Hindawi
Security and Communication Networks
Volume 2021, Article ID 2356284, 12 pages
https://doi.org/10.1155/2021/2356284

WILEY | Hindawi

Research Article

Disguise of Steganography Behaviour: Steganography Using Image Processing with Generative Adversarial Network

Mingjie Li, Zichi Wang, Haoxian Song, and Yong Liu

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Mingjie Li; mingjie8699@126.com

Received 20 July 2021; Accepted 16 November 2021; Published 8 December 2021

Academic Editor: Hao Peng

The deep learning based image steganalysis is becoming a serious threat to modification-based image steganography in recent years. Generation-based steganography directly produces stego images with secret data and can resist the advanced steganalysis algorithms. This paper proposes a novel generation-based steganography method by disguising the stego images into the kinds of images processed by normal operations (e.g., histogram equalization and sharpening). Firstly, an image processing model is trained using DCGAN and WGAN-GP, which is used to generate the images processed by normal operations. Then, the noise mapped by secret data is inputted into the trained model, and the obtained stego image is indistinguishable from the processed image. In this way, the steganographic process can be covered by the process of image processing, leaving little embedding trace in the process of steganography. As a result, the security of steganography is guaranteed. Experimental results show that the proposed scheme has better security performance than the existing steganographic methods when checked by state-of-the-art steganalytic tools, and the superiority and applicability of the proposed work are shown.

1. Introduction

Data hiding is an important technique to solve security problems and protect data. Steganography is an important branch of data hiding, which can be divided into image steganography [1, 2], audio steganography [3, 4], and video steganography [5, 6], according to different carriers. Image steganography aims to conceal secret data within cover images transmitted through public channels without causing suspicion [7, 8]. As the adversary, the purpose of image steganalysis is to reveal the presence of steganography by detecting the images transmitted on public channels [9, 10]. Image steganography and steganalysis technology have achieved rapid development in opposition. Traditional image steganography methods are mainly based on embedding, in which secret data is embedded into a cover image by modification [11, 12].

At present, the most advanced approach is content adaptive steganography (e.g., HILL [13], SUNIWARD [14], and WOW [15]). The main idea is to minimize the heuristically defined embedding distortion to achieve high undetectability [16]. In recent years, GAN (generative adversarial network) is gradually being combined with traditional steganography [17–20]. These methods are still modification based steganography, in which the modification traces will inevitably be left during embedding and be detected by a modern steganalyzer [21]. Modern steganalysis mainly uses supervised machine learning to classify cover and stego images. Features are extracted from an image training set to train a steganalytic model firstly and are then used to identify suspicious images [22–24]. The ensemble classifier [25] is commonly used to improve the detection accuracy, which has been proved to be effective in steganalysis. In recent years, the performance of deep learning based image steganalysis is good, in which the feature extraction and classifier are combined together [26–28]. Qian et al. [26] firstly use the KV kernel in the traditional method to filter the image to generate a residual image and use a Gaussian function to extract more effective features, and the final performance is slightly inferior to SRM [22]. In XuNet [27], Tanh activation function is used in the first two layers of the network model, 1×1 convolution is used in the upper

layer, and batch normalization layer is added behind the convolution layer to prevent the model from converging to a local minimum. The performance of the XuNet model is comparable to that of the rich model. YeNet [28] puts 30 kV kernels in the SRM into the first layer of the model to extract residual information, combines with channel selection information, and uses truncated linear units to filter the features. The performance exceeds SRM and its variant algorithms. Up to now, the deep learning based image steganalysis is becoming a serious threat to modified steganography.

Steganography without embedding (SWE) means that there is no need to modify the carrier to realize information hiding. Compared with embedding-based steganography, it leaves no trace of modification and has the ability to resist these state-of-the-art deep learning based image steganalyzers [21]. As a type of SWE, generation-based steganography has been developed in recent years [21, 29–31], in which the stego image is produced by secret data directly. In [29–31], image generation steganography method is designed by mapping the secret information from semantic features of the cover images. The disadvantage is the small steganographic capacity and the stego image generated is a fixed type of texture synthesis image or fingerprint image, which may attract the attention of the warder. Hu et al. [21] propose an image generation steganography method by mapping secret information to random noise according to certain rules based on DCGAN.

In some scenarios, images are processed (e.g., histogram equalization and sharpening) for better visual quality. Since these operations are widely used in image processing, the processed images will not cause noticeable suspicion. Thus, the processed images can be used for steganography. However, the undetectability of steganography will decrease when the cover image is processed [12]. Therefore, we propose a novel generation-based steganography method in this paper, by disguising steganography as image processing operations instead of modifying the processed image to resist these state-of-the-art steganalyzers. Specifically, we first train image processing generation model based on DCGAN and WGAN-GP, and then the model can generate high-quality processed images through random noise mapped by secret data. In this way, the steganographic process can be covered by the process of image processing, and the purpose of leaving no embedded trace in the process of image steganography can be realized. Compared with [29, 30], the stego images generated have better diversity. In addition, our method has better security performance than existing steganographic methods when checked by state-of-the-art steganalytic tools.

2. Related Work

This part introduces the related work of image generation steganography we propose. Since the steganographic method we propose is based on DCGAN and WGAN-GP, this section first introduces the basic principles of GAN and its variants, as well as their respective advantages and disadvantages. Then, the applications of GAN and its variants in the field of image steganography in recent years are described.

2.1. GAN and Its Variants. GAN is a deep unsupervised learning model proposed in 2014 [32]. It is characterized by being able to generate samples as close as possible to the data distribution of the target samples. GAN uses game theory to train two neural networks: generator model G and discriminator model D. G converts the random noise that obeys the prior distribution into generated samples, so that the distribution of the generated sample that obeys is as close to the distribution of the real data as possible; D tries to determine whether the input samples are real samples or generated samples. The training process of GAN can be summarized as a Minimax game: discriminator D tries to maximize the probability of correctly distinguishing real samples from generated samples; generator model G tries to maximize the probability that discriminator model D cannot distinguish generated samples. The objective function when training GAN is shown in (1), where $G(z)$ represents the sample generated by generator G according to input random noise vector $\mathbf{z}$, and $D(x)$ represents the probability that discriminator model D determines that sample x is the real sample:

$$\min_G \max_D L(D,G) = E_{x\sim \mathrm{Pdata}(x)}[\log D(X)] + E_{z\sim \mathrm{Pnoise}(z)}[\log(1 - D(G(Z)))]. \quad (1)$$

GAN has become a research hotspot since it was proposed. However, there are some problems in GAN, such as unstable training, disappearance of gradients, and mode collapse, so there are constantly new varieties, such as DCGAN [33], WGAN [34], and WGAN-GP [35], which greatly promote the development of GAN.

DCGAN is a better improvement after GAN. The improvement is mainly in the network structure. DCGAN almost completely uses convolution layer instead of full link layer. The discriminator is almost symmetric to the generator. The entire network does not have pooling layer and upsampling layer. In fact, convolution with step length is used instead of upper sampling to increase the stability of training. The network structure of DCGAN is widely used to improve the stability of GAN training and the quality of generated results [33].

Although DCGAN has a good structure, it still cannot improve the stability of GAN training, and it needs to balance the training process of G and D carefully. Unlike DCGAN, WGAN mainly improves GAN from the perspective of loss function, and WGAN theoretically solves the problems of model training instability and model collapse

and generates more diverse samples [34]. WGAN-GP model puts forward a new method of continuity restriction gradient penalty, which solves the problem of gradient disappearance and gradient explosion and has faster convergence speed than the standard WGAN [35]. In addition, WGAN-GP model can generate higher quality samples without much parameter adjustment in the whole training process.

2.2. GAN for Image Steganography. In recent years, steganalysis has gradually combined with deep learning and has achieved a great degree of development. This poses a great threat to image steganography [36]. And research on deep learning based image steganography has also made some progress. These tasks have mainly achieved image steganography through the training of neural networks on large-scale sample sets. Zhang et al. [36] add the gradient of the loss function of the neural network steganalysis of the target as the antinoise to the input image, forming the adversarial sample to resist the detection of steganalyzer. However, the disadvantage of the adversarial sample is that it is difficult to extract secret information [19]. Tang et al. [18] propose an automatic steganographic learning framework ADSL-GAN based on the STC algorithm [37]. ADSL-GAN achieves the adaptive embedding of secret information into the carrier image, but its performance is slightly inferior to that of S-UNIWARD. The variant of the ADSL-GAN model is produced later [19]. Since the ADSL-GAN model and its variant still embed secret information by modifying image pixel values, they will inevitably be detected by more advanced steganalyzers because they leave traces of modification. Volkhonskiy et al. [17] design an image steganographic model SGAN based on DCGAN. The model includes generator G, discriminator D, and steganalyzer S. The three parts update the parameters of each other through alternating training to make the SGAN model get better performance. SGAN generates carrier images by generating adversarial networks. The steganographic process still uses traditional steganographic algorithms. SGAN does improve the security of the model, but the carrier image generated is not real enough, which facilitates attracting the attention of channel listeners. In this paper, we try to implement image generation steganography in the process of image processing by using different GAN models and detect the security of the stego images.

3. Proposed Image Steganographic Scheme

The flowchart of the proposed method is shown in Figure 1. Firstly, the original image set is processed by common image processing methods of histogram equalization and sharpening to form processed image sets. Then, the generation model based on DCGAN and WGAN-GP is trained with the formed processed image sets. Next, we train corresponding extraction models according to the recovery error of random noise vector to extract secret data from the stego image generated by the generation models. At last, along with the method of [21], the sender divides the secret data to be transmitted into several segments, maps each segment to the noise z_i through the mapping rules, and then inputs z_i into the trained generation model to generate the processing stego image: the receiver uses the trained extraction model to extract the noise vector $\mathbf{z}'$ from the received processed stego image and recovers the secret data according to the reverse mapping rules.

3.1. Training Image Processing Generation Model. Since our idea is to embed secret information in the process of image processing, the image processing generation model needs be trained firstly. The generation model uses GAN excellent variants—DCGAN and WGAN-GP. Before the generation model training, histogram equalization and sharpening are performed on the original image set to form two processed image sets. Then, the two processed image sets are used to train DCGAN and WGAN-GP, respectively, until the two models converge and generate high-quality images.

3.2. Training the Extraction Model. The purpose of the extraction model is to extract the noise vector from the processed stego image generated by the generation model and then recover the secret data according to the mapping rules. The structure of the extraction model is shown in Figure 2. It is similar to the structure of the discriminator D in DCGAN [33] and is mainly composed of convolutional neural networks and fully connected networks. For the purpose of enhancing the ability of nonlinear learning and improving the speed of network convergence, each layer of convolutional neural network of the extraction model network uses a Leaky Relu activation function and batch normalization. In order to satisfy the output noise value of the extraction model between −1 and 1, we set the output layer function of the network as tanh.

The training process of the extraction model is as follows: Firstly, we generate random noise vector $\mathbf{z}$ with the same dimensions as z_i between −1 and 1 continuously. Then, random noise vector $\mathbf{z}$ is input into the trained image processing generation model to generate processed stego image, and then the processed stego image is input to the extraction model to extract the noise vector $\mathbf{z}'$. Here, the square loss between the input random noise vector $\mathbf{z}$ and the recover noise vector $\mathbf{z}'$ extracted from the extraction model is defined as the loss function of the extraction model. During the training of the extraction model, when its loss function is small enough and the network converges, we can use it to extract noise vectors from the processed stego images.

3.3. Communication of Secret Information. We refer to the method of reference [21] to map secret information into a range of random noise values. Firstly, the sender divides the secret information s into s_i segments and then maps each segment of the secret information s_i into corresponding noise vector z_i according to the mapping rules. In the mapping process, the divided secret information s_i is mapped to a noise value with a given interval according to the following equation:

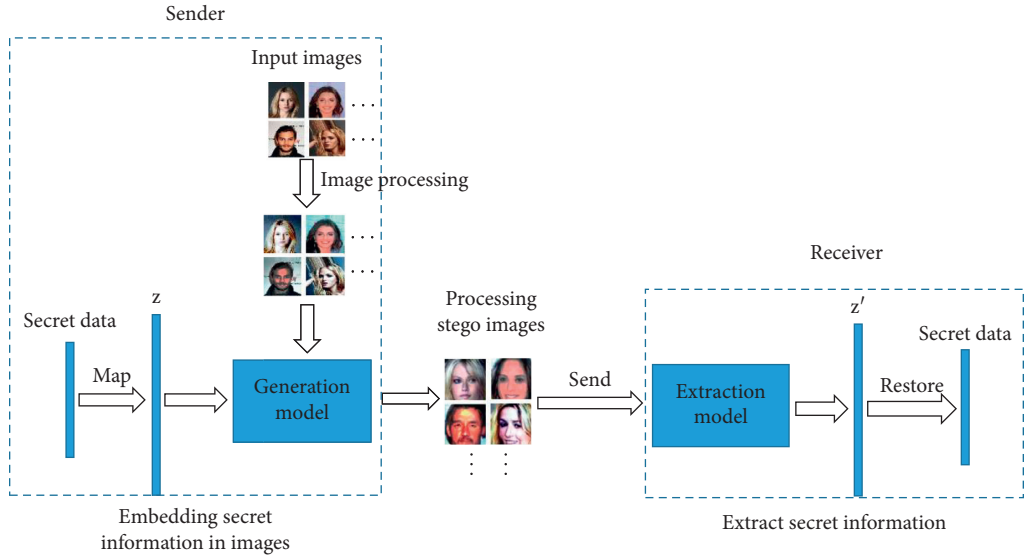

Figure 1: The flowchart of the proposed steganographic framework in the image processing process.

$$r = \text{random}\left(\frac{m}{2^{\sigma-1}} - 1 = \delta, \frac{m+1}{2^{\sigma-1}} - 1 - \delta\right). \quad (2)$$

In the equation, the function random (x, y) generates the mapping noise vector r by the interval x, y. m is the value of the secret information bits to be mapped, and σ is a positive integer variable, representing the number of secret information bits mapped by a random noise. δ represents the gap between the divided intervals, and a small amount of deviation is allowed when extracting secret information from the processed stego images, so that the accuracy of secret information extraction can be ensured.

Subsequently, the sender generates processed stego images through the mapped noise values and the trained image processing generation model. The sender transmits the processed stego image generated to the receiver. After receiving the processed stego images, the receiver extracts the corresponding noise vector through the trained extraction model and restores the corresponding secret information according to the reverse mapping rules. Here, since the mapping rule refers to Formula (2), the information extraction process also requires Formula (2) and its parameters.

4. Experimental Results and Analysis

In this paper, we use the Celebrities dataset to train the models. This dataset contains about 200,000 faces and is one of the commonly used datasets. Before the experiment, all the images are cropped to 64×64 pixels firstly, and then we perform histogram equalization and sharpening on the cropped datasets with MATLAB to form two new datasets, dataset 1 and dataset 2. Then, we use dataset 1 and dataset 2 to train the image processing generation models based on DCGAN and WGAN-GP, respectively. The models are trained using minibatch stochastic gradient descent (SGD) with a minimum batch size of 64. The initial learning rates of the DCGAN and WGAN-GP models during training are set to 0.0002 and 0.0001, respectively. In addition, Adam optimization method is used to adjust the learning rate of the models automatically during the training process, which can make the network model easier to converge and learn network parameters smoothly. During the experiment, the model of GPU used is NVIDIA Tesla P100-PCIE-16GB and the experimental environment is TensorFlow.

The experiment consists of four parts: In the first part, we train image processing generation models based on DCGAN and WGAN-GP with dataset 1 and dataset 2, respectively. In this part, we compare the convergence process of two network models on two training sets and the quality of the generated images. In the second part, we train the extraction model based on the output of the trained image processing generation model. In the third part, we generate processed stego images through mapped noise values and the trained image processing generation model and use the trained extraction model to extract the noise vector and recover the secret information and then test the recovery accuracy of the embedded secret information in the process of image processing. In the fourth part, we test the security of the proposed steganographic method with several state-of-the-art steganalysis methods today.

4.1. Training Image Processing Generation Model. In the process of training image processing generation model, we train DCGAN and WGAN-GP models with dataset 1 and dataset 2, respectively, with the purpose of generating high-quality histogram equalization and sharpening images. In the experiment, the dimension of random noise vector is 100, which is input into the generation model. And we set the initial learning rate of 0.0002; the minibatch is 64 to train the DCGAN model for 100 epochs, and WGAN-GP model is trained for 200 epochs; the initial learning rate is 0.0001, and the minibatch is 64. The loss curve of training generation model is shown in Figure 3.

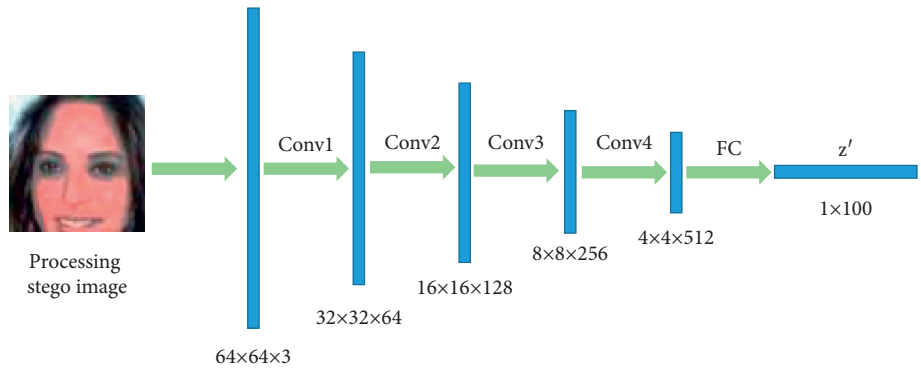

Figure 2: The structure of the extraction model.

Through comparative experiments, we observe that the DCGAN model is easier to converge than the WGAN-GP model during the training process, but the parameters of the DCGAN model need to be adjusted continuously during the training process; the WGAN-GP is more stable in the training process; the parameters do not need to be adjusted during the training process. And, during training, the histogram equalization processing generation model is easier to converge than the image sharpening processing generation model. The processed stego images generated by different types of image processing generation models are shown in Figure 4. As can be seen from Figure 4, the processed image generated by the image processing generation model based on WGAN-GP is more real and of higher quality than the processed image generated by the image processing generation model based on DCGAN.

4.2. Training the Extraction Model. Since the extraction model is mainly composed of convolutional neural network and fully connected network, therefore it needs a large dataset to train in the training process. In this paper, a large number of random noise vectors are input to the trained image processing generation model to generate processed stego images firstly, and then we train the extraction model by minimizing the mean square error between the corresponding input noise vector and the extracted noise vector. During training, the input images of the extraction model are processed stego images generated by the trained image processing generation model, with size of 64×64 pixels, and the optimization method is Adam optimization. In addition, we set the initial learning rate of the extraction model to 0.0002 and the minibatch to 100. Finally, we recorded 500 batches as an iteration, and loss curves of the extraction models corresponding to the four image processing generation models after 100 iterations are shown in Figure 5. It can be seen from the change curve of the loss function of the extraction model that, during the initial training period, the loss function curve fluctuates violently, which means that the ability to extract the noise vector is weak. In addition, it can be seen from Figure 5 that the loss function of the extraction model corresponding to the histogram equalization generation model converges faster than the loss function of the extraction model corresponding to the image sharpening generation model. When the extraction model of the histogram equalization generation model is trained for 80 iterations, and the extraction model of the image sharpening generation model is trained for 90 iterations, their corresponding loss function curves are gradually stable, which means that the ability to extract the noise vector has become very strong.

4.3. Secret Information Recovery Accuracy. For a steganographic method, it is important to extract the embedded information completely [38]. We define the recovery accuracy as the ratio of the number of secret information bits which correctly recovered from the processed stego image to the number of original secret information bits. Here, we test the important index of recovery accuracy of secret information. It can be seen from [21] that different σ values have a certain effect on the secret information recovery accuracy. Therefore, during this part of the experiment, we extract the secret information of the processied stego images generated by the four image processing generation models under different σ and δ is 0.01, study the corresponding secret information recovery accuracy, and compare it with the steganographic method proposed in [21] under the same conditions. The results are shown in Figure 6. It can be seen from Figure 6 that the smaller the payload, the higher the accuracy of the secret information recovery. In addition, under the same conditions, the recovery accuracy of the sharpening stego images generated is higher than that of the histogram equalization stego images generated. Finally, the recovery accuracy of the processed stego images generated by our proposed image processing generation model is slightly higher than that of the stego images generated in [21] under the same conditions.

In addition, in order to test the robustness of the proposed steganographic method, we add 10 dbw Gaussian noise interference to the processed stego images generated and test the recovery accuracy of the processed stego images generated by our proposed image processing generation model under the condition of σ being 1 and δ being 0.01; the test results are shown in Table 1. Comparing Figure 6 and Table 1, we can see that adding Gaussian noise to the processed stego images generated can indeed reduce the accuracy of the recovery of secret information to a certain extent, but the reduction is very small.

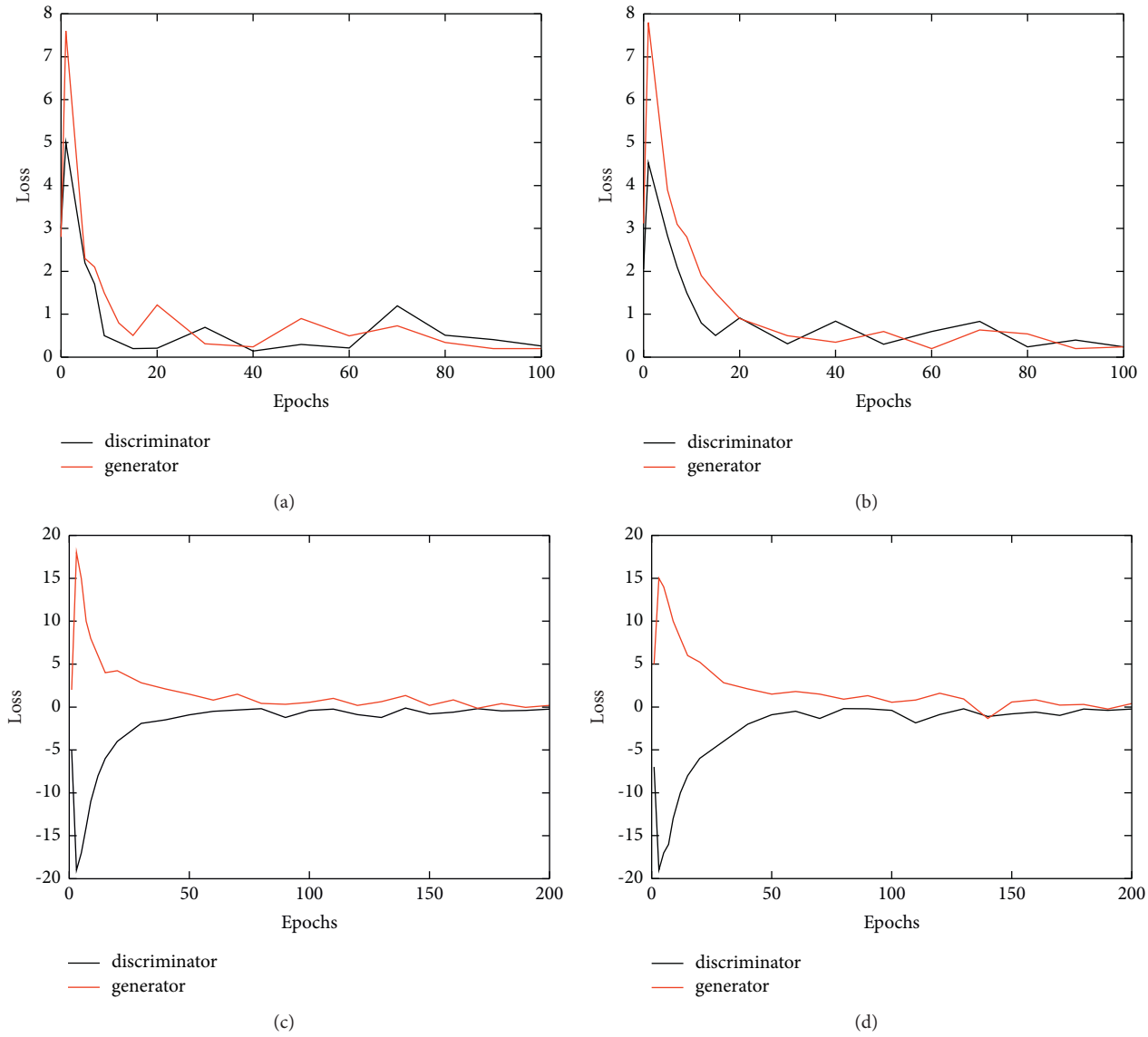

Figure 3: The loss curve of training image processing generation model. (a) Loss curve during training of generation model for image histogram equalization processing based on DCGAN. (b) Loss curve during training of generation model for image sharpening processing based on DCGAN. (c) Loss curve during training of generation model for image histogram equalization processing based on WGAN-GP. (d) Loss curve during training of generation model for image sharpening processing based on WGAN-GP.

4.4. Security Analysis. The characteristic of SWE method is that there is no difference between the cover images and the stegos image, so its undetectability is usually relatively strong [29, 30]. Since the steganographic method proposed in this paper is also a kind of SWE, therefore, in theory, the steganographic method can effectively resist the detection of steganalysis. Next, to study the undetectability of the steganographic method proposed in this paper, we use state-of-the-art image steganalysis methods to detect the processed stego images generated. The SRM proposed in [22] is a state-of-the-art feature extractor, which combines with an ensemble classifier [25] and constitutes a high performance steganalyzer in spatial domain. Ye et al. [28] proposed a CNN-based spatial steganalysis model, which is characterized by the initialization of the first layer with high pass filters and uses the activation function TLU and selection-channel-aware scheme for steganalysis. We use the trained image processing generation model to generate 5000 processed stego images and detect them with these state-of-the-art steganalyzers. This experiment is divided into the following two parts: In the first part of the experiment, 5000 processed cover images are taken from dataset 1 and dataset 2, respectively, and then they are embedded with several state-of-the-art spatial steganographic algorithms HILL, S-UNIWARD, and WOW with different payloads to generate the corresponding stego images. Next, we use these cover images and corresponding stego images to train the steganalyzer and test 5000 stego

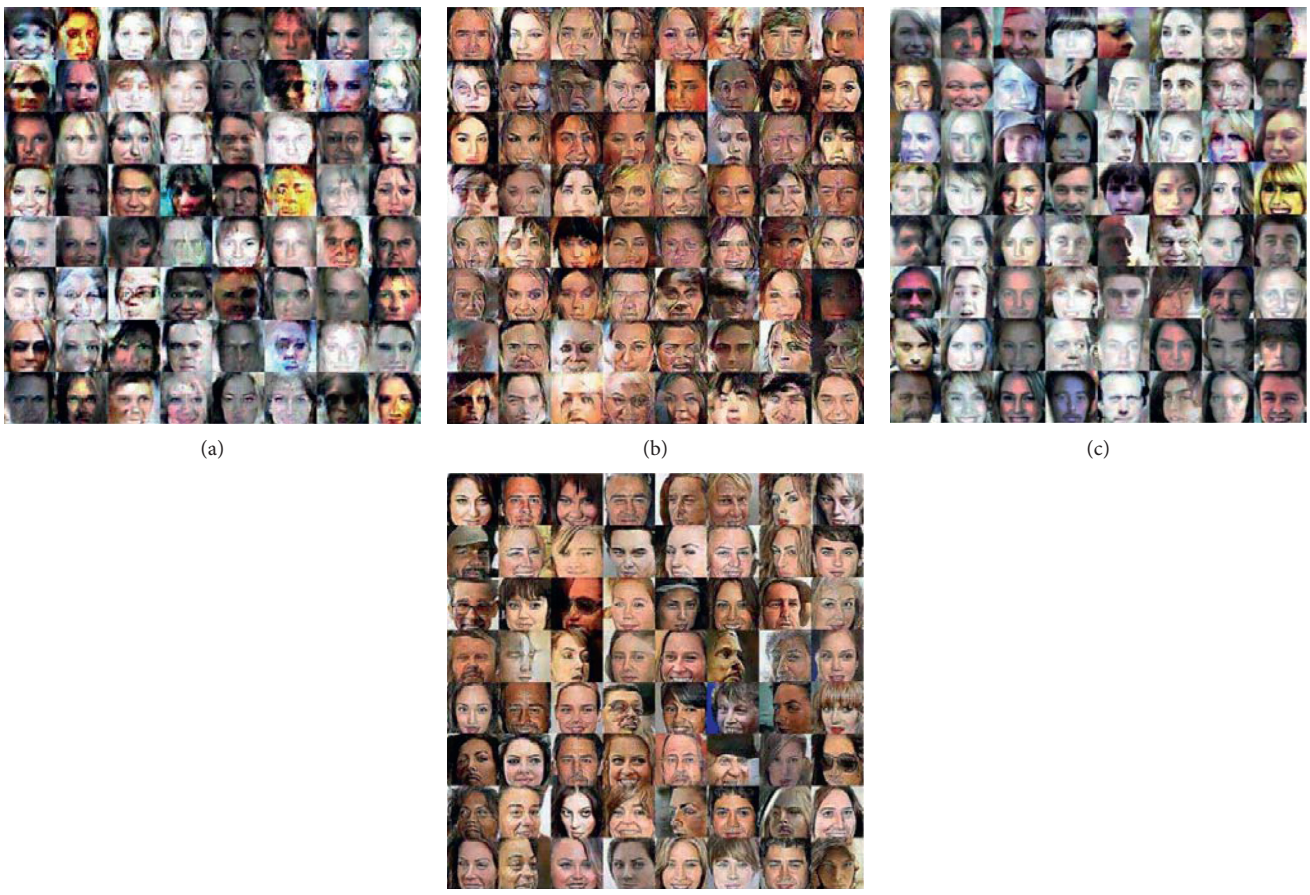

Figure 4: Minibatch generated processed images output from image processing generation model. (a) Samples of histogram equalization image processing generated by DCGAN model. (b) Samples of sharpening image processing generated by DCGAN model. (c) Samples of histogram equalization image processing generated by WGAN-GP model. (d) Samples of sharpening image processing generated by WGAN-GP model.

images generated by the image processing generation model. The criterion of evaluating the performance of the processed stego image set is to obtain the minimum total error P_E on the test set [28]:

$$P_E = \min\left(\frac{P_{FA} + P_{MD}}{2}\right), \tag{3}$$

where P_{FA} represents false alarm rate and P_{MD} represents missed detection rate. Testing error (P_E) of the processed stego image generated from different image processing steganography with different steganalyzers is shown in Tables 2 and 3. Column of Algorithm 1 of Table 2 and

Table 3, DCGAN1, DCGAN2, DCGAN3, WGAN-GP1, WGAN-GP2, and WGAN-GP3, respectively, represent the results of the test of processed stego images generated by the two image processing generation models with different steganalyzers. Among them, steganalyzers are trained by HILL, S-UNIWARD, and WOW steganographic algorithms with different payloads. And testing error rate comparison curve of the processed stego image generated from different image processing generation models with different steganalyzers is shown in Figure 7.

Figure 7 shows the undetectability comparisons of the processed stego images generated from different image processing generation models with different steganalyzers intuitively. Tables 2 and 3 depict all numerical values. The experimental results indicate that the use of traditional steganographic method to embed the secret information in the processed cover image to form the processed stego image has poor resistance to detection. Among them, with the increase of the steganographic payload, the undetectability of the processed stego images gradually deteriorates, especially when the payload increases to 0.5 bpp and the detection error P_E approaches 0. And through comparative experiments, we can observe that the processed stego images generated by the image processing generation model proposed in this paper can significantly improve the undetectability and, as the payload increases, the improvement effect is more obvious. In addition, the processed stego images generated by the image processing generation model based on WGAN-GP have stronger undetectability than the processed stego images generated by the image processing generation model based on DCGAN under the same conditions.

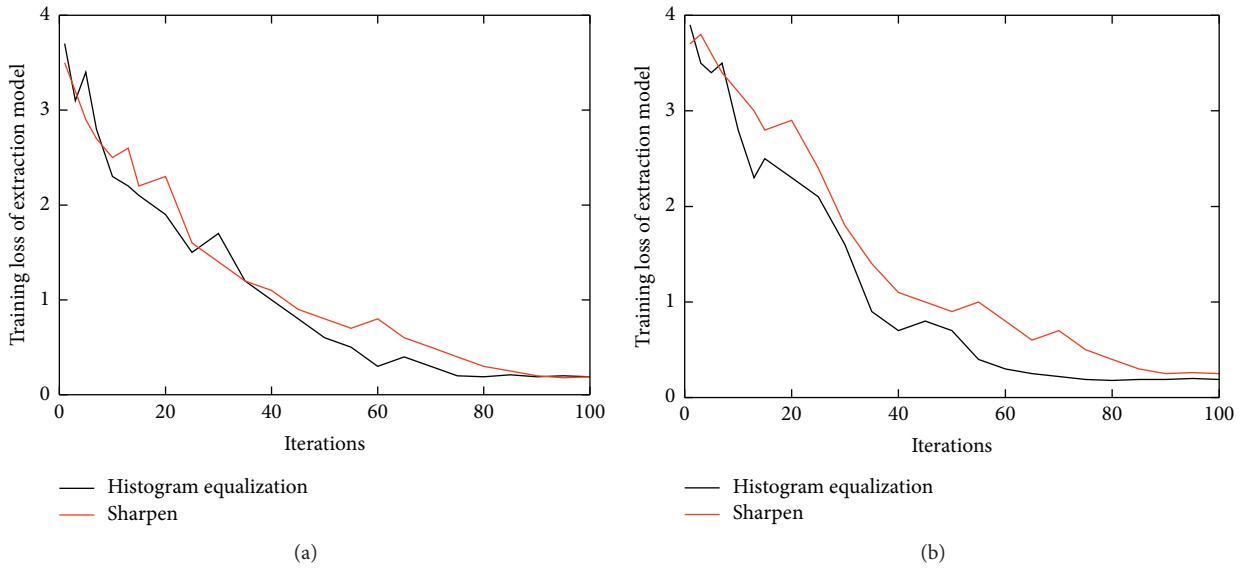

FIGURE 5: The loss curve of training extraction model. (a) Training loss of extraction model of image processing generation model based on DCGAN. (b) Training loss of extraction model of image processing generation model based on WGAN-GP.

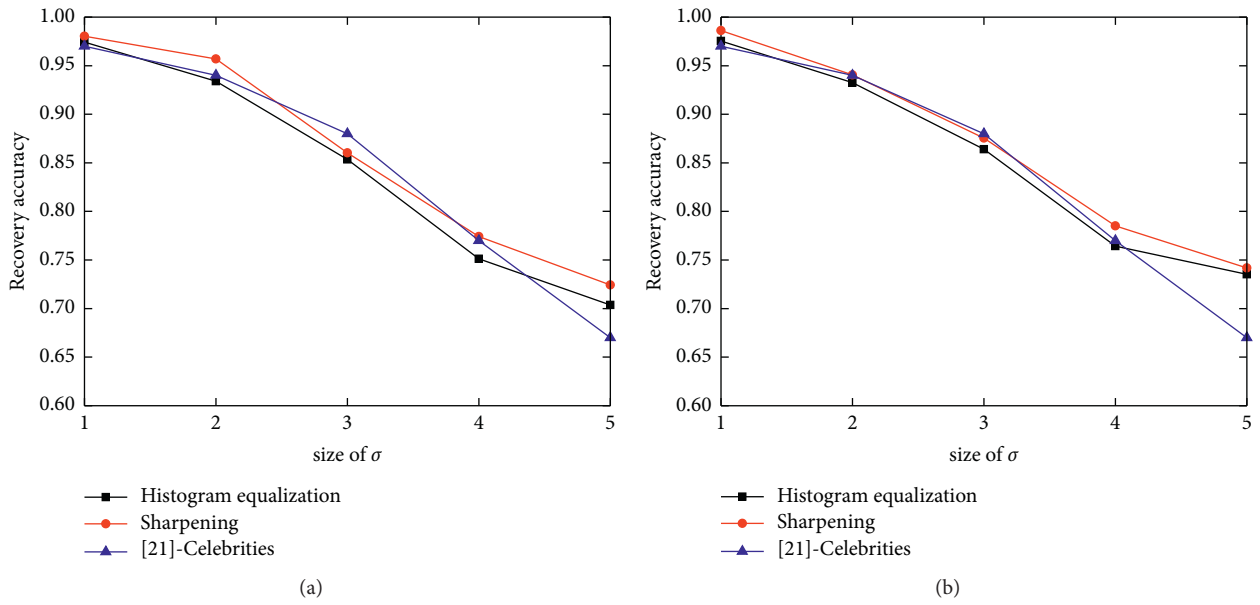

FIGURE 6: Recovery accuracy of secret information corresponding to different image processing models. (a) Recovery accuracy of secret information of image processing generation model based on DCGAN. (b) Recovery accuracy of secret information of image processing generation model based on WGAN-GP.

During the second part of the experiment, we use the processed cover images and the corresponding processed stego images generated from the image processing generation model proposed in this paper to train the steganalyzer and detect the processed stego images generated. The testing error (P_E) of the processed stego image generated from different image processing generation models with different steganalyzers is shown in Table 4.

According to [21], we know that the steganographic capacity of the processed stego image generated by the image processing generation model according to the mapping rule between secret information and noise is greater than or equal to 37.5, and the size of the processed stego image generated is 64×64, so the payload of the processed stego image generated is greater than or equal to 0.0732 bpp. It can be seen from Table 2 that the undetectability (P_E) of the processed stego image is gradually weakened with the increase of steganographic payload, so the undetectability of the processed stego image generated with payload of 0.0732 bpp is less than that generated with payload of 0.05 bpp. Therefore, it can be concluded that, under the condition of payload of 0.0732 bpp, the undetectability of the histogram equalization stego image

Table 1: The recovery accuracy of the processed stego image generated from different image processing generation model under the condition of Gaussian noise interference.

Image processing	Generation model	Recovery accuracy
Histogram equalization	DCGAN	0.862
	WGAN-GP	0.896
Sharpening	DCGAN	0.915
	WGAN-GP	0.956

Table 2: Testing error (P_E) of the processed stego image generated from different image processing generation model against SRM and ensemble classifier.

Image processing	Algorithm	Payload (bpp)					
		0.05	0.1	0.2	0.3	0.4	0.5
Histogram equalization	HILL	0.0165	0.0018	0.0008	0.0001	0	0
	DCGAN1	**0.4800**	**0.4791**	**0.4183**	**0.3995**	**0.3803**	**0.3590**
	WGAN-GP1	**0.4816**	**0.4698**	**0.4194**	**0.4039**	**0.3915**	**0.3765**
	S-UNIWARD	0.0140	0.0013	0.0002	0	0	0
	DCGAN2	**0.4482**	**0.4423**	**0.4105**	**0.3916**	**0.3877**	**0.3509**
	WGAN-GP2	**0.4601**	**0.4514**	**0.4109**	**0.4012**	**0.3890**	**0.3663**
	WOW	0.0095	0.0007	0.0001	0	0	0
	DCGAN3	**0.4383**	**0.4328**	**0.4039**	**0.3806**	**0.3777**	**0.3508**
	WGAN-GP3	**0.4444**	**0.4437**	**0.4192**	**0.3983**	**0.3796**	**0.3651**
Sharpening	HILL	0.2249	0.0837	0.0082	0.0014	0.0004	0.0002
	DCGAN1	**0.4869**	**0.4844**	**0.4187**	**0.4002**	**0.3918**	**0.3673**
	WGAN-GP1	**0.4952**	**0.4935**	**0.4467**	**0.4108**	**0.4003**	**0.3796**
	S-UNIWARD	0.1760	0.0434	0.0023	0.0003	0.0001	0.0001
	DCGAN2	**0.4496**	**0.4483**	**0.4127**	**0.4009**	**0.3925**	**0.3824**
	WGAN-GP2	**0.4698**	**0.4494**	**0.4327**	**0.4025**	**0.3988**	**0.3925**
	WOW	0.1740	0.0427	0.0028	0.0002	0	0
	DCGAN3	**0.4477**	**0.4424**	**0.4089**	**0.4015**	**0.3975**	**0.3912**
	WGAN-GP3	**0.4628**	**0.4622**	**0.4182**	**0.4021**	**0.3992**	**0.3962**

Table 3: Testing error (P_E) of the processed stego image generated from different image processing generation models against Ye-net.

Image processing	Algorithm	Payload (bpp)					
		0.05	0.1	0.2	0.3	0.4	0.5
Histogram equalization	HILL	0.0101	0.0009	0.0002	0.0002	0	0
	DCGAN1	**0.4290**	**0.4240**	**0.4024**	**0.3881**	**0.3640**	**0.3489**
	WGAN-GP1	**0.4427**	**0.4409**	**0.4388**	**0.3932**	**0.3881**	**0.3674**
	S-UNIWARD	0.0018	0.0006	0.0002	0	0	0
	DCGAN2	**0.4238**	**0.4186**	**0.4008**	**0.3889**	**0.3662**	**0.3497**
	WGAN-GP2	**0.4349**	**0.4343**	**0.4217**	**0.3924**	**0.3706**	**0.3515**
	WOW	0.0008	0.0007	0	0	0	0
	DCGAN3	**0.4116**	**0.4107**	**0.3872**	**0.3771**	**0.3654**	**0.3508**
	WGAN-GP3	**0.4345**	**0.4226**	**0.3918**	**0.3866**	**0.3694**	**0.3513**
Sharpening	HILL	0.1417	0.0132	0.0017	0.0006	0	0
	DCGAN1	**0.4351**	**0.4348**	**0.4104**	**0.3893**	**0.3689**	**0.3502**
	WGAN-GP1	**0.4501**	**0.4489**	**0.4376**	**0.3964**	**0.3883**	**0.3694**
	S-UNIWARD	0.1106	0.0105	0.0003	0	0	0
	DCGAN2	**0.4308**	**0.4296**	**0.4011**	**0.3998**	**0.3803**	**0.3617**
	WGAN-GP2	**0.4494**	**0.4363**	**0.4111**	**0.4034**	**0.3968**	**0.3708**
	WOW	0.0979	0.0093	0.0018	0	0	0
	DCGAN3	**0.4500**	**0.4127**	**0.4001**	**0.3802**	**0.3743**	**0.3529**
	WGAN-GP3	**0.4589**	**0.4311**	**0.4109**	**0.3956**	**0.3892**	**0.3705**

generated by the image processing generation model is higher than that of the processed stego image generated by traditional steganographic methods by comparing Table 4 with Tables 2 and 3. However, the undetectability of the processed image generated is still relatively poor. This problem has also been raised in [21]. The main reason is that there is a certain degree of distortion in the image generated by GAN model; that is, there is a certain gap between the image generated and the original input image. However, in the practical application, we can take the method of keeping the processed stego image set

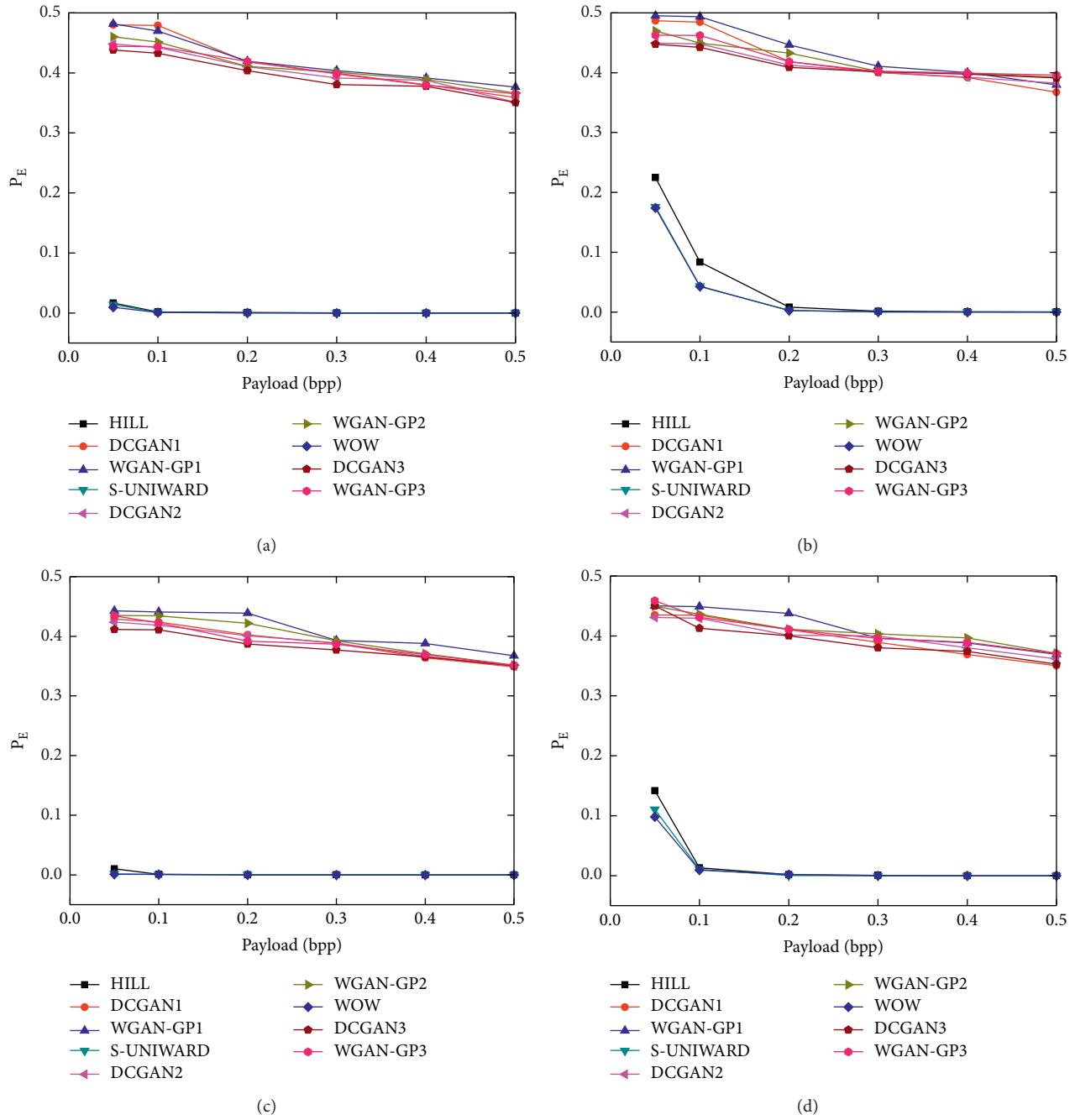

Figure 7: Testing error rate comparison curve of the processed stego image generated from different image processing generation models with different steganalyzers. (a) Testing histogram equalization stego images with SRM. (b) Testing sharpening stego images with SRM. (c) Testing histogram equalization stego images with Ye-net. (d) Testing sharpening stego images with Ye-net.

generated confidential to ensure the security of the processed stego image generated by the image processing generation model proposed in this paper.

4.5. Steganographic Capacity. In the experiment, the dimension of the input random noise vector is 100; therefore, the steganographic capacity of each processed stego image generated is $100 \times \sigma$, according to the mapping rules. We know that the current steganographic capacity of SWE is smaller than that of embedded steganography [21]. The proposed generation based steganography is a type of SWE, and the steganographic capacity is relatively small. However, since the size of the processed stego image generated is smaller, the relative steganographic capacity (steganographic capacity/size of the stego image) is larger.

4.6. Further Scope. Since our secret information extraction model is based on convolutional neural networks instead of

Table 4: The testing error (P_E) of the processed stego image generated from different image processing generation models with different steganalyzers.

Image processing	Generation model	Steganalyzer	
		SRM	Ye-net
Histogram equalization	DCGAN	0.0342	0.0226
	WGAN-GP	0.0519	0.0337
Sharpening	DCGAN	0.0407	0.0336
	WGAN-GP	0.0634	0.0518

artificially designed extraction rules, our proposed steganographic method cannot recover 100% of secret information due to the loss of convolutional neural network training. Reference [21] has the same problem. In the future, we may try to combine Reed-Solomon error-correction codes to further improve the accuracy of secret information extraction. In addition, since our steganographic method is based on GAN, the processed stego images generated by proposed steganographic method still have a certain degree of distortion. In the future, we will try different methods to generate more realistic processed stego images.

5. Conclusion

This paper proposes a new steganographic method, that is, embedding secret information in the process of image histogram equalization and sharpening processing through the trained histogram equalization generation model and sharpening generation model. Among them, the image processing generation models can generate high-quality processed images through random noise; then, we try to generate stego images directly from noise mapped by secret information during image processing. And the secret information can be extracted through the extraction model. To prove the security of the steganographic scheme proposed in this paper, we use the current state-of-the-art steganalyzer to detect the processed stego images generated. Experimental results prove that the steganographic method has better security performance to resist detection by state-of-the-art steganalyzer.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] B. Jana, D. Giri, S. K. Mondal, and P. Pal, "Image steganography based on cellular automata," *International Journal of Pure and Applied Mathematics*, vol. 83, no. 5, pp. 701–715, 2013.

[2] P. Chowdhuri, B. Jana, and D. Giri, "Secured steganographic scheme for highly compressed color image using weighted matrix through DCT," *International Journal of Computers and Applications*, vol. 43, no. 1, pp. 38–49, 2021.

[3] J. Wu, B. Chen, W. Luo, and Y. Fang, "Audio steganography based on iterative adversarial attacks against convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2282–2294, 2020.

[4] S. Jiang, D. Ye, J. Huang, Y. Shang, and Z. Zheng, "Smart-Steganogaphy: light-weight generative audio steganography model for smart embedding application," *Journal of Network and Computer Applications*, vol. 165, Article ID 102689, 2020.

[5] A. Banerjee and B. Jana, "A secure high-capacity video steganography using bit plane slicing through (7, 4) hamming code," *Advanced Computational and Communication Paradigms*, Springer, New York, NY, USA, pp. 85–98, 2018.

[6] Y. Yao and N. Yu, "Motion vector modification distortion analysis-based payload allocation for video steganography," *Journal of Visual Communication and Image Representation*, vol. 74, Article ID 102986, 2021.

[7] L Bin, .W Ming, .L Xiaolong, .T Shunquan, and .H Jiwu, "A strategy of clustering modification directions in spatial image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1905–1917, 2015.

[8] .J Tao, .S Li, .X Zhang, and .Z Wang, "Towards robust image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 594–600, 2018.

[9] .S Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 9, Article ID 10437, 2018.

[10] W. Lu, R. Li, L. Zeng, H. Chen, and Q. Yun, "Binary image steganalysis based on histogram of structuring elements," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 3081–3094, 2019.

[11] Z. Wang, Z. Qian, X. Zhang, M. Yang, and D. Ye, "On improving distortion functions for JPEG steganography," *IEEE Access*, vol. 6, Article ID 74917, 2018.

[12] Z. Wang, X. Zhang, and Z. Qian, "Practical cover selection for steganography," *IEEE Signal Processing Letters*, vol. 27, pp. 71–75, 2019.

[13] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function forspatial image steganography," in *Proceedings of the Image Processing (ICIP) 2014 IEEE International Conference on 2014*, pp. 4206–4210, IEEE, Paris, France, Octobar,2014.

[14] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[15] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS) 2012*, pp. 234–239, IEEE, Costa Adeje - Tenerife, Spain, December 2012.

[16] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2015.

[17] D. Volkhonskiy, B. Nazarov, and Burnaev, "Steganographic generative adversarial networks," 2017, https://arxiv.org/abs/1703.05502.

[18] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[19] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE*

Transactions on Information Forensics and Security, vol. 14, no. 8, pp. 2074–2087, 2019.

[20] C. Li, X. Sun, Z. Zhou, and Y. Yang, "Real-time image carrier generation based on generative adversarial network and fast object detection," *Journal of Real-Time Image Processing*, vol. 17, no. 3, pp. 655–665, 2020.

[21] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38303–38314, 2018.

[22] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[23] V. Holub and J. Fridrich, "Random projections of residuals for digital image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1996–2006, 2013.

[24] T. Denemark, V. Sedighi, V. Holub, .R Cogranne, and .J Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS) 2014*, pp. 48–53, IEEE, Atlanta, GA, USA, December, 2014.

[25] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2011.

[26] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Proceedings of the Media Watermarking, Security, and Forensics 2015*, vol. 9409, February 2015, Article ID 94090.

[27] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[28] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[29] Z. L. Zhou, Y. Cao, and X. M. Sun, "Coverless information hiding based on bag-of-words model of image," *Journal of Applied Sciences*, vol. 34, no. 5, pp. 527–536, 2016.

[30] S. Zheng, L. Wang, B. Ling, and D. Hu, "Coverless information hiding based on robust image hashing," in *Proceedings of the International Conference on Intelligent Computing*, pp. 536–547, Springer, Liverpool, England, August.

[31] Z. Zhang, G. Fu, R. Ni, J. Liu, and X. Yang, "A generative method for steganography by cover synthesis with auxiliary semantics," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 516–527, 2020.

[32] I. Goodfellow, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, December 2014.

[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," https://arxiv.org/abs/1511.06434.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," https://arxiv.org/abs/1701.07875.

[35] I. Gulrajani, F. Ahmed, and M. Arjovsky, "Improved training of wasserstein gans," *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017, https://scholar.google.co.in/citations?user=mZfgLA4AAAAJ&hl=en&oi=sra.

[36] Y. Zhang, .W Zhang, and .N Yu, "Adversarial examples against deep neural network based steganalysis," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, Innsbruck, Austria, June, 2018.

[37] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[38] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.