applied sciences

MDPI

Article

Optimizing Image Watermarking with Dual-Tree Complex Wavelet Transform and Particle Swarm Intelligence for Secure and High-Quality Protection

Abed Al Raoof Bsoul [1,*] and Alaa Bani Ismail [2]

[1] Information Systems Department, Yarmouk University, Irbid 21163, Jordan
[2] Computer Science Department, Yarmouk University, Irbid 21163, Jordan; ambaniismaeel@just.edu.jo
* Correspondence: raoofbsoul@yu.edu.jo

Abstract: Watermarking is a technique used to address issues related to the widespread use of the internet, such as copyright protection, tamper localization, and authentication. However, most watermarking approaches negatively affect the quality of the original image. In this research, we propose an optimized image watermarking approach that utilizes the dual-tree complex wavelet transform and particle swarm optimization algorithm. Our approach focuses on maintaining the highest possible quality of the watermarked image by minimizing any noticeable changes. During the embedding phase, we break down the original image using a technique called dual-tree complex wavelet transform (DTCWT) and then use particle swarm optimization (PSO) to choose specific coefficients. We embed the bits of a binary logo into the least significant bits of these selected coefficients, creating the watermarked image. To extract the watermark, we reverse the embedding process by first decomposing both versions of the input image using DTCWT and extracting the same coefficients to retrieve those corresponding bits (watermark). In our experiments, we used a common dataset from watermarking research to demonstrate the functionality against various watermarked copies and peak signal-to-noise ratio (PSNR) and normalized cross-correlation (NCC) metrics. The PSNR is a measure of how well the watermarked image maintains its original quality, and the NCC reflects how accurately the watermark can be extracted. Our method gives mean PSNR and NCC of 80.50% and 92.51%, respectively.

Keywords: watermarking; dual-tree complex wavelet transform; particle swarm optimization algorithm; imperceptibility

check for updates

Academic Editors: Héctor Manuel Pérez-Meana and Mariko Nakano-Miyatake

Received: 30 November 2024
Revised: 19 January 2025
Accepted: 23 January 2025
Published: 27 January 2025

Citation: Bsoul, A.A.R.; Ismail, A.B. Optimizing Image Watermarking with Dual-Tree Complex Wavelet Transform and Particle Swarm Intelligence for Secure and High-Quality Protection. *Appl. Sci.* **2025**, *15*, 1315. https://doi.org/10.3390/app15031315

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Online piracy has caused significant financial losses and job displacement in some countries, amounting to over USD 12 billion annually and 70,000 lost jobs according to the Recording Industry Association of America (RIAA) [1]. The increased use of the internet positively correlates with rising privacy concerns for digital media usage. To alleviate the effects of pirating of such media, digital watermarking is one solution. Digital watermarking is a form of data-hiding technology [2] that involves masking pre-designed information into digital media such as images, videos, or audio. There are various types of digital watermarking schemes, including those based on spatial and frequency domains, and they are classified as robust, semi-fragile, and fragile methods [3]. Of course, a given technique has its own advantages and disadvantages. However, the use of a specific method depends on the application itself.

The internet's widespread use has made it quick and effortless to distribute digital media such as audio, video, and images. However, this convenience has also resulted in critical issues involving copyright infringement and authentication. Watermarking is one of the most effective and widely used techniques to address these concerns. Typically, image watermarking techniques follow a general set of steps that include a watermark embedder and a watermark detector, as depicted in Figure 1.

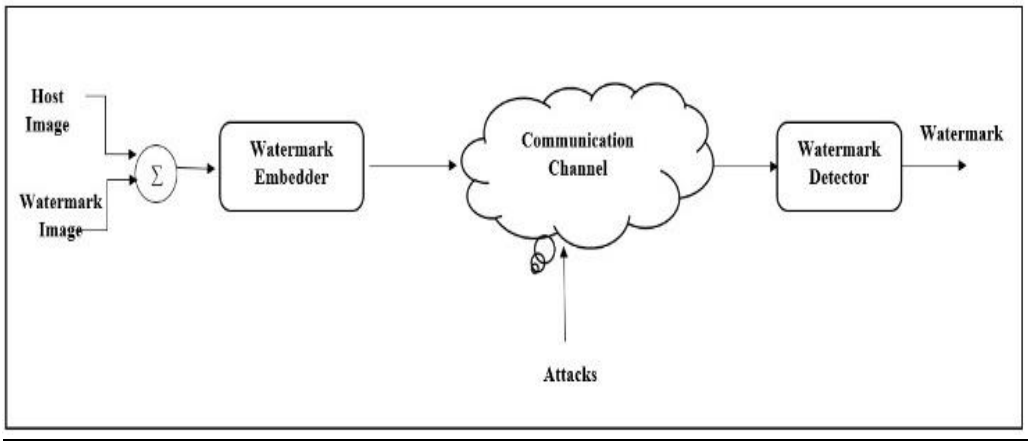


Figure 1. The architecture of a digital image watermarking system.

Watermarking systems need to have specific properties tailored to their intended applications, and as such there is no standard set of properties that all watermarking systems must meet. However, one of the most critical requirements is imperceptibility [4]. Image watermarking involves embedding secret data into a host image without degrading the quality of the watermarked image (imperceptibility). Imperceptibility requires that the quality of the original image is preserved even after inserting the watermark, such that it remains invisible to human eyes [5]. A watermarking approach is considered imperceptible if the host and watermarked images are visually indistinguishable. A robust watermarking scheme should not introduce artifacts into the images, and to assess imperceptibility, objective methods such as the peak signal-to-noise ratio (PSNR) are commonly used [6,7].

In addition to immanence, the watermarking method must also meet robustness requirements. A robust watermark is one that can largely survive variations in the host image (maintaining content with a level of computer independence) whilst other non-domain attack vectors are in play, i.e., cropping, resizing (scaling), contrast adjustment, and JPEG compression [4]. Balancing these numbers to find the watermark's secret formula, one that is strong enough but which does not cause severe degradation of the image, can be difficult. Thus, we should evaluate watermarking algorithms that are good enough to resist attacks with caution, and also aim to prevent significant changes in the quality of the embedded image.

The methods of watermarking that are best suited for proof of copyright and ownership in digital images are robust watermarking techniques. These are intended to withstand several image processing attacks that would either tamper with or erase the watermark. Therefore, robust watermarking techniques are employed to support intellectual property rights, including but not limited to digital image copyright, content authentication, and ownership proof [3]. Sensitive watermark-enabled methods are useful when one requires copyright protection, whilst fragile watermarking methods provide image authentication to confirm the integrity of an image without manipulation or modification. Semi-fragile watermarking is effective and strong, but it is also sensitive to higher image processing operations, and it can be employed for several tamper-detecting schemes [3].

Frequency domain watermarking schemes are generally preferred over the spatial domain watermarking approaches due to their inherent resistance against image processing attacks [8]. Also, as an embedding capacity extension is achieved, the frequency domain can provide more space while still achieving full imperceptibility [4]. However, the use of frequency domain techniques would incorporate some noise in the watermarked image which may be perceivable [9]. This has been countered by some researchers who came up with the idea of using optimization methods to select the best coefficients to embed watermarks, like particle swarm optimization (PSO) [10,11] and genetic algorithms (GAs) [12,13]. An alternative solution to increase the invisibility of watermarks is based on multiresolution decomposition techniques, such as dual-tree complex wavelet transform (DTCWT) [14,15].

In recent years, GAs have been successfully applied, in particular for optimization problems in watermarking. GA-based wavelet-based image watermarking schemes are given in [9]. The embedding energy of a DWT-based watermarking scheme has also been optimized by means of an artificial neural network. A fuzzy clustering technique was used in [16] to determine the optimal embedding locations for a DCT-based watermarking scheme. These AI techniques enhance the performance of watermarking schemes and their parameters (capacity, strength, or location).

In 2017, Liu et al. [17] proposed an image watermarking scheme in the contourlet transform domain, in connection with a probabilistic neural network. They distilled the watermark extraction to a great degree of robustness. Hamamoto and Kawamura [18] proposed an approach using an autoencoder to train two distinct neural networks (embedder and extractor) with a digital image using DCT coefficients. In [19], Ding et al. proposed a generalized approach of using deep neural networks in digital watermarking. They presented a method that obtained great robustness under common image processing attacks, such as filtering, compression, and geometric transformations.

Since the selection of transformation coefficients for embedding affects any watermarking method, researchers have exploited evolutionary algorithms to optimize the selection criteria. Researchers have used genetic algorithms widely in the watermarking of images in the last few years, and it has become an important research branch for many researchers [12,20–22]. Moreover, PSO has been proven to give promising accuracies in terms of robustness and impeccability. As such, many studies have exploited this fact in the literature, with particular highlights being outstanding approaches such as the work carried out in [11,23,24]. Other interesting approaches can be found in the literature, incorporating nature-inspired optimization algorithms such as artificial bee colony optimization [25,26], the bat algorithm [27], and gray wolf optimization [28].

Despite the advances in digital watermarking techniques, significant challenges remain in balancing imperceptibility and robustness. Conventional methods like DWT, DCT, or DFT usually leave visible artifacts on images or fail in preserving the integrity of the watermark under common malicious attacks (e.g., cropping, compression, and scaling). Although optimization algorithms have been applied in watermarking, their integration with newer wavelet-based transforms, like DTCWT, has not been fully explored. This study will address these gaps by exploring the following research questions:

- How can the use of DTCWT and PSO be incorporated to improve the imperceptibility of watermarked images while retaining high robustness?
- How well does the proposed method compare with traditional techniques like DWT and DCT in terms of PSNR (peak signal-to-noise ratio) and NCC (normalized cross-correlation) under standard image-processing manipulations such as compression, cropping, and scaling?
- What are the practical challenges involved in this watermarking scheme when applying it to real-world digital media applications, and how can these limitation be addressed?

The motivation underlying this work is to contribute to the domain of image watermarking and to propose an invisible and fragile method using DTCWT along with PSO algorithms. The main theoretical contribution of this work is the exploitation of shift invariance, redundancy, and directionality selection properties of DTCWT, which when applied in image applications give better results than traditional transform techniques such as DWT, DCT or DFT. These features make the watermark more robust against a number of image processing attacks, while preserving exceptional image quality. In order to improve the method and reduce any distortion of the watermarked images, PSO is used as an optimization algorithm. The proposed application of PSO has been integrated very well in research, and in this work, it is used to determine better coefficient embedding with optimal performance that will lead to good robustness and less visual distortion in the watermarked image. Additionally, DTCWT has a parallel dual-tree structure that enables efficient computation and is perfectly reconstructable, and therefore, it is well suited for real-world applications requiring both quality and security [29]. However, the proposed method faces limitations in computational complexity, applicability to color images, and resistance to advanced attacks.

2. Methodology

2.1. Overall Research Design

The proposed watermarking system is composed of two main phases: watermark embedding and watermark extraction. The system works as follows: The watermark image and the host image are first pre-processed and then input into the watermarking system. In the embedding phase, the host image is transformed into the frequency domain by applying DTCWT. The embedding process is optimized to choose the most suitable coefficients for watermark insertion. The watermark is then inserted into the least significant bit (LSB) of the selected coefficients. In the extraction phase, the watermark image is extracted by reversing the embedding process. To evaluate the proposed approach, the similarity between the watermark image and the extracted watermark is measured by computing the normalized cross-correlation (NCC). Additionally, the imperceptibility of the proposed approach is evaluated by calculating the peak signal-to-noise ratio (PSNR) of the watermarked images. Figure 2 illustrates the framework of the proposed approach.

2.2. Dataset

In this study, we used a standard dataset commonly employed in watermarking research. The experiments were performed on seventeen standard testing images: Lena, Cameraman, Baboon, Airplane, Boat, House1, House2, House3, Barbara, Elaine, Pirate, Zelda, Living room, Kiel, Lighthouse, Lake, and Pepper. All images were 512 $\times$ 512 in size, grayscale, and had an 8-bit depth.

The watermark image used in the experiments is the black and white logo for Yarmouk University, with a size of 100 $\times$ 109 bits. The logo of Yarmouk University is illustrated in Figure 3.

To ensure the reproducibility of the experiments, all images are publicly available and widely accepted in digital watermarking research. Further details about the dataset source can be provided upon request.

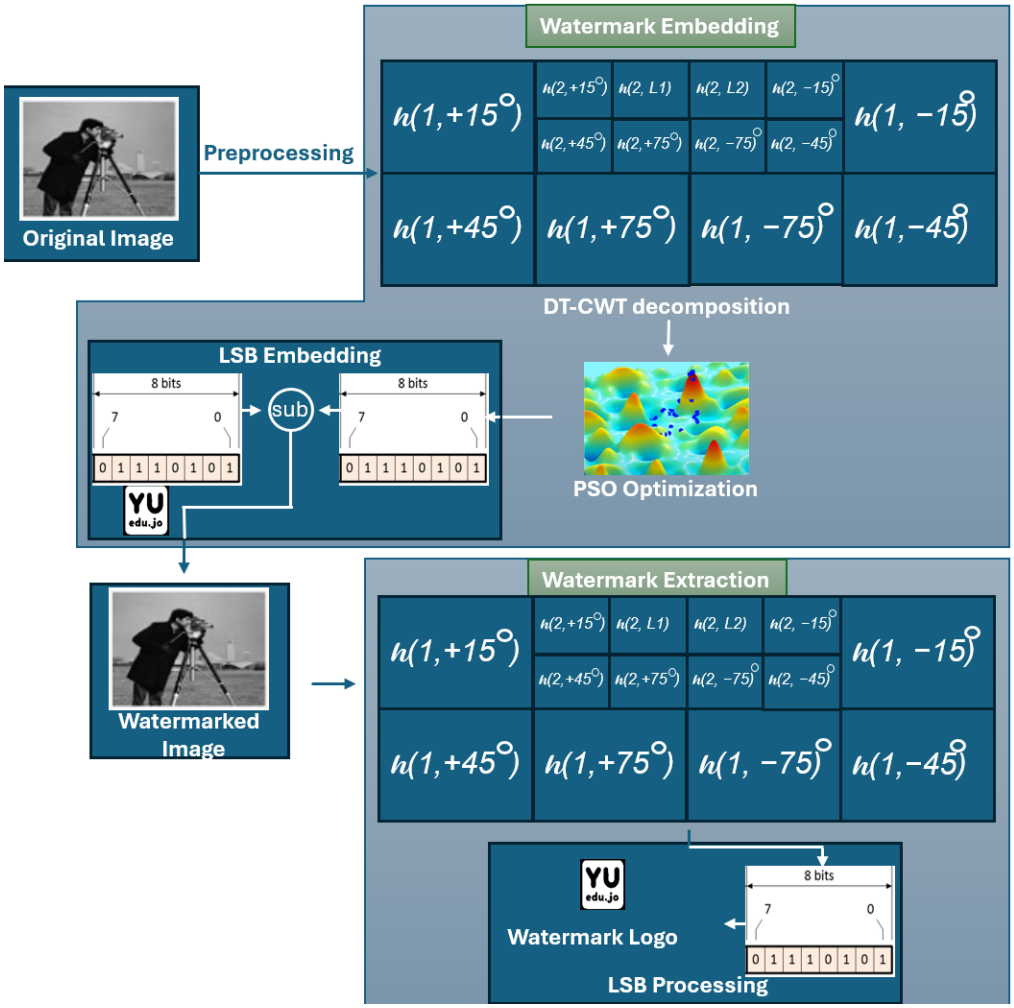


Figure 2. The overall design of the proposed method.



Figure 3. The logo used for embedding in this research. The actual logo size is 100 × 112, and the logo is black and white.

2.3. Watermark Embedding

Watermark embedding is the process of inserting the watermark image into the host image. In this research, this step can be split into three main sub-steps.

2.3.1. Frequency Domain Transformation

The insertion of the watermark image is performed in the frequency domain, and this involves converting the host image to the frequency domain using DTCWT. This transformation technique has gained attention due to its advantageous properties over DWT. The most notable properties are approximate shift invariance, directionally selective filters, and perfect reconstruction [29]. DTCWT utilizes two real DWT trees to produce the real and imaginary parts of the transformed image. After applying DTCWT, the image is decomposed into eight sub-bands, including Low–Low (LL), Low–High (LH), High–

Low (HL), and High–High (HH), for both the real and imaginary parts of a one-level decomposition. Figure 4 illustrates a multi-level decomposition for the LL sub-band. The coefficients from the LL sub-band are selected for embedding the watermark image.

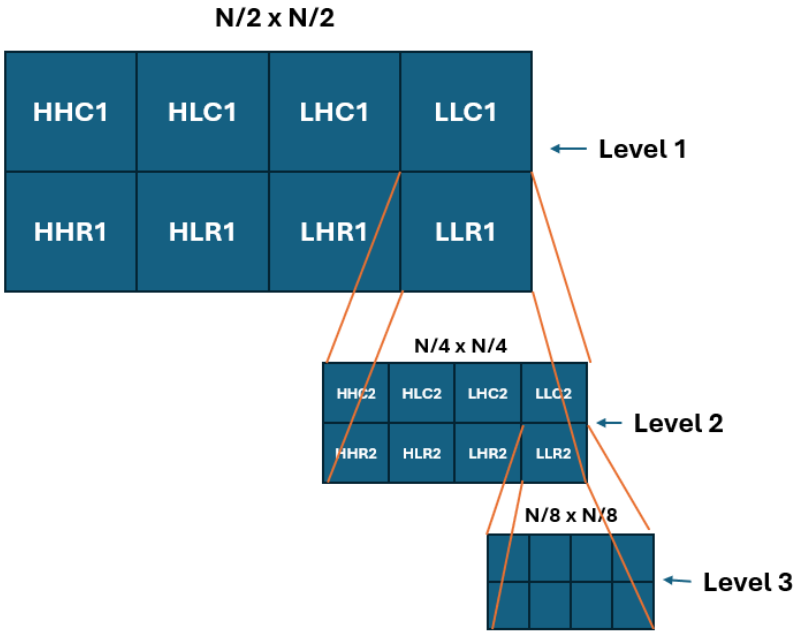


Figure 4. DTCWT sub-bands using multi-level decomposition.

The dual-tree complex wavelet transform differs from DWT in that it uses two orthogonal parallel transforms of the real wavelet, with the first DWT tree producing the real part and the second tree producing the imaginary part. Each tree contains a unique set of orthogonal filters. This results in DTCWT being able to obtain coefficients of six directions, namely $\pm 15^{\circ}$, $\pm 45^{\circ}$, and $\pm 75^{\circ}$, which solves the problem of poor directionality. At each level of decomposition, DTCWT produces two complex low-frequency sub-bands and six high-frequency complex detailed sub-bands, which are oriented by these angles. This property provides superior perceptual characteristics compared to DWT and enables the development of a watermarking approach with high imperceptibility. In mathematical terms, the low-frequency coefficients can be expressed as shown in Equation (1) [29].

$$x(\lambda, L, u, v) = \mathrm{Re}(x(\lambda, L, u, v)) + j\mathrm{Im}(x(\lambda, L, u, v)) \quad (1)$$

where λ is the scaling factor (wavelength), L is the decomposition level, Re is the real part of the tree, Im is the imaginary part of the tree, u and v indicate the location of the coefficient in each sub-band, and j represents the square root of -1.

In this equation, the image is factorized using both real and imaginary parts to preserve information during processing.

The high-frequency coefficients can be written as in Equation (2).

$$\begin{gathered} x(\hat{}, \theta, u, v) = \mathrm{Re}(x(\hat{}, \theta, u, v)) + j\mathrm{Im}(x(\hat{}, \theta, u, v)) \\ \theta \in \{-75, -45, -15, +15, +45, +75\} \end{gathered} \quad (2)$$

In the embedding process, the frequency component at locations u and v in the decomposition is represented by the complex coefficient after applying the DTCWT. Therefore, by modifying the least significant bits of these coefficients, the watermark data can be embedded without significantly altering the image's appearance.

2.3.2. Optimizing the Embedding Process

To embed a watermark in an image, we should first select frequency domain coefficients from the decomposed host image. Usually, this process is applied sequentially on the coefficients. However, in this research, we will use the particle swarm intelligence algorithm to optimize the selection of the coefficients. The optimization algorithm fitness function is used for minimizing degradation in the watermarked image.

PSO is a type of stochastic optimization algorithm and its process is inspired by the social interaction behavior of flocking birds or schooling fish, which differs significantly from traditional individual cognitive elements. This algorithm consists of a population of candidate solutions (particles) which exist in the search space and are given a fitness value for the maximization function. The PSO algorithm focuses on the purpose of minimizing or maximizing the objective function. All particles update their position based on their own experience (cognitive behavior) and development, as well as the best location reached so far by all other particles (social behavior).

For a brief overview of the PSO algorithm, the methodology was introduced in [30]. Each particle in the swarm consists of three 3-dimensional vectors, the current position (x_i), the best position reached by that particle ($pbest_i$), and the velocity (v_i), as shown in Equation (3). The velocity vector (v_i) controls the optimization process and enables information exchange between particles as well as utilizing the historical knowledge of each particle, as given in Equation (4). By adding the velocity coordinates (v_i) to the current position (x_i), the updated position of each particle (x'_i) is calculated.

$$v_i = \omega\, v_i + C_1 \varphi_1\, (p_{besti} - x_i) + C_2 \varphi_2 (g_{best} - x_i) \quad (3)$$

$$x'_i = x_i + v_i \quad (4)$$

where inertia weight (ω) is a constant that determines the balance between global and local search, while $\varphi 1$ and $\varphi 2$ represent uniformly distributed random values within [0, 1]. Additionally, positive acceleration constants C1 and C2 are used to scale the contribution of cognitive and social components, respectively, in the particle's movement.

As an example, suppose that a particle's current velocity is $v_i = 2$, its personal best is at position 5, and the global best is at position 8. If the inertia weight is $\omega = 0.7$ and C_1 and C_2 are both set to 1.5, with random values $\varphi_1 = 0.9$ and $\varphi_2 = 0.7$, the updated velocity would be

$$v_i = 0.7\ (2) + 1.5\ (0.9)\ (5 - x_i) + 1.5\ (0.7)\ (8 - x_i)$$

This gives the new velocity based on how far the particle is from its personal best and the global best, allowing it to move closer to an optimal solution.

PSO is applied to the coefficients obtained from the decomposition step. The pseudocode for finding the optimal set of coefficients using PSO is shown in Algorithm A1 (Appendix A).

As the algorithm shows, the coefficients are divided into *m* swarms, where each swarm has several particles. Each coefficient in a swarm is considered as a particle. The objective function is calculated over each particle in each swarm, and the best value in swarm *i* is denoted as $pbest_i$, while the best objective function over all swarms is considered as *gbest*. The inertia is calculated for each iteration and used as a weight contribution of the previous velocity. The updated velocity is computed from the weighted velocity in the previous iteration, along with the distance of particle *i* from the *pbesti* and the distance of particle *i* from *gbest*.

After finding out the velocity of particle *i*, we can generate a random number from [0, 1] and compare the obtained number with the sigmoid value of the new velocity. If the

random number is less than the sigmoid value, we will choose that particle (coefficient) as an embeddable particle by setting it to 1. Then, the *pbest* and *gbest* are updated and the iteration continues.

After optimizing the selected coefficients in the transformed image using PSO, we calculated their mean, variance, skewness, and kurtosis, as well as the entropy and the energy of the selected coefficients. We found that the selected coefficients exhibit a non-Gaussian distribution. As a result, this fact will help in identifying the embeddable coefficients during the reconstruction of the stego-image.

2.3.3. Optimizing PSO Parameters for Computational Efficiency

Particle swarm optimization (PSO) plays a critical role in determining the coefficients used for watermark embedding in the proposed method. However, the choice of PSO parameters significantly impacts not only the quality of the watermarking results but also the computational efficiency of the optimization process. To address this, we conducted a focused analysis of the key parameters involved in PSO: inertia weight (ω), cognitive component (C1), social component (C2), swarm size, and the number of iterations.

The optimization of PSO parameters was approached with the dual objectives of minimizing computational time while maintaining high performance in terms of imperceptibility and robustness. The following parameter ranges and configurations were examined based on insights from the existing literature and empirical observations:

- Inertia Weight (ω): Tested within the range [0.4, 0.9] to balance exploration and exploitation.
- Cognitive Component (C1): Varied between 0.5 and 2.0 to evaluate the effect of individual particle learning.
- Social Component (C2): Varied between 0.5 and 2.0 to assess the influence of swarm-wide cooperation.
- Swarm Size: Analyzed for sizes ranging from 20 to 50 particles to explore trade-offs between computational cost and optimization performance.
- Number of Iterations: Evaluated over a range of 50 to 200 iterations to identify the point of convergence for the objective function.

2.3.4. Insertion in the Least Significant Bit (LSB)

A digital image is composed of small dots called pixels, and each pixel is a series of zeros and ones that represent a given color. When masking the least significant bits of the color value with message bits, embedding takes place. This process is called the least significant bit (LSB) technique, and it is a widely used method in image manipulation. The more bits used in masking, the more noticeable the distortion in the image. On the other hand, using more bits in the masking process provides more embeddable data. As a result, a compromise should be made to balance the quality of the resulting watermarked image and the amount of data embedded. The process of LSB is repeated in the given manner until the whole message has been embedded. Although it has been used in various applications, such as information hiding, copyright protection, and authentication, LSB has some limitations, especially when implemented in a time domain. For example, the use of advanced steganalysis techniques may result in the whole method becoming corrupted. Therefore, to overcome these limitations, researchers have developed more complex steganography techniques that provide better security and greater capacity for hiding data while maintaining imperceptibility.

One such technique is the transformation of the image to the frequency domain and the application of LSB on selected coefficients of the transformation. By utilizing other image domains, larger amounts of data can be hidden while still maintaining excellent

transparency. As more steganalysis techniques are introduced, the acquisition of more advanced and secure steganography techniques becomes of paramount importance.

2.3.5. Watermarking Evaluation

According to the applied method for watermarking, the performance can change dramatically. To find out the performance of a given watermark method, one can measure the method's transparency. As artifacts are introduced during the watermarking process, the host image's quality is affected. Transparency is defined as the amount of information disclosed after the watermark operation. In practice, transparency is usually measured by evaluating the imperceptibility of the algorithm, which in turn is measured by an objective image quality metric called peak signal-to-noise ratio (PSNR). Measuring the PSNR of a method is straightforward and simple, and as a result, PSNR has become a widely used metric for watermarking methods. The PSNR is evaluated by comparing the host image I of size $N \times M$, with the watermarked image $_K$, as follows [6]:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{Max}^2}{\frac{1}{M \cdot N}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - K(i,j))^2}\right) \tag{5}$$

where

- Max: Maximum possible pixel value (e.g., 255 for 8-bit images).
- M and N: Dimensions of the image (rows and columns).
- I(i,j) Pixel value at position (i,j) in the original image.
- K(i,j): Pixel value at position (i,j) in the distorted or compressed image.

Alongside transparency, the degree of similarity between the watermarked image and the original image is also crucial in assessing the performance of a watermarking technique. Evaluating and measuring the similarity between the extracted watermark and the original one is essential in determining the level of distortion introduced by the method. Normalized cross-correlation (NCC) is used to measure image similarity between the original and extracted images in watermarking.

NCC is computed as follows: W, W′ are the original watermark image and the extracted watermark, respectively, while $\mu_W.\mu_{W'}$ represent the means of the original and extracted watermark.

$$NC\left(W.W'\right) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i\,.\,j) - \mu_W][W\prime(i\,.\,j) - \mu_{W\prime}]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[W'^{(i\,.\,j)} - \mu_{W'}\right]^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i\,.\,j) - \mu_W]^2}} \tag{6}$$

NCC is particularly useful in evaluating the accuracy of the watermark extraction process.

3. Results

The proposed scheme was evaluated using the dataset described in Section 2.2, and the validation of the proposed scheme was performed as described in Section 2.3.5, by calculating the PSNR between the host and watermarked images and the NCC between the embedded logo image and the extracted image logo.

The implementation of the approach was carried out in Python on a laptop computer equipped with an Intel Core i7 processor, with 32 GB RAM, and running on the Windows 10 operating system.

To embed the watermark logo into the host image, the host image was decomposed using DTCWT. The filters "near_sym_b" and "qshift_b" were empirically found to produce the best coefficients, and a decomposition level of four was chosen. The PSO algorithm

described in Section 2.3.2 was utilized to select the best embeddable coefficients from the LL sub-band.

The parameters of the PSO algorithm were optimized for computational efficiency using empirical settings. Specifically, the cognitive component (C1) and social component (C2) were both tested for a value which varied between 0.5 and 2.0. The inertia weight (ω), which controls the search direction, was tested in the range of a maximum weight (W_{max} = 0.9) and a minimum weight (W_{min} = 0.4). Additionally, $\varphi 1$ and $\varphi 2$ were assigned uniformly distributed random values within [0, 1]. The number of swarms was analyzed in sizes ranging from 20 to 50, and the number of iterations was evaluated in the range of 50 to 200 iterations in order to discover the convergence point of our objective function.

The findings of the PSO parameters' optimization underscore the importance of carefully selecting PSO parameters to enhance computational efficiency. Figure 5 demonstrates that $\omega = 0.7$ is the optimal inertia weight, achieving the best balance between computational efficiency and performance quality. At this ω value, PSNR reaches its peak while computational cost is at its minimum, highlighting it as the ideal value. The red dashed line effectively emphasizes this point. Figure 6 shows that $C1 = 1.5$ is the optimal cognitive component, achieving the highest PSNR while maintaining a decreasing computational cost. The red dashed line emphasizes this ideal balance between performance and efficiency.

The optimization results identified $C2 = 1.5$ as the optimal value, as illustrated in Figure 7. This value achieved the highest PSNR, while the computational cost increased gradually. The optimization process was also applied to the swarm size and number of iterations of PSO parameters to determine the optimal values that minimized the computational cost. The results of these optimization processes are presented in Figures 8 and 9.

It should be noted that this insertion process may introduce artifacts and distortion into the watermarked image. To evaluate the impact of these distortions, the PSNR was measured for the watermarked image. A visual comparison between some standard cover images before and after the watermark was applied, along with the PSNR between the two images, is provided in Table 1.

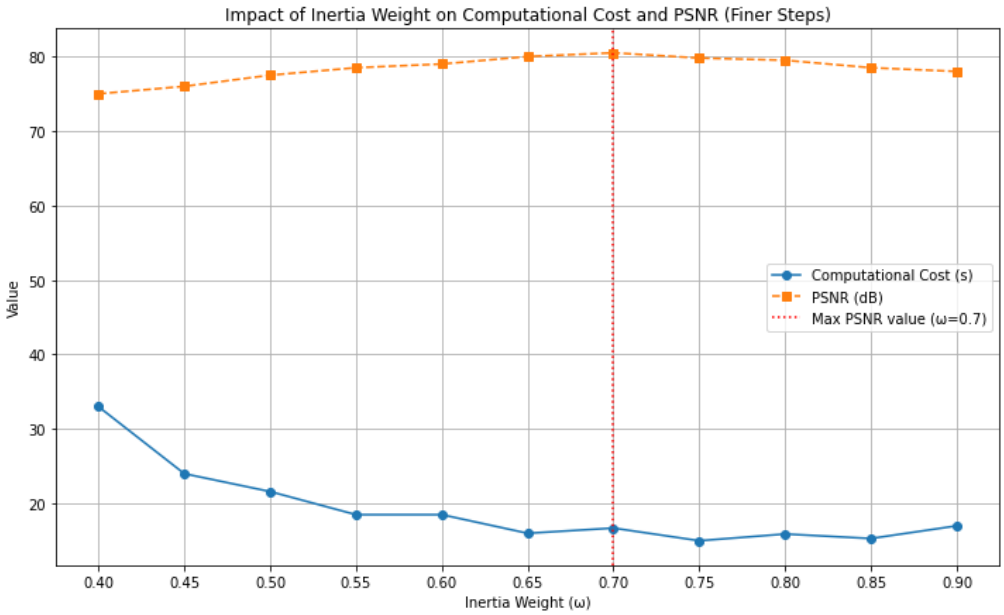


Figure 5. Impact of inertia weight (ω) on computational cost and PSNR.

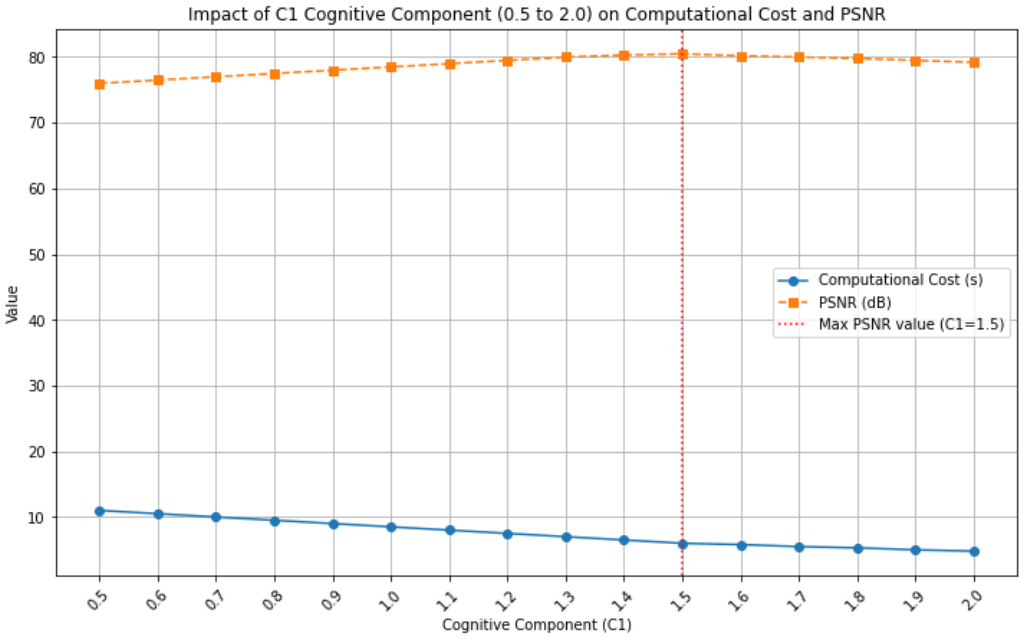


Figure 6. Impact of cognitive component (*C*1) on computational cost and PSNR.

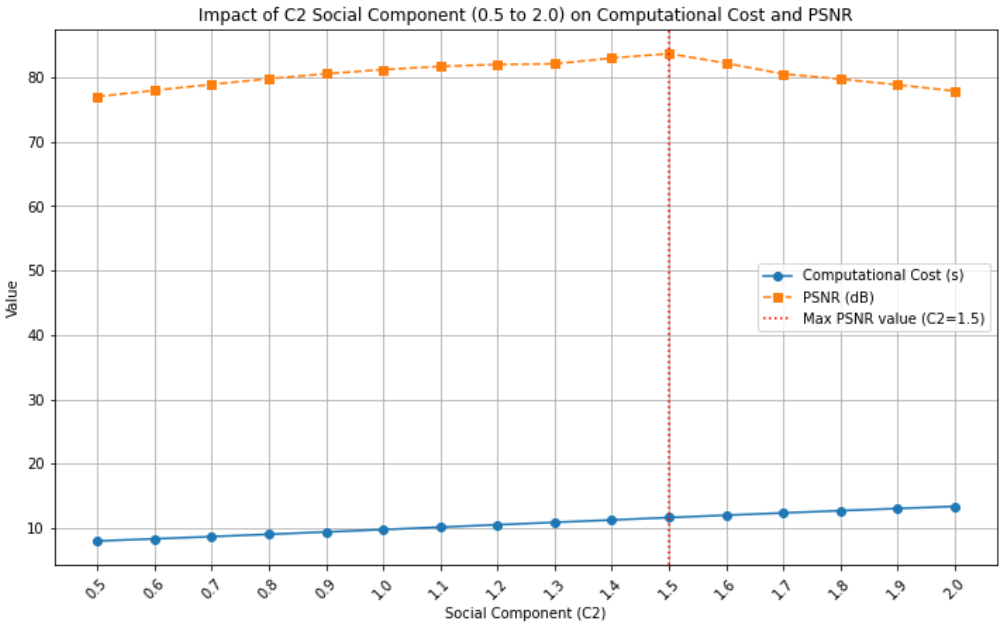


Figure 7. Impact of social component (*C*2) on computational cost and PSNR.

As shown in Table 1, the proposed method achieved a high PSNR between the host and watermarked images, indicating that the watermark was embedded effectively. The shift-invariant properties of DTCWT were found to be crucial in achieving this high PSNR. These properties allow for robust watermark embedding while preserving the quality of the host image.

To verify the embedding algorithm, we performed an extraction of the logo embedded in the images and measured the NCC between the original image logo and the extracted logo image. The results are provided in Table 2.

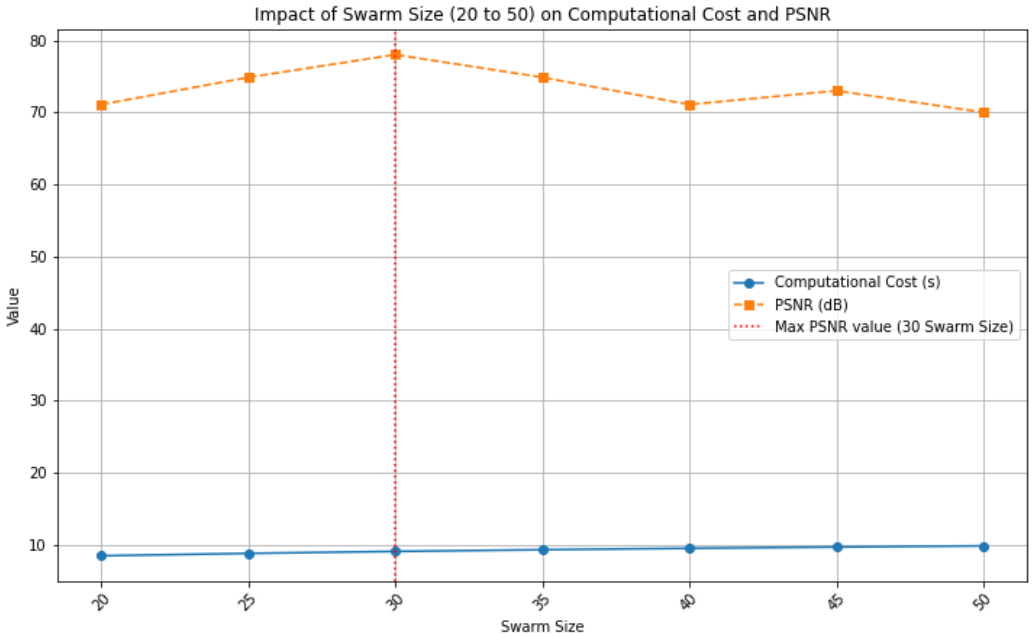


Figure 8. Impact of swarm size on computational cost and PSNR.

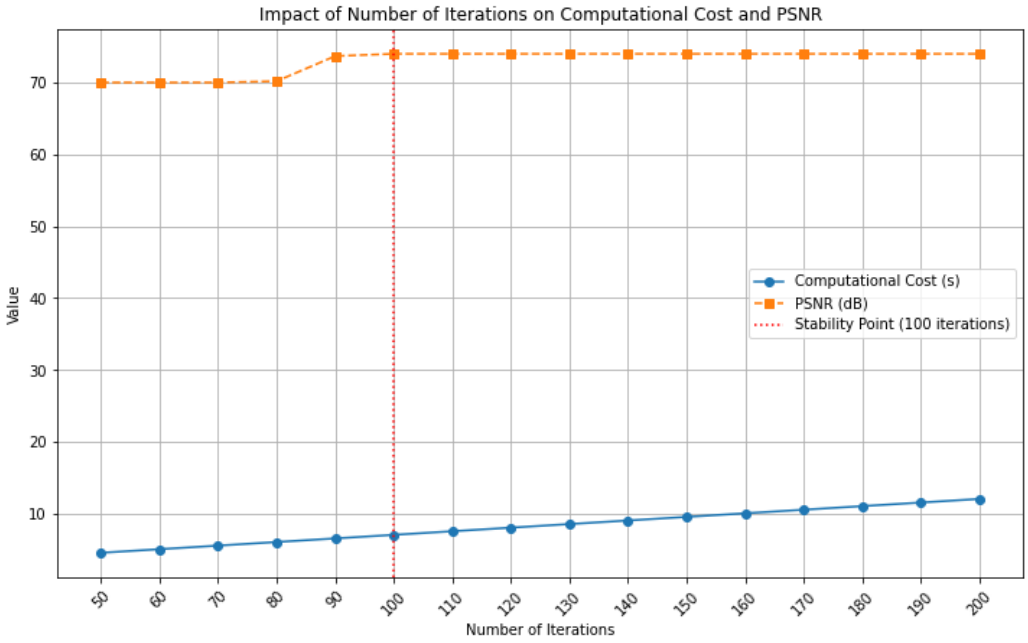
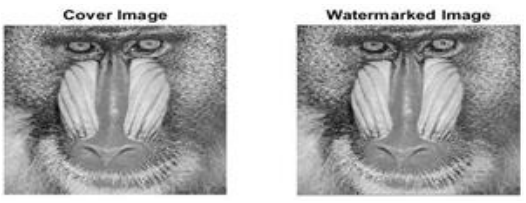


Figure 9. Impact of number of iterations on computational cost and PSNR.

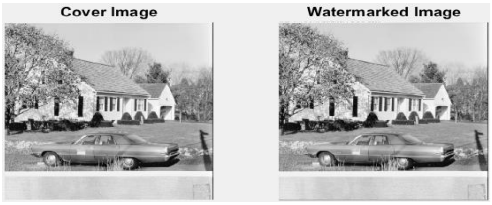
Table 1. The original host images along with the watermarked image for visual comparison; filters used are near_sym_b and qshift_b, while the decomposition level is four.

			
(**a**) PSNR = 80.65.		(**b**) PSNR = 81.17.	

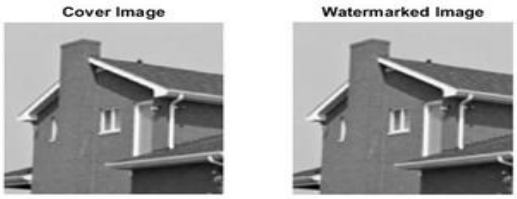
Table 1. *Cont.*



(**c**) PSNR = 80.14.



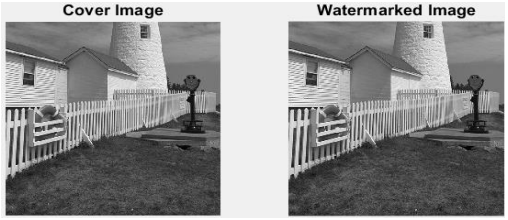
(**d**) PSNR = 80.4.



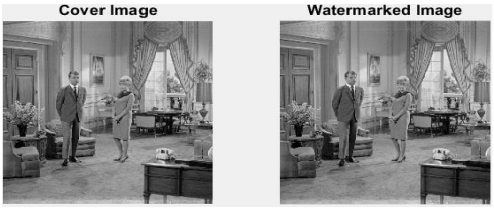
(**e**) PSNR = 80.42.



(**f**) PSNR = 80.46.



(**g**) PSNR = 80.2.



(**h**) PSNR = 80.1.

Table 2. NCC values for extracted watermarks.

Image	Extracted Logo	NCC
Leena		93.3%
Cameraman		94.9%
Mandrill		89.2%

Table 2. *Cont.*

Image	Extracted Logo	NCC
Homecar		91.1%
House		96.1%
Airplane		90.7%
Lighthouse		91.3%
Bedroom		93.5%

Table 2 examines the robustness of our watermarking method through various images. The high NCC values indicate that the watermarking technique is effective at embedding and extracting the watermark even after possible distortions or manipulations to the image. The consistency in NCC values across different images means that the method is functionally robust for image content or complexity.

To assess the performance of the proposed PSO-based watermarking method, a comparative analysis was conducted against other classical optimization approaches commonly used in watermark optimization. The comparison highlights the strengths and trade-offs of the proposed method. The evaluation focused on three primary metrics, PSNR, NCC, and computational cost, which collectively reflect the efficiency and effectiveness of the optimization process. The findings are summarized in Table 3.

Table 3. Comparison of the proposed PSO–DTCWT method and classical approaches.

Method	PSNR (dB)	NCC	Computational Cost
GA-Based Method [12]	47.81	97.84%	High (requires DCNN-SVR correction)
DE-Based Method [31]	69.25	93.30%	High (due to embedding additional data)
ACO + LGBA Method [32]	46.4–49.8	99% for most attacks	Significantly improved with LGBA
Proposed PSO–DTCWT	80.5	92.51%	Lower (due to parameter optimization)

The results in Table 3 highlight the superior performance of the proposed PSO–DTCWT method compared to the GA-based, differential evolution (DE)-based, and ant colony optimization with light gradient boosting algorithm (ACO + LGBA) methods. The proposed method achieves the highest imperceptibility, with a PSNR of 80.5 dB.

The PSNR significantly surpasses that of the other methods, whose values range from 46.4 to 69.25 dB. Although the PSO–DTCWT method's robustness (NCC of 92.51%) is slightly lower than that of the ACO + LGBA method (99%) and the GA-based method (97.84%), it remains competitive and sufficient for reliable watermark extraction. Furthermore, the PSO–DTCWT method demonstrates a clear advantage in computational efficiency due to parameter optimization, unlike the GA- and DE-based methods, which incur higher costs from additional processes such as DCNN-SVR correction and the embedding of extra data. These findings indicate that the PSO–DTCWT method achieves the best balance between imperceptibility, robustness, and computational cost, making it a practical and efficient solution for real-world applications.

4. Discussion and Conclusions

Using dual-tree complex wavelet transform (DTCWT) and particle swarm optimization (PSO), the watermarking method proposed revealed drastic improvements in terms of imperceptibility and resilience against attack, with a high capacity to embed large bits of messages into target images. Higher PSNR values indicate less degradation in the quality of host images due to watermarking operation. The technique shows good performance in preserving the image's quality up to a mean PSNR value of 80.50% across the dataset.

Moreover, the average NCC value of 92.51% indicates high accuracy in extracting watermarks from an original image compared to its corresponding retrieved version (Table 2). This highlights the robustness of the proposed method in authenticating and verifying forgeries performed on watermarked images.

Utilizing DTCWT for frequency domain transformation confers benefits such as shift invariance and directional selectivity, which are pivotal in obtaining high imperceptibility. The PSO algorithm's optimization of coefficient selection augmented the watermark's robustness against frequent image manipulation attacks. Overall, combining DTCWT and PSO in the proposed watermarking method delivers a highly effective solution for secure and superior image watermarking.

To assess the performance of the proposed PSO-based watermarking method, a comparative analysis was conducted against other classical optimization approaches commonly used in watermark optimization. The comparison highlights that the proposed PSO-based watermarking method achieves the highest imperceptibility (PSNR of 80.5 dB) compared

to the GA-based, DE-based, and ACO + LGBA methods. While its robustness (NCC of 92.51%) is slightly lower than the GA-based (97.84%) and ACO + LGBA methods (99%), it remains competitive. The PSO-based method also demonstrates significantly lower computational cost due to parameter optimization, making it an efficient and practical solution for watermarking applications.

By leveraging PSO's ability to find optimal solutions in uncertain environments and DTCWT's capacity for multi-dimensional data analysis, the proposed approach can effectively address uncertainties, improve efficiency, and provide robust solutions across various industrial applications.

The proposed watermarking method has limitations in terms of computational complexity, applicability to color images and advanced attack resistance. Future work must be considered to minimize the effect of these limitations and widen the use of the algorithm in other domains. Future research can be applied to improve such limitations in 3D gray scale and color images, optimize PSO parameters, and apply the method in real-time and IoT environments.

Author Contributions: Conceptualization, A.A.R.B.; methodology, A.A.R.B. and A.B.I.; software, A.B.I.; validation, A.A.R.B.; formal analysis, A.A.R.B. and A.B.I.; investigation, A.A.R.B. and A.B.I.; resources, A.A.R.B.; data curation, A.B.I. and A.A.R.B.; writing—original draft, A.B.I.; writing—review and editing, A.A.R.B.; data visualization, A.B.I.; supervision, A.A.R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Algorithm A1: PSO to find the best coefficients for watermark embedding

```
Input:  sub band coefficients <c1, c2, ... cm>.
Output:  Particles matrix where row i and column j of the matrix is set
to 1 if the corresponding coefficient is selected by the optimization.

// Initialize algorithm variables
pbest = array of size num_swarms x num_of_watermark_bits, initialized to
all zeros
gbest = array of size 1 x num_of_watermark_bits, initialized to all
zeros
pbest_fitness = array of size num_swarms, initialized to all zeros
gbest_fitness = 0
max_iterations = user-defined maximum number of iterations
num_swarms = user-defined number of swarms
Wmax = user-defined maximum weight factor
Wmin = user-defined minimum weight factor
K = user-defined number of selected particles
iteration = 1
```

```
// Evaluate initial fitness values
For i = 1 to m
     F_i = calculate objective function of particle c_i
     If F_i is better than F(pbest_i) then
         pbest_i = c_i
         pbest_fitness(i) = F_i
     EndIf
     If F_i is better than gbest_fitness then
         gbest = c_i
         gbest_fitness = F_i
     EndIf
EndFor

// Particle Swarm Optimization loop
while iteration <= max_iterations do
      For i = 1 to num_swarms do
         // Calculate new particle velocities
         W = Wmax - (Wmax - Wmin) * iteration / max_iterations
         For j = 1 to num_of_watermark_bits do
              r1 = random number between 0 and 1
              r2 = random number between 0 and 1
              velocity_i_j = W * velocity_i_j + C1 * r1 * (pbest_i_j -
particles_i_j) + C2 * r2 * (gbest_j - particles_i_j)
              // Update particle position based on sigmoid function
              if rand() < 1 / (1 + exp(-velocity_i_j)) then
       particles_i_j = 1
              else
       particles_i_j = 0
              EndIf
         EndFor

         // Select subset of particles based on K
         M = sum(particles_i)
         if M > K then
              index = find(particles_i == 1)
              index_to_remove = index(randperm(length(index), M - K))
              particles_i(index_to_remove) = 0
         else if M < K then
              index = find(particles_i == 0)
              index_to_add = index(randperm(length(index), K - M))
              particles_i(index_to_add) = 1
         EndIf
```

```
            // Update personal best and global best particles
            fitness_i = calculate objective function of particle i
            if fitness_i > pbest_fitness(i) then
                 pbest_fitness(i) = fitness_i
                 pbest_i = particles_i
            EndIf
            if fitness_i > gbest_fitness then
                 gbest_fitness = fitness_i
                 gbest = particles_i
            EndIf
       EndFor
       iteration = iteration + 1
EndWhile

// Output selected particles
Output particles
```

References

1. Siwek, S.E. *The True Cost of Sound Recording Piracy to the US Economy*; Institute for Policy Innovation Lewisville: Lewisville, TX, USA, 2007.
2. Rizzi, M.; D'Aloia, M.; Longo, A. Digital watermarking for healthcare: A survey of ECG watermarking methods in telemedicine. *Int. J. Comput. Sci. Eng.* **2020**, *23*, 235–249. [CrossRef]
3. Kumar, R.; Singh, P.K.; Yadav, J. Digital image watermarking technique based on adaptive median filter and HL sub-band of two-stage DWT. *Int. J. Comput. Aided Eng. Technol.* **2023**, *18*, 290–310. [CrossRef]
4. Begum, M.; Uddin, M.S. Digital image watermarking techniques: A review. *Information* **2020**, *11*, 110. [CrossRef]
5. Garg, P.; Kishore, R.R. Performance comparison of various watermarking techniques. *Multimed. Tools Appl.* **2020**, *79*, 25921–25967. [CrossRef]
6. Setiadi, D.R.I.M. PSNR vs SSIM: Imperceptibility quality assessment for image steganography. *Multimed. Tools Appl.* **2021**, *80*, 8423–8444. [CrossRef]
7. Rustad, S.; Andono, P.N.; Shidik, G.F. Digital image steganography survey and investigation (goal, assessment, method, development, and dataset). *Signal Process.* **2023**, *206*, 108908.
8. Evsutin, O.; Dzhanashia, K. Watermarking schemes for digital images: Robustness overview. *Signal Process. Image Commun.* **2022**, *100*, 116523. [CrossRef]
9. Yuan, Z.; Su, Q.; Liu, D.; Zhang, X. A blind image watermarking scheme combining spatial domain and frequency domain. *Vis. Comput.* **2021**, *37*, 1867–1881. [CrossRef]
10. Garg, P.; Kishore, R.R. Optimized color image watermarking through watermark strength optimization using particle swarm optimization technique. *J. Inf. Optim. Sci.* **2020**, *41*, 1499–1512. [CrossRef]
11. Husain, A.; Mishra, A.D.; Jena, S.K. A robust approach for digital watermarking of satellite imagery dataset. *Int. J. Swarm Intell.* **2022**, *7*, 82–93. [CrossRef]
12. Barlaskar, S.A.; Singh, S.V.; Monsley, K.A.; Laskar, R.H. Genetic algorithm based optimized watermarking technique using hybrid DCNN-SVR and statistical approach for watermark extraction. *Multimed. Tools Appl.* **2022**, *81*, 7461–7500. [CrossRef]
13. Zhu, T.; Qu, W.; Cao, W. An optimized image watermarking algorithm based on SVD and IWT. *J. Supercomput.* **2022**, *78*, 222–237. [CrossRef]
14. Huang, T.; Xu, J.; Yang, Y.; Han, B. Robust zero-watermarking algorithm for medical images using double-tree complex wavelet transform and Hessenberg decomposition. *Mathematics* **2022**, *10*, 1154. [CrossRef]
15. Nouioua, N.; Seddiki, A.; Ghaz, A. Blind Digital Watermarking Framework Based on DTCWT and NSCT for Telemedicine Application. *Trait. Du Signal* **2020**, *37*, 955–964. [CrossRef]
16. Balasamy, K.; Suganyadevi, S. A fuzzy based ROI selection for encryption and watermarking in medical image using DWT and SVD. *Multimed. Tools Appl.* **2021**, *80*, 7167–7186.
17. Liu, J.-X.; Wen, X.; Yuan, L.-M.; Xu, H.-X. A robust approach of watermarking in contourlet domain based on probabilistic neural network. *Multimed. Tools Appl.* **2017**, *76*, 24009–24026. [CrossRef]
18. Hamamoto, I.; Kawamura, M. Digital watermarking method by encoder using neural network. *IEICE Tech. Rep.* **2017**, *117*, 17–22.

19. Ding, W.; Ming, Y.; Cao, Z.; Lin, C.-T. A generalized deep neural network approach for digital watermarking analysis. *IEEE Trans. Emerg. Top Comput. Intell.* **2021**, *6*, 613–627. [CrossRef]
20. Kumsawat, P.; Attakitmongcol, K.; Srikaew, A. A new approach for optimization in image watermarking by using genetic algorithms. *IEEE Trans. Signal Process.* **2005**, *53*, 4707–4719. [CrossRef]
21. Wu, Q.; Qu, A.; Huang, D.; Ma, L. Robust and blind audio watermarking scheme based on genetic algorithm in dual transform domain. *Math. Probl. Eng.* **2021**, *2021*, 3378683. [CrossRef]
22. Thanki, R. Genetic algorithm-based intelligent watermarking for security of medical images in telemedicine applications. In *Intelligent Data Security Solutions for e-Health Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 185–204.
23. Laxmanika; Singh, P.K. Robust and imperceptible image watermarking technique based on SVD, DCT, BEMD and PSO in wavelet domain. *Multimed. Tools Appl.* **2022**, *81*, 22001–22026.
24. Awasthi, D.; Srivastava, V.K. Performance enhancement of SVD based dual image watermarking in wavelet domain using PSO and JAYA optimization and their comparison under hybrid attacks. *Multimed. Tools Appl.* **2023**, *82*, 35685–35717. [CrossRef]
25. Sharma, S.; Sharma, H.; Sharma, J.B. Artificial intelligence based watermarking in hybrid DDS domain for security of colour images. *Int. J. Intell. Eng. Inform.* **2020**, *8*, 331–345. [CrossRef]
26. Garg, P.; Kishore, R.R. An efficient and secured blind image watermarking using ABC optimization in DWT and DCT domain. *Multimed. Tools Appl.* **2022**, *81*, 36947–36964. [CrossRef]
27. Pourhadi, A.; Mahdavi-Nasab, H. A robust digital image watermarking scheme based on bat algorithm optimization and SURF detector in SWT domain. *Multimed. Tools Appl.* **2020**, *79*, 21653–21677. [CrossRef]
28. Gupta, M.; Saraswat, M. Grey wolf optimisation-based colour image watermarking. *World Rev. Entrep. Manag. Sustain. Dev.* **2020**, *16*, 648–664. [CrossRef]
29. Selesnick, I.W.; Baraniuk, R.G.; Kingsbury, N.C. The dual-tree complex wavelet transform. *IEEE Signal Process. Mag.* **2005**, *22*, 123–151. [CrossRef]
30. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [CrossRef]
31. Salimi, L.; Haghighi, A.; Fathi, A. A novel watermarking method based on differential evolutionary algorithm and wavelet transform. *Multimed. Tools Appl.* **2020**, *79*, 11357–11374. [CrossRef]
32. Sharma, V.K.; Mir, R.N. An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 615–626. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.