Convolutional neural network-based image watermarking using discrete wavelet transform

Alireza Tavakoli1 · Zahra Honjani2 · Hedieh Sajedi2

Received: 14 September 2022 / Accepted: 21 February 2023 / Published online: 22 March 2023
© The Author(s), under exclusive licence to Bharati Vidyapeeth’s Institute of Computer Applications and Management 2023

Abstract With the growing popularity of the Internet, digital images are used and transferred more frequently. Although this phenomenon facilitates easy access to information, it also creates security concerns and violates intellectual property rights by allowing illegal use, copying, and digital content theft. Using watermarks in digital images is one of the most common ways to maintain security. Watermarking is proving and declaring ownership of an image by adding a digital watermark to the original image. Watermarks can be either text or an image placed overtly or covertly in an image and are expected to be challenging to remove. This paper proposes a combination of convolutional neural networks (CNNs) and wavelet transforms to obtain a watermarking network for embedding and extracting watermarks. The network is independent of the host image resolution, can accept all kinds of watermarks, and has only 11 layers while keeping performance. Performance is measured by two terms; the similarity between the extracted watermark and the original one and the similarity between the host image and the watermarked one.

Keywords Watermarking · Convolutional neural networks · Wavelet transform · Neural networks

1 Introduction

Protecting art ownership was always an issue since painters used to sign their works, which was not safe. Due to widespread Internet use, artistic works are becoming digital, so the threat of privacy, copyright, and illegal usage has increased. Authentication is not limited to artistic works but includes identity document (ID) cards and source tracking. In order to solve these issues, digital watermarking is an effective method of hiding information that can be used for authentication purposes.

The digital watermarking technique involves embedding information (watermark) into the content and extracting it when needed. Various attacks could be applied to the watermarked content to damage or remove the embedded watermark, which shows the importance of robust watermarking. Even though watermarking can be applied to a wide range of content, including video [1], audio [2], or images, in this paper, we will focus on images.

Until recently, digital watermarking was based on deterministic algorithms. Typical methods are based on embedding the watermark in one of the discrete cosine transform (DCT) [3], discrete wavelet transform (DWT) [4–6], discrete Fourier transform (DFT) [7], or quantization index modulation [8], or a combination of these [9, 10].

Due to the inflexibility of deterministic algorithms, small changes in watermarked images lead to inaccurate extracted watermarks. The inflexibility leads to using neural networks as a robust technique for digital watermarking [11, 12]. Even though these methods solve the robustness problem, some non-deterministic networks still perform poorly, and others have complex structures.

This paper investigates the combination of deterministic and neural network methods to improve accuracy and performance. As the results suggest, we have improvements in
some attacks. In addition, the proposed network enhances the quality of watermarked images by using DWT as a preprocessing technique.

The rest of the paper is organized as follows: In Section 2, we discuss relevant previous studies. Then in Section 3, we outline the basic concept, wavelet transform, used for the proposed algorithm. The proposed network structure is explained in Section 4. Section 5 discusses the training technique, the experimental results, and the comparison with literature [12], and [13]. Moreover, this paper is concluded in Section 6.

2 Related works

Various image watermarking schemes have previously been done to provide robustness, authenticity, and imperceptibility. Earlier, we mentioned some papers proposing watermarking algorithms which use DWT, DFT, and neural networks. Here we will explore some of them in more detail.

In paper [4], original images and watermarks are split into RGB channels to be used as inputs in the other embedding process. In the original image, the blue channel has been subjected to DWT and Singular Value Decomposition (SVD). A first Arnold transformation is applied to the blue channel of the watermark image, which acts as a key. In addition to Arnold’s transformation, SVD is applied to the watermark image, and then the image is embedded with the watermark. In addition to being semi-blind, this method is also robust against attacks due to the use of the key during the watermark extraction.

As another deterministic approach, Xiangui Kang proposed a blind DWT-DFT composite image watermarking algorithm that is robust against both affine transformation and Joint Photographic Experts Group (JPEG) compression [10]. This paper used watermark structure, two-dimensional interleaving, and synchronization techniques to improve the robustness.

In recent years, residual blocks, fully connected layers, DCT layers, and CNN have been used as watermarking networks. An end-to-end autoencoder in ReDMark [12] simulates a DCT layer in its deep network. The concatenation of the input image and watermark is used as input to the DCT layer, and after transformation, the embedding is performed. The proposed structure is robust against JPEG attacks since the DCT layer is used.

Jae-Eun Lee proposed a robust watermarking neural network that included an attack simulation and was adjusted to the resolution of the host image and the watermark [13]. It is composed of convolutional layers with average pooling layers. This method also includes a strength factor to adjust the tradeoff between invisibility and robustness. This method uses the images’ luma component (Y) to hide the watermark. A watermark is converted into a layer the same size as the original image. The watermark is then merged with the processed image in the leading network. Despite the impressive results, the proposed network is relatively complex. In contrast to mentioned articles, this paper presented a non-deterministic method with higher robustness than discussed papers.

3 Discrete wavelet transform

Each image has smooth regions interrupted by edges. Edges provide much information about the image. In the Fourier transform, a signal is decomposed into sin waves that are not localized in time. Therefore, Fourier transforms cannot efficiently represent sudden changes (particularly edges), and wavelet transforms are preferred. A wavelet transform is a way of representing a function by a specific orthonormal series. A wavelet is a wave-like oscillation that is localized in time. Different types of wavelet transform exist based on the mother function $\psi(t)$ and the scaling function $\varphi(t)$. One of the most widely-used wavelets is Haar whose $\psi(t)$ and $\varphi(t)$ are shown in Figure 1.

In addition, the discrete wavelet transform decomposes an image into four wavelet subbands, LL, LH, HL, and HH, which were examined independently to compare their capacity for hiding information. The results are demonstrated in Section 5 [14].

4 Proposed method

The proposed network consists of 5 main parts; watermark preprocess network, image preprocess network, embedding network, attack layer, and extraction network. Throughout this paper, we will discuss each part in detail, but Figure 2 shows the overall structure of the network. In addition to these parts, we also use the wavelet transform for embedding and extraction. Different shapes of images can be accepted through this structure because it is based on CNN layers. The codes of the proposed method are available on GitHub1. Since DWT must be implemented in the model, we have used [15] to implement DWT layers that can run on the same hardware as the rest of the machine learning pipeline without impacting performance.

1 https://github.com/alirezatwk/Convolutional-Neural-Network-Based-Image-Watermarking-using-Discrete-Wavelet-Transform.
4.1 Host image preprocessing network

We select the first channel of the original image and scale it into the [0, 1] interval. Next, it is decomposed by the Haar wavelet. Then the LL layer is used as the input of the host image preprocessing network. The LL layer is fed into a convolutional layer with 3x3 filters whose strides equal one. According to Figure 3, the layer consists of 64 filters with the same padding so that the output shape is the same as the host image.

4.2 Watermark preprocessing network

As shown in Figure 4, the first step is to reshape array-like watermarks to 2D watermarks so that applying convolutional layers will be possible. The preprocessing network includes three network blocks. The blocks comprise convolutional transpose, batch normalization, ReLU activation, and average pooling. Convolutional transposes have 512, 128, and 1 filters, respectively. The output’s data type
and shape are the same as the host image, so we can concatenate them in the following networks.

### 4.3 Embedding network

First, the preprocessed image is concatenated with the preprocessed watermark. The concatenation is then used as input for the four-block embedding network, as shown in Figure 5. The first three blocks include a convolutional layer with 64 filters, a batch normalization, and a ReLU activation function. The last block consists of a convolutional layer with one filter and a Tanh activation function to achieve an appropriate output size.

---

**Fig. 3** Host image preprocessing network structure

**Fig. 4** Watermark preprocessing network structure

**Fig. 5** Embedding network structure
As mentioned earlier, in the host image preprocessing network, the LL channel was passed to the embedding network. Now we change the main LL with the watermarked one and apply a wavelet inverse transform which results in the watermarked image.

### 4.4 Attack simulator

Watermarked images might suffer from malicious attacks. In order to increase its robustness, an attack simulator was added to the network. This simulator executes different types of attacks with the ratio in Table 1 on each mini-batch.

| Attack          | Info | Ratio |
|-----------------|------|-------|
| No attack       | –    | 1/3   |
| Salt and Pepper | $p = 0.1$ | 1/6   |
| Gaussian noise  | $\sigma = 0.15$ | 1/6   |
| JPEG            | quality= 50  | 1/6   |
| Dropout         | $p = 0.3$ | 1/6   |

Figure 6 shows how explained attacks affect the original image.

### 4.5 Extraction network

The watermark is embedded in the LL channel, so the first step is to extract the LL channel. The extracted channel is then used as input for the three-block extracting network, as shown in Figure 7. The first two blocks include two convolutional layers with 128 and 256 filters, a batch normalization, and a ReLU activation function. The last block consists of a convolutional layer with one filter and a Sigmoid activation function. All convolutional layers in this section will have a stride of 2 to adjust the output shape to the watermark shape.

### 4.6 Loss function

The two-term loss function is used for training the network. The similarity between a watermarked image and the host
image is measured by the mean square error (MSE) loss function shown in Equation 1.

$$L_1 = \frac{1}{MN} \sum_{i,j} [I_H(i,j) - I_{W_I}(i,j)]^2$$

(1)

where $M$ and $N$ are the resolution of the host image, $I_H$ is the host image, and $I_{W_I}$ is the watermarked one.

Also, the mean absolute error (MAE) function is used as the loss function between the extracted watermark and input watermark displayed in Equation 2.

$$L_2 = \frac{1}{X} \sum_{i} |WM_O(i) - WM_E(i)|$$

(2)

where $X$ is the size of the watermark, $WM_O$ is the original watermark, and $WM_E$ is the extracted one.

As shown in Equation 3, a combination of these functions determines the loss function of the whole network.

$$L_3 = \lambda_1 L_1 + \lambda_2 L_2$$

(3)

coefficients ($\lambda_1$ and $\lambda_2$) are hyper-parameters that adjust invisibility and robustness.

5 Experiments

Previously, we described the network in detail. This section explains training materials, such as datasets, training facilities, and evaluation metrics.

5.1 Dataset

COCO dataset\(^2\) was used containing 328,000 color images in different sizes [16]. Randomly, 40000 images were chosen as training data and 1000 as testing dataset. All images in the dataset were resized to (256, 256, 3) to prepare for feeding into the network. For the watermark, a random generator was used to generate 256-bit watermarks for each instance. By generating random watermarks for each instance, the network became more generalized.

5.2 Training

The proposed watermarking network was trained in a Laptop with an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz, 32 GB RAM, and the NVIDIA GeForce GTX 1650 GPU. Every mini-batch consists of 10 images and 10 new watermarks. There were 60 epochs. 33.0 for image loss weight ($\lambda_1$) and 0.2 for watermark loss weight ($\lambda_2$). The learning rate was 0.001.

5.3 Metrics

Two metrics were used to estimate robustness and watermark invisibility. Most previous works used peak-signal-to-noise ratio (PSNR) to evaluate the invisibility of the watermarked image. PSNR measures the similarity between the watermarked image and the host image by Equation 4.

$$\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

(4)

Here, $\text{MAX}_I$ is the maximum possible pixel value of the image. Also, $\text{MSE}$ is the mean square difference between the host image and the watermarked one.

Due to the importance of the extracted watermark, another metric is required to assess the robustness of the network. In this study, bit error ratio (BER) was used, which is shown in Eq. 5.

$$\text{BER} = \frac{100}{X} \sum_{i=1}^{X} D(WM_O(i), WM_E(i))$$

(5)

$$D(x,y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}$$

where $X$ is the size of the watermark, $WM_O$ is the original watermark and $WM_E$ is the extracted one.

5.4 Results

In this section, the algorithm described above is examined from various perspectives. First, the effect of the attack simulator is described. Then there is a comparison between the accuracy of the wavelet channels. In the end, it will be compared to state-of-art methods.

---

\(^2\) [http://images.cocodataset.org/zip/train2017.zip](http://images.cocodataset.org/zip/train2017.zip)
5.4.1 Attack simulator analysis

In order to evaluate the functionality of the attack simulator, the model was trained without it, and Table 2 shows a significant difference which suggests the critical role of the attack simulator in network robustness.

5.4.2 Wavelet channels analysis

DWT bands have different metrics and efficiency values, as suggested by Shaila R Hallur in [17]. The purpose of this section is to analyze the results of each band in order to choose the most proficient model. These experiments are based on the same parameters setting with different DWT bands. According to Figure 8, the HH band is preferred for the PSNR criterion.

Nevertheless, in the BER comparison, LL has superior performance, as shown in Figure 9. Further investigations are conducted on the HH band in the remainder of the paper.

5.4.3 Comparison with state-of-the-arts methods

As mentioned, for evaluating the invisibility of the watermarked images, the PSNR of the watermarked images and the host images on the test dataset are calculated, and the average of these numbers results in 40.1 dB. As shown in Figure 10, the difference between the first and second columns is quite negligible.

The measured BER on output watermarks without any attack is 0.0003. However, applying the attacks, this number increased to 9.58.

Table 3 shows the network’s robustness under each attack. These performances are compared with state-of-art methods. Regarding Table 3, using wavelet transform in the structure keeps PSNR, but there is an improvement in the BER metric. Additionally, the proposed network has performed better under Gaussian noise and Salt and Pepper attacks.

6 Conclusion and future work

In this paper, we propose a watermarking neural network that can work well for attacked images and has fewer convolutional layers than previous networks due to wavelet transform. Compared with [13], the overall BER has decreased. The PSNR metric of the proposed model is slightly higher than this one; in addition, the model is smaller in size and works better in attacked images.

According to Arpit Bansal in [18], considering some special attacks in the network leads to poor performance under new attacks. We claimed deterministic methods are inflexible, and thus we used a neural network. However, we are still using deterministic methods in the attack simulator. As a result, the attack simulator can be updated to a more general form in future efforts. It is also possible to investigate famous neural network architectures in order to improve results.

Table 2 Effect of the attack simulator on robustness

| Attack                  | Attack-simulator free model | Model with attack simulator |
|-------------------------|-----------------------------|-----------------------------|
| PSNR                    | 50.9                        | 40.1                        |
| No attack (BER)         | 0.014                       | 0.0003                      |
| Salt & pepper (BER)     | 46.44                       | 1.4231                      |
| Gaussian noise (BER)    | 43.55                       | 7.5382                      |

Fig. 8 PSNR comparison of wavelet layers

Fig. 9 BER comparison of wavelet layers
References

1. Patel R, Lad K, Patel M, Desai M (2021) An efficient dct-sbpm based video steganography in compressed domain. Int J Inf Technol 13(3):1073–1078
2. Joshi Saurav, Pande Jeetendra, Singh BK (2018) Watermarking of audio signals using iris data for protecting intellectual property rights of multiple owners. Int J Inf Technol 10(4):559–566
3. Xu ZJ, Wang ZZ, Lu Q (2011) Research on image watermarking algorithm based on DCT. Procedia Environ Sci, 10:1129–1135. 2011 3rd International Conference on Environmental Science and Information Application Technology ESIAT
4. George J, Varma S, Chatterjee M (2014) Color image watermarking using DWT-SVD and Arnold transform. In 2014 Annual IEEE India Conference (INDICON), pages 1–6
5. Pradhan C (2020) Robust and blind watermarking using Arnold 4D cat map in discrete wavelet. Int J Inf Technol 12(2):593–597
6. Giri KJ, Bashir R (2018) A block based watermarking approach for color images using discrete wavelet transformation. Int J Inf Technol 10(2):139–146
7. Li C, Zhang Z, Wang Y, Ma B, Huang D (2015) Dither modulation of significant amplitude difference for wavelet based robust watermarking. Neurocomputing 166(C):404–415
8. Hwai HT, Chang YJ, Chen SH (2014) A progressive QIM to cope with SVD-based blind image watermarking in DWT domain. In 2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP), pages 421–425
9. Nejati F, Sajedi H, Mohammadi M (2019) Fragile watermarking for image authentication using QR factorization and fourier transform. In 2019 5th International Conference on Web Research (ICWR), pages 45–49
10. Kang X, Huang J, Shi YQ, Lin Y (2003) A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. IEEE Trans Circuits Syst Video Technol 13(8):776–786

Table 3 Comparing proposed and state-of-art methods

| Attack               | ReDMark  | [13]          | Proposed |
|----------------------|----------|---------------|----------|
| PSNR                 | 40.24    | 40.58         | 40.1     |
| No attack (BER)      | 0.7015   | 0.0003        |          |
| Salt and pepper (BER)| 9.1      | 3.1888        | 1.4231   |
| Gaussian noise (BER) | 14.5     | 27            | 7.5382   |
| Drop out (BER)       | 8        | 4.7194        | 11.7051  |
| JPEG (BER)           | 25.4     | 0.6696        | 27.2482  |

Fig. 10 Examples of input images, output images, and their differences
11. Kandi H, Mishra D, Sai Gorthi SRK (2017) Exploring the learning capabilities of convolutional neural networks for robust image watermarking. Comput Secur 65:247–268
12. Ahmadi M, Norouzi A, Karimi N, Samavi S, Emami A (2020) ReDMark Framework for residual diffusion watermarking based on deep networks. Expert Syst Appl 146:113157
13. Lee JE, Seo YH, Kim DW (2020) Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. Appl Sci 10(19):6854
14. Singh K, Ranade SK, Singh C (2017) Comparative performance analysis of various wavelet and nonlocal means based approaches for image denoising. Optik 131:423–437
15. Versaci F (2021) Wavetr: A fast 2D wavelet transform for machine learning in keras. ICPR International Workshops and Challenges. Pattern Recognition. Springer International Publishing, Cham, pp 605–618
16. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Lawrence Zitnick C (2014) Microsoft COCO: common objects in context. In: Fleet David, Pajdla Tomas, Schiele Bernt, Tuytelaars Tinne (eds) Computer Vision - ECCV 2014. Springer International Publishing, Cham, pp 740–755
17. Hallur SR, Kuri S, Sudi GS, Kulkarni GH (2015) A robust digital watermarking for gray scale image. Int J Technol Res Eng 2(10):2440–2443
18. Bansal A, Chiang PY, Curry MJ, Jain R, Wigington C, Manjunatha V, Dickerson JP, Goldstein T (2022) Certified neural network watermarks with randomized smoothing. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 1450–1465. PMLR

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.