Hindawi
Security and Communication Networks
Volume 2021, Article ID 9119478, 11 pages
https://doi.org/10.1155/2021/9119478

Research Article

Destroy the Robust Commercial Watermark via Deep Convolutional Encoder-Decoder Network

Wei Jia,[1] Zhiying Zhu ,[2] and Huaqi Wang[3]

[1]*School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China*
[2]*School of Computer Science, Fudan University, Shanghai 200433, China*
[3]*School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Zhiying Zhu; zyzhu19@fudan.edu.cn

Received 6 September 2021; Accepted 10 November 2021; Published 6 December 2021

Academic Editor: Weiwei Liu

Nowadays, robust watermark is widely used to protect the copyright of multimedia. Robustness is the most important ability for watermark in application. Since the watermark attacking algorithm is a good way to promote the development of robust watermark, we proposed a new method focused on destroying the commercial watermark. At first, decorrelation and desynchronization are used as the preprocessing method. Considering that the train set of thousands of watermarked images is hard to get, we further use the Bernoulli sampling and dropout in network to achieve the training instance extension. The experiments show that the proposed network can effectively remove the commercial watermark. Meanwhile, the processed image can result in good quality that is almost as good as the original image.

1. Introduction

With the development of Internet technology and smartphones, the copyright protection of digital images has become increasingly important. Digital watermarking technology is an important branch of information hiding, and it provides a solution for the copyright protection of multimedia products. Watermark can be classified into visible watermarks and invisible watermarks in terms of visibility. A common commercial watermark is visualized. The classic style is a logo with a degree of transparency. The emphasis is on copyright declaration, but it is not safe. Infringement can be reached directly by intercepting or erasing by image processing software. Invisible watermark can be subdivided into robust watermark, semi-fragile watermark, and fragile watermark according to the difference in robustness. Images containing fragile watermark can be easily located after being tampered with, while semi-fragile watermark is robust to certain attacks and only vulnerable to certain specific attacks. Robust watermark is the most widely studied and used watermark. As it is most resistant to attacks, the robust watermark can be extracted after many kinds of attacks, so it is often used in OSN for copyright protection Voloshynovskiy et al. [1]. The QIM Chen and Wornell [2] algorithm quantifies the original cover into several different index intervals by different quantifiers, which is also the embedding process. The watermarking will be extracted according to the quantitative index interval of modulated data. The receiver can detect hidden data by the shortest distance method when the channel interference is not serious. The spread spectrum code with pseudorandom and cross-correlation properties plays a key role in SS Dixon [3] algorithm, and the energy distribution of embedded watermarking signal is extended to a wider spectrum, which improves the security and robustness capability. ULPM Kang et al. [4] eliminates the interpolation distortion and expands the embedding space. A discrete log-polar point can be obtained by performing the ULPM to the frequency index in the Cartesian system, and the data of which are then embedded to the corresponding DFT coefficient in the Cartesian system. Figure 1 shows the general steps of using the robust watermark in OSN. After transmission through the lossy channel, the robust watermark can still be extracted correctly to protect the copyright.

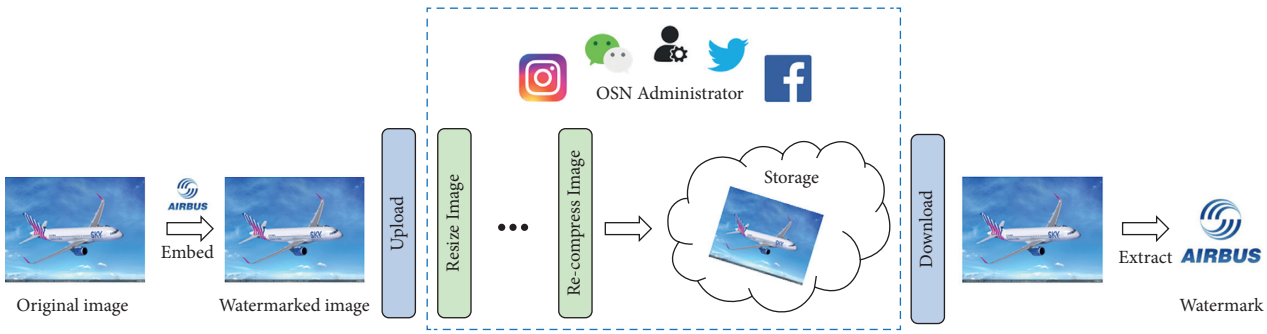


Figure 1: Robust watermark used in OSN. "···" used in the image represents the unknown operations.

The original robust watermarking technology was also developed from least significant bit (LSB). In addition, the method based on image pixel histogram is also a representative research in the early stage Coltuc and Bolon [5]. Same as steganography, the embedding domain of image watermark is also divided into spatial domain and transform domain. Compared with embedded in spatial domain, watermark embedded in transform domain usually has less impact on image vision with the same robustness. In the transform domain, the discrete cosine transform (DCT) attracts the most research attention, which is mainly due to the widespread use of JPEG format images Parah et al. [6]. In addition to DCT, the commonly used transform domains include Fourier–Mellin transform (FMT), singular value decomposition (SVD), and discrete wavelet transform (DWT) Li et al. [7]. In practical applications, watermark needs to consider the application scenarios using different transform domains or composite combinations. In addition to the research in laboratories, the commercial watermark Digimarc is also very typical. This application is integrated with software Adobe Photoshop in the form of a plug-in.

Unlike steganography, which hardly considers active attacks, watermark faces a variety of attacks. Images transmitted in OSN have their compression operation. Image cropping, video capture tools, and the addition of mosaics and textures are also very common. In terms of types, anti-watermarking can be roughly divided into removal attacks Su and Girod [8]; geometric attacks D'angelo et al. [9] D'Angelo [10]; cryptographic attacks Cox and Linnartz [11]; and protocol attacks Craver et al. [12]; Kutter et al. [13]. Cryptographic attacks and protocol attacks are mainly aimed at watermarks that use cryptographic theories, and the hidden vulnerabilities in the transmission of watermarking images. Due to limited applications, they have been rarely studied nowadays. However, removal attacks and geometric attacks can still be seen in cutting-edge watermarking research. The most common one is JPEG compression. After the lossy compression process, the watermarked image loses its information, and the amount of loss is related to the compression strength. The geometric attack is to geometrically warp the image to change the position of the original pixel coefficients and break the rules of the watermarking algorithm.

In the field of information security, we often research the two sides of the problem. The research of new watermarking attacks is not only to put forward better standards for measuring the robustness of watermarks but also to prevent watermarking algorithms from being applied to illegal transmission. We need better watermarking attack technology, which can destroy the watermark more effectively than the traditional methods while ensuring that the quality of the processed image will not be affected too much. In today's popularization of deep learning research, we should consider using new related technologies to update methods.

In this study, we proposed a new watermarking attack framework focused on destroying the commercial watermark. The advantages of our proposed method lie in the following twofold:

(i) We proposed a preprocessing method, which includes two parts: decorrelation and desynchronization.

(ii) The results of experiments show the excellent attack ability of our network, and compared with traditional attacks, the processed image maintains good image quality.

The rest of this study is organized as follows. Section 2 reviews the related work with our method. The proposed scheme is specified in detail in Section 3. Section 4 provides the experimental results, and Section 5 concludes the study.

2. Related Work

2.1. Attack Methods on Digital Watermark. The watermarking attack technology has been developed for many years. We mainly introduce two traditional watermark attacking methods: removal attacks and geometric attacks. It is not necessary to know the principle of watermarking algorithm to remove attacks and geometric attacks. Destroying the secret carrier is essentially destroying the watermarking information probabilistically. The increasingly developed watermarking technology has also developed adequate countermeasures.

The removal attack aims to completely remove the watermark from the protection carrier, and it is the most used attack method with the most categories. The removal attack can be divided into denoising attack Shukla et al. [14], remodulation attack, and lossy compression attack Wallace [15], mainly using filtering, coding, and other technologies Langelaar et al. [16]. The basic idea of the denoising attack is

to assume that the watermark is a layer of additive noise in the carrier, which theoretically defines the expected goal of removing the watermark. Considering the speed requirements of real-time attack applications, Geng, Zhang, Chen, Fang and Yu [17] used a denoising network based on DnCNN Zhang et al. [18] architecture to remove watermark. The inputs of the network are the watermarked image preprocessed, and the corresponding original images are as labels. In this network, the residual features extracted by CNN are considered as the watermark, and the time cost of estimate is short enough to meet most real-time requirements and destroys the correlation between the watermarked image and the real watermark, which causes the burden of the watermarking decoder. Lossy compression attacks use the JPEG compression method to compress the three YUV components of the image. Different from the removal attack to eliminate the embedded watermark, the geometric attack method is used usually to spatially warp the carrier image to change the original pixel position. The purpose is to make the watermarking extraction algorithm and the embedded information lose synchronization, to complete the destruction of the watermark.

The other methods of attack include cryptographic attacks and average joint attacks. The attack methods of cryptographic attacks are very similar to the early methods of decrypting passwords, and the calculation is very complicated. A cryptographic attack method can only target the watermark of a specific method. The most representative one is the oracle attack method. Average attacks and joint attacks are mostly used on video watermark Deguillaume et al. [19], Pereira and Pun [20]. Video is composed of continuous images, and there is a strong correlation between frames, especially in static scenes. Both the average and joint attack operations require the use of a large number of video frames as the datasets.

According to Zhu, Kaplan, Johnson, and Fei-Fei [21], a triple convolutional network of encoder, decoder, and discriminator is used. The carrier information and embedded information are input into the encoder to obtain the coded image. The decoder is responsible for reconstructing information from the coded image, and the discriminator guarantees the quality of the watermarked image. The coded image is attacked, so that the decoder can extract watermarking information from the attacked image and realize the robustness of the watermark. This method is stable and excellent in the face of various traditional attack methods. We can also realize that the research on anti-watermarking is far from enough in this study. We should consider more comprehensive and complete attack methods to promote the development of watermark.

2.2. Digimarc Watermark. Digimarc is a commercial watermark, usually used as a plug-in integrated into the application software, the most common of which is Adobe Photoshop. Like the watermarks mentioned in the introduction, the Digimarc watermark is visually invisible with strong robustness. When embedding a watermark, the available options include image information, image attributes, and watermarking durability. The first two are differences in information content, and watermarking durability is the most critical option. The software divides the durability into four levels. The levels are denoted by a in the experimental part below. As the number increases, the embedded watermark is more robust, and the corresponding impact on the vision of cover is also increased. When extracting information, the software will display two results, one is the content of the embedded information and the other is the strength of the extracted watermark. The strength of the watermark cannot be quantified, but it can be roughly divided into six levels, namely very strong, strong, medium, weak, very weak, and none. If it is only the weakening of the intensity, the expected purpose cannot be achieved. Only when the result is displayed as none, the watermark is considered to be destroyed.

3. Proposed Framework

Different from the watermarking algorithms studied in laboratory, as a commercial watermark, the principle of Digimarc is unknown. Therefore, we cannot design and optimize the network in a targeted manner according to the selection of their embedding domain and the process of embedding and extracting information, such as the use of high-pass filters. Used as a plug-in, the Digimarc watermark is a complete black box, and it is difficult to have prior knowledge that can be relied on.

Figure 2 shows the framework of Digimarc watermark attacking model. To ensure the success of the attack, the input watermarked image is preprocessed before the training starts. The preprocessing includes two operations of decorrelation and synchronization. In addition, different from the datasets usually used in network training, it is hard to get thousands of images with the same type of commercial watermark. Therefore, we use a deep convolutional encoder-decoder network based on single-image training. The network is based on the idea of removing attacks, regarding the Digimarc watermark as additive noise on the cover, and allows the original image to be used as a learning target. The main body is made up of an encoder-decoder network. To fully avoid the overfitting of small data training, some neurons are dropped out during the training and testing phases, and the Bernoulli rules are followed. In the early stage of training, the train set is expanded by the Bernoulli sampling to further improve the generalization ability.

3.1. The Preprocessing of Watermarked Images. Before training the encoder-decoder network, we first proposed a preprocessing method. The preprocessing of the input image mainly includes two parts: decorrelation and desynchronization. The flow chart is shown in Figure 3. Assuming that the length and width of the watermarked image I_W are H and W, respectively, we applied a Wiener filter with $\varphi \times \varphi$ size filter kernel in decorrelation. Starting from minimizing the mean square error, the purpose is to reduce the correlation between the watermark signal and the original carrier. Assuming that the original carrier image is I_C and

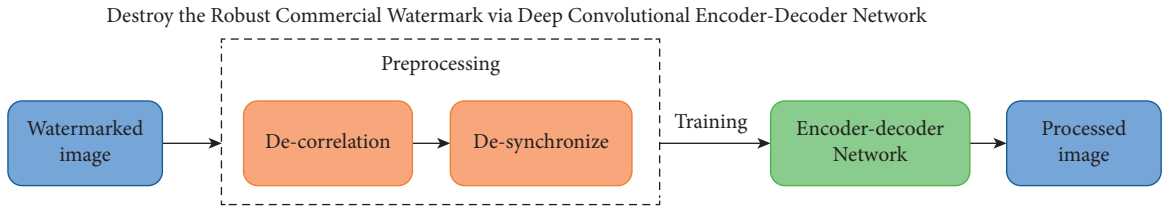


Figure 2: The framework of Digimarc watermark attacking model.

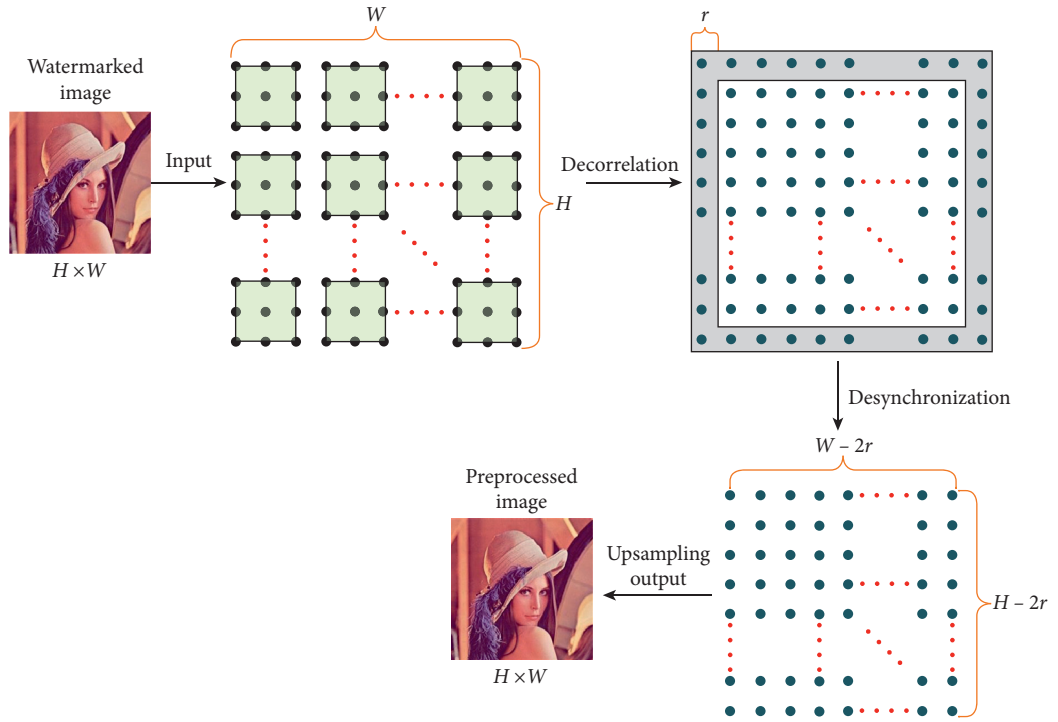


Figure 3: Preprocessing operations on watermarked images.

the reconstructed image after decorrelation is I_M, the mean square error between the two images can be expressed as follows:

$$\begin{aligned} \mathrm{MSE} &= E\left[I_C(n) - I_M(n)\right]^2 \\ &= E\left[I_C^2(n)\right] - 2E\left[I_C(n)I_M(n)\right] + E\left[I_M^2(n)\right], \end{aligned} \tag{1}$$

where $E[\cdot]$ represents mathematical expectation, $n = 1, 2, \ldots, H \times W$. The right part of equation (1) can be further derived as follows:

$$\begin{aligned} E\left[I_C^2(n)\right] &= R_{I_C}(0), \\ E\left[I_C(n)I_M(n)\right] &= E\left[I_C(n)\sum_{k=-\infty}^{\infty} h(k)I_M(n-k)\right] \\ &= \sum_{k=-\infty}^{\infty} h(k)R_{I_C I_W}(k), \\ E\left[I_M^2(n)\right] &= \sum_{k=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(k)h(m)R_{I_W}(m-k). \end{aligned} \tag{2}$$

Among them, $R(\cdot)$ represents the correlation function, and h is the filter element. If h_0 is the optimal filter that meets the minimum mean square error, then h can be expressed as follows:

$$h(n) = h_0(n) + g(n). \tag{3}$$

Among them, g is the error. Substituting h_0 into equation (1), we can get the minimum mean square error 0. Combining the above equation, we can get the following:

$$\begin{aligned} \mathrm{MSE} &= \mathrm{MSE}_0 + f_1 + f_2, \\ f_1 &= 2\sum_{m=-\infty}^{\infty} g(m)\left[\sum_{k=-\infty}^{\infty} h_0(k)R_{I_W}(m-k) - R_{I_C I_W}(k)\right], \\ f_2 &= \sum_{m=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} g(m)g(k)R_{I_W}(m-k). \end{aligned} \tag{4}$$

Since MSE must be greater than or equal to MSE_0, it is easy to know that f_2 must be greater than zero, so f_1 needs to be equal to zero.

$$R_{I_C I_W}(m) = \sum_{k=-\infty}^{\infty} h_0(k) R_{I_W}(m-k). \tag{5}$$

Converting equation (5) into the form of power spectrum, the filter expression at the energy level can be obtained as follows:

$$H_0(s) = \frac{\Phi I_C I_W(s)}{\Phi I_W(s)}. \tag{6}$$

Equation (6) is the key to reducing the correlation between the watermark and the original carrier. To explain from the perspective of image pixels, the local neighborhood of the pixel can be used to estimate the mean and standard deviation of the filter h under the filter kernel size φ.

$$\begin{aligned} \mu &= \frac{1}{\varphi^2} \sum_{i,j \in \eta} b_{i,j}, \\ \sigma &= \sqrt{\frac{1}{\varphi^2} \sum_{i,j \in \eta} b_{i,j}^2 - \mu^2}. \end{aligned} \tag{7}$$

Among them, $b_{i,j}$ is the (i, j) pixel in I_W, so the decorrelated image I_M can be expressed as follows:

$$I_M = I_W \otimes H. \tag{8}$$

Among them, $\otimes$ represents the convolution operation, and H is the convolution kernel of the filter. The filter core of $\varphi = 3$ is used in this scheme. The neighborhood of pixel 8 in the natural image has the strongest correlation. Choosing this size is enough to complete the initial weakening of the correlation between the watermark and the carrier image. In Figure 3, the filter kernel is represented as a green square. After convolving the black pixels of I_W, the blue pixels in I_M are obtained.

Further, the image I_M is desynchronized, the pixel collection matrix in I_M by M is indicated, and the pixel matrix of the image I_T after desynchronization can be expressed as follows:

$$T = M\{r\colon W - r, r\colon H - r\}. \tag{9}$$

We set $r = 1$ here, and finally, I_T will be upsampled to the original image size to complete the preprocessing of the watermarked image.

3.2. Data Augmentation. Many image processing methods, such as denoising, restoration, and super-resolution, often have a similar goal, which is to minimize the difference between the image generated and the original image. The goal can usually be expressed by the following equation:

$$\tilde{y} = \min_y E(y; x) + R(y), \tag{10}$$

where y represents the generated image and x represents the original image. $E(y; x)$ is a optimization goal, which will be changed according to different requirements. The most common optimization goal is $\|y - x\|^2$. $R(y)$ is a regularizer, which is generally obtained as a priori information based on a large amount of dataset. The choice of regularizer, which usually captures a generic prior on natural images, is more difficult and is the subject of much research. In this work, we replace the regularizer $R(y)$ with the implicit prior captured by the neural network as follows:

$$\begin{aligned} \theta^* &= \arg\min_\theta E\left(f_\theta(z); x\right), \\ \tilde{y} &= f_{\theta^*}(z). \end{aligned} \tag{11}$$

The minimizer θ^* is obtained using an optimizer such as gradient descent starting from a random initialization of the parameters.

In DIP Ulyanov et al. [22], the author believes that the same training results can be obtained by relying on the parameters of the network itself through only one image used for repeated iterations. DIP proposed a hand-designed priori function. In some cases, the performance is comparable to that of networks based on large datasets. The most urgent problem for single-image training is overfitting. If the network is regarded as the Bayesian estimation, then the prediction accuracy can be represented by the mean square error (MSE). The variance in the MSE will increase sharply because the training samples are particularly small, which will make the model lose its generalization ability. Therefore, the focus of network design is to reduce the variance to solve the problem of overfitting.

While learning only by a single image, to make full use of the information of the train sets, we first extended with the single image. We use the Bernoulli sampling to generate a large number of samples. These samples essentially contain image content, but are different from the original image, and have a good optimization effect on training tasks. The Bernoulli distribution can be explained simply by the coin tossing problem, and it describes a binary problem. For example, for a variable x, there are two possible values of 0 and 1, and the probability of $x = 1$ can be represented in formula (12). When expressed as the Bernoulli distribution, it can be written as formula (13).

$$P(x = 1 \mid \rho) = \rho(0 \le \rho \le 1), \tag{12}$$

$$P(x \mid \rho) = (1 - \rho)^{1-x} \rho^x. \tag{13}$$

For a sample set, although the samples are independent of each other, but conform to the same distribution, the likelihood function can be expressed by the following formula:

$$\begin{aligned} P(S \mid \rho) &= \prod_{m=1}^{M} P\left(x_m \mid \rho\right) \\ &= \prod_{m=1}^{M} (1 - \rho)^{1 - x_m} \rho^{x_m}, \end{aligned} \tag{14}$$

where S represents the sample set, $S = \{x_1, x_2, \ldots, x_m\}$.

In the field of image computing, it can be represented by a simple mathematical model. In this study, we suppose that the original image and the watermarked image are I_C and I_W, respectively, end expanded into corresponding sample pairs with a total of M copies. The sample pairs can be represented as $\left\{\tilde{I}_c^m\right\}_{m=1}^M$ and $\left\{\tilde{I}_W^m\right\}_{m=1}^M$.

$$\left\{\widetilde{I}_{\text{label}}^{m}\right\}_{m=1}^{M} = B_n \odot I_{\text{label}\,m=1}^{M}, \tag{15}$$

where label represents C or W. In practical applications, P_m is the Bernoulli distribution with the same size as the tensor and the probability conforms to ρ. Denote the element as K, and (15) can be redefined as follows:

$$\widetilde{I}_{\text{label}}^{m}[k] = \begin{cases} I_{\text{label}}[k], & \rho, \\ 0, & 1-\rho. \end{cases} \tag{16}$$

Let $g_\theta(\cdot)$ be the objective mapping function, and then, the loss function can be expressed as follows:

$$\min_\theta \sum_{m=1}^{M} \left\| g_\theta\left(\widetilde{I}_W^m\right) - \widetilde{I}_C^m \right\|_{B_n}^2. \tag{17}$$

In this subsection, we use the Bernoulli sampling to increase the training samples and initially solve the overfitting problem. The loss of each pair of samples is only performed on the mask represented by the current Bernoulli distribution. Because the mask tensor is completely selected at random, when we obtain a sufficient number, we can use the sum of the loss of all samples to measure the perceptual loss on the overall image.

3.3. Network Architecture. Inspired by Quan, Chen, Pang, and Ji [23], we propose to use the network depicted in Figure 4. This model has been proved useful in denoising task. Different from the existing denoising methods, this network can only use the input noisy image itself for training, which meets our need perfectly. It is mainly composed of two parts: encoder and decoder. Assuming that the original image I_C is embedded in the Digimarc watermark by software, the watermarked image I_W is obtained. The size of each image is $H \times W \times C$, H and W represent the length and width of the image, and C is the number of image channels. To prevent overfitting, before I_W is sent to the network for training, it is first sampled by the Bernoulli sampling to get the sample set. Then, after a partial convolution (PConv, partial convolutional), and activated by LeakyReLU, 3×3 PConv is used for pixel normalization.

The whole encoder is composed of six coding units. The first five coding units have a similar structure, and they are all composed of a local convolution layer activated by LeakyReLU and a maximum pooling layer with a sampling kernel of 2×2 size with the step of 2. The feature number of each layer is set to 64, and finally, the result is output by the last coding unit, and then, it is upsampled to the decoder.

The decoder contains a total of five decoding units, the first four are collectively referred to as decoding unit A and the last one is decoding unit B. For each decoding unit, the encoding feature map is connected first, and after that, two 3×3 convolution layers are applied. The convolutional layers are also activated by LeakyReLU. To further solve the overfitting problem, "dropout" is applied to delete some neurons and then continues to enter the next decoding unit through an upsampling. The numbers of convolution kernels of the decoding unit A are all 128, and the number of channels of the convolutional layer included in the decoding unit B is 96, 64, 32, and C, and finally, the output image is generated.

The convergence goal of the network is given in formula (17) in the previous section. As can be seen from the introduction in this section, the single-image training network must first face the problem of its own overfitting and then the performance index. Although it is difficult to achieve the performance of a network based on large datasets by learning from a single image under the full conditions, it can play a very good effect when the data source is lacking.

4. Experimental Results and Analysis

4.1. The Method of Test. Assuming that $g_\theta(\cdot)$ is the mapping function model obtained by training, if the test image is processed by this model, it is equivalent to a mapping of the sampled instance from test set. Generally, a network with "dropout" will scale the weight of the model through related rules during testing. Therefore, to better optimize the performance, in the actual testing phase, we use the Bernoulli sampling and "dropout" again on the model to generate a set of submodels and then average the final results. We define the submodel set of $g_\theta(\cdot)$ after K Bernoulli sampling as $\{g_\theta^k(\cdot)\}_{k=1}^{k}$ and the generated image I_G can be expressed as follows:

$$I_G = \frac{1}{K} \sum_{k=1}^{K} g_\theta^k(B_{N+k} \cdot I_{\text{Pre}}). \tag{18}$$

4.2. Experimental Settings. In this section, we conduct experiments to analyze the performance of proposed network. The goal of this study is the commercial Digimarc watermark, and the embedding and extraction of the watermark are carried out on the Photoshop software platform. The datasets we use are Set9 and Set14 Bevilacqua et al. [24]; all images are color and resized as 512×512. The examples of test image are shown in Figure 5. As mentioned in Subsection 2.2, the most important variable in the embedding process is the durability of the watermark, where the variable S refers to the strength of the watermark. Each group of experiments will test the watermarked images of four intensities separately. The objective evaluation indicators include the strength of the watermark after attack, PSNR, and SSIM. PSNR and SSIM are two common metrics for assessing the quality of the reconstructed image. Higher PSNR and SSIM values generally indicate a better quality of the image Hore and Ziou [25]. We therefore used the average PSNR and SSIM values to assess the quality of processed images. In addition, to better reflect the excellent performance of the proposed network, we have also compared the proposed method with StirMark. The most famous software in the traditional anti-watermarking technology is StirMark software. The integration of this application covers almost all traditional watermark attacking methods, which can be described as a benchmark. Because of the variety of

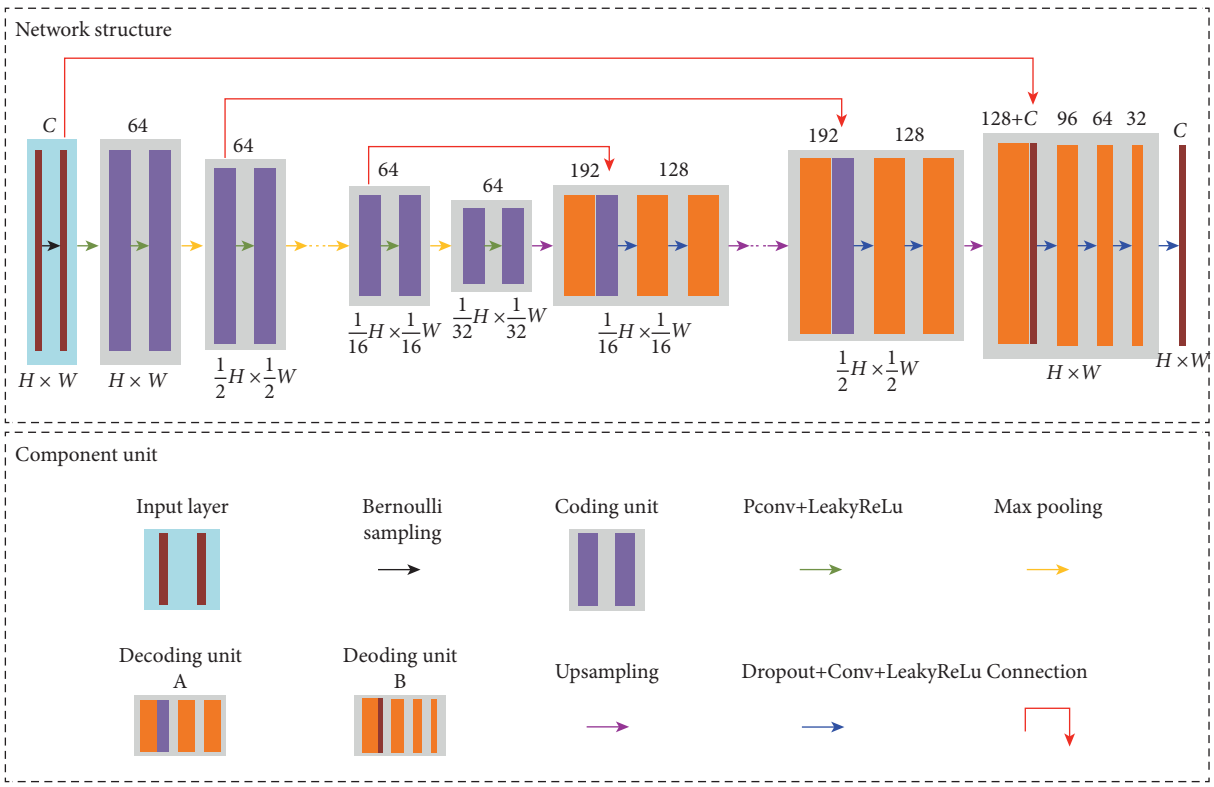


Figure 4: Overview of the network architecture.

(a) (b) (c) (d) (e)

Figure 5: The examples of test image. (a) F16. (b) Lena. (c) Man. (d) Pepper. (e) Baboon.

Table 1: Strength of watermark extracted after different methods of attack.

	Test images		$F16_{a1}$	$F16_{a2}$	$F16_{a3}$	$F16_{a4}$
StirMark	Proposed method		None	None	None	None
	Wang et al. [26]		None	None	None	None
	Cropping	$c = 0.75$	Weak	Medium	Medium	Medium
		$c = 0.5$	Very weak	Weak	Weak	Weak
		$c = 0.25$	None	None	None	None
	Rotation	$r = 180^{\circ}$	Medium	Medium	Medium	Medium
		$r = 150^{\circ}$	Medium	Medium	Medium	Medium
		$r = 90^{\circ}$	Medium	Medium	Medium	Medium
	Scaling	sc = 0.5	Weak	Weak	Weak	Weak
		sc = 1.5	Medium	Strong	Strong	Strong
		sc = 2	Medium	Medium	Medium	Medium

Table 2: Visual effects after being processed by various attacks. The strength of extracted watermark is in the ().

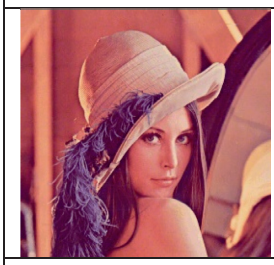
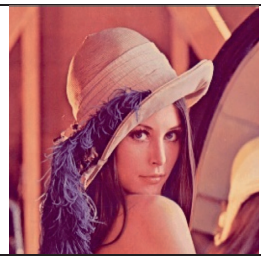
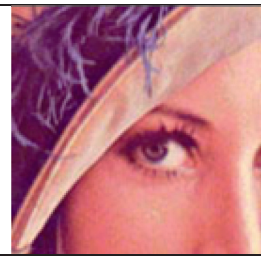
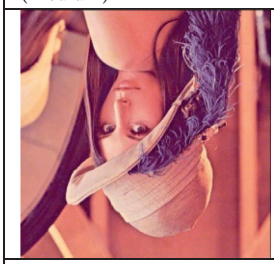
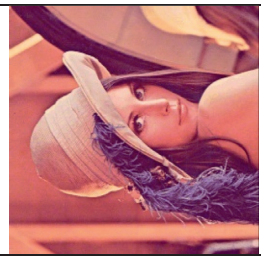
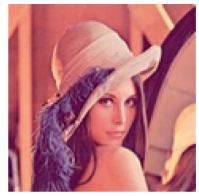
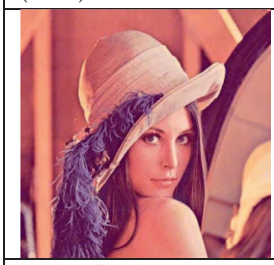
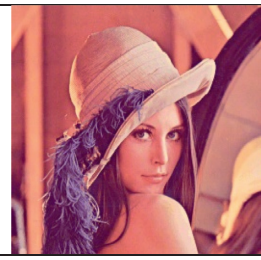
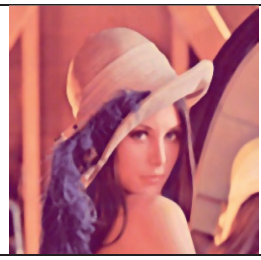
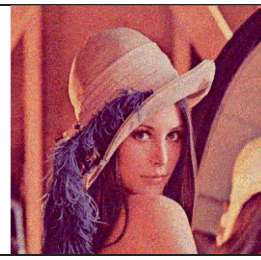
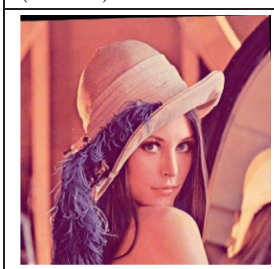
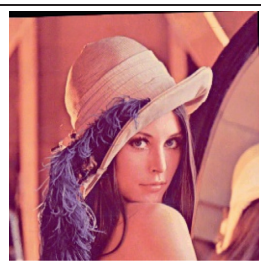
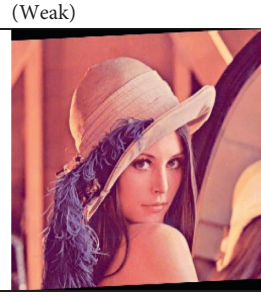
Lena_a3	The proposed method	Cropping c =0.75	Cropping c =0.25
	(None)	(Medium)	(None)
			
Rotation r =180	Rotation r =90	Scaling sc =0.5	Scaling sc =0.75
(Medium)	(Medium)	(Very weak)	(Medium)
			
JPEG Compress QF =50	JPEG Compress QF =30	Desynchronization SS =3	Desynchronization SS =2
(Weak)	(Very weak)	(Strong)	(Medium)
			
Median filter k =5	Median filter k =9	Gaussian noise =4	Gaussian noise =8
(Strong)	(Weak)	(Weak)	(Very weak)
			
Random distortion d =0.95	Random distortion d =1.05	Affine transformation f =5	Affine transformation f =2
(Medium)	(Medium)	(Very weak)	(Weak)
			

Table 3: Performance comparison in terms of removal effect and image quality for test images.

Test images		Man_{a4}			$Pepper_{a4}$			$Baboon_{a4}$		
		Strength of watermark	PSNR	SSIM	Strength of watermark	PSNR	SSIM	Strength of watermark	PSNR	SSIM
Proposed method		**None**	**39.48**	**0.974 4**	**None**	**36.76**	**0.915 4**	**None**	**36.35**	**0.941 3**
JPEG	$Q = 30$	None	24.28	0.786 6	Very weak	23.04	0.762 7	Very weak	22.96	0.780 5
	$Q = 50$	Very weak	24.76	0.824 8	Weak	24.45	0.800 2	Weak	23.61	0.827 5
	$Q = 70$	Weak	25.18	0.860 4	Weak	24.76	0.838 7	Weak	24.13	0.862 6
Gaussian noise	$\sigma = 2$	Medium	26.61	0.670 3	Medium	26.50	0.541 7	Medium	26.48	0.777 1
	$\sigma = 4$	Weak	20.94	0.430 8	Weak	20.61	0.286 7	Weak	20.51	0.535 1
	$\sigma = 8$	Very weak	14.83	0.217 1	Very weak	14.49	0.129 3	Weak	14.55	0.276 8
Median filter	$k = 2$	Strong	22.77	0.779 9	Medium	23.55	0.876 0	Medium	21.40	0.779 7
	$k = 5$	Strong	22.59	0.668 7	Medium	23.58	0.819 0	Medium	20.53	0.548 3
	$k = 7$	Medium	22.10	0.594 4	Medium	23.42	0.779 4	Medium	19.92	0.422 5

watermark attacking method types, only the representative part is selected in the experiment of this article.

We initial the Adam optimizer with a learning rate of $1e - 4$ to train the model and train our network on a single GPU NVIDIA GTX1080ti for 1.5×10^5 iterations. The time cost of each training is about 3.5 hours. The LeakyReLU activation function with a hyperparameter of 0.1, the Bernoulli with a sampling probability of 0.1, and a decoder with a "dropout" rate of 0.3 are used in training. In the test phase, a total of 100 dropouts were performed on the model, and the actual result is the average of the 100 processing results.

4.3. Performance Evaluation. In this section, we will show the results of attack on watermarking of different strengths through our method and the comparison experiments with Wang, Qian, Feng, and Zhang [26] and StirMark. We choose the "F16" in Set9 as the test image and embed the watermark with the strength a with 1, 2, 3, and 4, and then, we will get the four images to be attacked. Table 1 shows the results after processing with the method proposed, and the comparison with Wang et al. [26] and several common geometric attacks in StirMark. Let the parameter of the cropping attack be c, and the value is $\{0.75, 0.5, 0.25\}$; the rotation attack parameter be r, and the value is $\{180^\circ, 150^\circ, 90^\circ\}$; and the scaling attack parameter be sc, and the value is $\{0.5, 1.5, 2\}$. It can be seen from the table that the Digimarc watermark has a very high correction ability in resisting rotation attacks. In a scaling attack, the reduction operation weakens the watermark more than the zoom operation. The cropping attack has a higher destruction rate than the other two methods, and it can completely eliminate the watermark in the case of great damage to the image. For the Digimarc watermark, the weakening of the strength does not mean that the watermarking information disappears. Only the effect of the attack makes the software identification result "none," and this attack is meaningful. Although the method proposed in Wang et al. [26] can also meet the requirement, the proposed method is more convenient when training. In our opinion, the visual quality of the image should not be affected too much while the watermark is erased. Different from StirMark, our method can completely erase watermark of different durability while maintaining good image quality.

To show the visual effect of our method more intuitively, we take the Lena image as the test image and embed the Digimarc watermark with intensity 3 in it. Table 2 shows the results of $Lena_{a3}$ under each attack. The quality factor of JPEG compression is defined as QF, the mean value of Gaussian noise is 0, and the standard deviation is defined as σ. The kernel size of the median filter is k. The parameters of desynchronization attack, random distortion attack, and affine transformation are denoted by ss, d, and f, respectively.

It can be seen from Table 2 that although traditional attacks have different methods, the results will inevitably cause serious visual distortion, and the watermark still exists in this case. Taking the most common JPEG compression attack in reality as an example, the image compression factor in online social networks (OSN) will be around 70, and even when the QF is set as 30, the Digimarc watermark cannot be completely removed. The quality of the watermarked image processed by our method is almost unchanged from that before processing. Table 3 used two objective evaluation indicators, PSNR and SSIM, to illustrate the image quality after processing.

In Table 3, three test images of Man, Pepper, and Baboon are selected for watermark embedding. When the embedding intensity is 4, "Man_{a4}," "$Pepper_{a4}$," and "$Baboon_{a4}$" are obtained, respectively. The result shows the comparison of the watermarking strength and visual quality of the test image after several methods of attack. Table 3 further verifies the robustness of Digimarc watermark itself. Even if the standard deviation of Gaussian noise is set to 8, the watermark still cannot be removed. As a comparison, our method can not only remove the watermark perfectly, but also improve the image quality compared with other methods. Taking the "Man" as an example, compared with our method, the PSNR is improved by 24.65 dB, and the SSIM is improved by 0.757 3. The image quality has reached a level that cannot be detected by the human eye.

We use four intensities to embed watermark on the five test images. After all the objects are processed by the proposed method, no watermark can be extracted from the images. Table 4 also shows the image quality results of all the attacked images, which meets the expectation that the attack is invisible.

Table 4: Image quality of watermarked images with different intensities after being attacked.

		F16	Lena	Man	Pepper	Baboon
$a = 1$	PSNR	40.10	39.68	41.29	38.69	37.29
	SSIM	0.956 0	0.953 0	0.982 1	0.934 3	0.946 6
$a = 2$	PSNR	39.36	38.74	40.37	37.75	36.69
	SSIM	0.950 3	0.946 2	0.978 3	0.926 4	0.942 4
$a = 3$	PSNR	38.80	38.01	39.95	37.27	36.48
	SSIM	0.945 6	0.940 7	0.976 6	0.920 9	0.942 6
$a = 4$	PSNR	38.37	37.20	39.48	36.76	36.35
	SSIM	0.941 8	0.932 3	0.974 4	0.915 4	0.941 3

5. Conclusions

In this study, we propose an attack model against the commercial Digimarc watermark. We have solved the problem that the principle of commercial watermark is unknown and the amount of training data is scarce. The attack rate of our method is very high, so that the watermarking information after the attack cannot be extracted completely, not only weakening the strength of the watermark. The experimental results show that our method has a remarkable improvement in attack performance and image quality compared with various attack methods in StirMark. In the future, we will explore how to add the process of extracting the watermark to the back propagation, which may achieve better results.

Data Availability

No data were used in the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modelling: towards a second generation watermarking benchmark," *Signal Processing*, vol. 81, no. 6, pp. 1177–1214, 2001.

[2] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[3] R. C. Dixon, *Spread Spectrum Systems: With Commercial Applications*, John Wiley & Sons, Hoboken, NJ, USA, 1994.

[4] X. Kang, J. Huang, and W. Zeng, "Efficient general print-scanning resilient data hiding based on uniform log-polar mapping," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 1–12, 2010.

[5] D. Coltuc and P. Bolon, "Robust watermarking by histogram specification," in *Proceedings of the 1999 International Conference on Image Processing (Cat. 99CH36348)*, pp. 236–239, IEEE, Kobe, Japan, October 1999.

[6] S. A. Parah, J. A. Sheikh, N. A. Loan, and G. M. Bhat, "Robust and blind watermarking technique in DCT domain using inter-block coefficient differencing," *Digital Signal Processing*, vol. 53, pp. 11–24, 2016.

[7] 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.

[8] J. K. Su and B. Girod, "Power-spectrum condition for energy-efficient watermarking," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 551–560, 2002.

[9] A. D'angelo, M. Barni, and N. Merhav, "Stochastic image warping for improved watermark desynchronization," *EURASIP Journal on Information Security*, vol. 2008, pp. 1–14, Article ID 345184, 2008.

[10] A. D'Angelo, *Characterization and quality evaluation of geometric distortions in images with application to digital watermarking*, Ph.D. thesis, Springer, Berlin, Germnay, 2009.

[11] I. J. Cox and J. P. M. G. Linnartz, "Some general methods for tampering with watermarks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 587–593, 1998.

[12] S. A. Craver, N. D. Memon, B. L. Yeo, and M. M. Yeung, "Can invisible watermarks resolve rightful ownerships?" in *Storage and Retrieval for Image and Video Databases V*International Society for Optics and Photonics, Bellingham, WA, USA, 1997.

[13] M. Kutter, S. V. Voloshynovskiy, and A. Herrigel, "Watermark copy attack," in *Security and Watermarking of Multimedia Contents II*, pp. 371–380, International Society for Optics and Photonics, Bellingham, WA, USA, 2000.

[14] A. K. Shukla, R. K. Pandey, S. Yadav, and R. B. Pachori, "Generalized fractional filter-based algorithm for image denoising," *Circuits, Systems, and Signal Processing*, vol. 39, no. 1, pp. 363–390, 2020.

[15] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[16] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Removing spatial spread spectrum watermarks by non-linear filtering," in *Proceedings of the 9th European Signal Processing Conference (EUSIPCO 1998)*, pp. 1–4, IEEE, Rhodes, Greece, September 1998.

[17] L. Geng, W. Zhang, H. Chen, H. Fang, and N. Yu, "Real-time attacks on robust watermarking tools in the wild by CNN," *Journal of Real-Time Image Processing*, vol. 17, pp. 1–11, 2020.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[19] F. Deguillaume, G. Csurka, and T. Pun, "Countermeasures for unintentional and intentional video watermarking attacks," in *Security and Watermarking of Multimedia Contents II*, pp. 346–357, International Society for Optics and Photonics, Bellingham, WA, USA, 2000.

[20] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarks," in *Proceedings of the International Workshop on Information Hiding*, pp. 199–210, Springer, Cambridge, UK, May 1999.

[21] J. Zhu, R. Kaplan, J. Johnson, and L. F. Fei, "Hidden: hiding data with deep networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 657–672, Munich, Germany, September 2018.

[22] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, Salt Lake City, UT, USA, June 2018.

[23] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: learning self-supervised denoising from single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1890–1898, IEEE, Seattle, WA, USA, June 2020.

[24] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference (BMVC)*, Guildford, England, September 2012.

[25] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, IEEE, Istanbul, Turkey, August. 2010.

[26] H. Wang, Z. Qian, G. Feng, and X. Zhang, "Defeating data hiding in social networks using generative adversarial network," *EURASIP Journal on Image and Video Processing*, vol. 2020, pp. 1–13, 2020.