Robust Image Watermarking using DWT, DCT, and PSO with CNN-Based Attack Evaluation

Mohammad Alia
Department of Cybersecurity
Al-Zaytoonah University of Jordan
dr.m.alia@zuj.edu.jo

Adnan Hnaif
Department of Cybersecurity
Al-Zaytoonah University of Jordan
adnan_hnaif@zuj.edu.jo

Aseel Alrawashdeh
Department of Computer Science
Al-Zaytoonah University of Jordan
202017113@std-zuj.edu.jo

Yousef Jaradat
Department of Electrical Engineering
Al-Zaytoonah University of Jordan
y.jaradat@zuj.edu.jo

Mohammad Masoud
Department of Electrical Engineering
Al-Zaytoonah University of Jordan
m.zakaria@zuj.edu.jo

Ahmad Manasrah
Department of Mechanical Engineering
Al-Zaytoonah University of Jordan
ahmad.mansrah@zuj.edu.jo

Ahmad AlShanty
Department of Cybersecurity
Al-Zaytoonah University of Jordan
a.alshanty@zuj.edu.jo

Abstract: *Digital content protection is one of the most significant research areas that lies at the intersection of cyber-security and multimedia processing. Protect multimedia content from copyright violation, unauthorized use, replication, and online content theft is needed. Digital Image Watermarking is utilized to preserve the copyright of different digital images from forgery. Various techniques had been developed in this regard with two main issues, the method robustness and the resistance against various types of attacks like, Salt and Pepper noise, filtering and blurring. Current digital watermarking techniques may reduce the quality of the original digital media content if it is not robust. The purpose of this research is to create a robust image watermarking technique against different attack types such as salt and pepper noise and Gaussian noise, ensuring the image content is protected. Specifically, this study proposed a new mechanism for Image watermarking based on combining Discrete Wavelet Transform (DWT), and Discrete Cosine Transform (DCT). Additionally, Particle Swarm Optimization (PSO) was applied to perform the optimization for both the embedding and extraction processes. At the final stage we assess the proposed approach against some types of attacks such as Additive White Gaussian Noise (AWGN). The Denoising Convolutional Neural Network, (DnCNN) was used to evaluate the mechanism against AWGN. For testing, we utilized measures such as Peak Signal-to-Noise Ratio (PSNR) and Normalized Correlation Coefficient (NCC). The experimental results of implementing our proposed method for embedding and extracting watermarks across various host image sizes were encouraging, achieving a PSNR ratio of 0.998 and an NCC of 1 in the absence of attacks. Additionally, our evaluation revealed that a specific type of denoising attack, while damaging the watermark (though not completely), actually improved the image quality. It is also important to highlight that our findings surpassed those reported in existing literature, with the PSNR and NCC values serving as evidence of this superior performance.*

Keywords: *Particle swarm optimization, discrete wavelet transform, discrete cosine transform, denosing network, additive white gaussian noise, peak signal-to-noise ratio, normalized correlation coefficient.*

Received March 29, 2024; accepted October 9, 2024
https://doi.org/10.34028/iajit/21/6/1

1. Introduction

With the widespread development of network technology, digital content can rapidly propagate through transmission and storage. The Copyright violations, unauthorized use, replication, and online content theft maybe occurred into that content. Therefore, a robust technique for digital data protection should be developed. To identify the legitimate owner of digital works, verify for the integrity and the authenticity of data, control copying, and accomplish the copyright protection. In order to meet the demands or challenges of the current and the upcoming threats in protecting digital information [3, 28].

Watermarking is a branch from steganography which aims to safeguard digital media's intellectual property against unauthorized copying or access by incorporating a watermark (either visible or invisible) into the content. This watermark can stay next to the data and be used whenever there is any doubt about the media's authenticity (e.g., the hidden watermark refers to the original owner) [2]. Digital watermarking involves embedding (inserting) information into any content such as documents or images, usually called digital signature or watermark, [1, 4, 5]. To imagine the nature of watermark, it's like a visible "seal" over an image. Digital Watermarking is the most effective method for securing digital data when compared to similar techniques like cryptography, steganography, and digital rights management [7].

Digital Watermarking (DW) is one of the information security hiding methods that solves these issues. DW involves embedding/ inserting data called digital signature or watermark into the original digital media content to provide content protection and anti-piracy toolkit. However, the addition of the watermark images into the original host image, may reduce the quality of that image if it was not robust enough. And it may be exposed to attack. So that, the main problem that have been solved in this research is to reduce the noise of the host image and to develop a robust image watermarking technique against different attacks types such salt and pepper noise and Gaussian noise. Thus, the content will be protected.

In terms of digital watermarking techniques, DW can be classified based on domain. Spatial Domain, watermarking slightly modifies the value of pixels in randomly chosen areas of images; the watermark is embedded in the original image. The spatial domain watermarking techniques are straightforward and offer a large amount of data. The ability to embed a single watermark several times through the embedding process has another benefit. If any single watermark survives the attack, the goal of watermarking is achieved. In the spatial domain, no transformation or conversion is carried out. Least Significant Bit (LSB) and Local Binary Pattern (LBP) , are common methods of spatial domain-based [22]. LSB is the simplest approach to watermarking in the spatial domain. Digital watermarking security challenges were handled by [10] using the LSB method, the input image is first converted to binary bits using the LSB substitution methods, after which the rightmost bits of each pixel value are changed to watermark bits. As a result, the quality of the watermarked image is decreased by this technique, which directly alters the pixel values. LBP where the host image is transformed into non-overlapping blocks (square). The spatial association between the middle pixel and its surrounding pixels is then identified for each block. The watermark is then extracted and inserted in accordance with these pixels [9].

The Discrete Cosine Transform (DCT) is a method used in S techniques. Unlike direct embedding of watermarks into the original image, this approach first transforms the original image. The watermark is then embedded into the coefficients of this transformed image. To retrieve the original data, an inverse transformation is applied to these coefficients. Techniques in the transform domain are highly resistant to various attacks, offering strong robustness and making the watermark less perceptible to image manipulations and data processing attacks. DCT transforms the representation of data from the time domain to the frequency domain. It generates a two-dimensional matrix of coefficients from an image [21]. DCT is a preferred method in fields such as data compression, pattern recognition, and several spatial domain applications due to its efficiency and speed. The DCT-based technique segments the image into non-overlapping blocks of a specified size and applies DCT to each block [13].

The Discrete Wavelet Transformation (DWT) is a technique used in the domain of transform domain watermarking. It excels in creating images that offer multi-resolution views, facilitating the analysis of visual data by allowing observations at different resolutions. DWT works by splitting the image into components of high and low frequencies. The lower frequency components are then further divided iteratively until the desired result is obtained. Applying DWT results in the division of the image into four sub-bands, effectively converting the image from its original pixel domain to a specific frequency domain. It has been shown that the wavelet coefficients perform better than most conventional methods in comparison to other schemes. DWT operates by decomposing every one-dimensional signal into two elements: the detail coefficient and the approximation. This decomposition is achieved using low-pass filters for the signal's low frequencies and high-pass filters for its high frequencies, producing what are known as DWT coefficients. These coefficients can then be used to reconstruct the original watermark image, a process known as Inverse DWT (IDWT) [22]. The four sub-bands produced, LL, LH, HL, and HH, represent the image’s approximation, horizontal, vertical, and diagonal details, respectively. The approximation sub-band, which contains the low-frequency component and holds the bulk of the image's information, is ideally suited for watermark insertion. The original host image can be reconstructed using the Inverse Discrete Wavelet Transform (IDWT). DWT also provides scalability features [13].

The basic features of digital watermarking are, firstly, Robustness: DWs that can be used for copyright protection are said to be robust if they can withstand a specific class of transformations. The robustness criterion focuses on two aspects, namely

1. If the watermark is still present after data distortion.
2. Whether the watermark detector can detect it [26].

Secondly, Imperceptibility: The imperceptibility, which refers to the similarity of the original and watermarked images, can be thought of as a metric of the perceptual transparency of a watermark. Thirdly, Security: The watermark security indicates that it should be difficult to remove or change the watermark without affecting the cover image. Fourthly, Capacity or data payload: Images must have an appropriate quantity of information associated to them. Data payload is the phrase used to describe the embedded information in watermarked images. Data payload is defined as bits encoded in a watermark for a specific amount of time or work [13]. Fifthly, Computational cost: It displays the cost of watermark embedding process into a cover, and also the retrieving of it from the digital cover [26]. Sixthly, Transparency: Digital watermarking shouldn't degrade the quality of the original image after it has been watermarked.

The digital image watermarking technique based on DCT was enhanced through the use of the Particle Swarm Optimization (PSO) Algorithm. This method represents a novel approach in distributed and collective intelligence for solving problems, particularly in the optimization arena, without the need for central control or the development of a global model. Particle Swarm Optimization is characterized by its robustness as an optimization technique. It involves a group of potential solutions, termed as a swarm of particles, navigating the parameter space. Their movement and the paths they create are influenced by their own best achievements and those of their neighboring particles. PSO has emerged as a widely adopted algorithm in academic research, notable for its dual modes of agent interaction: direct and indirect. This optimization strategy excels in tackling issues where the ideal solution lies within a multidimensional parameter space. Swarm-based methods have gained recognition as nature-inspired algorithms. This is because, despite the individual simplicity and limited capabilities of the agents (the swarm's members), their collective behaviors and interactions enable effective problem-solving. Such algorithms are celebrated for their ability to deliver fast, dependable, and cost-effective solutions to complex challenges [8, 15, 29].

Generative models, or GANs, were created in 2014 Goodfellow *et al*. [18]. As it has been used successfully for a variety of real-world applications, it has recently attracted significant attention in capturing rich data distribution, such as images, audio or video and generating new samples. The fundamental principle of a GAN is to use a “generator” and a “discriminator” to simplify indirect training. The generator learns to produce more realistic data samples while the discriminator learns to recognize between actual samples and fake samples produced by the generator.

The image may be corrupted by noise during the transmissions over the internet. The noise is added to the image like lossy compression, wrong memory locations, the pipeline of camera imaging (like shot noise), scattering, and other unfavorable atmospheric circumstances [27]. It is also added by noise sources nearby the image capturing devices, as well as by impurities in the devices themselves and from their proximity. The image denoising process is the estimation of clean images from its noisy observations. This process is much related to image inpainting, artefacts reduction, and blur. Also, watermark removal is also recognized as preprocessing tasks in computer vision applications like image segmentation. Based on probability distribution, the noise is modeled as Gaussian, Gamma, Poisson, etc. In the past decades, image denoising methods have received a lot of attention, initially Nonlinear and non-adaptive filters were used. After that, image denoising has been successfully incorporated into Machine Learning (ML) techniques, such as sparse-based algorithms [19], diffusion-based methods and Support Vector Machine (SVM). Despite the fact that the majority of the approaches mentioned above produced reasonably performance in image denoising, they had some disadvantages [23], including the necessity for test phase optimization methods, manual parameter setup, and a specific model for single denoising tasks.

Recently, Deep Learning (DL) algorithms were able to get beyond these limitations as architectures became more adaptable [20]. In 1980 [16], the original Deep learning algorithms were developed first in image processing, and were used in denoising by [32]. Specifically, the latent clean image was recovered using a neural network that also had additive noise and the common shift-invariant blur function. The neural network then utilized weighting parameters to eliminate complex noise [11].

The Convolutional Neural Network (CNN) algorithm is a type of deep learning algorithm that has been applied to various tasks such as image super-resolution, de-blurring, and denoising. CNNs are easier to train than Artificial Neural Networks (ANNs) because they feature sparse connectivity in each convolutional layer, unlike ANNs. This sparse connectivity contributes to CNNs' superior performance in image resolution enhancement [14] and their greater representational capabilities compared to traditional methods like sparse representation, which loses 2D structural information by converting image matrices into vectors. Conversely, CNNs maintain 2D structural information during both training and testing phases through the use of convolution operations that consider the local neighborhood of pixels with 2D masks [12, 31]. This study examines denoising CNN models such as DnCNN, which incorporates a batch normalization layer and residual learning connections. However, DnCNN faces challenges with gradient explosion and slow convergence rates.

This research aims to develop a robust image watermarking technique utilizing DWT and DCT. It focuses on refining the search process through PSO to identify the optimal block for watermark embedding. Additionally, the study seeks to improve the watermarking algorithm's resilience against various attack types, including white Gaussian noise. Another objective is to evaluate the effectiveness of the DnCNN model in removing noise and enhancing the watermarked image's quality and smoothness. The research will present the outcomes of our algorithm, test it across various images, and compare these results with those found in existing literature. It will also demonstrate the robustness of our algorithm by using metrics such as Peak Signal-to-Noise Ratio (PSNR) and Normalized Correlation Coefficient (NCC).

The rest of this study is organized as follows: Section 2 presents literature review section 3 describes the proposed methodology. Section 4 shows results and discussion, and finally, section 5 concludes the study results and discusses the potential future directions.

2. Literature Review

Yadav *et al*. [30] explored the application of PSO to improve the outcome of image watermarking using a combination of DWT and DCT. In their study, the watermark was embedded into the DWT-DCT coefficients when they exceeded a specific threshold, targeting low-frequency areas of the image to enhance robustness. In addition to DWT-DCT, Singular Value Decomposition (SVD) was also tested for watermarking. The effectiveness of PSO was compared to a modified inertia weight-based version of the algorithm, showing improved results in terms of watermark undetectability and robustness against attacks. The simulation results indicated minimal differences between the original and watermarked images, highlighting the effectiveness of PSO in optimizing the watermarking process. [20] Investigated the impact of denoising attacks using Fully Convolutional Neural Networks (FCNN) on watermarked images. The study employed an encoder-decoder architecture to reduce noise while preserving the fine details of the image structure. Although the FCNN was effective at maintaining high image quality, it significantly reduced the robustness of the watermark, making it vulnerable to attacks. The study demonstrated that while this denoising method could enhance image quality, it compromised the durability of the watermark against all tested methods, ultimately weakening the protection provided by the watermark. Quan *et al*. [24] developed a novel approach to watermarking Deep Neural Networks (DNNs), particularly focusing on low-level image processing tasks. Their method involved creating a black-box watermarking mechanism for pre-trained models by leveraging the overparameterization of DNNs. To further verify the presence of the watermark, they introduced an auxiliary model that visualized the embedded watermark. Their experimental results indicated that the watermarking approach had little to no negative impact on model performance and remained resilient to a variety of attacks, making it a robust solution for neural network watermarking. Geng *et al*. [17] proposed a removal attack using Convolutional Neural Networks (CNN) designed for real-time applications. Due to the need for speed in such scenarios, they adopted a simple yet efficient CNN model, trained on a dataset of watermarked images, to remove the watermark. Their findings showed that the CNN model was capable of effectively removing watermarks without significantly degrading the quality of the original image. AL-Nabhani *et al*. [6] aimed to enhance the invisibility of watermarked images. Their approach involved using a DCT with a Haar filter to insert the watermark into specific coefficient blocks, without needing the original image for extraction. They utilized a probabilistic neural network to retrieve the watermark and evaluated the algorithm using PSNR and NCC. The results were promising, achieving a PSNR of 68.27 dB and an NCC of 0.9779, indicating high invisibility and robustness of the watermarked images against common attacks such as Gaussian noise, JPEG compression, rotation, and cropping.

3. The Proposed Methodology

This study investigates the effectiveness and reliability of a fully FCNN for image denoising. We introduced an algorithm that employs two watermarking steps: DWT and DCT, aiming to achieve robust watermarking. Our methodology involves two key phases: embedding and extraction, to insert and retrieve the watermark, respectively. The process of watermarking modifies the wavelet coefficients in certain subbands, followed by the application of the DCT transform to these subbands. Figure 1 shows the workflow of the proposed algorithm:

In the proposed method, we start by loading both the watermark and host images as inputs. The watermark image is converted into a binary format and transformed into a bitstream, as illustrated in Figure 1.

Initially, the algorithm selects the most suitable color channel from the red, green, or blue options by evaluating each channel's PSNR values to determine the optimal one. Following this, the algorithm verifies the availability of sufficient blocks for the watermarking process and employs a random strategy to segment these blocks. Upon successful segmentation, the image is divided into 8×8 non-overlapping blocks, also referred to as multi-resolution coefficient sets. PSO is then utilized to identify the most appropriate block for embedding the watermark by analyzing each block's suitability to conceal the watermark based on PSO findings. The PSO algorithm is applied to each block of the image, iterating through each bit of the watermark image (denoted as ‘n’ bits). The fitness function, represented by the PSNR value, evaluates the robustness and efficiency of the watermarking method. Following this, both DWT and DCT are employed to facilitate the watermark hiding process and to compute the PSNR. Initially, the first level of DWT is applied, followed by DCT. It is important to note that this step occurs concurrently with the execution of the PSO. Through these processes, the watermark image is successfully embedded into the host image.

We subjected the watermarked image to a specific type of attack, namely white Gaussian noise, to evaluate the method's imperceptibility and robustness. Following this, we explored the impact of a DnCNN on the denoising process.

For our experiments, we chose the 'Lena' image, measuring 512 by 512 pixels, as the host image.

Additionally, a grey-scale image measuring 20 by 50 pixels, featuring the word 'copyright', was utilized as the watermark. The embedding process consists of several distinct steps:

- *Step* 1. Select Host image which is (Lena image) for example and watermarked image (copyright image).
- *Step* 2. Apply 1-level DWT on the host image. Aiming to decompose it into four non-overlapping blocks. By decomposing the image into blocks, it allows for the watermark to be embedded without disrupting the original image.
- *Step* 3. The next step is to Create 4×4 blocks from the 4 coefficient sets. This step is necessary as it provides a finer resolution for watermarking while also making the watermark more imperceptible and difficult to detect. This step helps to create blocks with the same size and the same amount of pixels, making it easier to embed the watermark.
- *Step* 4. For each block, the DCT is applied. This is accomplished by modifying the coefficients of the DCT blocks in the watermarked region to add the watermark. The modified coefficients are then quantized and encoded to produce the watermarked image.
- *Step* 5. The watermark image is reformulated in grey scale of 0s and 1s vector. In this step of the algorithm, the watermark image is converted into a 0s and 1s vector. This vector is essentially a binary representation of the watermark image, with 0s representing “off” or “black” and 1s representing “on” or “white”.

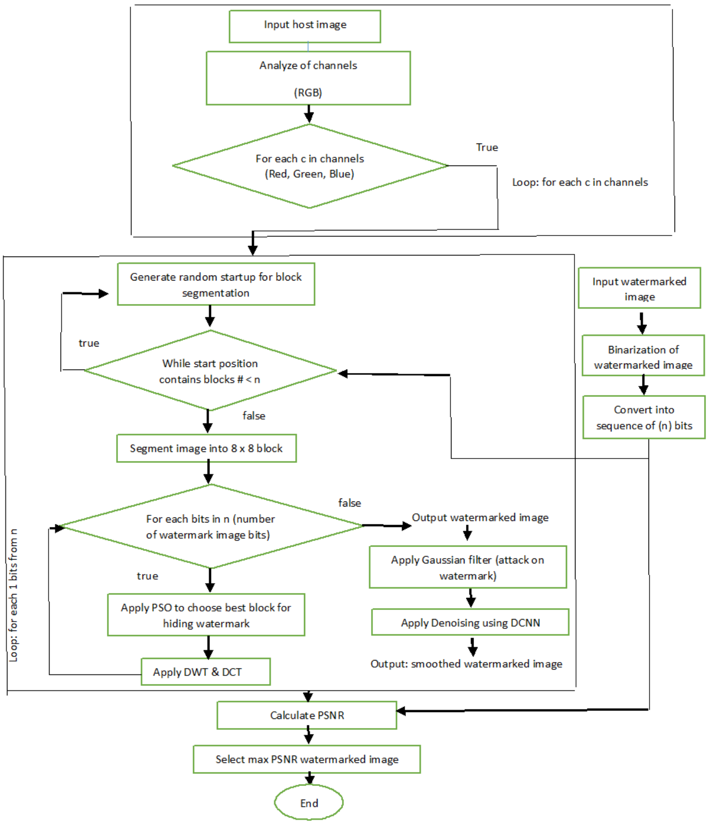

Input host image
Analyze of channels
(RGB)
For each c in channels
(Red, Green, Blue)
True
Loop: for each c in channels
Generate random startup for block segmentation
true
While start position contains blocks # < n
false
Segment image into 8 x 8 block
For each bits in n (number of watermark image bits)
false
true
Apply PSO to choose best block for hiding watermark
Apply DWT & DCT
Loop: for each 1 bits from n
Input watermarked image
Binarization of watermarked image
Convert into sequence of (n) bits
Output watermarked image
Apply Gaussian filter (attack on watermark)
Apply Denoising using DCNN
Output: smoothed watermarked image
Calculate PSNR
Select max PSNR watermarked image
End

Figure 1. Flowchart of the algorithm.

Now, create distinct sequences using a key randomly. These sequences will be used to embed the watermark bits, bit 0 is embedded using one sequence (PN 0), while bit 1 is embedded using the other sequence (PN 1). Noting that, the number of mid-band elements in the DCT-transformed, DWT coefficient sets must match the number of elements in each of the two pseudorandom sequences in order to ensure proper embedding of the watermark.

Embed the two sequences, in the DCT-transformed 4×4 blocks of the chosen DWT coefficient set of the host image with a gain factor of α. Noting that, Just the mid-band DCT coefficients are subject to embedding, not the other coefficients in the DCT block.

After the modifications of mid-band coefficients to embed the watermark bits, we apply the inverse DCT for each block. This step is necessary to obtain the watermarked image from the modified mid-band coefficients. The Inverse Discrete Cosine Transform (IDCT) is the reverse process of the DCT. It is used to reconstruct the original image from the modified mid-band coefficients.

Finally, apply the inverse DWT on the DWT transformed image to produce the watermarked host image, including the updated coefficients. In other words, apply an inverse DWT transformation to the modified blocks.

The following workflow chart in Figure 2 explain the steps of the embedding procedure.

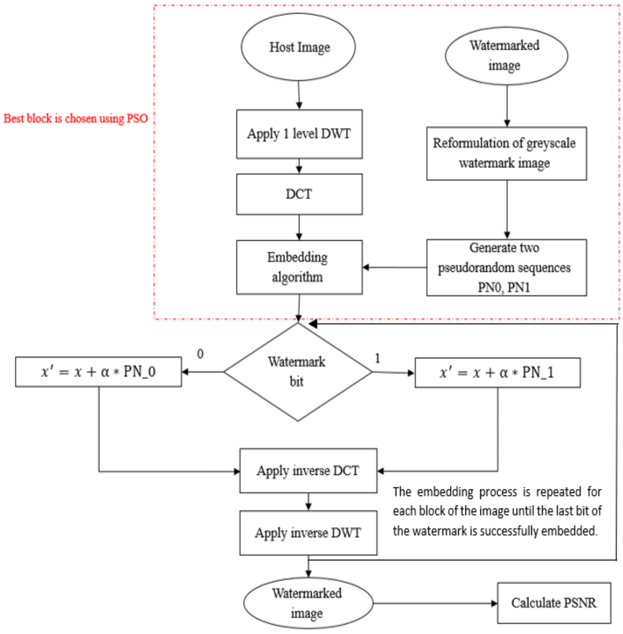

Best block is chosen using PSO
Host Image
Watermarked image
Apply 1 level DWT
Reformulation of greyscale watermark image
DCT
Embedding algorithm
Generate two pseudorandom sequences PN0, PN1
Watermark bit
0
1
$x' = x + \alpha * PN\_0$
$x' = x + \alpha * PN\_1$
Apply inverse DCT
Apply inverse DWT
The embedding process is repeated for each block of the image until the last bit of the watermark is successfully embedded.
Watermarked image
Calculate PSNR

Figure 2. Flowchart of the embedding process.

In terms of watermark extraction procedure. No need for the host image for watermark extraction using the Combined DWT-DCT algorithm, which is a blind watermarking algorithm. In our case, the key to generate sequence is known, also the location that was generated previously as starting point and the hiding pattern of the intended blocks.

- *Step* 1. Select the saved location and watermarked image.
- *Step* 2. Apply 1-level DWT on the host image. Aiming to decompose it into four non-overlapping blocks.

Divide four coefficient sets into 4×4 blocks and apply DCT on each block.

Regenerate the (PN_0 and PN_1) which are the 2 pseudorandom sequences using the same key which used in the watermark embedding procedure. This ensures that the same key is used throughout the process and that the original watermark is accurately reproduced.

Calculate the correlation between the mid-band coefficients and the two sequences (PN 0 and PN 1) for each block in the coefficient sets. The extracted watermark bit is regarded as 0 if the correlation with PN 0 was higher than the correlation with PN 1, otherwise it is regarded as 1.

The bits of the extracted watermark are used to reconstruct the watermark, and then we compare among the extracted and original watermarks, if the two watermarks are similar, then the watermark is successfully extracted.

Figure 3 shows a summarization for the extraction process for the watermarked image.

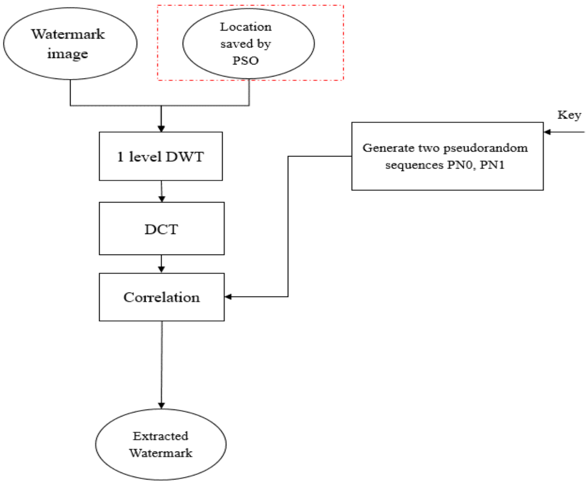


Figure 3. Flowchart of the extraction process.

The implementation of PSO in this approach is summarized as follows:

- Define population Size (20% from the blocks size).
- Define the fitness function (PSNR).
- Define the evaluation criteria (Highest PSNR).
- Start population randomly (n agents).

Loop till the PSNR not change within specific number of iterations (i.e., 3 iterations).

Best block is chosen after DWT (level-1) and DCT is applied based on the evaluation criteria after a set of iterations.

Denoising Convolutional Neural Network (DnCNN) is a rapid and simple method that can be used to smooth the image and remove noise from images. It's a pretrained denoising neural network that was designed by [18] to predict residual image, which is the difference among the latent clean image and the noisy observation. In other words, DnCNN removes the latent clean image implicitly in the hidden layers. Additionally, it introduced the batch normalization to stabilize and enhance and the DnCNN performance. The difference of using this method over the other discriminative denoising algorithms that those methods usually train a particular model for (AWGN) at a certain noise level, but this model is able to handle Gaussian denoising with unknown noise level such as blind Gaussian denoising case.

Moreover, DnCNNs are well-suited to this task because of their ability to learn features from large datasets of noisy images, and their capacity to learn representations which are more robust to noise than traditional methods. DnCNNs also have the ability to generalize to unseen examples, which is invaluable in many real-world applications. Additionally, DnCNNs are able to work in parallel with other denoising methods such as wavelet-based methods, providing a complementary approach for more effective denoising.

Using the residual learning method, DnCNN implicitly removes the latent clean image from the hidden layers. With the use of this feature, the DnCNN model is able to handle a number of generic image denoising tasks, including Gaussian denoising and single-image super-resolution.

$$B = A + V \tag{1}$$

Where V is the noise and A is the anticipated clean image. The mean squared difference between input A and the residual image resulting from noisy input is:

$$Error_x = \frac{1}{2x}\sum_{i=1}^{x} ||R(B_i;\ x)(\ B_i - A)\ ||^2 \tag{2}$$

In this study, we utilized MATLAB's DnCNN to evaluate the potential of neural networks for improving image quality and denoising, while ensuring the watermark remains intact. The choice of CNN is motivated by several factors, including their deep architecture, which enhances their capability and flexibility in processing image features. Additionally, significant progress has been made in CNN training techniques, such as Batch Normalization (BN), residual learning, and the use of the Rectifier Linear Unit (ReLU). These advancements contribute to enhanced denoising results and faster training times. We employed MATLAB's "denoisingNetwork" function to load a pretrained DnCNN model. Subsequently, the "denoiseImage" function was used to process a noisy 2-D single-channel image through the DNCNN network. The process of denoising an image with the pretrained DNCNN network is depicted in Figure 4.

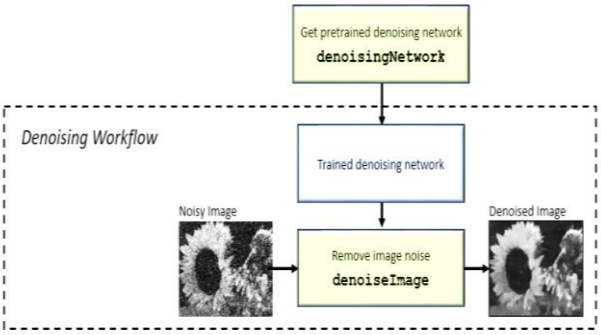


Figure 4. Workflow of denoise image using DnCNN.

To eliminate Gaussian noise, we used the pre-trained DnCNN network that is already built in. Several limitations and challenges apply to removing noise using the pretrained network, as the following:

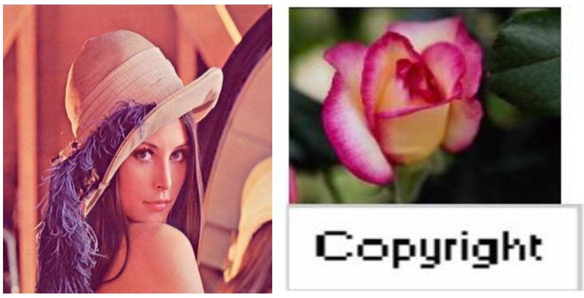
Only 2-D single-channel images can be used for noise removal. In this work we solved this limitation because we are working with 3-D images or several color channels. By handling each channel separately, the algorithm identifies only additive Gaussian noise, with a constrained standard deviation range. This pretrained denoising network consists of 59 layers including input and output layers (with a mix of Batch Normalization layers, convolutional layers, and RELU layers). The final layer is a regression layer. Moreover, we used Gaussian filter and salt and pepper filter to assess the ability of neural network in denoising regarding these types of attack.

One of the most important preprocessing steps is image binarization, which significantly reduces the amount of data subjected to further analysis and speeds up that analysis. Binarization calculates the threshold value that separates background and stroke pixels. In comparison to 256 levels of information for a greyscale or color image, using two levels of information minimizes the computational load. Compared to a greyscale image, a binary image is easier to process. So that, the primary benefit of binary images is that they reduce computational load and boost system effectiveness. We perform the binarization step of the watermark image at the beginning, due to its importance before the embedding and denoising steps.

To assess the robustness of our algorithm, we tried to perform Gaussian noise attack. Which can generate white noise into the watermark image, this attack is very simple and common. Salt and pepper filter (commonly known as impulse noise), is a kind of noise that result in pointed and unexpected distortions in the image. Salt and pepper filter had been used usually to check the robustness of watermarking algorithm. The effect of this filter is similar to sprinkling black and white dots on the image.

4. Implementations, Results and Discussion

After the text Here are some example from test set that we used to test our approach, these examples are already built-in in MATLAB. Figure 5 shows some examples from the existed images in Matlab that can be used to test various algorithms.



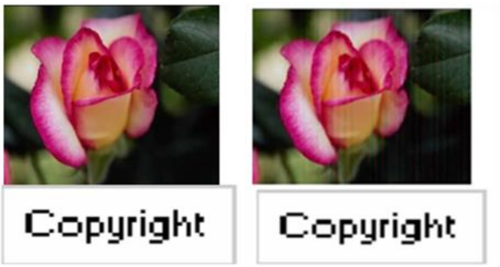
a) Lena image, rose image b) Copyright image.

Figure 5. Lena and copyright image.

The proposed watermarking method is tested with varying image sizes. The test included images with 256×256 and 512×512 sizes. Below are the images for three samples, namely the Lena profiles, and rose images before and after incorporating the copyright watermark. As shown in Figures 6 and 7.

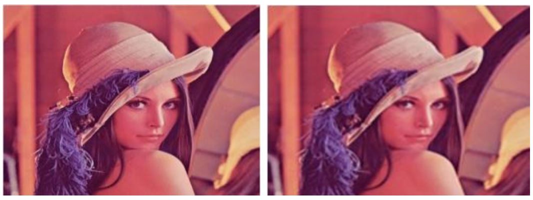
Figure 6 shows a digital image before and after watermark embedding. The image on the left is the original image without any watermark. The image on the right shows the same image with a strong digital image watermark embedded in it. The watermark is virtually undetectable to the human eye.

Figure 7 also shows a Lena image before and after watermark embedding. The image on the left is the original one without any watermark. The image on the right shows the same image with copyright image watermark embedded in it. We can see from the illustration that the image quality is very good even after embedding. And that is the evident of the robustness of the algorithm used for embedding.



a) Host rose image. b) Watermarked rose image.

Figure 6. Digital coyright watermarking image.



a) Lena image before embedding. b) Watermarked Lena image.

Figure 7. Digital stego-image.

All the testing of the algorithm was made using MATLAB, and the following table consists the PSNR results that the proposed algorithm attained on several sizes of test images.

From Table 1, the performance results of the watermark extraction showed that the algorithm keeps the quality of the watermarked image after embedding. This conclusion is supported by the function values used to quantify image quality. Given that, the PSNR values showed that the watermarked image and the original image are identical, the value of PSNR for Lena host images was 0.9807, and it was 0.9811 for rose image. The watermarked image produced by the proposed approach has good imperceptibility.

Table 1. PSNR value results.

Cover image	Watermark	Image Size	Proposed algorithm (PSNR) of the watermarked image
Lena	copyright	512x512	0.9807
Rose	copyright	256x256	0.9811

Using MATLAB functions, several types of noise are purposefully added, distorting the watermarked host image like Gaussian noise and salt and pepper. A PSNR value is calculated in two ways, the first one among the watermarked image and the original image after noise addition. And also, among the original host image and the image after denoising (host image). A low PSNR value indicate that the image contains greater distortion. NCC is used to measure the correlation coefficient among the extracted watermark and the original watermark. A high NC value indicates that the extracted watermark is closer to the original watermark. Despite the host image's PSNR ratio, it is clear from the findings in the preceding table that the NC value of extracted watermark is high and reasonably stable, which is necessary to verify the owner's identity on the image (see Table 2).

Table 2. Experiment results using PSNR and NCC metrics for robustness and fidelity.

Authors	Extraction algorithm based on	Cover image size	Watermark size	Image denoising attack	PSNR of extracted watermark	NCC of extracted watermark
AL-Nabhani *et al.* [6]	DCT+ Probabilistic neural network	512 x 512	64 x 64	No attack	68.27 db	0.9779
		512 x 512	64 x 64	Gaussian noise (G=20)	67.04 db	0.9753
		512 x 512	64 x 64	Gaussian noise (G=50)	67.48 db	0.9863
		512 x 512	64 x 64	JPEG compression (Q=50)	57.20 db	0.7520
Chopra *et al.* [10]	DCT +DWT	256 x 256	20 x 50	No Attack	41.1613 db	1
		256 x 256	20 x 50	JPEG Compression (77%)	38.6302 db	1
		256 x 256	20 x 50	Salt and Pepper (20%)	35.3782 db	0.9983
Kumar [22]	DWT	512 x 512	8 x 8	No Attack	49.06 db	1
Rani *et al.* [25]	DWT + SVD	512 x 512	8 x 8	Gaussian noise (5%)	20.05 db	0.9764
	DWT + SVD	512 x 512	8 x 8	JPEG compression (Q=40)	28.50 db	0.9984
Proposed approach	DCT + DWT + PSO	512 x 512	20 x 50	No Attack	70.50 db	0.6761
		512 x 512	20 x 50	Gaussian noise (G=20)	66.24 db	0.9807
		512 x 512	20 x 50	Gaussian noise (G=30)	66.9 db	0.9800
		512 x 512	20 x 50	Salt and Pepper (20%)	34.0732 db	1
		256 x 256	20 x 50	Gaussian noise (G=20)	66.02 db	0.9811
		256 x 256	20 x 50	Gaussian noise (G=30)	66.48 db	0.9802
		256 x 256	20 x 50	Salt and Pepper (20%)	34.1752 db	1

According to the simulation results of the proposed method as shown in the previous table, the PSNR is 70.05 db with no attacks and all the NCCs are higher than 0.98 with different attacks types. When compared to the existing literature, the proposed approach shows good resilience against a variety of attacks, including noise. In particular, the NCC values are at least 0.98 higher than those in noise attacks especially, Gaussian noise.

However, in the proposed method the image was noised with Gaussian and denoised using DnCNN, and this makes PSNR not very well without attack as with attacks. But it makes a big difference in its robustness against several types of attack with different variations. Because it's already undergone the white Gaussian noise. Therefore, the quality of the image is extremely enhanced after this process. Table 2 shows the PSNR ratios of the host image after denoising. Before the extraction of the watermark.

From the Table 3 values, we can see that the images are extremely enhanced in our algorithm before the extraction of the watermark is done, and this is the strongest point in its performance. This happens due the AWG addition before.

Table 3. PSNR of the host image.

Image	PSNR	NCC
Lena image	0.96	1
Rose image	0.957	1

There are some limitations for this study such as: the algorithm is restricted to colored images only, it cannot deal with grayscale, and also our mechanism is restricted to 1-2 levels of DWT, because of size restrictions. Future recommendation including taking into consideration more types of attacks to test the robustness and fidelity of the proposed approach more deeply. Moreover, we plan to perform some pre-analysis for the images before the DCT hiding process to enhance the efficiency of the watermarking embedding and reconstruction.

5. Conclusions

In this study, we introduce a robust and efficient digital image watermarking technique that leverages DWT and DCT. PSO was utilized to optimize both the embedding and extraction processes of the watermark. In the final phase, we tested our method's resilience against various attacks, including Additive White Gaussian Noise (AWGN) and the salt and pepper filter, using the DnCNN denoising network, which is based on CNN, to assess its effectiveness against noise.

To determine the robustness of our algorithm, we employed the NCC and the PSNR as our primary metrics. The experimental results (refer to Table 2) from embedding and extracting watermarks across different host image sizes were encouraging. Additionally, our evaluations revealed that certain denoising attacks, while partially damaging the watermark, actually improved the overall image quality. The improvements in image quality were substantiated by the metrics utilized in our assessment.

Acknowledgement

This work was funded by Al-Zaytoonah University of Jordan under grant number (2023-2022/17/41).

References

[1] Abraham J. and Paul V., "An Imperceptible Spatial Domain Color Image Watermarking Scheme," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 1, pp. 125-133, 2019. https://doi.org/10.1016/j.jksuci.2016.12.004

[2] Ahvanooey M., Li Q., Shim H., and Huang Y., "A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents," *Security and Communication Networks*, pp. 1-22, 2018. https://doi.org/10.1155/2018/5325040

[3] Al-Fayoumi M., Al-Mimi H., Veisi A., Al-Aqrabi H., Daoud M., and Eftekhari-Zadeh E., "Utilizing Artificial Neural Networks and Combined Capacitance-Based Sensors to Predict Void Fraction in Two-Phase Annular Fluids Regardless of Liquid Phase Type," *IEEE Access*, vol. 11, pp. 143746-143747, 2023.

[4] Alia M. and Suwais K., "A Novel Steganography Scheme Based on Fractal Set," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 128-136, 2020.

[5] Al-Madi R., Saleh K., and Tarawneh M., "A Comprehensive Review on Digital Watermarking Security and Performance Evaluation," *Journal of Computational Security*, vol. 29, no. 1, pp. 102-121, 2024.

[6] AL-Nabhani Y., Jalab H., Wahid A., and Noor R., "Robust Watermarking Algorithm for Digital Images Using Discrete Wavelet and Probabilistic Neural Network," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 4, pp. 393-401, 2015.

[7] Alsarayreh M., Alia M., and Abu-Maria K., "A Novel Image Steganographic System Based on Exact Matching Algorithm and Key-Dependent Data Technique," *The International Arab Journal of Information Technology*, vol. 95, no. 5, pp. 1212-1224, 2017.

[8] Blum C. and Merkle D., *Swarm Intelligence: Introduction and Applications*, Springer Science and Business Media, 2008.

[9] Chang J., Chen B., and Tsai C., "LBP-based Fragile Watermarking Scheme for Image Tamper Detection and Recovery," *in Proceedings of the International Symposium on Next-Generation Electronics*, Kaohsiung, pp. 173-176, 2013.

[10] Chopra D., Gupta P., Sanjay G., and Gupta A., "LSB based Digital Image Watermarking for Gray Scale Image," *IOSR Journal of Computer Engineering*, vol. 6, no. 1, pp. 36-41, 2012.

[11] Chunwei T., Lunke F., Wenxian Z., Yong X., Wangmeng Z., and Chia-Wen L., "Deep Learning on Image Denoising: An Overview," *Neural Networks*, vol. 131, pp. 251-275, 2020. https://doi.org/10.1016/j.neunet.2020.07.025.

[12] Cruz C., Foi A., Katkovnik V., and Egiazarian K., "Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1216-1220, 2018. DOI: 10.1109/LSP.2018.2850222

[13] Dixit A. and Dixit R., "A Review on Digital Image Watermarking Techniques," *International Journal of Image, Graphics and Signal Processing*, vol. 9, no. 4, pp. 56-66, 2017.

[14] Dong C., Loy C., He K., and Tang X., "Image

Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295-307, 2015.

[15] Engelbrecht A., *Fundamentals of Computational Swarm Intelligence*, Wiley, 2005.

[16] Fukushima K., "A Self-Organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, pp. 193-202, 1980. https://doi.org/10.1007/BF00344251

[17] Geng L., Zhang W., Chen H., Fang H., and Yu N., "Real-Time Attacks on Robust Watermarking Tools in the Wild by CNN," *Journal of Real-Time Image Processing*, vol. 17, pp. 631-641, 2020.

[18] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y., "Generative Adversarial Nets," *in Proceedings of the Conference on Advances in Neural Information Processing Systems*, Montreal, pp. 2672-2680, 2014. https://doi.org/10.48550/arXiv.1406.2661

[19] Gu S. and Timofte R., "A Brief Review of Image Denoising Algorithms and Beyond," *Inpainting and Denoising Challenges*, pp. 1-21, 2019. https://doi.org/10.1007/978-3-030-25614-2_1

[20] Hatoum M., Couchot J., Couturier R., and Darazi R., "Using Deep Learning for Image Watermarking Attack," *Signal Processing: Image Communication*, vol. 90, 2021. https://doi.org/10.1016/j.image.2020.116019

[21] Jana M. and Jana B., "A new DCT Based Robust Image Watermarking Scheme Using Cellular Automata," *Information Security Journal: A Global Perspective*, vol. 31, no. 5, pp. 527-543, 2022.

[22] Kumar A., "A Review on Implementation of Digital Image Watermarking Techniques Using LSB and DWT," *in Proceedings of the Information and Communication Technology for Sustainable Development Conference*, New Delhi, pp. 595-602, 2020.

[23] Lucas A., Iliadis M., Molina R., and Katsaggelos A., "Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20-36, 2018. DOI:10.1109/MSP.2017.2760358

[24] Quan Y., Teng H., Chen Y., and Ji H., "Watermarking Deep Neural Networks in Image Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 1852-1865, 2020.

[25] Rani A., Bhullar K., Dangwal D., and Kumar S., "A Zero-Watermarking Scheme using Discrete Wavelet Transform," *in Proceedings of the 4th International Conference on Eco-Friendly Computing and Communication Systems*, Manchester, pp. 603-609, 2015. https://doi.org/10.1016/j.procs.2015.10.046

[26] Singh A., Kumar B., Singh G., and Mohan A., "Digital Image Watermarking: Concepts and Applications," *Medical Image Watermarking: Techniques and Applications*, pp. 1-12, 2017. https://doi.org/10.1007/978-3-319-57699-2_1

[27] Sridhar S., *Digital Image Processing*, Oxford Publications, 2016.

[28] Wadhera S., Kamra D., Rajpal A., Jain A., and Jain V., "A Comprehensive Review on Digital Image Watermarking," *arXiv Preprint*, vol. arXiv:2207.06909, 2022.

[29] Yadav N., Rajpoot D., and Dhakad S., "Optimization of Watermarking in Image by Using Particle Swarm Optimization Algorithm," *in Proceedings of the 6th International Conference on Signal Processing and Communication*, Noida, pp. 85-90, 2020.

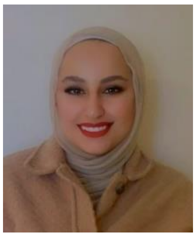
[30] Yadav N., Rajpoot D., and Dhakad S., "Enhancing Digital Image Watermarking Using Deep Learning Approaches," *Journal of Image Processing and Security*, vol. 25, no. 3, pp. 451-467, 2023.

[31] Zhang K., Zuo W., Chen Y., Meng D., and Zhang L., "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *in Proceedings of the IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142-3155, 2017. doi: 10.1109/TIP.2017.2662206

[32] Zhou Y., Chellappa R., and Jenkins B., "A Novel Approach to Image Restoration Based on a Neural Network," *in Proceedings of the International Conference on Neural Networks*, San Diego, pp. 269-276, 1987.

Mohammad Alia is a Professor of Computer Science at AL-Zaytoonah University of Jordan. He received his PhD from the University Sains Malaysia (USM), Penang, Malaysia, 2008. His research interests include public key cryptosystems, fractals, image processing and steganography, wireless networks and machine learning.

Adnan Hnaif is a full Professor at the Cybersecurity Department, Faculty of Science and information Technology, Al-Zaytoonah University of Jordan. He received his Ph.D. Degree in Network Security from University Sains Malaysia-National Advanced IPv6 Centre and Excellence (NAV6) in 2010. His researches focus on the Network Security and Network Monitoring.



Aseel Alrawashdeh received the Master Degree in Computer Science, Al-Zaytoonah University of Jordan, Jordan, in 2023. Her research interests in Cyber Security.

Yousef Jaradat is a Professor of Electrical and Computer Engineering at Al-Zaytoonah University of Jordan. He received his Ph.D. from New Mexico State University, New Mexico, USA, in 2012. His research interests include Wireless Networks, Network Modeling and Simulation, AI and Machine Learning, Computer Security and Quantum Computing.

Mohammad Masoud is a Professor of electrical Engineering at Al-Zaytoonah University of Jordan. He received his Ph.D. in Communication Engineering and Information Systems from Huazhong University of Science and Technology (HUST), Wuhan, China in 2012. He is a reviewer in many computer and communication journals. His research interests include Computer Network Measurements, Network Security, Machine Learning, Software Defined Networking (SDN), Embedded Systems, Control Theory and Cyber Physical Systems (CPS).

Ahmad Manasrah is currently an Associate Professor of Mechanical Engineering at Al-Zaytoonah University of Jordan. He received his Ph.D. Degree from The University of South Florida. He was a research assistant and a member of Rehabilitation Engineering and Electromechanical Design Lab at the USF. He is also a member of ASHRAE, Jordan. His interests include Renewable Energy, Smart Energy Technology Mechanical Control, and Education.

Ahmad Alshanty received the Ph.D. Degree in Computer Engineering from the Department of Computer Engineering, Girne American University, Cyprus, in 2017. Alshanty is an Assistant Professor specializing in Cybersecurity Department at Al-Zaytoonah University. With a solid academic background and a Ph.D. in Networking Security. His research interests and areas of expertise include: WSN Security, IDS, IoT Security and Security Protocols.