Hindawi
Security and Communication Networks
Volume 2023, Article ID 9894683, 1 page
https://doi.org/10.1155/2023/9894683

Retraction

Retracted: Detecting Digital Watermarking Image Attacks Using a Convolution Neural Network Approach

Security and Communication Networks

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. Alzahrani, "Detecting Digital Watermarking Image Attacks Using a Convolution Neural Network Approach," *Security and Communication Networks*, vol. 2022, Article ID 4408336, 12 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4408336, 12 pages
https://doi.org/10.1155/2022/4408336

Research Article

Detecting Digital Watermarking Image Attacks Using a Convolution Neural Network Approach

Ali Alzahrani

Department of Computer Engineering, King Faisal University, Al Hofuf, MB-400-Alahsa-31982, Saudi Arabia

Correspondence should be addressed to Ali Alzahrani; aalzahrani@kfu.edu.sa

Received 18 April 2022; Revised 13 May 2022; Accepted 17 May 2022; Published 6 June 2022

Academic Editor: Mohammad Ayoub Khan

In scientific research, one of the most significant problems of recent years has been and continues to be the protection of digital material. The advancement of Internet technology has allowed for the illicit duplication, authentication, and distribution of digital material by unauthorized individuals. For this reason, a variety of watermarking systems have been investigated for a variety of purposes, including broadcast monitoring, intellectual property protection, content authentication, and copy control. There are various types of the watermarking image attacks that impact the quality of the images; therefore, it is critical to ensure that watermarked digital images can withstand these kinds of attacks. Hence, novelty of the proposed research is to develop approaches to detect these attacks which becomes very important to guarantee a sufficient quality of watermarking images. In this paper, a deep learning method based on a convolution neural network (CNN) algorithm was proposed to detect various types of watermarking attacks, namely, median filter, Gaussian filter, salt-and-pepper, average filter, motion blur, and no attack, to improve the watermarking quality. Evaluation metrics such as peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and the normalization correlation (NC) were employed to examine the invisibility and robustness of the watermarking images. The empirical results of the CNN model show good performance for detecting watermarking attacks with different sizes (256, 128, and 64). The accuracy percentage of the testing process was 98%. A highly efficient CNN approach was developed. Very high performance of NC was found in the detection of the salt-and-pepper attack (99.02%, 99.97%, and 99.49% with respect to watermarking image sizes of 256×256, 128×128, and 64×64, respectively). The study concludes that the CNN model is able to detect watermarking attacks successfully.

1. Introduction

Multimedia communication has become very simple, efficient, and cost effective in today's world of the Internet. Digital multimedia may be readily tampered with and manipulated by the wrong person. Digital watermarks have been suggested as a method of protecting multimedia data against infringement of intellectual property rights. There are several different watermarking techniques that are intended primarily for copyright protection and data authentication. When it comes to copyright protection, it is critical to be able to identify who is the legitimate owner of a picture. Even if the watermarked data is processed, duplicated, or redistributed, the embedded information should be decipherable from the watermarked data [1]. Digital technology has a variety of potential uses. Copyright protection and dissemination are among the benefits of watermarking. Tracing, authentication, and approved access control are examples of the application and use of such technologies, as well as clandestine communication.

Regarding watermarking, picture watermarking in particular has garnered a great deal of attention from the scientific community. When compared to audio and video watermarking, the majority of the research is devoted to picture watermarking. There might be three possible explanations for this. First, test pictures are readily available. Second, images contain sufficient redundant information to allow for the easy embedding of watermarks. Finally, any effective image watermarking technique may be updated to work with video as well. Images are represented/stored in

both the spatial and transform domains, depending on the application. The picture in the transform domain is represented in terms of frequencies, while the image in the spatial domain is represented in terms of pixels. In layperson's terms, the transform domain refers to the process of segmenting a picture into various frequency bands. When converting a picture to a frequency representation, we may apply a variety of reversible transforms, such as the discrete Fourier transform (DFT), Discrete Cosine Transform (DCT), and discrete wavelet transform (DWT), among others [2, 3].

Spread Spectrum (SS) and Quantization Index Modulation (QIM) are two forms of watermarking technologies. Spread Spectrum (SS) is a type of digital signal processing. In addition to being additive, SS and QIM are also substitutive. Comparing Spread Transform Dither Modulation (STDM) to other QIM versions, it can be seen that it is highly robust to both quantization error and random noise [4]. It is conceivable to use a combination of QIM's robustness and STDM's effectiveness. Watermarking methods are distinguished by a variety of features, the most important of which are payload, resilience, and fidelity. As a consequence, transforms such as Singular Value Decomposition (SVD), DCT, and DWT are often used. The DCT is more stable when compared to the spatial domain. This is especially true when it comes to simple image processing techniques such as brightness adjustment, blurring, and low pass filtering. Interesting results can also be obtained by using the DWT transformation, which enhances the sturdiness of the photos that have been watermarked. When creating a picture at different resolutions and processing the image from high resolution to low resolution, it is the DWT domain that is responsible. Increasing the amount of energy used to conceal a watermark in a photograph will enhance the image's durability. With the increasing popularity of SVD, it is becoming increasingly difficult to counterfeit. In general, SVD has a moderate degree of resistance against the majority of types of watermarking attacks [5, 6].

With digital picture watermarking, data is implanted into a multimedia product and afterwards removed from it or identified inside the watermarked product. Using these technologies, tamper-resistant imaging is ensured, as is authentication, content verification, and picture integration [7]. Changing the format of the watermarked data or displaying or converting the watermarked data into a different file format is not a simple process. As a result, after an attack, it is feasible to deduce information about the change from the watermark left behind. It is critical to be able to distinguish between digital watermarking and other methods such as encryption [8]. Digital picture watermarking methods may also be used to survive digital-to-analog conversion, compression, file format changes, re-encryption, and decryption, as well as other types of data loss. Because of these tasks, it may be used as a substitute for (or in conjunction with) cryptography. By using the material as intended, the information is integrated in the content and cannot be deleted [9].

The term "steganography" comes from the Greek word "steganos," which means "hidden writing." This method hides communication and alters a picture in such a way that only the sender and the intended recipient are aware of the message that has been transmitted. The use of this approach makes the process of detection more complex. As an alternative to encrypting communications, steganography may be used to conceal them inside other inoffensive-looking items, preventing their discovery. As a result, steganography can be used as a privacy and security technique in addition to encryption. As a result of the fast expansion of the Internet and computer networks, steganography has the potential to be exploited as a tool for transferring information and planning terrorist acts [10]. When using steganography, a cover picture is hidden from view; however, when using watermarking, a message is embedded into the real content of a digital signal, which is then embedded into the signal itself. Consequently, an eavesdropper will be unable to delete or change a message in order to acquire an output message. In order to safeguard material from unwanted access, it is necessary to embed information within the original picture itself. Unauthorized individuals will have a difficult time detecting and removing digital picture watermarking. A variety of algorithms have been developed for implementing this approach in both the spatial and frequency domains, each with its own set of advantages and limitations.

With the advent of the Internet, it has never been easier or cheaper to share multimedia content with others. The incorrect individual may easily tamper with and alter digital multimedia. Using digital watermarks to guard multimedia material against theft of intellectual property has been proposed. Therefore, developing system for detecting different attacks that effect the digital watermarking images is a main motif of this research.

Watermarking is the process of embedding the owner's information (watermark [WM]) into material, and the resulting file is either kept or shared. This approach is intended to assert ownership by extracting the encoded WM information when it is required. Various technologies have been explored and created in accordance with the technologies now in use, the application sector, and so on. A number of algorithms for performing WM embedding, extracting the WM according to the embedding process, and modifying the WM have been suggested [3–8, 11], but they have not yet been implemented. If the WM must be rendered invisible, a standard technique embeds it in the domain of the discrete cosine transform (DCT), the discrete wavelet transform (DWT) domain [12], the discrete Fourier transform (DFT) domain [13], or the quantization index modulation (QIM) domain [14–17], respectively. As a general rule, watermarking may be compromised by a malicious assault aiming to corrupt or erase the embedded WM information, as well as by a nonmalicious attack caused by unavoidable procedures such as those used to store or distribute material. Accordingly, either algorithmically or deterministically generated WM embedding may be achieved. The extraction of WM, on the other hand, is a different matter. It is possible that the WM-embedded host data may be destroyed as a result of malicious or nonmalicious assaults, and the embedded WM data will also be

affected. As a result, it may not be acceptable to extract the WM in an algorithmic or deterministic manner, and a more statistical approach may provide better results. For these and other reasons, recent research has tended to execute watermarking using a neural network (NN) [18–24], which is a kind of artificial neural network that learns from experience. The main contributions of this research are as follows:

(1) Developing an approach to detect the various types of watermarking images to enhance robustness.

(2) Testing the CNN model with different sizes of watermarking images (256×256, 128×128, and 64×64).

2. Materials and Methods

In this section, the proposed system of detecting digital watermark attacks by using a CNN model is presented (see Figure 1).

2.1. Deep Learning Approaches. Artificial neural networks (ANNs) with several layers are referred to as "deep learning" or "deep neural networks." With their ability to manage a large quantity of data, they have been a prominent instrument in the literature over the last several decades [25]. Recently, there has been an increase in the desire to have deeper hidden layers, particularly in pattern and picture recognition [26, 27]. The CNN is a well-known deep neural network. When matrices are conjoined in a linear operation, it is referred to as convolution. In addition to the convolutional and nonlinear layers, there are also pooling and fully connected layers in the CNN. There are parameters for the convolutional and fully connected layers, but there are none for the pooling or nonlinearity layers [28, 29].

Image processing and speech recognition are just two of the many disciplines in which CNN has achieved revolutionary accomplishments in the last decade. The greatest advantage of CNNs is the reduction in the number of parameters needed in an ANN [30, 31]. Researchers and developers are now looking at larger models to handle difficult tasks that were previously impossible with conventional ANNs. To solve an issue using a CNN, one needs to avoid having characteristics that are spatially dependent. There is no need to pay attention to the location of the faces in the photographs while using a face detection program. The only thing that matters is that the algorithm finds them, no matter where they are in the photos. As input travels through the network's layers, CNNs are able to extract abstract characteristics.

2.1.1. Convolution. In convolution, all pixels in the picture or frame are treated the same, which is a per-pixel action. As a result, as the size of a picture or frame grows, so do the complexity and time needed to conduct convolution operations. A filter (also known as a kernel) is a two-dimensional real number matrix or a matrix that is less than the input picture or frame dimensions. Filter coefficients might vary depending on the application.

Convolution is conducted as follows: The filter glides over an image or frame from the top right corner and passes over each pixel until the lower left corner is reached. The filter's receptive field is the same size as the filter's image area. This is achieved by multiplying each kernel number by the appropriate picture or frame value and then summarizing the result. If we have an RGB image (three channels), three filters should be used for the convolution operation. As a result, the filter dimensions are $3 \times 3 \times 3$, and each 3×3 filter is utilized for a single channel to form a single feature map in the output, as illustrated in Figure 2.

2.1.2. Stride. When we slide the filter across an image or frame matrix, we may define stride as the number of pixels by which we slide the filter. As a result, when we say stride = 1, we imply that we only move the filter one pixel over the input picture; however, when we say stride = 2, we indicate that we hop the filter two pixels over the input image with each step. By looking at the areas in the previous example, it might be inferred that the node of the following layer has a large number of overlaps with its neighbors, which is simply not the case. It is possible to alter the overlap by adjusting the stride. Figure 3 shows the stride of the proposed system.

$$O = 1 + \frac{N - F}{S}, \tag{1}$$

where N is the image size, the fitter size is indicated by F, and S denotes the stride size.

2.1.3. Padding. A downside of the convolution stage is the loss of information that can be present on the image's borders. As a result of the filter sliding, they never get the opportunity to be viewed. As a simple and effective solution, zero padding may be used. Zero padding is introduced into the input matrix border to regulate the output of the convolution operation and to include the border pixels of the input picture or frame at the same time. The filter may be dragged across the whole input matrix in this situation. The padding layer is shown in Figure 4.

$$O = 1 + \frac{N + 2P - F}{S}, \tag{2}$$

where P is number of layers; N is input image; F is filter layer.

2.1.4. Features of the CNN. The model gains invariance translations as a result of the weight sharing and aids in the filtering of the learnt feature, independent of the spatial characteristics of the feature. In order to determine the optimal values for the filter that will be utilized in the convolution process, Figure 5 shows the convolution layer of the CNN model.

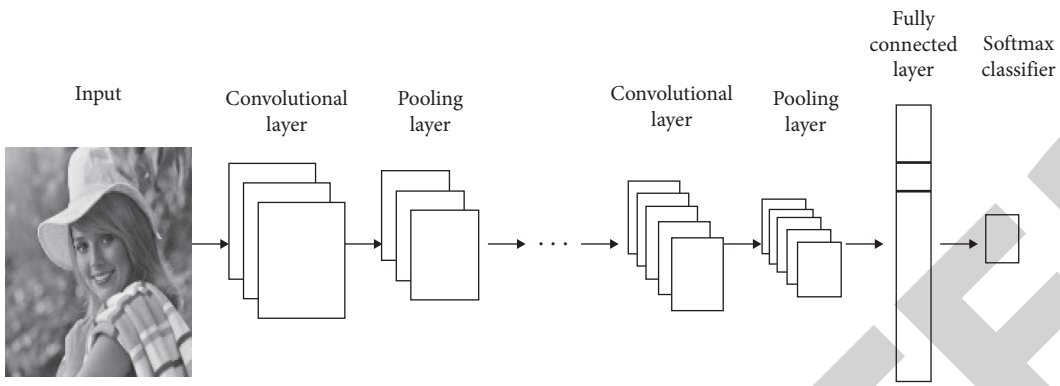

Figure 1: CNN model.

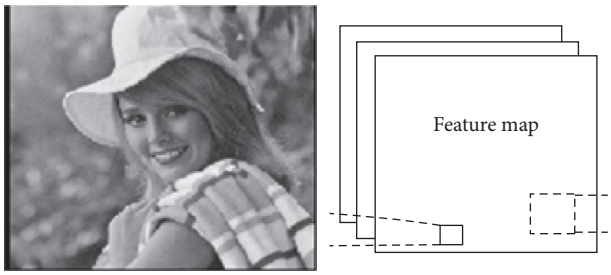

Figure 2: Filter of convolution operation.

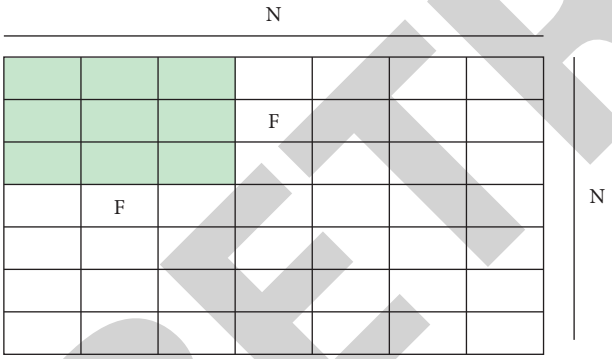

Figure 3: Size stride.

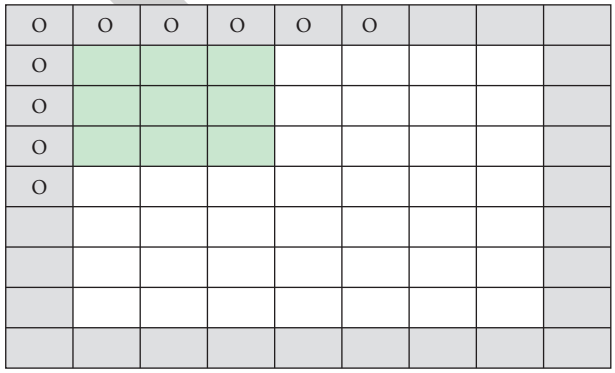

Figure 4: Padding operation.

$$\begin{aligned} \text{net}(i,j) &= (x * w)[i,j] \\ &= \sum_{m}^{i} \sum_{n}^{j} x[m,n] \times w[i-m, j-n]. \end{aligned} \tag{3}$$

where $\text{net}(i,j)$ is output, x is input image, w is kernel or filter, and $*$ is convolution operation.

The nonlinearity layer follows the convolution. The produced output may be adjusted or turned off using the nonlinearity. In order to restrict or saturate the output, this layer is used. There is always a convolution layer nested inside the nonlinearity layer. Sigmoid and tanh have been the most common nonlinearity functions for many years now. Figure 6 illustrates the most frequent nonlinearity functions. ReLU has simpler definitions in both function and gradient, as shown in the following two equations:

$$\begin{aligned} \text{ReLU}(x) &= \max(0, x), \\ \frac{d}{dx}\text{ReLU}(x) &= \{1 \, if \, x > 0; \; 0 \, \text{otherwise}\}. \end{aligned} \tag{4}$$

In the backpropagation, saturation functions like sigmoid and tanh are problematic. Known as the "vanishing gradient," a gradient signal gradually disappears as the depth of a neural network's architecture increases. Since the gradient of those functions is almost negative everywhere outside the center, this is what ends up happening. For the positive input, however, the ReLU has a constant gradient. Despite the fact that the function cannot be differentiated, this does not matter for implementation purposes.

2.1.5. Pooling Layer. There are two main reasons why CNN employs pooling. First, the output feature map of pooling has a predetermined size, which is necessary for the classification process to function properly. Example: If you have 512 filters and you apply maximum pooling to each of them, you will obtain a result that is 256 dimensions in size, regardless of the size of your filters. Second, pooling may be utilized in conjunction with nonequal filters and strides to increase efficiency. For example, a 3×3 max-pooling with a stride of 2 maintains certain overlaps between the regions under consideration. Furthermore, for the reasons stated above, the pooling layer reduces network overfitting by

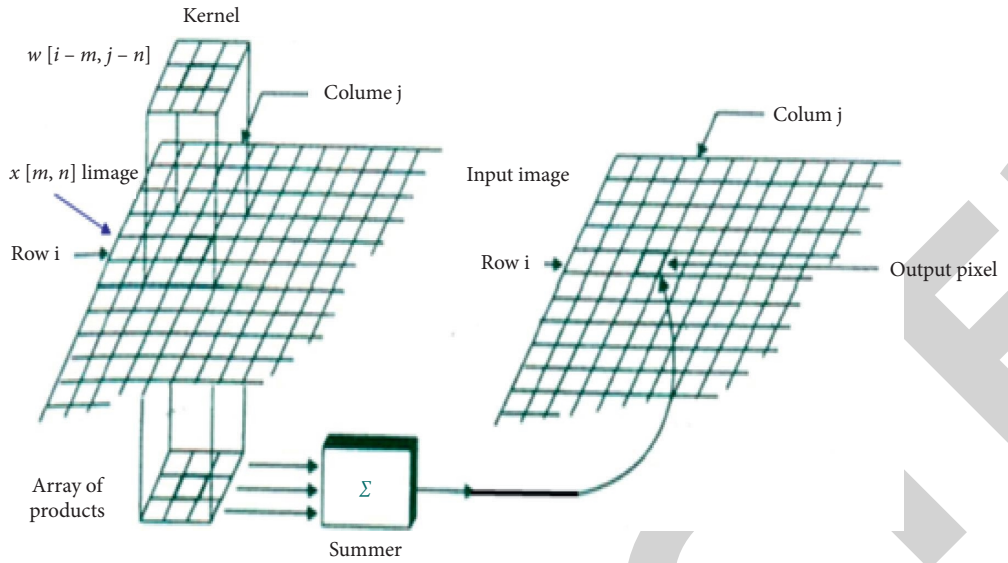

Figure 5: Convolution layer.

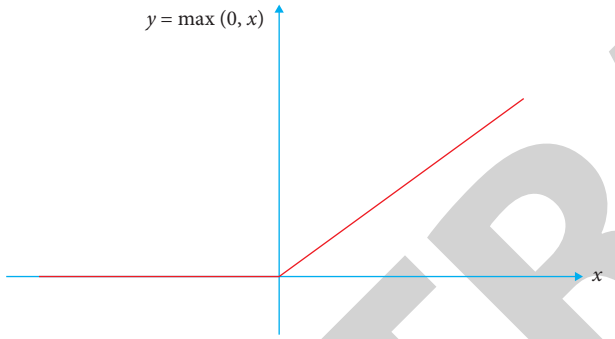

Figure 6: ReLU function.

reducing the number of parameters and computations utilized in the network and also scales the staple representation of the input picture to make it more readable. As a result of pooling, the network is motivated to be very invariant to even minor changes in the input picture, such as transformations, distortions, and translations.

2.1.6. Fully Connected Layer. The fully connected layer is comparable to the way neurons are placed in a standard neural network in that it is completely coupled to everything else. This picture demonstrates each of the nodes in the final frames of the convolutional or ReLU or pooling layer before it is linked as a vector to the first layer from the fully connected layer, as shown in the previous image. These are the parameters that are most often employed with the CNN inside these layers, and they require a significant amount of time to train.

2.1.7. SoftMax Layer. The softmax function (also known as the normalized exponential function) is often regarded as the most effective means of displaying categorical distributions. When the softmax function is called, it receives as input an N-dimensional vector of units, with each unit represented by an arbitrary real value. As opposed to this, the result is an M-dimensional vector (N M) with real values ranging between 0 and 1 million (0 and 1). In this case, the big value is converted to a real number close to one, while the tiny value is changed to a real number near zero. To be valid, the total of all values of output must equal one (1). As a result, the output has a high likelihood of remaining unchanged. Figure 7 shows the softmax layer. The important parameters of CNN model for detecting watermarking attacks are presented in Table 1.

$$O_i = \frac{e^{z_i}}{\sum_{i=1}^{M} e^{z_i}}, \tag{5}$$

where O_i is output number, i, z_i is output i, and M is total number.

2.2. Measurement Performance. Performance measurements such as the PSNR, NC, and SSIM were applied to analyze the results of the deep leaning model.

$$\mathrm{PSNR}(C, C^*) = 101g\frac{C_{\max}^2}{\mathrm{MSE}}, \tag{6}$$

$$\mathrm{MSE} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}(C_i\, j - C_i^*,\, j)^2, \tag{7}$$

$$\mathrm{SSIM}(C, C^*) = \frac{\mu c \mu c^* + d_1}{\mu^2 c + \mu^2 c + d_1}.\frac{\sigma cc^* + d_2}{\sigma^2 c + \sigma^2 c * + d_2}, \tag{8}$$

$$\mathrm{NC} = \frac{\sum_{I=1}^{N}\sum_{J=1}^{N} W_{i,j}\, w_{i,j}^*}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j}^2}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j}^2}}, \tag{9}$$

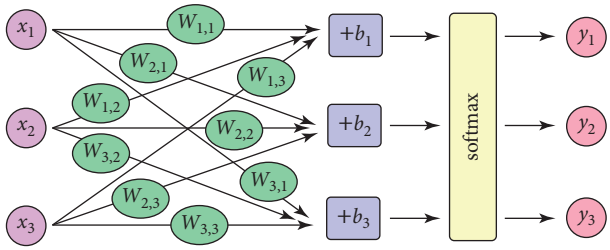

Figure 7: Softmax layer of CNN model.

Table 1: Important parameters of the CNN model.

Parameters	Values
Kernel size	3
Max-pooling size	2
Drop out	0.50
Fully connected layer	128
Activation function	ReLu function
Optimizer	Adam
Epochs number	20
Batch size number	50

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \times 100\%, \quad (10)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%, \quad (11)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \quad (12)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%. \quad (13)$$

where the evaluation metrics, namely, PSNR, SSIM, and NC, were examined. M is the output from the CNN model, C_i is the input image, the watermarking image is denoted by C_j, and μc and σ^2 are average values and variance, respectively. TP denotes true positive, TN is true negative, FP is false positive, and FN is false negative.

3. Experiment Results

In this study, the convolution neural network model was applied to detect various types of watermark attacks. For detecting these attacks, 50 watermarking images were examined during the training process to test the performance of the CNN model. In this research, we have considered the median filter, Gaussian filter, salt-and-pepper, average filter, motion blur, and no attack. The evaluation metric SSIM was used to evaluate the quality of the digital image after extracting the watermarking, and the image robustness was examined using the NC metric. The MATLAB programming language was used to run the CNN approach for detecting the watermarking attacks. The system employed a computer with 8 GB RAM and running the Windows 10 operating system.

3.1. Results Discussion. In this section, we have embedded a grayscale image watermark as the input image. The grayscale image is shown in Figure 8.

Figure 8(a) is the original image, with a size of 512×512, and the embedded image is shown in Figure 8(b), with a size of 256×256. The most common attacks are based on the watermarking images, to analyze the robustness and quality of the image. Table 2 shows the results of the CNN model for extracting the attacks by different sizes of watermarking images.

The watermarking images of varying sizes show good performance; however, watermarking with a smaller size (64×64) gave the best results. The extraction of the embedded image from the cover image was evaluated by using NC for finding the robustness between grayscale images with different sizes. Figure 9 shows the performance of the CNN model for different sizes of watermarking images.

Table 3 shows the results of the CNN model for detecting various types of attacks on the watermark images with a size of 256×256. These attacks are used to create an issue on the watermarked images; therefore, developing the system for managing and predicting attacks can help in improving the quality of watermarking. It can be seen that the CNN model achieved high performance for detecting median filter (98.62%), Gaussian filter (98.17%), and average filter (98.13%) attacks.

Table 4 shows the results of the CNN model in extracting the watermark from the cover image with a size of 256×256. It can be seen that the salt-and-pepper was closest with respect to the NC metric (99.02) where no attack image was 99.99%. The lowest score was seen for Gaussian filter (90.23). A graphical representation of the performance of the CNN model to detect attacks on watermark images with a size of 256×256 is presented in Figure 10.

Table 5 shows the results of the CNN model for detecting various attacks on original images with a size of 128×128. It is observed that the median filter, Gaussian filter, and average filter achieved high performance (98.61%, 98.10%, and 98.66%, respectively), while the performance of the CNN model for detecting motion blur was considerably less (94.13%).

The results of the CNN model in extracting the watermark image from the cover image with a size of 128×128 are shown in Table 6. The CNN model achieved good results for detecting salt-and-pepper attacks (99.97) but achieved poorer results in the detection of a motion blur attack. Figure 11 shows the quality of images extracted for each type of attack.

Table 7 shows the results of the proposed system by employing the NC metric to determine the level of robustness of the parameter values of the attacked watermark images. The proposed system achieved good robustness for enhancing the quality of embedded images.

The obtained results of the CNN model for detecting attacks on watermarking images with a size of 64×64 are presented in Table 8. The NC metrics were used to evaluate the results of extracting digital watermark images from the original cover images using various attack types. It can be seen that the CNN obtained good results for detecting salt-

(a) (b)

Figure 8: Grayscale watermark image; (a) cover image; (b) watermark image.

Table 2: Results of the CNN model for extracting the attacks by different sizes of watermarking image.

Images	Size	PSNR	SSIM	NC
Watermark image	256 × 256	37.61	99.91	
Watermark image	128 × 128	43.39	99.97	
Watermark image	64 × 64	49.21	99.99	
Extracted watermark image	256 × 256			99.99
Extracted watermark image	128 × 128			99.99
Extracted watermark image	64 × 64			99.97

Figure 9: Performance of CNN model for various sizes of watermark images.

Table 3: Results of the CNN model attack with a watermark image size of 256 × 256.

Attacks	PSNR	SSIM
Median filter	36.32	98.62
Gaussian filter	35.72	98.17
Salt-and-pepper	33.50	97.11
Average filter	35.66	98.13
Motion blur	32.35	97.09
No attack	37.61	99.99

Table 4: Results of extraction of watermark image from cover image with a size of 256 × 256.

Attacks	NC
Median fitter	96.89
Gaussian fitter	90.23
Salt-and-pepper	99.02
Average filter	98.54
Motion blur	95.23
No attack	99.99

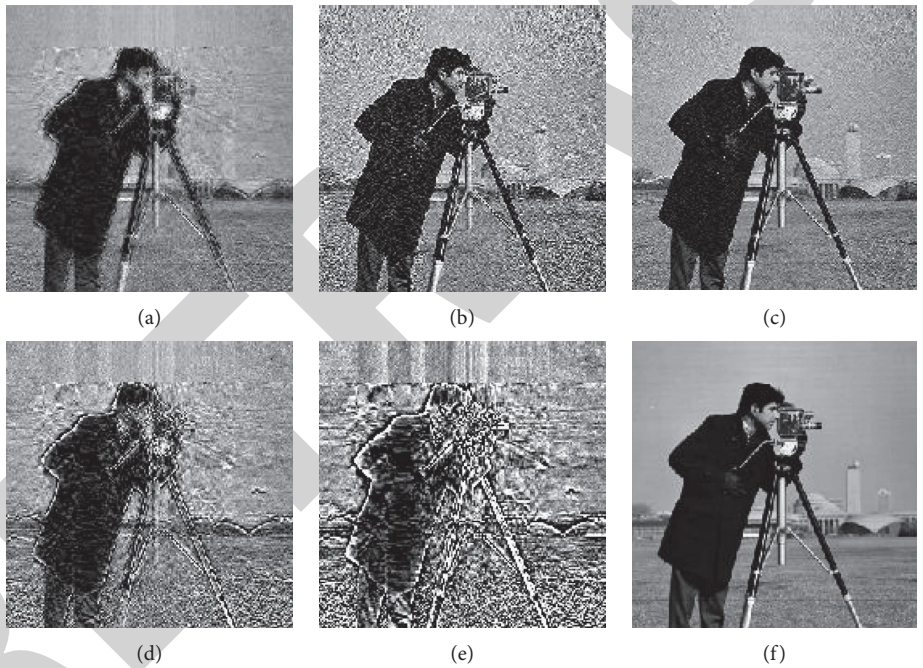

Figure 10: Extract attack from watermark images with a size of 256 × 256; (a) median filter; (b) Gaussian filter; (c) salt-and-pepper; (d) average filter; (e) motion blur; (f) no attack.

Table 5: Results of the CNN model attack with a watermark image size of 128 × 128.

Attacks	PSNR	SSIM
Median fitter	39.43	98.61
Gaussian fitter	38.06	98.10
Salt-and-pepper	34.06	97.03
Average filter	37.96	98.66
Motion blur	31.07	94.13
No attack	43.39	99.99

Table 6: Results of extraction of watermark image from cover image with a size of 128 × 128.

Attacks	NC
Median fitter	96.89
Gaussian fitter	92.83
Salt-and-pepper	99.97
Average filter	92.63
Motion blur	90.65
No attack	99.99

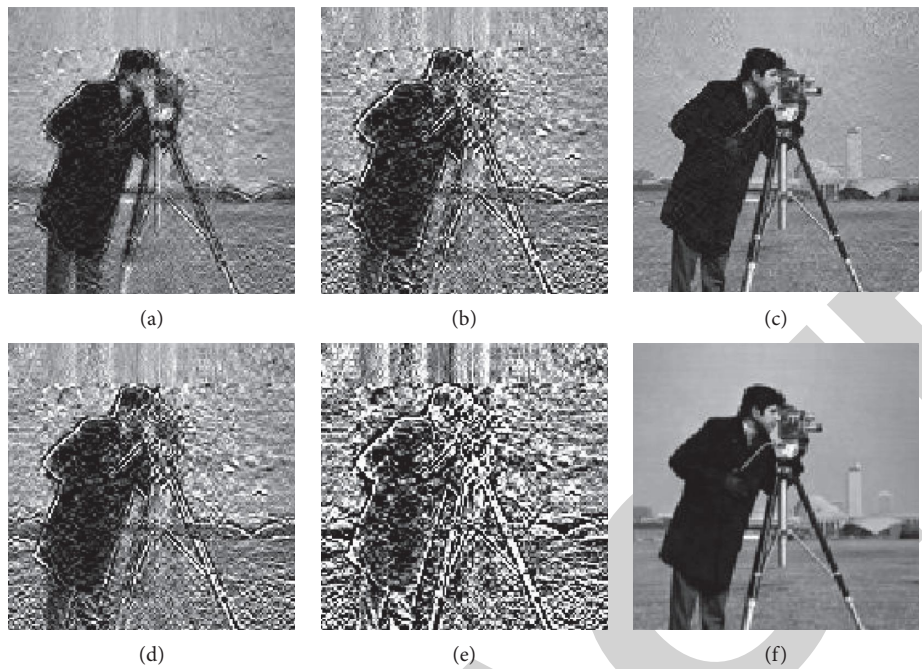

Figure 11: Extract attack from watermark images with a size of 128 × 128; (a) median filter; (b) Gaussian filter; (c) salt-and-pepper; (d) average filter; (e) motion blur; (f) no attack.

Table 7: Results of the CNN model attack with a watermark image size of 64 × 64.

Attacks	PSNR	SSIM
Median fitter	40.89	98.61
Gaussian fitter	38.96	98.10
Salt-and-pepper	35.20	96.99
Average filter	38.83	98.99
Motion blur	31.89	90.23
No attack	49.21	99.99

Table 8: Results of extraction of watermark image from cover image with a size of 64 × 64.

Attacks	NC
Median fitter	98.87
Gaussian fitter	95.15
Salt-and-pepper	99.49
Average filter	93.13
Motion blur	91.25
No attack	99.99

and-pepper attacks. The performance of the CNN model for detecting attacks on watermarking images with a size of 64 × 64 is shown in Figure 12.

Table 9 summarizes the results of the CNN model in the training process. It can be seen that the CNN model achieved high accuracy performance (98%) for detecting various types of watermarking extraction attacks.

Figure 13 displays the performance of the CNN model with respect to the NC metric for detecting watermarking attacks, where the *Y*-axis shows the performance of the CNN model and the values of extracting watermarking images are shown on the *X*-axis. The graphic representation shows a high percentage of the CNN model to detect salt-and-pepper and median filter attacks with different sizes of image (256 × 256, 128 × 128, and 64 × 64). The obtained results show the efficient performance of the CNN model.

Furthermore, with the suggested system of CNN model for detecting various attacks from watermarking images, we compared proposed results with other systems; it was noted that the findings from the proposed system attained great

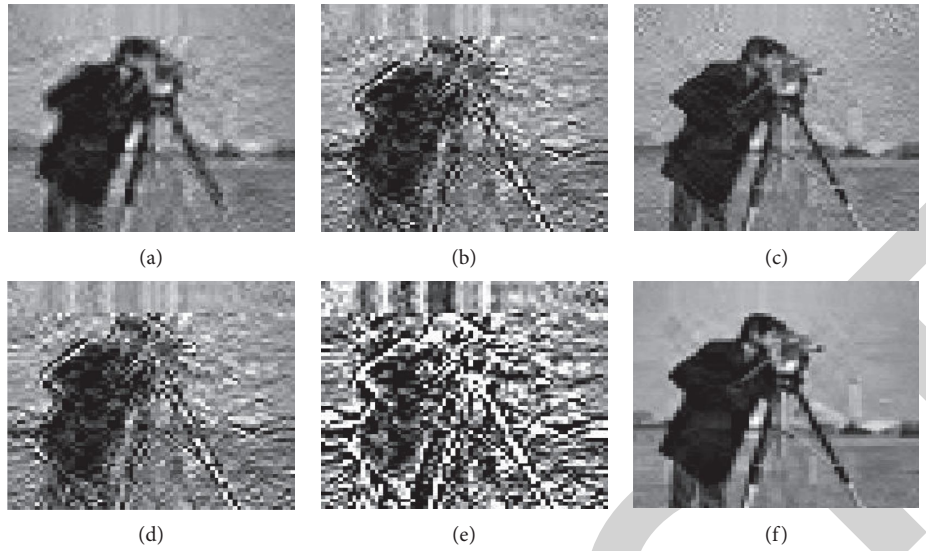

Figure 12: Extract attack from watermark images with a size of 64 × 64; (a) median filter; (b) Gaussian filter; (c) salt-and-pepper; (d) average filter (e) motion blur; (f) no attack.

Table 9: Results of the CNN model.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
CNN	98	99.05	99.22	96.64

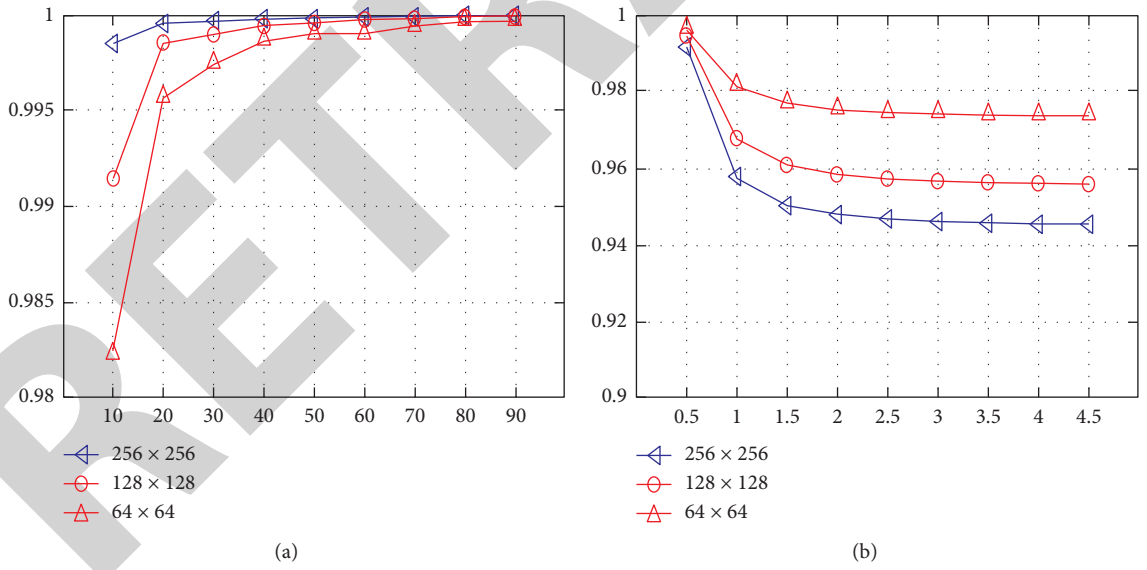

Figure 13: Performance of CNN model; (a) salt-and-pepper; (b) median filter.

Table 10: Comparison results with existing models.

Ref.	Attacks	NC metric (%)	SIMM metric (%)	Model
Ref. [32]	Median	99.66	87.07	Deep learning
	Salt-and-pepper noise	89	61.02	
	Average filter	082	41.08	
Ref. [33]	Median	Not used		DWT method
	Salt-and-pepper noise	57.75		
	Average filter	Not used		
Ref. [34]	Median	Not used		SVD
	Salt-and-pepper noise	98.24		
	Average filter			
Ref. [35]	Median	95.8	97.19	DW-SVD
	Salt-and-pepper noise	98.24	97.39	
	Average filter	90.63	96.68	
Proposed system	Median	98.87	98.61	CNN model
	Salt-and-pepper noise	99.49	98.10	
	Average filter	95.15	96.99	

accuracy in identifying different watermarked attacks. Table 10 shows the significant results of CNN model against the different watermarking systems.

4. Conclusion

In order to secure the hidden information included inside digital media and to maintain ownership of certain media data, digital watermarking is used. Many ways have been developed to ensure that the watermark is both active and resistant to various types of assault while still being secure. Aspects of digital image watermarking procedures may be divided into two primary groups based on the extraction methods used: blind watermarking and nonblind watermarking. In this paper, deep learning based on the convolution neural network approach was applied to detect various attacks, namely, median filter, Gaussian filter, salt-and-pepper, average filter, motion blur, and no attack. Various types of original watermark were processed as training for building the CNN model. The testing process used grayscale digital watermarking images of various sizes (256×256, 128×128, and 64×64) for testing the proposed system. The experiment results revealed that the CNN model shows high values of reliability for detecting the various types of watermark attacks. The CNN-LSTM model will be proposed in future for enhancing the existing results.

Data Availability

The dataset has been collected from standard repository, https://www.kaggle.com/datasets/felicepollano/watermarked-not-watermarked-images.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported through the Annual Funding Track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (AN000238).

References

[1] Miyazakim and A. Okamoto, *Analysis of Watermarking Systems in the Frequency Domain and its Application to Design of Robust Watermarking Systems*, pp. 506–509, Kyushu University, Fukuoka, Japan, 2001.

[2] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, "A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 776–786, 2003.

[3] J. George, S. Varma, and M. Chatterjee, "Color image watermarking using DWT-SVD and Arnold transform," in *Proceedings of the 2014 Annual IEEE India Conference (INDICON)*, pp. 1–6, Pune, India, December 2014.

[4] W. C. Chen, H. Y. Chen, C.-D. Hsu, J.-Y. Wu, and F.-J. Tsai, "No association of vitamin D receptor GeneBsmI polymorphisms with calcium oxalate stone formation," *Molecular Urology*, vol. 5, no. 1, pp. 7–10, 2001.

[5] Q. Li, C. Yuan, and Y. Zhong, "Adaptive DWT-SVD domain image WatermarkingUsing human visual model," in *Proceesings of the 9th International Conference on Advanced Communication Technology*, vol. 3, pp. 1947–1951, Baton Rouge, LA, USA, 2007.

[6] A. K. Singh, M. Dave, and A. Mohan, "Hybrid technique for robust and imperceptible image watermarking in DWT-DCT-SVD domain," *National Academy Science Letters*, vol. 37, no. 4, pp. 351–358, 2014.

[7] H. Tao, L. Chongmin, J. Mohamad Zain, and A. N. Abdalla, "Robust image watermarking theories and techniques: a review," *Journal of Applied Research and Technology*, vol. 12, no. 1, pp. 122–138, 2014.

[8] Y. Zhang, "Digital watermarking technology: a review," in *Proceedings of the ETP International Conference on Future Computer and Communication*, pp. 250–252, Wuhan, China, June 2009.

[9] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, "Digital watermarking and steganography," *The Morgan Kaufmann Series in Multimedia Information and Systems*, Morgan Kaufmann Publishers, Burlington, MA, USA, 2nd edition, 2008.

[10] A. Mohanarathinam, S. Kamalraj, G. K. D. Prasanna Venkatesan, R. V. Ravi, and C. S. Manikandababu, "Digital watermarking techniques for image security: a review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 8, pp. 3221–3229, 2019.

[11] W. Aiken, H. Kim, and S. Woo, "Neural network laundering: removing black–box backdoor watermarks from deep neural networks," *Computers & Security*, vol. 1878, no. 1, Article ID 012062, 2021.

[12] X. Liu, F. Li, B. Wen, and Q. Li, "Removing backdoor–based watermarks in neural networks with limited data," in *Proceedings of the IEEE Internationl Conference Pattern Recognition*, Article ID 10149, Milan, Italy, January 2021.

[13] Y.-S. Lee, Y.-H. Seo, and D.-W. Kim, "Blind image watermarking based on adaptive data spreading in n-level DWT subbands," *Security and Communication Networks*, vol. 2019, Article ID 8357251, 11 pages, 2019.

[14] C. Li, Z. Zhang, Y. Wang, B. Ma, and D. Huang, "Dither modulation of significant amplitude difference for wavelet based robust watermarking," *Neurocomputing*, vol. 166, pp. 404–415, 2015.

[15] J. Ouyang, G. Coatrieux, B. Chen, and H. Shu, "Color image watermarking based on quaternion Fourier transform and improved uniform log-polar mapping," *Computers & Electrical Engineering*, vol. 46, pp. 419–432, 2015.

[16] R. Mehta, V. P. Vishwakarma, and N. Rajpal, "Lagrangian support vector regression based image watermarking in wavelet domain," in *Proceedings of the 2015 2nd International Conference on SPIN*, pp. 854–859, Noida, India, February 2015.

[17] H. Hu, Y. Chang, and S. Chen, "A progressive QIM to cope with SVD-based blind image watermarking in DWT domain," in *Proceedings of the 2014 IEEE China Summit & International Conference on Signal and Information Processing*, pp. 421–425, Xi'an, China, July 2014.

[18] H. Kandi, D. Mishra, and S. R. K. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," *Computers & Security*, vol. 65, pp. 247–268, 2017.

[19] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: hiding data with deep networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 657–672, Xi'an, China, July 2018.

[20] M. Ahmadi, A. Norouzi, S. M. Soroushmehr et al., "Framework for residual diffusion watermarking on deep networks," https://arxiv.org/abs/1810.07248.

[21] S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, and H.-K. Lee, "Finding robust domain from attacks: a learning framework for blind watermarking," *Neurocomputing*, vol. 337, pp. 191–202, 2019.

[22] X. Zhong and F. Y. Shih, "A robust image watermarking system based on deep neural networks," 2019, https://arxiv.org/abs/1908.11331.

[23] B. Wen and S. R. O. M. Aydore, "A robust watermarking system using adversarial training," 2019, https://arxiv.org/abs/1910.01221.

[24] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1509–1517, Nice, France, October 2019.

[25] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, https://arxiv.org/abs/1511.08458.

[26] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Applied Sciences*, vol. 10, no. 10, p. 3443, 2020.

[27] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, MA, USA, 2015.

[28] E. M. Senan, A. Alzahrani, M. Y. Alzahrani, N. Alsharif, and T. H. H. Aldhyani, "Automated diagnosis of chest X-ray for early detection of COVID-19 disease," *Computational and Mathematical Methods in Medicine*, vol. 2021, 10 pages, 2021, https://doi.org/10.1155/2021/6919483, Article ID 6919483.

[29] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020.

[30] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[31] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 1, 2021.

[32] Z. H. Wei, P. Qin, and Y. Q. Fu, "Perceptual digital watermark of images using wavelet transform," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 4, pp. 1267–1272, 1998.

[33] S. M. Poonam and S. M. Arora, "A DWT-SVD based robust digital watermarking for digital images," *Procedia Computer Science*, vol. 132, pp. 1441–1448, 2018.

[34] A. Joseph and K. Anusudha, "Robust watermarking based on DWT SVD," *International Journal of Signal and Image Processing*, vol. 1, pp. 147–164, 2013.

[35] A. Alzahrani, "Enhanced invisibility and robustness of digital image watermarking based on DWT-SVD," *Applied Bionics and Biomechanics*, vol. 2022, Article ID 5271600, 13 pages, 2022.