c_1r_1\left(p_{(i,localb)}^{(t)} - x_i^{(t)}\right) + c_2r_2\left(p_{(globalb)}^{(t)} - x_i^{(t)}\right)$$

Position of particle (*i*) is adjusted as :

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$$

i is the *i*
i is the i^{th} particle
t is the generation counter
$v_i^{(0)}$ set randomly
w adds to the inertia of the particle
c_1 and c_2 are the acceleration coefficients

r_1 and r_2 are the random number in between 0 and 1
$p_{(i,localb)}^{(t)}$ is the local best of i^{th} particle
$p_{(globalb)}^{(t)}$ is the global best

➢ $\boldsymbol{wv_i^{(t)}}$ is the **momentum** part consists of –
❖ Inertia component
❖ Memory of previous flight direction
❖ Prevents particle from drastically changing direction
➢ $\boldsymbol{c_1r_1\left(p_{(i,localb)}^{(t)} - x_i^{(t)}\right)}$ is **cognitive** part –
❖ Quantifies performance relative to past performances
❖ Memory of previous best solution
❖ Nostalgia
➢ $\boldsymbol{c_2r_2\left(p_{(globalb)}^{(t)} - x_i^{(t)}\right)}$ is the **social** part-
❖ Quantifies performance relative to neighbors
❖ envy

Bat optimization (BO)

BO can be mathematically modeled as:

$$f_i = f_{min} + (f_{max} - f_{min}) \times \alpha$$

$$v_i^{(t)} = v_i^{(t-1)} + \left(x_i^{(t)} - x_{globalb}\right) f_i$$

$$x_i^{(t)} = x_i^{(t-1)} + v_i^{(t)}$$

Frequencies of all the bats should be assigned uniformly by random values from [f_{min}, f_{max}].

Local solution around best solution:

$$x_{new} = x_{old} + \beta s^t$$

Loudness can be mathematically be modeled as:

$$s_i^{(t)} = \varepsilon s_i^{(t-1)}$$

Pulse rate is updated as:

$$\gamma_i^{(t+1)} = \gamma_i^{(0)}[1 - e^{(-\mu t)}]$$

α ϵ [0, 1]; $v_i^{(t)}$ is the velocity of i^{th} particle for t^{th} iteration
$x_i^{(t)}$ is the position of i^{th} particle for t^{th} iteration
β ϵ [-1, 1]; $s_i^{(t)}$ is the average loudness
x_{new} is the new solution space evaluated from old global solution space x_{old}
If **loudness is fixed to 0** and **pulse emission rate to 1** then BO behaves as PSO

Fig. 8 Steps used in PSO and BO techniques

Particle size: 5; Cognitive factor (C1): 2; Social Coefficient: 2; Inertia weight: 0.9; Velocity: [0.5,1].

Number of iterations: 21; Elapsed Time: 5.15 s.

Parameters used in BO:

Particle size: 5; loudness: [1, 2]; pulse rate: [0, 1]; ε: 0.5 (reducing factor for loudness); μ: 0.9 (pulse rate increasing factor); Number of iterations: 17; Elapsed Time: 4.76 s.

6 Results and discussion

The proposed watermarking method uses NIfTI image [10] as the input host image of size 256×256. NIfTI images consists of different slices and behaves as the stack of papers as shown in Fig. 11 and in this work all the 21 slices are used as host image. It comes in two different file extensions:.nii and.nii.gz. while it does not contain any patient information. The hybrid watermark is generated initially then used for watermarking. The size of the hybrid watermark image is 128×256. Table 1 shows the performance parameters used in this proposed scheme.

6.1 Imperceptibility analysis

For imperceptibility analysis, the proposed method uses PSNR (dB), mean square error (MSE), SSIM, KLD and Jensen JSD. The acceptable value of PSNR is greater than 27 dB without attack [11]. The value of MSE should to tends to zero for higher imperceptibility. The value of SSIM should be close to 1.0000 or greater than 0.9 for better imperceptibility. Lesser the value of KLD [8] and JSD [8] higher is the imperceptibility. Table 2 shows the values of performance parameters for PSO and Table 3 is for BO. In this proposed work, the value of PSNR is greater than 50 dB for PSO and BO in all the

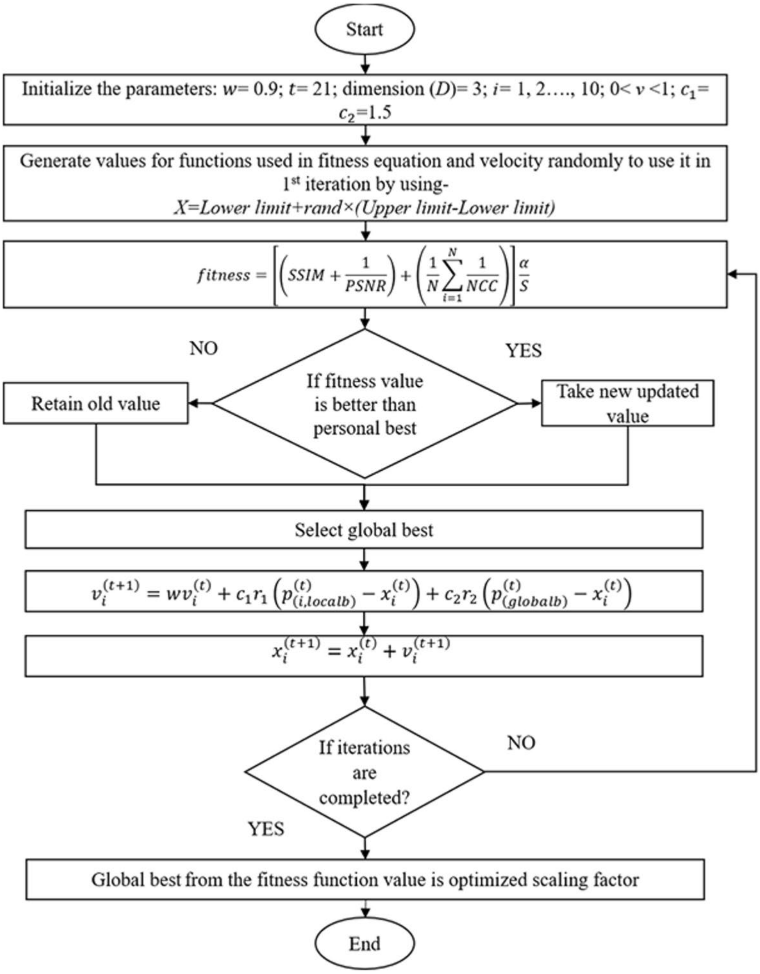

Start
Initialize the parameters: w= 0.9; t= 21; dimension (D)= 3; i= 1, 2…., 10; 0< v <1; c_1= c_2=1.5
Generate values for functions used in fitness equation and velocity randomly to use it in 1st iteration by using-
X=Lower limit+rand×(Upper limit-Lower limit)
$fitness = \left[\left(SSIM + \frac{1}{PSNR}\right) + \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{NCC}\right)\right]\frac{\alpha}{S}$
NO
YES
If fitness value is better than personal best
Retain old value
Take new updated value
Select global best
$v_i^{(t+1)} = wv_i^{(t)} + c_1r_1\left(p_{(i,localb)}^{(t)} - x_i^{(t)}\right) + c_2r_2\left(p_{(globalb)}^{(t)} - x_i^{(t)}\right)$
$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$
If iterations are completed?
NO
YES
Global best from the fitness function value is optimized scaling factor
End

Fig. 9 Flow chart of PSO technique

slices. The value of MSE is also close to 0 in all the cases. The value of SSIM is greater than 0.9 in all the cases and the values of KLD and JSD are also acceptable for PSO and BO. The value of PSNR is higher in case of BO. The value of SSIM is almost same in PSO and BO except for S5 (higher in case of BO). The values of KLD and JSD are more efficient in case of BO. So, by observing the Tables 2 and 3 it is concluded that the proposed method is highly imperceptible and BO is superior than PSO. Figure 12 shows the Comparison of PSNR (dB) values for different slices and for PSO and BO.

6.2 Robustness analysis

For robustness analysis, the proposed method uses NCC and bit error rate. The value of NCC should be close to 1.0000 without attack. Tables 2 and 3 shows the values of NCC for all the

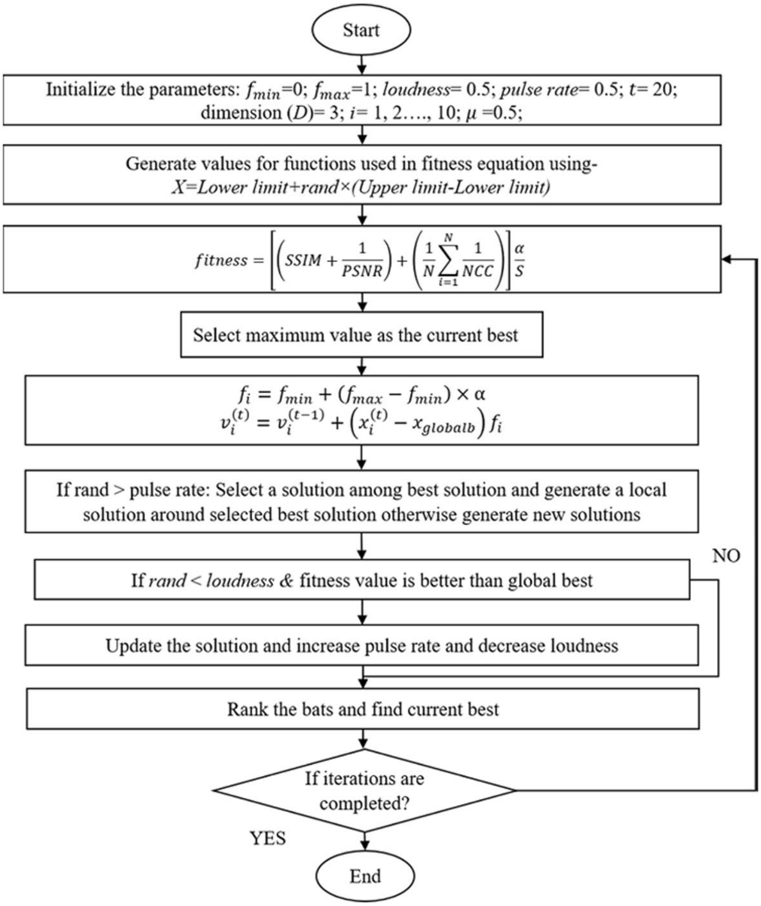

Start
Initialize the parameters: f_{min}=0; f_{max}=1; loudness= 0.5; pulse rate= 0.5; t= 20; dimension (D)= 3; i= 1, 2…., 10; μ =0.5;
Generate values for functions used in fitness equation using- X=Lower limit+rand×(Upper limit-Lower limit)
$fitness = \left[\left(SSIM + \frac{1}{PSNR}\right) + \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{NCC}\right)\right]\frac{\alpha}{S}$
Select maximum value as the current best
$f_i = f_{min} + (f_{max} - f_{min}) \times \alpha$
$v_i^{(t)} = v_i^{(t-1)} + \left(x_i^{(t)} - x_{globalb}\right) f_i$
If rand > pulse rate: Select a solution among best solution and generate a local solution around selected best solution otherwise generate new solutions
If rand < loudness & fitness value is better than global best
NO
Update the solution and increase pulse rate and decrease loudness
Rank the bats and find current best
If iterations are completed?
YES
End

Fig. 10 Flow chart of BO technique

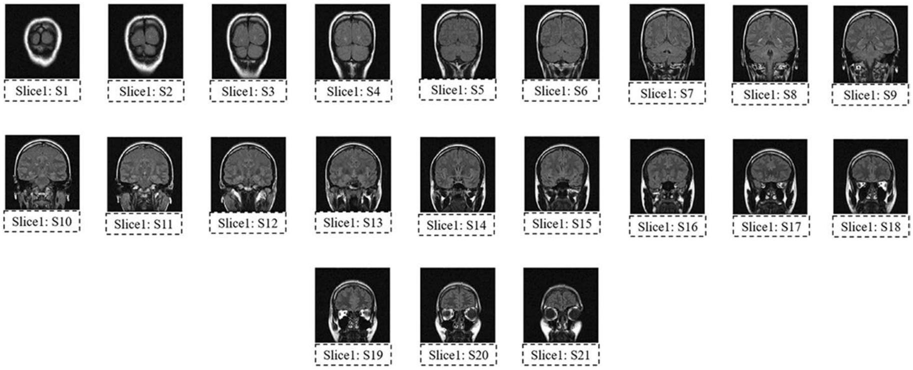


Fig. 11 Various slices of input NIfTI image

Table 1 Performance parameters

Parameters	Formula	Description
Peak signal to noise ratio (PSNR) [5, 8]	$PSNR = 10log_{10}\left(\frac{P^2}{\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[H_I(i,j)-W_I(i,j)]^2}{M\times N}}\right)$	where P is the maximum pixel value, $M\times N$ is the size of the host image, $H_I(i,j)$ is the host image, $W_I(i,j)$ is the watermarked image, and i, j is used for pixels. The denominator term inside logarithm operator is MSE
Structural similarity index measure (SSIM) [5, 8]	$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}$	where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images $x, y, and\ C_1\ and\ C_2$ are the regularization constant for the luminance and contrast
Normalized correlation coefficient (NCC) [5, 8]	$NCC = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}W(i,j)W_r(i,j)}{\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}W^2(i,j)}\sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}{W_r}^2(i,j)}}$	where $W(i,j)$ is the original watermark logo, and $W_r(i,j)$ is the recovered watermark logo
Bit error rate (BER) [8]	$BER = \frac{EIB}{WB}$	Calculate the error rate between the original watermark and the extracted watermark. EIB is the incorrect bits in extracted watermark, WB is the bits of original watermark image
Kullback–Leibler distance (KLD) [8]	$KLD = \sum A \times \log\left(\frac{A}{B}\right)$	where A and B are the probabilities of the input and watermarked images
Jensen Shannon distances (JSD) [8]	$JSD = 0.5 \times KLD(A,M) + 0.5 \times KLD(B,M)$	where $M = 0.5\times(A+B)$

Table 2 Values of performance parameters (PSO-without attack)

Input slices (S)	PSNR (dB)	MSE	SSIM	NCC	BER	KLD	JSD
S1	50.3585	9.4995e-06	0.9999	1.0000	0.0030	0	0.0031
S2	50.3361	9.5805e-06	0.9999	1.0000	0.0033	0	0.0026
S3	50.3919	9.3934e-06	0.9998	1.0000	0.0030	0	0.0025
S4	50.5004	9.2958e-06	0.9999	1.0000	0.0030	0	0.0025
S5	50.4921	9.1359e-06	0.9997	1.0000	0.0037	0	0.0029
S6	50.6881	9.0274e-06	0.9999	1.0000	0.0034	0	0.0031
S7	50.7452	8.6520e-06	0.9999	1.0000	0.0040	0.0284	0.0067
S8	50.8976	8.5414e-06	0.9998	1.0000	0.0040	0.0298	0.0070
S9	50.7460	8.7264e-06	0.9998	1.0000	0.0030	0.0326	0.0076
S10	50.6836	8.8534e-06	0.9998	1.0000	0.0035	0.0306	0.0072
S11	50.7754	8.7529e-06	0.9999	1.0000	0.0028	0.0300	0.0071
S12	50.8138	8.5417e-06	0.9999	1.0000	0.0037	0.0300	0.0071
S13	50.6992	8.7649e-06	0.9999	1.0000	0.0033	0.0356	0.0083
S14	50.7070	8.7618e-06	0.9999	1.0000	0.0035	0.0358	0.0084
S15	50.7624	8.7638e-06	0.9999	1.0000	0.0037	0.0356	0.0084
S16	50.7098	8.7387e-06	0.9999	1.0000	0.0031	0.0204	0.0049
S17	50.6974	8.9283e-06	0.9999	1.0000	0.0038	0.0369	0.0086
S18	50.6714	8.8891e-06	0.9999	1.0000	0.0041	0.0354	0.0083
S19	50.5449	9.0962e-06	0.9999	1.0000	0.0031	0.0326	0.0077
S20	50.6042	9.0761e-06	0.9999	1.0000	0.0034	0.0336	0.0080
S21	50.4998	9.1643e-06	0.9999	1.0000	0.0044	0	0.0041

slices and for PSO and BO. In all the cases the NCC value is 1.0000, it shows that the proposed method is highly robust. Figures 13 and 14 shows the extracted watermark images and corresponding NCC values against various attacks for PSO and BO respectively. The NCC values are greater than 0.9 in all the cases. The least NCC value is for sharpening attack (0.9963). The least NCC value is for sharpening attack (0.9963). The value of NCC is higher for BO in cases of Salt and pepper noise (0.01), Histogram equalization, and Motion blur (5, 5). The value of NCC is higher for PSO in cases of Average filter (7×7), and Median filter (7×7). Rest in all the cases the values of NCC are similar. Table 4 shows the NCC values against various attacks with PSO. Table 5 shows the NCC values against various attacks with BO. Tables 6 and 7 shows the NCC values against hybrid attacks with PSO and BO respectively.

6.3 Time complexity analysis

In Table 8, the time taken during the process of hybrid watermark generation, watermark embedding and extraction of watermarks are evaluated. The minimum time is taken by S1 (1.7436 s) and the maximum time is for S20 (1.7521 s). The overall timing analysis shows that the proposed method is computationally efficient and also suitable for real-time applications.

Table 3 Values of performance parameters (BO-without attack)

Input slices (S)	PSNR (dB)	MSE	SSIM	NCC	BER	KLD	JSD
S1	51.0722	8.0401e-06	0.9999	1.0000	0.0030	0	0.0030
S2	51.0486	8.1087e-06	0.9999	1.0000	0.0033	0	0.0024
S3	51.1068	7.9503e-06	0.9998	1.0000	0.0029	0	0.0024
S4	51.2103	7.8677e-06	0.9999	1.0000	0.0029	0	0.0024
S5	51.2086	7.7324e-06	0.9998	1.0000	0.0033	0	0.0027
S6	51.3932	7.6405e-06	0.9999	1.0000	0.0030	0	0.0030
S7	51.4603	7.3228e-06	0.9999	1.0000	0.0034	0.0283	0.0065
S8	51.6051	7.2292e-06	0.9998	1.0000	0.0033	0.0294	0.0068
S9	51.4582	7.3858e-06	0.9998	1.0000	0.0033	0.0324	0.0075
S10	51.3957	7.4933e-06	0.9998	1.0000	0.0035	0.0306	0.0072
S11	51.4842	7.4091e-06	0.9999	1.0000	0.0031	0.0300	0.0071
S12	51.5279	7.2294e-06	0.9999	1.0000	0.0034	0.0300	0.0071
S13	51.4136	7.4183e-06	0.9999	1.0000	0.0030	0.0355	0.0081
S14	51.4208	7.4157e-06	0.9999	1.0000	0.0032	0.0356	0.0081
S15	51.4718	7.4174e-06	0.9999	1.0000	0.0034	0.0354	0.0081
S16	51.4244	7.3962e-06	0.9999	1.0000	0.0029	0.0202	0.0045
S17	51.4056	7.5566e-06	0.9999	1.0000	0.0033	0.0368	0.0085
S18	51.3831	7.5235e-06	0.9999	1.0000	0.0039	0.0353	0.0081
S19	51.2587	7.6988e-06	0.9999	1.0000	0.0029	0.0324	0.0076
S20	51.3141	7.6818e-06	0.9999	1.0000	0.0030	0.0334	0.0078
S21	51.2146	7.7564e-06	0.9999	1.0000	0.0030	0	0.0040

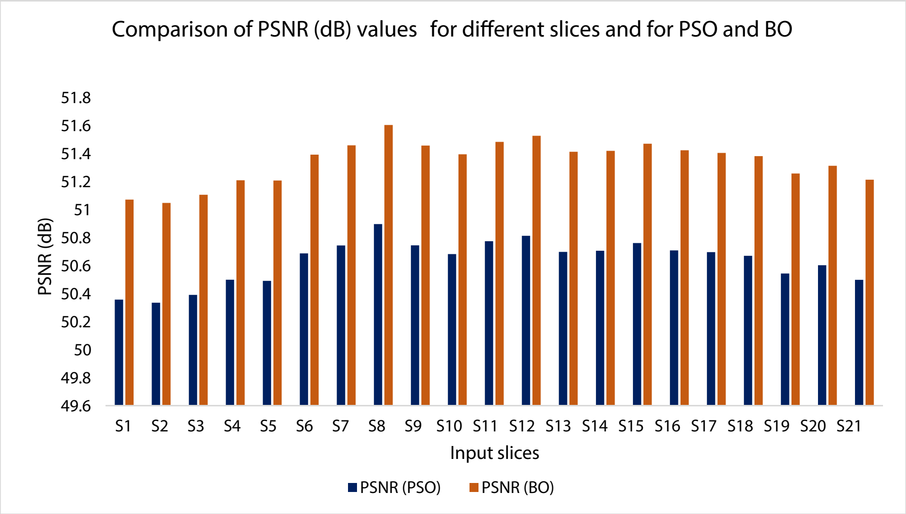


Fig. 12 Comparison of PSNR (dB) values for different slices and for PSO and BO

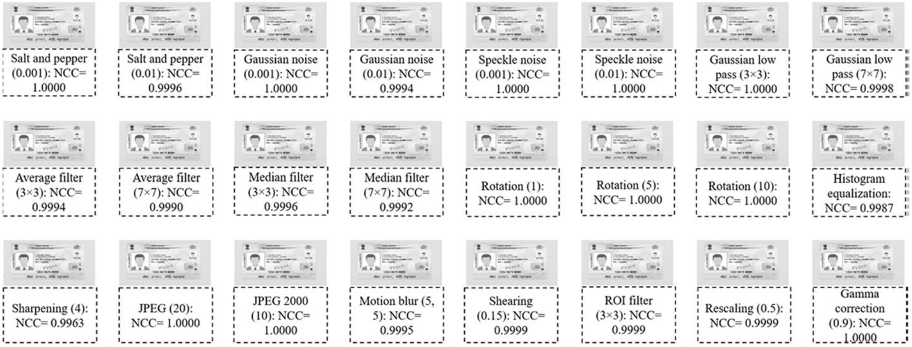

Salt and pepper (0.001): NCC= 1.0000
Salt and pepper (0.01): NCC= 0.9996
Gaussian noise (0.001): NCC= 1.0000
Gaussian noise (0.01): NCC= 0.9994
Speckle noise (0.001): NCC= 1.0000
Speckle noise (0.01): NCC= 1.0000
Gaussian low pass (3×3): NCC= 1.0000
Gaussian low pass (7×7): NCC= 0.9998
Average filter (3×3): NCC= 0.9994
Average filter (7×7): NCC= 0.9990
Median filter (3×3): NCC= 0.9996
Median filter (7×7): NCC= 0.9992
Rotation (1): NCC= 1.0000
Rotation (5): NCC= 1.0000
Rotation (10): NCC= 1.0000
Histogram equalization: NCC= 0.9987
Sharpening (4): NCC= 0.9963
JPEG (20): NCC= 1.0000
JPEG 2000 (10): NCC= 1.0000
Motion blur (5, 5): NCC= 0.9995
Shearing (0.15): NCC= 0.9999
ROI filter (3×3): NCC= 0.9999
Rescaling (0.5): NCC= 0.9999
Gamma correction (0.9): NCC= 1.0000

Fig. 13 Extracted watermark images and corresponding NCC values against various attacks (S1-PSO)

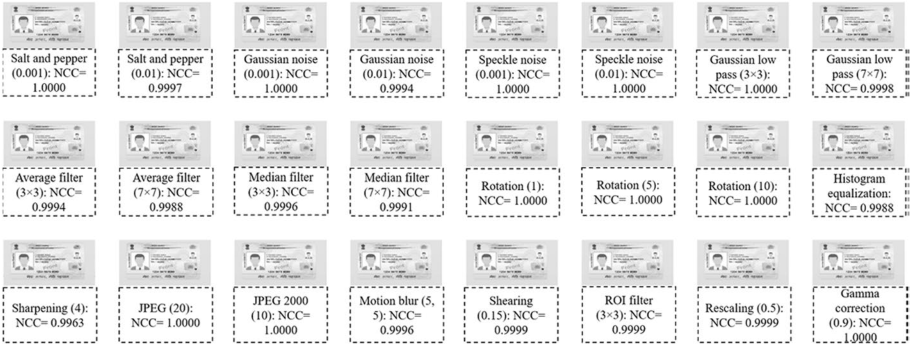

Salt and pepper (0.001): NCC= 1.0000
Salt and pepper (0.01): NCC= 0.9997
Gaussian noise (0.001): NCC= 1.0000
Gaussian noise (0.01): NCC= 0.9994
Speckle noise (0.001): NCC= 1.0000
Speckle noise (0.01): NCC= 1.0000
Gaussian low pass (3×3): NCC= 1.0000
Gaussian low pass (7×7): NCC= 0.9998
Average filter (3×3): NCC= 0.9994
Average filter (7×7): NCC= 0.9988
Median filter (3×3): NCC= 0.9996
Median filter (7×7): NCC= 0.9991
Rotation (1): NCC= 1.0000
Rotation (5): NCC= 1.0000
Rotation (10): NCC= 1.0000
Histogram equalization: NCC= 0.9988
Sharpening (4): NCC= 0.9963
JPEG (20): NCC= 1.0000
JPEG 2000 (10): NCC= 1.0000
Motion blur (5, 5): NCC= 0.9996
Shearing (0.15): NCC= 0.9999
ROI filter (3×3): NCC= 0.9999
Rescaling (0.5): NCC= 0.9999
Gamma correction (0.9): NCC= 1.0000

Fig. 14 Extracted watermark images and corresponding NCC values against various attacks (S1-BO)

6.4 Capacity analysis

The capacity of the watermarking technique is evaluated by obtaining the ratio of the number of pixels in the watermark image and the number of pixels in the host image [27]. In the proposed method, RDWT is applied to input host image of size 256×256. After applying RDWT the size of the input image remains the same. During the embedding process, the RDWT is applied to the watermark image, so the maximum embeddable watermark size is 256×256. Therefore, the capacity of our proposed technique is (256×256)/(256×256) = 1.0.

6.5 Feature authentication

A distinctive feature point descriptor and local invariant rapid feature point detector are known as SURF. The initial stage of processing assigns an orientation within a sphere centered on the keypoint after determining its location [9]. The SURF detector uses integral images and the determinant of the Hessian matrix as a foundation to increase the speed of feature discovery.

Another binary descriptor is BRISK. A sample strategy is then used in the keypoint's neighborhood after keypoints have been chosen [13]. Short-distance pairs and long-distance pairs are two subgroups of the pairs of pixels surrounding the keypoint. The orientation of the feature point is established by computing local intensity gradients from

Table 4 Value of NCC against various attacks for different input images (PSO)

Attacks		S1	S3	S5	S7	S9	S11	S13	S15	S21
Salt and pepper	0.001	1.0000	0.9999	0.9999	1.0000	0.9999	0.9999	0.9999	1.0000	1.0000
	0.01	0.9996	0.9995	0.9996	0.9994	0.9997	0.9993	0.9996	0.9995	0.9995
Gaussian noise	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.01	0.9994	0.9993	0.9995	0.9992	0.9994	0.9995	0.9995	0.9992	0.9992
Speckle noise	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.01	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Gaussian low pass filter	3×3	1.0000	0.9999	1.0000	1.0000	0.9999	0.9999	0.9999	1.0000	1.0000
	7×7	0.9998	0.9995	0.9994	0.9997	0.9996	0.9997	0.9997	0.9998	0.9998
Average filter	3×3	0.9994	0.9992	0.9993	0.9994	0.9994	0.9991	0.9993	0.9993	0.9994
	7×7	0.9990	0.9989	0.9989	0.9990	0.9990	0.9990	0.9989	0.9990	0.9990
Median filter	3×3	0.9996	0.9995	0.9994	0.9994	0.9995	0.9996	0.9995	0.9994	0.9996
	7×7	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992
Rotation	1 °	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	5 °	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	10°	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Histogram equalization		0.9987	0.9986	0.9985	0.9986	0.9987	0.9986	0.9986	0.9985	0.9987
Sharpening	4	0.9963	0.9959	0.9961	0.9960	0.9963	0.9963	0.9962	0.9961	0.9963
JPEG	20	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
JPEG 2000	10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Motion blur	5, 5	0.9995	0.9994	0.9992	0.9993	0.9995	0.9995	0.9994	0.9995	0.9995
Shearing	0.15	0.9999	0.9998	0.9995	0.9995	0.9996	0.9997	0.9998	0.9999	0.9998
Region of interest filtering	3×3	0.9999	0.9998	0.9998	0.9997	0.9997	0.9996	0.9997	0.9998	0.9998
Rescaling	0.5	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
Gamma correction	0.9	1.0000	0.9999	0.9999	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000

Table 5 Value of NCC against various attacks for different input images (BO)

Attacks		S1	S3	S5	S7	S9	S11	S13	S15	S21
Salt and pepper	0.001	1.0000	0.9999	1.0000	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000
	0.01	0.9997	0.9996	0.9997	0.9994	0.9997	0.9994	0.9995	0.9996	0.9997
Gaussian noise	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.01	0.9994	0.9995	0.9994	0.9994	0.9995	0.9995	0.9995	0.9992	0.9994
Speckle noise	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	0.01	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Gaussian low pass filter	3×3	1.0000	0.9999	1.0000	1.0000	1.0000	0.9999	0.9999	0.9999	1.0000
	7×7	0.9998	0.9997	0.9997	0.9997	0.9996	0.9998	0.9997	0.9997	0.9998
Average filter	3×3	0.9994	0.9993	0.9993	0.9994	0.9994	0.9991	0.9993	0.9994	0.9994
	7×7	0.9989	0.9990	0.9989	0.9990	0.9990	0.9989	0.9990	0.9991	0.9991
Median filter	3×3	0.9996	0.9996	0.9995	0.9994	0.9995	0.9996	0.9996	0.9995	0.9996
	7×7	0.9991	0.9992	0.9992	0.9991	0.9990	0.9993	0.9992	0.9991	0.9991
Rotation	1 °	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	5 °	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	10°	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Histogram equalization		0.9988	0.9988	0.9987	0.9987	0.9986	0.9985	0.9986	0.9987	0.9985
Sharpening	4	0.9963	0.9960	0.9961	0.9962	0.9963	0.9963	0.9961	0.9960	0.9963
JPEG	20	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
JPEG 2000	10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Motion blur	5, 5	0.9996	0.9994	0.9995	0.9995	0.9996	0.9995	0.9993	0.9994	0.9995
Shearing	0.15	0.9999	0.9998	0.9999	0.9998	0.9997	0.9996	0.9998	0.9999	0.9999
Region of interest filtering	3×3	0.9999	0.9998	0.9997	0.9997	0.9997	0.9998	0.9999	0.9998	0.9999
Rescaling	0.5	0.9999	0.9997	0.9997	0.9998	0.9997	0.9997	0.9999	0.9999	0.9999
Gamma correction	0.9	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000

Table 6 Value of NCC against hybrid attacks for different input images (PSO)

Attacks		S2	S4	S6	S8	S10	S12
Average filter, Gaussian low pass	3×3, 3×3	0.9941	0.9910	0.9942	0.9989	0.9982	0.9981
Salt and pepper, Gaussian noise	0.001, 0.01	0.9775	0.9755	0.9752	0.9772	0.9799	0.9779
Rotation, Shearing	2°, 0.15	0.9441	0.9543	0.9551	0.9472	0.9571	0.9570
Rotation, Salt and pepper	1°, 0.001	0.9990	0.9990	0.9991	0.9993	0.9991	0.9992
Shearing, Salt and pepper	0.15, 0.001	0.9588	0.9551	0.9599	0.9591	0.9499	0.9592
Salt and pepper, Rotation, Speckle	0.001, 1, 0.01	0.9972	0.9971	0.9970	0.9970	0.9969	0.9969

Table 7 Value of NCC against hybrid attacks for different input images (BO)

Attacks		S2	S4	S6	S8	S10	S12
Average filter, Gaussian low pass	3×3, 3×3	0.9943	0.9912	0.9942	0.9990	0.9983	0.9984
Salt and pepper, Gaussian noise	0.001, 0.01	0.9779	0.9754	0.9755	0.9778	0.9799	0.9779
Rotation, Shearing	1°, 0.15	0.9443	0.9549	0.9553	0.9478	0.9573	0.9571
Rotation, Salt and pepper	1°, 0.001	0.9991	0.9992	0.9991	0.9993	0.9991	0.9991
Shearing, Salt and pepper	0.15, 0.001	0.9589	0.9552	0.9599	0.9592	0.9499	0.9593
Salt and pepper, Rotation, Speckle	0.001, 1, 0.01	0.9973	0.9973	0.9971	0.9971	0.9970	0.9971

Table 8 Time analysis for different input images (in seconds)

Input Images	S1	S5	S10	S15	S20
Hybrid watermark generation + Embedding	1.4127	1.4210	1.4130	1.4129	1.4209
Extraction (Aadhar + MAC + Patient code)	0.3309	0.3310	0.3313	0.3308	0.3312
Total time	1.7436	1.7520	1.7443	1.7437	1.7521

long distance pairs. This orientation rotates short distance pairs. Short distance pair-wise brightness comparison test data are used to create a BRISK description.

SURF and BRISK features are used to detect that the copyright protection does not harm the ROI regions of brain image. If these regions are distorted or manipulated by the algorithm used then the scheme is inefficient. Table 9 shows the SURF and BRISK feature matching with and without attack. Figures 15 and 16 shows the number matched SURF and BRISK features for different slices (without attack) respectively. Figures 17 and 18 shows number of SURF and BRISK matched against rotation attack respectively. By evaluating the results of matched features, it is clear that the proposed technique does not distort the ROI of medical images (Fig. 19). Table 10 shows the extracted hybrid watermark images.

Table 9 SURF and BRISK feature matching with and without attack

Input slices	SURF feature matching points		BRISK feature matching points	
	Without attack	Rotation (90°)	Without attack	Rotation (90°)
S1	92	83	126	67
S2	102	107	119	65
S3	141	147	150	90
S4	170	173	131	88
S5	194	167	168	113
S6	216	235	175	133
S7	241	234	181	136
S8	241	234	293	179
S9	226	216	253	166
S10	261	247	330	181
S11	279	279	301	210
S12	283	268	299	178
S13	286	256	328	223
S14	247	247	289	156
S15	239	239	250	153
S16	218	216	349	173
S17	218	215	234	148
S18	250	220	278	193
S19	255	248	335	208
S20	216	207	329	183
S21	170	152	258	145

Algorithm 4 SURF and BRISK feature matching

```
I1= imread('host NIfTI image')
I2= imread('attacked watermarked image')
points1 = detectSURFFeatures(I1)
points2 = detectSURFFeatures(I2)
Visualization←points1 and points2
[features1, valid_points1] = extractFeatures(I1, points1)
[features2, valid_points2] = extractFeatures(I2, points2)
Visualization←valid_points1 and valid_points2
indexPairs = matchFeatures(features1, features2)
matchedPoints1 = valid_points1(indexPairs(:,1))
matchedPoints2 = valid_points2(indexPairs(:,2))
Visualization←matchedPoints 1 and matchedPoints 2
points3 = detectBRISKFeatures(I1,'MinContrast',0.001)
points4 = detectBRISKFeatures(I2,'MinContrast',0.001)
Visualization←points3 and points4
[features1, valid_points3] = extractFeatures(I1, points3)
[features2, valid_points4] = extractFeatures(I2, points4)
Visualization←valid_points3 and valid_points4
indexPairs = matchFeatures(features3, features4)
matchedPoints3 = valid_points3(indexPairs(:,1))
matchedPoints4 = valid_points4(indexPairs(:,2))
Visualization←matchedPoints3 and matchedPoints4
```

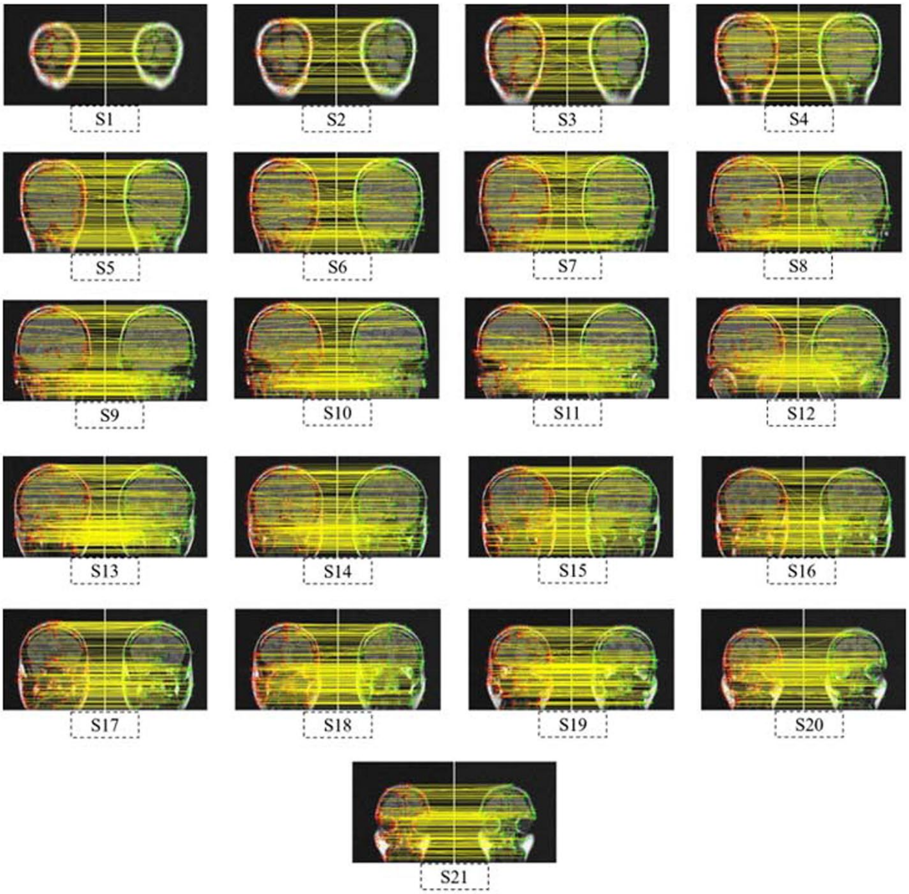


Fig. 15 Matched SURF features for different slices (Without attack)

7 Comparison of results

The comparison of the proposed method with other existing techniques are shown in this section. Table 11 shows the comparison of PSNR (dB) values in which the PSNR values for different schemes are 34.0455 in [1], 45.5891 in [2], 44.9488 in [5], 45.8186 in [7], and 47.12 in [15]. The presented work is having PSNR of 51.6051 dB which is much higher in comparison with other techniques, it depicts that the proposed work is highly imperceptible. Figure 20 shows the graphical comparison of PSNR values. Table 12 shows the comparison of NCC values and it is clear that proposed work is highly robust

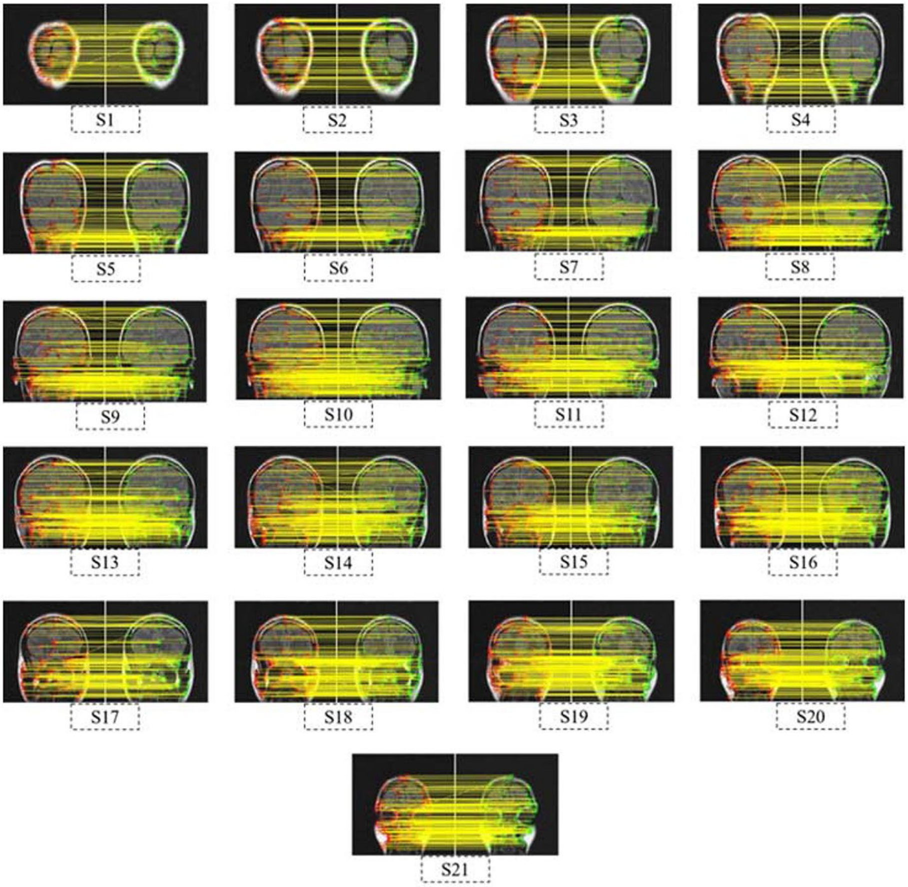


Fig. 16 Matched BRISK features for different slices (Without attack)

in comparison with other mentioned techniques. Table 13 shows the time complexity comparison. Table 14 shows the comparison of matched features against Rotation attack. The proposed method authenticates the features of the watermarked image more effectively in comparison with other existing technique.

8 Conclusion

PBNHWA technique uses PSO and BO to obtained the optimized scaling factor which maintains the balance between various performance parameters. For embedding RDWT, RSVD, DCT, and HD are used for achieving efficient performance results. The security of

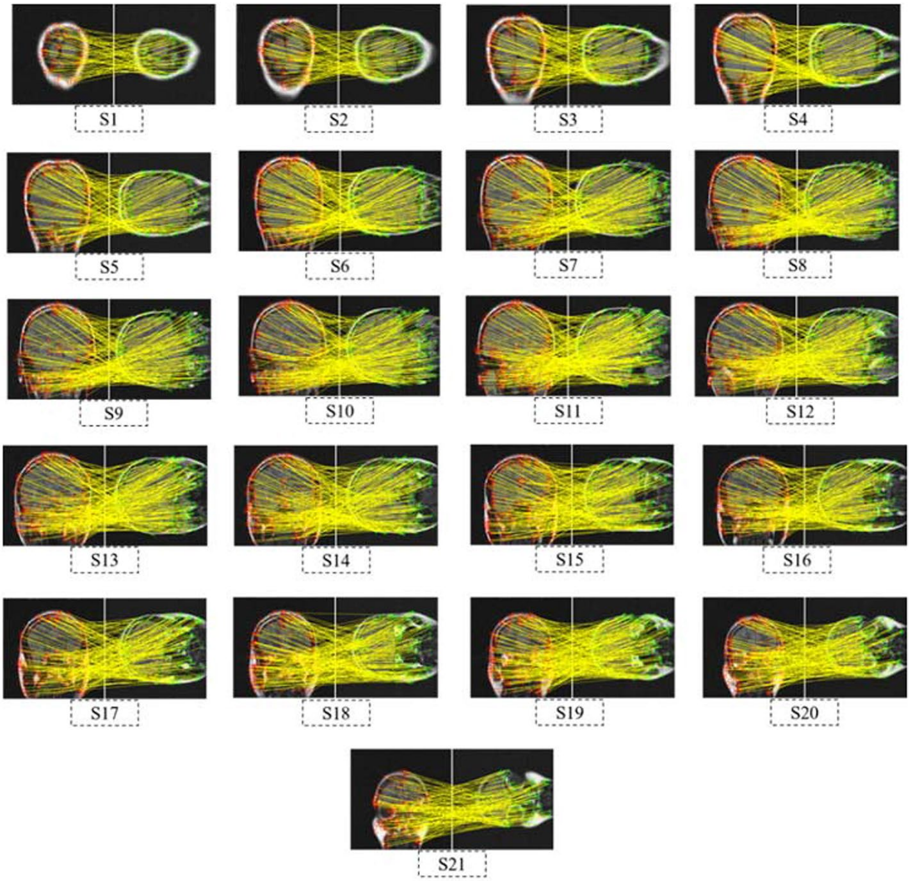


Fig. 17 Matched SURF features for different slices (Rotation-90°)

the proposed method is enhanced by using hybrid watermarks whereas SURF and BRISK based ROI are verified to demonstrate the superiority of the presented algorithm. The proposed method is tested under various attacks and found to be highly robust. The significant features of medical images are also successfully authenticated with no distortion for telemedicine applications. The proposed watermarking technique can be used for the copyright protection, security enhancement, improved imperceptibility, robustness, features authentication, and identity verification. Although, the proposed technique delivers sufficient robustness and imperceptibility but still there is a scope of improvement. The presented work can also be tested and verified by the hybrid optimization technique under different set of attacks. This task can be considered as the extension of the proposed work

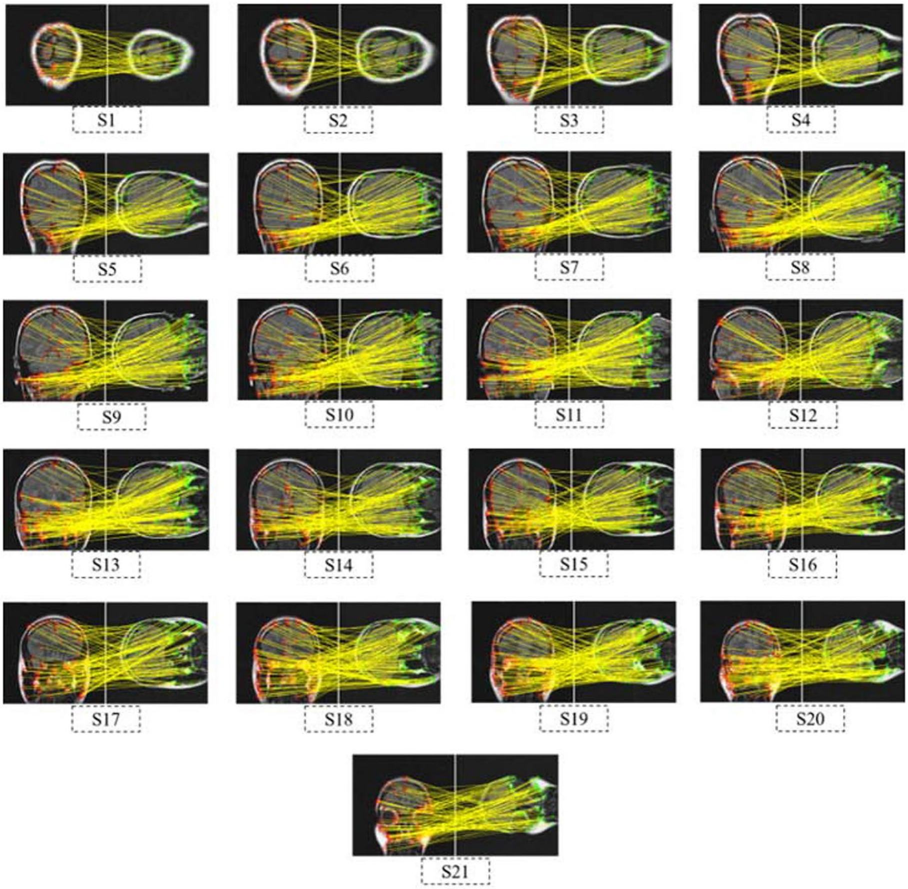


Fig. 18 Matched BRISK features for different slices (Rotation-90°)

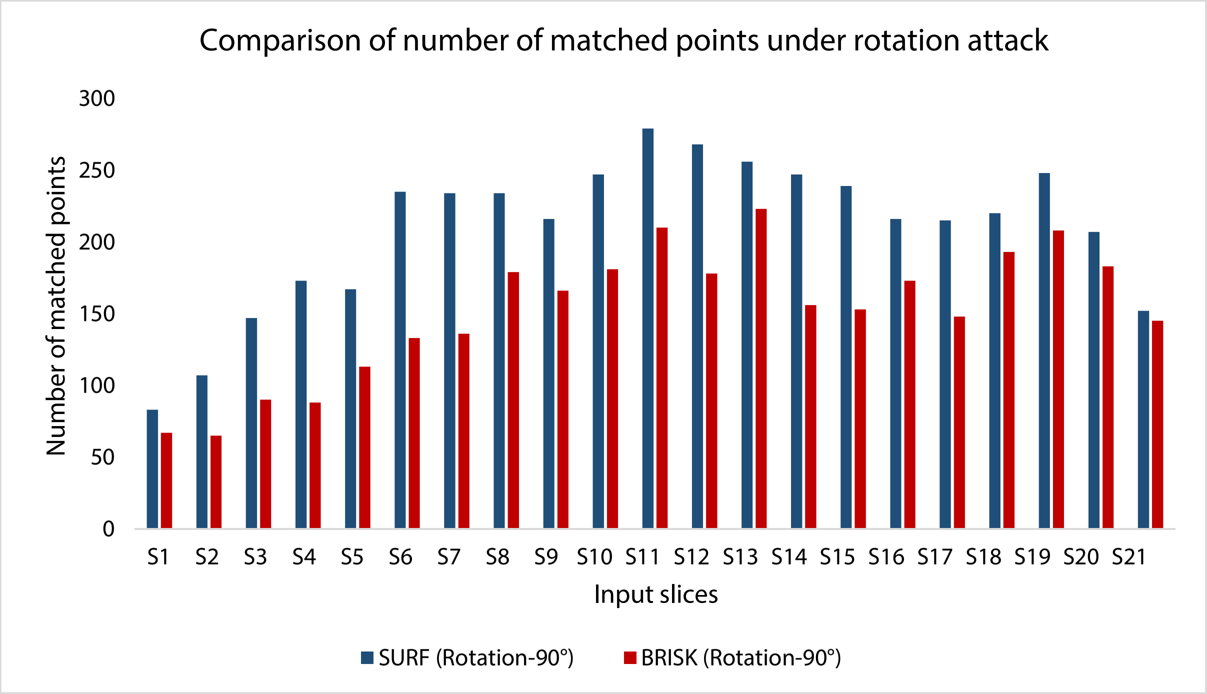


Fig. 19 Comparison of number of matched points under rotation attack (SURF and BRISK)

Table 10 Extracted hybrid watermark images

Attacks	PSO		BO	
	MAC	Patient code	MAC	Patient code
Salt and pepper noise 0.001	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Gaussian noise 0.001	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Speckle noise 0.001	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Gaussian low pass filter 7×7	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Average filter 3×3	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Median filter 3×3	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE
Rotation 10°	84A938D7266C	NAMEAGESEXD ISEASECODE	84A938D7266C	NAMEAGESEXD ISEASECODE

Table 11 Comparison of PSNR (dB) values for different existing schemes

Schemes	[1]	[2]	[5]	[7]	[15]	Proposed scheme (BO)
PSNR	34.0455	45.5891	44.9488	45.8186	47.12	51.6051

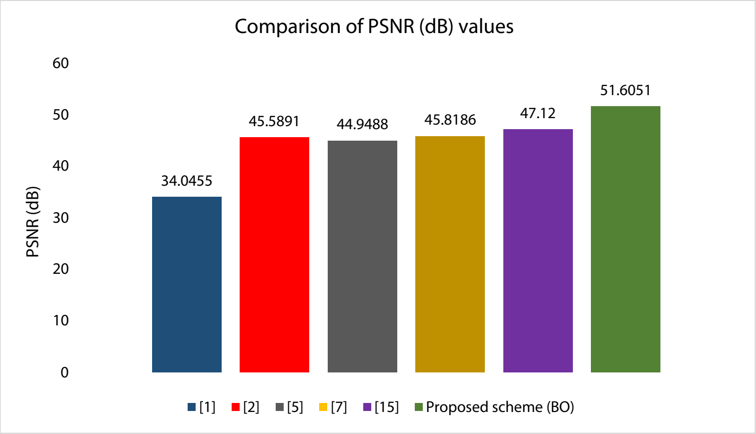


Fig. 20 Comparison of PSNR values against different schemes

Table 12 Comparison of NCC values with other existing techniques

Attacks		[1]	[2]	[5]	[15]	[20]	Proposed technique
No attack		0.9851	0.9985	0.9954	0.9997	0.9869	1.0000
Speckle noise	0.005	0.7564	0.9985	0.9939	-	0.9947	1.0000
	0.001	0.7564	0.9995	-	-	0.9947	1.0000
Salt and pepper	0.001	0.9251	0.9981	0.9939	0.9491	0.8761	1.0000
Median filter	2×2	0.6789	0.7549	0.9899	-	0.9099	0.9997
	3×3	-	0.7341	0.9932	-	0.9290	0.9996
Gaussian noise	0.001	0.5198	0.9920	0.9931	-	0.8311	1.0000
	0.01	-	0.9655	0.9930	-	-	0.9994
Sharpening	0.1	0.6381	0.8716	0.9932	-	0.8042	1.0000
	0.2	-	-	-	0.9347	-	1.0000
JPEG compression	50	0.9388	0.9825	0.9921		0.9626	1.0000
	70	-	-	-	0.9564	-	1.0000
Rotation	1°	-	-	0.9936		-	1.0000

Table 13 Comparison of time complexity

Schemes	[1]	[20]	Proposed
Embedding time	2.3141	2.0256	1.4127
Extraction time	1.3760	1.5987	0.3309
Total time	3.6909	3.6243	1.7436

Table 14 Comparison of number of matched features against Rotation (90°)

[8]	Proposed	
	SURF	BRISK
159 (Patient 1)	279 (S11)	210 (S11)
122 (Patient 2)	268 (S12)	178 (S12)
177 (Patient 3)	256 (S13)	223 (S13)

for future studies. Deep learning models can also be combined with traditional watermarking approaches to enhance the performance of the proposed work.

Funding This research received no specific grant from any funding agency.

Data availability My manuscript has no associated data.

Declarations

Ethics approval and consent to participate This article does not contain any studies with animals performed by any of the authors.

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Anand A, Singh AK (2020) An improved DWT-SVD domain watermarking for medical information security. Comput Commun 152:72–80. https://doi.org/10.1016/j.comcom.2020.01.038
2. Anand A, Singh AK (2022) Hybrid nature-inspired optimization and encryption-based watermarking for e-healthcare. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2022.3140862
3. Anand A, Singh AK, Zhou H (2022) ViMDH: Visible-imperceptible medical data hiding for internet of medical things. IEEE Trans Industr Inf 19(1):849–856
4. Awasthi D, Srivastava VK (2023) Hessenberg decomposition-based medical image watermarking with its performance comparison by particle swarm and JAYA optimization algorithms for different wavelets and its authentication using AES. Circ Syst Signal Process 8:1–32
5. Awasthi D, Srivastava VK (2022) LWT-DCT-SVD and DWT-DCT-SVD based watermarking schemes with their performance enhancement using Jaya and Particle swarm optimization and comparison of results under various attacks. Multimedia Tools Appl 81(18):25075–25099
6. Awasthi D, Srivastava VK (2023) Performance enhancement of SVD based dual image watermarking in wavelet domain using PSO and JAYA optimization and their comparison under hybrid attacks. Multimedia Tools Appl 82(23):1–33
7. Awasthi D, Srivastava VK (2022) Robust, imperceptible and optimized watermarking of DICOM image using Schur decomposition, LWT-DCT-SVD and its authentication using SURF. Multimedia Tools Appl 82:1–35
8. Awasthi D, Khare P, Srivastava VK (2023) BacterialWmark: telemedicine watermarking technique using bacterial foraging for smart healthcare system. J Electron Imaging 32(4):042107
9. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. Lect Notes Comput Sci 3951:404–417
10. Cox RW, Ashburner J, Breman H, Fissell K, Haselgrove C, Holmes CJ, Lancaster JL, Rex DE, Smith SM, Woodward JB, Strother SC (2004) A (sort of) new image data format standard: NiFTI-1. In: 10th Annual Meeting of Organisation of Human Brain Mapping, Budapest. http://nifti.nimh.nih.gov/nifti-1/documentation/hbm_nifti_2004.pdf
11. Khare P, Srivastava VK (2021) A novel dual image watermarking technique using momomorphic transform and DWT. J Intell Syst 30(1):297–311. https://doi.org/10.1515/jisys-2019-0046
12. Ji H, Yu W, Li Y (2016) A rank revealing randomized singular value decomposition (R3SVD) algorithm for low-rank matrix approximations. arXiv preprint arXiv:1605.08134. https://doi.org/10.48550/arXiv.1605.08134
13. Leutenegger S, Chli M, Siegwart RY (2011) BRISK: Binary robust invariant scalable keypoints. In: 2011 International conference on computer vision. Ieee, pp 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542
14. McLoone M, McCanny JV (2002) Efficient single-chip implementation of SHA-384 and SHA-512. In: 2002 IEEE International Conference on Field-Programmable Technology, 2002. (FPT). Proceedings. IEEE, pp 311–314. https://doi.org/10.1109/FPT.2002.1188699

15. Singh KU, Aljrees T, Kumar A, Singh T (2023) Secure NIfTI image authentication scheme for modern healthcare system. Appl Sci 13(9):5308
16. Singh KU, Kumar A, Singh T, Ram M (2022) Image-based decision making for reliable and proper diagnosing in NIFTI format using watermarking. Multimedia Tools Appl 81(27):39577–39603
17. Sinhal R, Sharma S, Ansari IA, Bajaj V (2022) Multipurpose medical image watermarking for effective security solutions. Multimedia Tools Appl 81(10):14045–14063
18. Sweldens W (1996) The lifting scheme: a custom-design construction of biorthogonal wavelets. Appl Comput Harmon Anal 3(2):186–200
19. Sweldens W (1998) The lifting scheme: a construction of second-generation wavelets. SIAM J Math Anal 29(2):511–546
20. Thakur S, Singh AK, Kumar B, Ghrera SP (2020) Improved DWT-SVD-based medical image watermarking through hamming code and chaotic encryption. In: Dutta D, Kar H, Kumar C, Bhadauria V (eds) Advances in VLSI, communication, and signal processing. Lecture notes in electrical engineering, vol 587. Springer, Singapore. https://doi.org/10.1007/978-981-32-9775-3_80
21. Yang XS, Hossein Gandomi A (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483
22. Tan J, Liao X, Liu J, Cao Y, Jiang H (2022) Channel attention image steganography with generative adversarial networks. IEEE Trans Netw Sci Eng 9(2):888–903. https://doi.org/10.1109/TNSE.2021.3139671
23. Liao X, Yu Y, Li B, Li Z, Qin Z (2020) A new payload partition strategy in color image steganography. IEEE Trans Circ Syst Video Technol 30(3):685–696. https://doi.org/10.1109/TCSVT.2019.2896270
24. Liao X, Yin J, Chen M, Qin Z (2022) Adaptive payload distribution in multiple images steganography based on image texture features. IEEE Trans Dependable Secure Comput 19(2):897–911. https://doi.org/10.1109/TDSC.2020.3004708
25. Liao X, Li K, Zhu X, Liu KJR (2020) Robust detection of image operator chain with two-stream convolutional neural network. IEEE J Sel Top Signal Process 14(5):955–968. https://doi.org/10.1109/JSTSP.2020.3002391
26. Yousif SF, Abboud AJ, Alhumaima RS (2022) A new image encryption based on bit replacing, chaos and DNA coding techniques. Multimedia Tools Appl 81(19):27453–27493. https://doi.org/10.1007/s11042-022-12762-x
27. Yousif SF, Abboud AJ, Radhi HY (2020) Robust image encryption with scanning technology, the El-Gamal algorithm and chaos theory. IEEE Access 8:155184–155209. https://doi.org/10.1109/ACCESS.2020.3019216
28. Salim MZ, Abboud AJ, Yildirim R (2022) A visual cryptography-based watermarking approach for the detection and localization of image forgery. Electronics 11(1):136. https://doi.org/10.3390/electronics11010136
29. Zhang L, Wei D (2019) Dual DCT-DWT-SVD digital watermarking algorithm based on particle swarm optimization. Multimedia Tools Appl 78:28003–28023. https://doi.org/10.1007/s11042-019-07902-9

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