IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 22, 2018, accepted January 2, 2019, date of publication January 7, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891247

Reversible Image Steganography Scheme Based on a U-Net Structure

XINTAO DUAN[1], KAI JIA[1], BAOXIA LI[1], DAIDOU GUO[1], EN ZHANG[1], AND CHUAN QIN[2]

[1]College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
[2]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Xintao Duan (duanxintao@126.com)

This paper was supported in part by the National Natural Science Foundation of China under Grant 61672354, in part by the Key Programs for Science and Technology Development of Henan Province under Grant 172102210335, and in part by the Key Scientific Research Projects in Henan Universities under Grant 16A520058.

ABSTRACT Traditional steganography methods often hide secret data by establishing a mapping relationship between secret data and a cover image or directly in a noisy area, but has a low embedding capacity. Based on the thought of deep learning, in this paper, we propose a new image steganography scheme based on a U-Net structure. First, in the form of paired training, the trained deep neural network includes a hiding network and an extraction network; then, the sender uses the hiding network to embed the secret image into another full-size image without any modification and sends it to the receiver. Finally, the receiver uses the extraction network to reconstruct the secret image and original cover image correctly. The experimental results show that the proposed scheme compresses and distributes the information of the embedded secret image into all available bits in the cover image, which not only solves the obvious visual cues problem, but also increases the embedding capacity.

INDEX TERMS Information security, reversible image steganography, deep learning, U-Net structure.

I. INTRODUCTION

In today's increasingly globalized era, cloud computing pro-vides individuals and organizations with enough online space to store multimedia data (e.g., documents, videos, and images) and to provide people a convenient way with access and data sharing over the network [1]. Since these multimedia data may contain private, valuable or even confidential information, preventing such important information from being dis-closed is an important and urgent issue for individuals and organizations. There are usually two common methods: data hiding and encryption to protect image content from leaks. Data hiding technology embeds information into carriers such as images, audios or videos, which not only protects the contents of secret files, but also hides the communication process itself, so as to be as free from attacks as possible [2].

Information hiding is to hide secret information in a host signal in an invisible way, and extract secret information when needed to achieve covert communication and copyright protection [3], [4]. It is mainly used for secret communication between specific parties, especially in fast-growing social networks, with rich images and videos as carriers, which provides more opportunities and challenges for information hiding. Carpentieri *et al.* [4] proposed a reversible information hiding and compression scheme for hyperspectral images. Carpentieri said that the proposed scheme represents the first one-pass frame designed specifically for hyper-spectral images. It can perform lossless data hiding and lossless compression of marker streams by leveraging the capabilities of the predictive paradigm. In terms of application scenarios, this work will play an important role in military applications and forensic science. However, the challenge of information hiding arises mainly because embedded secret messages can change the look and under-lying statistics of the cover image. The amount of change depends on two factors: first, the amount of information to be hidden. A common use is to hide text messages in images. The amount of hidden information is measured in units of bits per pixel (bpp). Usually, the amount of information is set to 0.4 bpp or less. The longer the message, the larger the bpp, so the cover image changes more [5], [6]. Second, the amount of change depends on the cover image itself. Hiding information in noisy, high-frequency filled image areas produces less human detectable perturbations than hiding in flat areas. In [7], it can be found in [7] how much information can be hidden by the cover image. Currently, common used image information hiding methods mainly include information hiding methods of spatial domain and transform domain. The spatial domain hiding method is the least significant bit (LSB) hiding method and the adaptive LSB hiding method [8], [9]. Such a method can embed the same payload as the LSB matching, but less modification for the cover image. Later, a spatial-universal wavelet relative distortion (S-UNIWARD) method [10] improves the LSB method and chooses to embed more information in the noisy or complex texture region of the cover image. Pevny *et al.* [11] proposed the highly undetectable steganography (HUGO) method, which is a new embedding algorithm for spatial domain digital images. The main design principle is to minimize the distortion function, which is defined based on the weighted sum of the difference between the feature vector extracted from the cover image and the stego image in the feature space of the subtracted pixel adjacency matrix (SPAM) [12]. Subsequently, Holub *et al.* proposed the wavelet obtained weights (WOW) method [13], which embeds the payload into the cover image while obeying the more complex rules of the image region texture, the more pixel values that will be modified in the region. The above methods all embed the secret information directly into the image pixel values. However, the spatial domain-based algorithm has little effect on the quality of the cover image and has a large embedding capacity. But, it usually has poor robustness. To improve robustness, the researchers proposed to embed secret information in the transform domain. Transform domain methods such as discrete Fourier transform (DFT) hiding method [14], discrete cosine transform (DCT) hiding method [15], discrete wavelet transform (DWT) hiding method [16] and so on. The traditional information hiding method is to embed the secret information by modifying the cover image. The stego image always leaves traces of modification, which makes it difficult for the cover image containing a secret information to fundamentally resist the detection of the statistical-based information hiding analysis algorithm.

Recently, reversible image steganography has attracted great attention from researchers because it can reconstruct the original version of the main image without loss after image steganography [1]–[4], [17]. In [17], a new reversible image steganography based on rhombus prediction and local complexity has been proposed by Nguyen *et al.* Nguyen said that the proposed scheme is divided into two steps: the first step is to evaluate the local complexity of each pixel to ensure the quality of stego image and achieve high-precision tamper detection. The second step calculates the prediction error by embedding the authentication code using rhombus prediction. Nguyen's proposed scheme is superior to previous scheme in terms of tamper detection and image quality. In view of the recent excellent results obtained by combining deep neural networks with steganalysis [18]–[21], there are relatively few attempts to incorporate neural networks into the hidden process itself [22]–[25]. Some of these researchers used deep neural networks (DNN) to use the binary representation of text messages in the image to select which LSBs were replaced. Some researchers use DNN to determine which bits are extracted in the steganographic cover image. In contrast, in our work, we used neural networks to implicitly simulate the distribution of natural images and embed larger information (full-size images) into the cover image compared to previous studies. By using a cover image of N×N×RGB pixels to completely hide the secret image of N×N×RGB pixels, instead of simply modifying the bits, and having the smallest distortion rate for the cover image (each color channel is 8 bits). The DNN determines where the secret information is placed and how it is effectively encoded, and the hidden messages are scattered among the bits in the cover image. Different from the deep steganography scheme [26], our method eliminates the cover image preprocessing process, the encoder network is directly used to encode the secret image into the cover image, and the simultaneously trained decoder network is used to extract the secret images present in the stego image. The network is only trained once during this process. To summarize, the major contributions of our work as below:

– Unlike [26], we only use two networks to achieve a hidden effect, a hiding network and an extraction network. For hiding networks, a U-Net structured convolutional neural network is used to achieve this goal. The cover image and the secret image are concatenated into a 6-channels tensor as an input to the hiding network.

– For the extraction network, there are 6 convolutional layers with a convolution kernel size of 3×3, except that the last layer uses the Sigmoid activation function, followed by a Batch Normalization (BN) layer and a ReLU activation layer. A stego image generated by a hiding network is used directly as an input to the extraction network.

– The hiding network and the extraction network are a full convolutional neural network. The input and output are images, and there is no fully connected layer. The shallower high resolution layer is used to solve the problem of pixel positioning for determining the image position of the cover image; the deeper layer is used to solve the problem of pixel classification, and is used to determine the position at which the secret image pixel can be encoded into the cover image. And extract the secret image and the cover image in the extraction network.

– The transmitted stego image is a real and meaningful image. Unlike texture or noise, it does not provide visual cues to the attacker, which greatly reduces the possibility of being attacked. However, given the large amount of hidden information, we prefer to find an acceptable compromise in the cover image hiding capacity and the secret image: the embedding rate is increased while reducing the distortion rate of the cover image and the secret image.

The rest of this paper is organized as follows: Section II presents the related work on image steganography and encoder and decoder in GAN and U-Net. Section III describes the proposed method. Section IV presents the experimental results and analysis. Conclusions are presented in Section V.

II. RELATED WORKS

A. IMAGE STEGANOGRAPHY BASED ON DEEP NEURAL NETWORK

Image steganography based on deep neural networks [26], which creatively uses neural networks to determine where to embed secret information in an image, rather than artificially modifying the LSB accordingly. This deep model contains three subnetworks: Pre-Network, Hiding Network and Reveal Network. Among them, Pre-Network preprocesses the secret image. There are two main functions: first, since the size of the secret image may be smaller than the cover image, Pre-Network will distribute the bits of the original M×M secret image to N×N (cover image size) pixels. The second is to convert the color-based pixel points of the original image into more valuable features that facilitate encoding the image, such as edge information in the image. The Hiding Network is the main structure of the method. The role of the network is to take the output of the Pre-Network and the cover image as input. Then, the output of the network is a container image (described as a stego image). The input size of the network is N×N, and the depth is the number of transformed feature channels of the RGB 3-channels plus the previously extracted secret image. Finally, the Reveal Network is used by the receiver of the image, it acts as a decoder, inputting the stego image by the Hiding Network, and outputting the recovered secret image. The network structure is shown in Fig. 1.

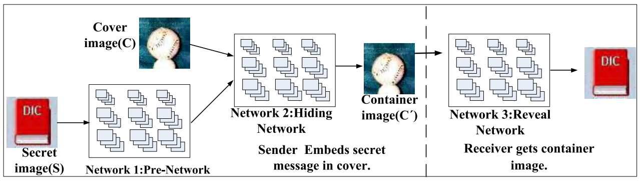


FIGURE 1. Image steganography frame based on deep neural network.

The goal of [26] is to hide an N×N×RGB size secret image into the same size of the carrier, and the disturbance to the carrier is as small as possible. In this way, the restriction that the previous secret information must be reconstructed with-out loss is relaxed, and there is a compromise between acceptable stego image quality and restored secret image quality. The existing steganography analysis method can detect stego image with a steganography rate as low as 0.1 bpp. The steganography rate in this paper is ten times or even 40 times higher than the previous method. Although visually difficult to detect, due to the large amount of hidden information, the probability of detecting stego image generated by this method is certainly not small. However, with the advantage of deep network, this network uses the convolutional neural network to hide the secret image directly into the carrier for the first time, which proposes a new way for the development of image steganography.

B. ENCODER AND DECODER IN GAN

GAN is a generation model proposed by Goodfellow *et al.* [27] in 2014. The idea comes from the two-person zero-sum game in game theory. GAN is mainly composed of a generator and a discriminator. Any differentiable function can be used to represent the GAN generator (G) and the discriminator (D) [28]. BEGAN [29] made further improvements to GAN. Based on the U-Net [30] model, a new generator and discriminator network was designed. The network uses the decoder as the generator G and the encoder as the discriminator D. In addition, the network uses the advantages of the residual network to initialize the network with the missing residuals, and for successively the same size layers, the layer inputs are combined with their outputs. Skip connection [31] were also introduced to aid gradient propagation [32]. After each upsampling step, the output is cascaded up to the same dimension as h0. This creates a layer of skip connection between the hidden state and each successive upsampling layer of the decoder, making network transmissions smoother. The overall idea is similar to U-Net [30]. In addition, a new way to evaluate the quality of the generator is proposed, so that even if the GAN uses a very simple network, without some training skills, it can achieve good training effects without worrying about model collapse and the problem of unbalanced training.

BEGAN's main work has the following aspects: First, a new simple and powerful GAN network structure is proposed, which can be quickly and stably converged using standard training methods without training techniques. Second, a balanced concept is proposed for the balance of G and D capabilities in GAN. The third is to provide a hyperparameter that balances the diversity of images and the quality of the generation. Finally, an estimate of the degree of convergence is proposed. This mechanism has only appeared in WGAN. The network structure is shown in Fig. 2.

C. APPLICATION AND DEVELOPMENT OF U-NET

U-Net [30], as its name suggests, which is named after its network structure is similar to the u-type. The early stages of development are mainly used for tasks such as medical image segmentation and semantic segmentation. It mainly uses the encode and decode methods to make the underlying information and high-level information merge. The network structure is shown in Fig. 3.

The blue arrows represent the convolution and activation functions, the grey arrows represent the copy, the red arrows represent the downsampling, the green arrows represent the upsampling and then the convolution, and conv 1×1 represents the 1×1 convolution operation. It can be seen that this network is not fully connected, only convolution and sampling. This is also an end-to-end network where the input is an image and the output is an image. The network on the left is the contracting path: downsampling using convolution and maxpooling. The network on the right is an expansive path: using upsampling combined with the feature

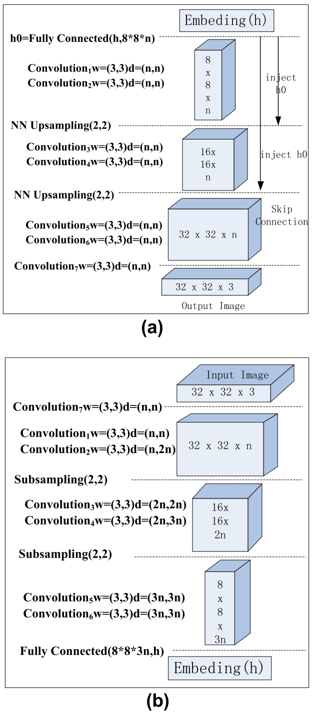

Embeding(h)
h0=Fully Connected(h,8*8*n)
Convolution$_1$w=(3,3)d=(n,n)
Convolution$_2$w=(3,3)d=(n,n)
8 x 8 x n
inject h0
NN Upsampling(2,2)
Convolution$_3$w=(3,3)d=(n,n)
Convolution$_4$w=(3,3)d=(n,n)
16x 16x n
inject h0
NN Upsampling(2,2)
Skip Connection
Convolution$_5$w=(3,3)d=(n,n)
Convolution$_6$w=(3,3)d=(n,n)
32 x 32 x n
Convolution$_7$w=(3,3)d=(n,n)
32 x 32 x 3
Output Image
(a)
Input Image
32 x 32 x 3
Convolution$_7$w=(3,3)d=(n,n)
Convolution$_1$w=(3,3)d=(n,n)
Convolution$_2$w=(3,3)d=(n,2n)
32 x 32 x n
Subsampling(2,2)
Convolution$_3$w=(3,3)d=(2n,2n)
Convolution$_4$w=(3,3)d=(2n,3n)
16x 16x 2n
Subsampling(2,2)
Convolution$_5$w=(3,3)d=(3n,3n)
Convolution$_6$w=(3,3)d=(3n,3n)
8 x 8 x 3n
Fully Connected(8*8*3n,h)
Embeding(h)
(b)

FIGURE 2. Structure diagram of BEGAN. (a) Generator/Decoder. (b) Encoder.

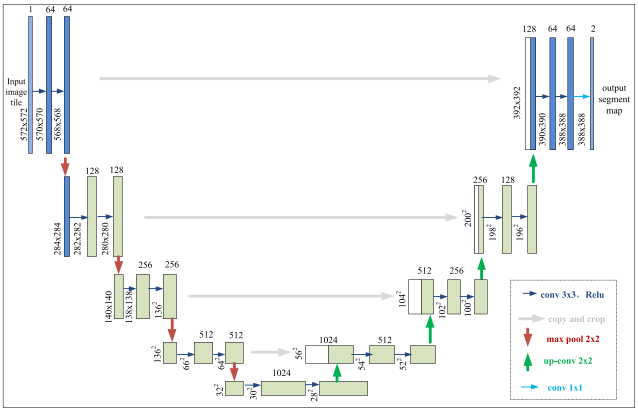


FIGURE 3. Structure diagram of original U-Net.

map of the pooling layer on the left contracting path, and then upsampling to a 392×392 size heatmap. Finally, after two convolutions, the final heatmap is reached. And then use a convolution kernel 1×1 convolution to classify, here is divided into two categories, so use two neurons to do the convolution operation, get the last two heatmap. For example, the first heatmap represents the score of the first category (i.e., each pixel corresponds to the first category has a score), the second heatmap represents the score of the second category, and then the input of the Softmax activation function, the probability is calculated. Compare the large softmax class and select it as input to the cross entropy for backpropagation training. The network includes 4 upsamplings and 4 downsamplings. The convolution kernel size is 3×3. max-pooling is used during pooling and the location information is preserved so that the location information can be restored while upsampling. This kind of network design can use a small amount of data sets for training tests and make a great contribution to medical image segmentation.

The design idea of this paper is similar to that of BEGAN and U-Net. The hiding network and extraction network of this paper also adopt the idea of decoder and encoder. The encoder network uses U-Net network structure directly to encode the secret image into the cover image, the simultaneously trained decoder network is used to extract the secret images present in the stego image. The specific process is described in the part III.

III. PROPOSED IMAGE STEGANOGRAPHY SCHEME

A. OVERALL DESIGN IDEAS

As illustrated in Fig. 4, the proposed steganography framework is different from many popular steganographic methods for encoding secret messages in the LSB of the cover image and the coverless information hiding method. Our method compresses and distributes secret images on all available bits on the cover image. We try different network structures to avoid the existence of secret information in steganalysis. Finally, the hiding network and the extraction network of this paper also adopt the idea of decoder and encoder. The specific functions are as follows:

Hiding network: The encoder network is used directly to encode the secret image into the cover image. Using a U-Net structured convolutional neural network, the cover image and the secret image are concatenated into a 6-channels tensor as an input to the hiding network.

Extraction network: The trained decoder network is used to extract the secret image existing in the stego images. The network has 6 convolutional layers with a convolution kernel size of 3×3, except that the last layer uses the Sigmoid activation function [33], and each layer is followed by a BN Layer and ReLU activation layer. A stego image generated by a hiding network is used directly as an input to the extraction network.

B. HIDING NETWORK STRUCTURE

As illustrated in Fig. 5, the specific network architecture parameter settings are similar with the U-Net network structure, the hiding network in this paper has a contraction phase and an expansion phase. The contraction phase is a typical convolutional neural network structure. At this time, unlike the U-Net network, the input of the network is 256×−256 cascading 6-channels feature tensor, which is completed by a 4×4 convolution layer in each downsampling process. Each convolution is followed by a LeakyReLU activation function and BN operation to speed up network training. In each downsampling step, we double the number of feature channels after convolution. After seven downsampling operations, the number of feature channels is 512 and

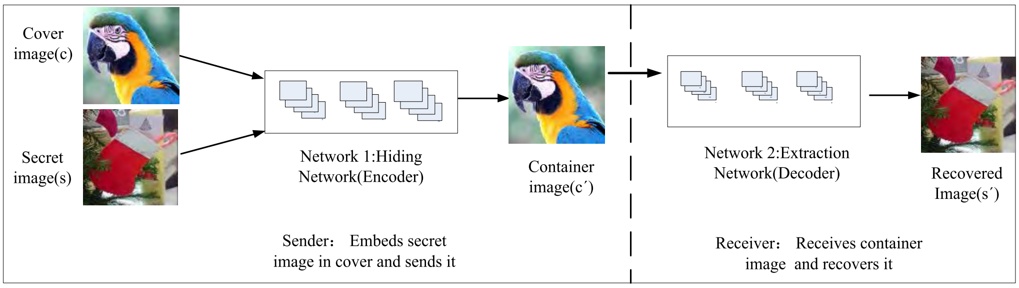
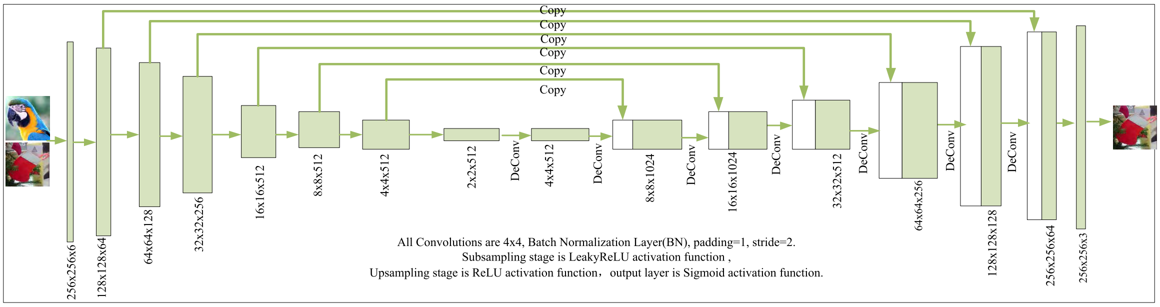


FIGURE 4. Flow chart of the proposed scheme.



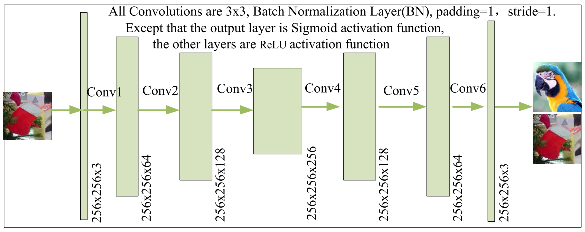
All Convolutions are 4x4, Batch Normalization Layer(BN), padding=1, stride=2.
Subsampling stage is LeakyReLU activation function ,
Upsampling stage is ReLU activation function，output layer is Sigmoid activation function.

FIGURE 5. Hiding network architecture diagram.

the size of the feature map is 2×2. In the expansion phase, the feature map is upsampled using a deconvolution layer (DeConv), which is also 4×4 in size and halve the number of feature channels. At this time, each upsampling operation is cascaded with the feature map from the contraction phase, so that the network learns the feature maps of different stages. In each upsampling process, a 4×4 convolutional layer is used, each convolution followed by a ReLU activation function and BN operation for accelerated network training. At the last level of the network, the 4×4 convolution is used to compress the convolved 64 feature vectors into a 3-channels feature map and the Sigmoid activation function to compute an output [33], which is the hidden stego image. It can be seen that the hiding network is a full convolutional neural network. The input and output are images, and there is no fully connected layer. The shallower high-resolution layer is used to solve the problem of pixel positioning, and the deeper layer is used to solve the problem of pixel classification. It is worth noting that in this paper, in order to achieve the same size of the input and output, it is very important to select the size of the input image block, so that when the image is hidden, the input image is downsampled to a 2×2 feature map through the contraction path. Therefore, a 256×256 image is used as input during the training phase, and the corresponding output is a stego image containing secret information.

C. EXTRACTION NETWORK STRUCTURE

As illustrated in Fig. 6, we designed a Convolutional Neural Network Architecture (CNN) to recover secret images from stego images generated by hiding networks, called extraction networks. We studied the architecture of the extraction net-work to accurately recover information from hiding networks. Unlike the hiding network structure, the designed network is a "plain network" with six convolution layers. In CNN, the dropout operation, activation function and pooling layer are used to enhance the nonlinear learning ability of the neural network. The purpose of CNN is to use nonlinear features to learn the fitting parameters. The weight parameters in each layer of the network are learned to fit the mapping between input and output. If the effects of these nonlinear operations are ignored, the effect of CNN is similar to the effect of linear multivariate equations. With this in mind, we designed the filter size for each convolutional layer to be 3×3, with each convolution layer followed by a ReLU activation function and BN operation without using the pooling layer and dropout operation. At the last level of the network, each 64-components feature vector is mapped to the desired number of categories using a 3×3 convolution, and the Sigmoid activation function [33] is used to compute the two outputs, which are secret image and cover image.



All Convolutions are 3x3, Batch Normalization Layer(BN), padding=1， stride=1.
Except that the output layer is Sigmoid activation function,
the other layers are ReLU activation function

FIGURE 6. Extraction network architecture diagram.

D. LOSS FUNCTION AND EVALUATION INDICATOR

In order to minimize the loss of the generated stego image c' and the original cover image c and the extracted secret image s' and the original secret image s, the paper forces hiding networks and extraction networks to continually optimize learning, ultimately minimizing reconstruction errors. The model loses the cost of loss between the stego image c' obtained by the reconstruction and the original cover image c and the extracted secret image s' and the original secret image s. The parameter $\Theta = \{w_i, b_i\}$ is continuously adjusted by backpropagation, for a set of real images X_j and the network reconstructed image $F_j(Y;\Theta)$, this paper uses mean squared error (MSE) as the cost function:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \| F^j(Y;\Theta) - X_j \|^2 \tag{1}$$

where n represents the number of training samples. Complete network training by minimizing the loss of equation (2):

$$L(c, c', s, s') = \| c - c' \| + \alpha \| s - s' \| \tag{2}$$

where c and s are the cover and the secret image, respectively, and α is the trade-off error. $\| c-c' \|$ and $\| s-s' \|$ are the cost of hiding network and extraction network, respectively. Here, the weight of the error term $\| c-c' \|$ of the hiding network is not shared with the weight of the extraction network, and the weight of the error term $\| s - s' \|$ is shared between the two networks. This ensures that the two networks adjust the network training by receiving this error term to minimize the error loss of the hiding network reconstructed secret image and the cover image, and to ensure that the information of the secret image is completely encoded on the cover image. The network uses the Adam optimization method and the back propagation algorithm [34] to minimize the MSE to adjust the parameters of the network. The network weight update process is:

$$\Delta_{k+1} = 0.9 \times \Delta_k - \eta \times \frac{\partial L}{\partial W_k^l}, \quad W_{k+1}^l = W_k^l + \Delta_{k+1}, \tag{3}$$

where Δ_k represents the last weight update value, l represents the number of layers in the network, and k represents the number of iterations of the network. η is the learning rate. W_k^lrepresents the weight of the k th iteration of the l th layer. $\partial L/\partial W_k^l$ represents the partial bias of the corresponding weight in the cost function. The weights are randomly initialized by a Gaussian distribution with a mean of 0 and a variance of 0.001. The model can automatically adjust the learning rate within the determined range during the training process, making the parameter learning relatively stable.

In this paper, two common evaluation indicators, peak signal to noise ratio (PSNR) [35] and structural similarity (SSIM) [36], are used to measure the difference between the image quality and the original image after reversible steganography.

As an objective measure of image quality, PSNR evaluates image quality by calculating the error between corresponding pixels. The PSNR unit is dB, and the larger the value, the smaller the image distortion. Calculated by using:

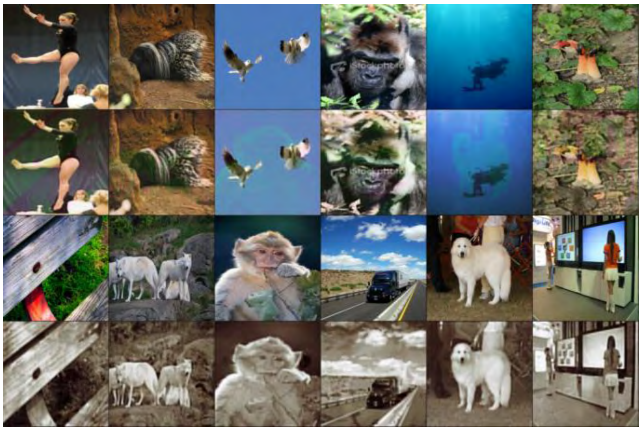


FIGURE 7. The model hides the effect in the middle of training.

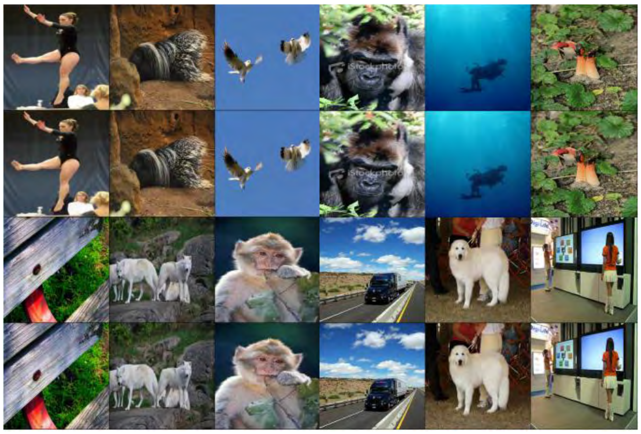


FIGURE 8. The model hide results after stable training.

$$PSNR = 10\log_{10}(\frac{(2^n-1)^2}{MSE}) \tag{4}$$

where MSE is the mean square error of the original image and the evaluated image, $(2^n - 1)^2$ is the square of the maximum value of the signal, and n is the number of bits of each sample value.

SSIM measures image similarity in three ways: brightness, contrast, and structure. The range of SSIM value is [0, 1]. The closer the SSIM value to 1, the smaller the distortion effect is. Calculated by using:

$$SSIM(X, Y) = l(X, Y) \cdot c(X, Y) \cdot s(X, Y) \tag{5}$$

$$l(X, Y) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \tag{6}$$

$$c(X, Y) = \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \tag{7}$$

$$s(X, Y) = \frac{2\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3} \tag{8}$$

among them: in this paper, X is represented as a stego image and an extracted secret image, respectively; Y is the original cover image and secret image, respectively; μ_X and μ_Y

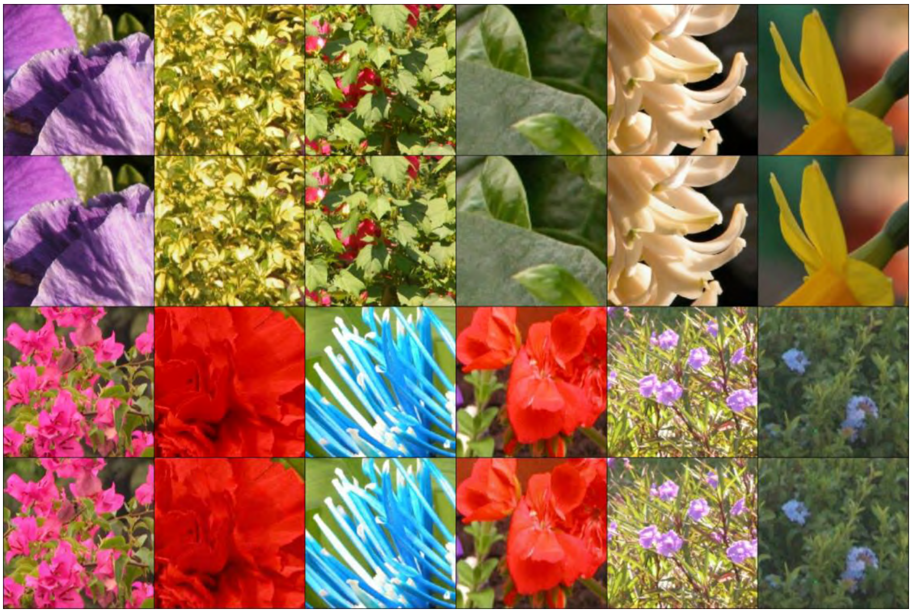


FIGURE 9. The model hide results on natural images after stable training.

represent the mean of the images X and Y, respectively; σ_X *and* σ_Y represent the standard deviation of the stego image and the cover image, the reversible extracted secret image and the original secret image; $\sigma_X\sigma_Y$ denotes the covariance of the stego image and the cover image, the reversible extracted secret image and the original secret image.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND DATASET

In this paper, 45,000 images for training and 5000 images for testing were collected as training set training network models from ImageNet. The initial learning rate of the network is set to 0.001, and the hyperparameter α is set to 0.75. The Adam optimization method is used to automatically adjust the learning rate so that the network parameters can be learned smoothly. The number of images per batch is set to 16, and the network trains 200 iterations. In the GPU is NVIDIA GeForce 1080 Ti, the experimental environment is Pytorch, and the application is Python 3.5 for simulation experiments. The training results of this model will verify the practicability of the proposed method from two aspects: subjective steganography result and hidden capacity.

B. SUBJECTIVE STEGANOGRAPHY RESULT

In view of the large amount of hidden information, we will experimentally prove that the method is to place supplementary information in the cover image instead of simply modifying the LSB. As illustrated in Fig. 7 and Fig. 8, in order to more intuitively compare the model to hide the image changes during the training process, we show the effect of the intermediate process of hiding and extracting under different iterations of the model. Fig. 7 and Fig. 8 list the results of hiding 6 images. From top to bottom, the first row is the cover image, the second row is the stego image (container), the third row is the secret image that needs to be hidden, and the fourth row is the extracted secret image. In contrast, in Fig. 7, in the early stage of training of the model, the loss of MSE is very large, and the hidden secret image outline information is clearly visible on the cover image, and the pixel loss is visually unacceptable. While Fig. 8 is in the model training stabilization phase, the MSE loss is minimized. After the cover image encodes the secret image, although all the secret image information is encoded and the cover image is reconstructed from the stego image, most of the reconstructed cover image looks almost identical to the original cover image, the visual effects change very little, and it is difficult to distinguish whether there are traces of modification, which are barely noticeable to the human eye. Similarly, after the secret image is decoded from the stego image, the visual change is very small, and the image distortion is almost imperceptible. As shown in Fig. 9, this paper selects natural images besides the ImageNet training set for testing. The steganography and extraction effects are still stable, and it is still difficult to distinguish whether there are traces of modification.

To further illustrate the effect of steganography, observe the difference between the cover image and the secret image before and after steganography. As illustrated in Fig. 10, shows the hidden results of this model. As illustrated in Fig. 10 (a, b, e, f), shows the cover image and stego image and their error histogram. Fig. 10 (c, d, g, h), shows the original secret image and the secret image extracted from the stego image and their error histogram. It can be found that the overall trend of the high and low frequency information of the image hardly changes, and there is almost no large pixel error. In addition, as shown in Table 1,

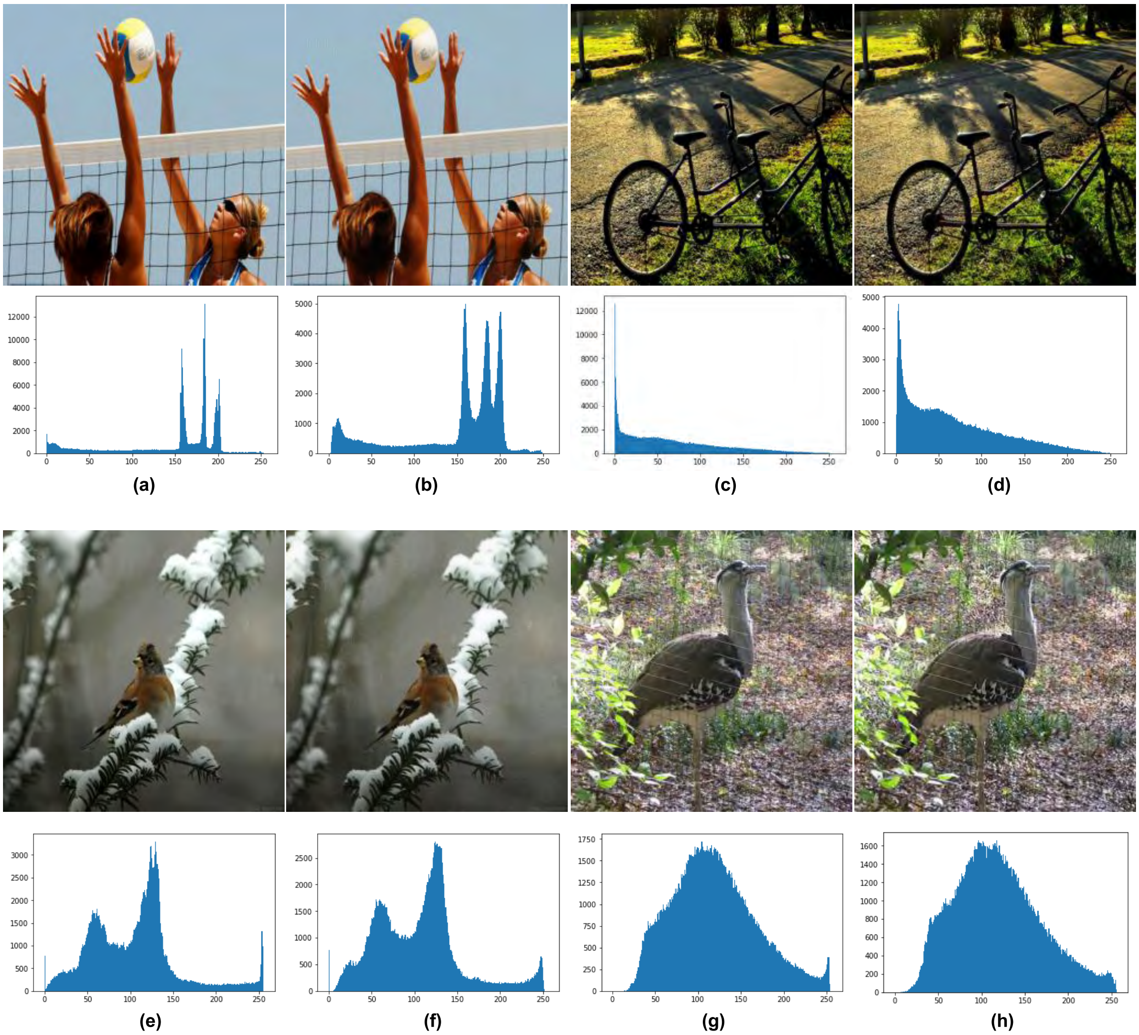


FIGURE 10. The difference between cover image and secret image before and after steganography. (a) Original cover image and its histogram. (b) Stego image and its histogram. (c) Original secret image and its histogram. (d) Extracted secret image and its histogram. (e) Original cover image and its histogram. (f) Stego image and its histogram. (g) Original secret image and its histogram. (h) Extracted secret image and its histogram.

in addition to visually verifying the four pairs of images listed in Fig. 10, we randomly select one thousand samples from the ImageNet dataset, including the cover image and the secret image before and after steganography. The PSNR and SSIM indicators were used to further analyze the degree of change of each image before and after steganography. From the results, it can be found that the PSNR and SSIM of the four pairs of images listed in Fig. 10 are very high, reached (39.9837/0.9728), (34.8328/0.9814), (40.2027/0.9668), (36.9231/0.9871), respectively. Under the ImageNet dataset, the average of PSNR and SSIM for the cover image reached (40.4716/0.9794), the average of PSNR and SSIM for the secret image reached (40.6665/0.9842). It is worth noting that the method is to place supplementary information in each pixel in the cover image instead of directly modifying the pixel value, and thus is visually difficult to find.

C. STEGANOGRAPHY CAPACITY

Different from many popular steganography methods for encoding secret messages in the LSB of the cover image and coverless information hiding method, Our method compresses and distributes secret images on all available bits on the cover image. At present, traditional embedding steganography and based on coverless information hiding of steganography capability are relatively low. Since our method is a new embedding hiding method, in order to make a more intuitive comparison, in this paper we simply compare the steganography with other most advanced embedding steganography methods and based on the non-embedding hiding method. There are non-embedding hiding methods including the cover-selection-based and the cover-synthesis-based methods. The comparison results are shown in Table 2, where the second column is the absolute steganography capacity (steganography capacity per image), the third column

TABLE 1. Comparison of PSNR(DB) and SSIM values of cover image and secret image after hinding and extraction process in our scheme.

Compare Image		Cover Image	Secret Image
Fig.10 (a,b)	PSNR	39.9837	--
	SSIM	0.9728	--
Fig.10 (c,d)	PSNR	--	34.8328
	SSIM	--	0.9814
Fig.10 (e,f)	PSNR	40.2027	--
	SSIM	0.9668	--
Fig.10 (g,h)	PSNR	--	36.9231
	SSIM	--	0.9871
ImageNet	PSNR(average)	40.4716	40.6665
	SSIM(average)	0.9794	0.9842

(Note: '--' in each row in Table 1 indicates null. For example, Fig.10 (a,b) is the cover image, which corresponds to the result is (39.9837/.9728) in the third column of the Cover Image column. Therefore, Fig.10 (a,b) has a null result in the fourth column of the Secret Image column.)

is the size of the stego image, and the last column is the relative steganography capacity (steganography capacity per pixel):

$$Relative\ capacity = \frac{Absolute\ capacity}{The\ size\ of\ the\ image} \tag{9}$$

TABLE 2. Comparisons of steganography capacities.

Schemes	Absolute capacity (bytes/image)	Image size	Relative capacity (bytes/pixel)
[37]	1.125	512×512	4.29e-6
[38]	3.72	≥512×512	1.42e-5
[39]	2.25	512×512	8.58e-6
[40]	25~100	480×640	8.14e-5~3.26e-4
[41]	64×64	800×800	6.40e-3
[42]	1535~4300	1024×1024	1.46e-3~4.10e-3
Ours	256×256	256×256	1

Since our method compresses and distributes secret image pixel information on all available bits of the stego image, the relative capacity is 1 byte/pixel and the result is shown in the last row of Table 2. Rows 1-3 show the steganography capacity of the cover-selection-based methods. Obviously, the relative capacity of our method is much larger than methods of the cover-selection-based. Rows 4-6 show the steganography ability of the cover-synthesis-based methods. Although the relative capacity of cover-synthesis-based methods is significantly improved compared to the relative capacity of the cover-selection-based methods, it is still lower than the steganography capacity of our method. In other words, our approach is superior to the most advanced cover-selection-based and cover-synthesis-based methods in terms of steganography capacity.

V. CONCLUSIONS

This paper discards the information embedding of the least significant bits of the image, but uses an end-to-end approach to hide one image onto another and has lower pixel distortion. Experimental results show that this method has significant advantages in both visual effect and steganography capacity. The next step in this paper will combine the process of image delivery with the generative adversarial networks, taking the form of passing image parameters to the receiver. The receiver extracts the transmitted secret image through the pre-trained model, and in the form of double encryption, ensures that the secret message cannot be detected by the attacker during the transmission process, and the information is secure.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] C. Qin, P. Ji, C.-C. Chang, J. Dong, and X. Sun, "Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery," *IEEE MultimediaMag.*, vol. 25, no. 3, pp. 36–48, Jul./Sep. 2018, doi: 10.1109/MMUL.2018.112142509.

[2] C. Qin, C.-C. Chang, and Y.-P. Chiu, "A novel joint data-hiding and compression scheme based on SMVQ and image inpainting," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 969–978, Mar. 2014.

[3] C. Qin, W. Zhang, F. Cao, X. P. Zhang, and C.-C. Chang, "Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Process.*, vol. 153, pp. 109–122, Dec. 2018.

[4] B. Carpentieri, A. Castiglione, A. De Santis, F. Palmieri, and R. Pizzolante, "One-pass lossless data hiding and compression of remote sensing data," *Future Gener. Comput. Syst.*, vol. 90, pp. 222–239, Jan. 2019.

[5] J. Fridrich and M. Goljan, "Practical steganalysis of digital images: State of the art," *Proc. SPIE*, vol. 4675, p. 465263, Apr. 2002.

[6] H. Ozer, I. Avcibas, B. Sankur, and N. D. Memon, "Steganalysis of audio based on audio quality metrics," *Proc. SPIE*, vol. 5020, pp. 55–66, Jun. 2003.

[7] F. Yaghmaee and M. Jamzad, "Estimating watermarking capacity in gray scale images based on image complexity," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, p. 851920, 2010.

[8] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[9] C.-H. Yang, C.-Y. Weng, and S.-J. Wang, H.-M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 488–497, Sep. 2008.

[10] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, pp. 1–13, 2014.

[11] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. 12th Int. Workshop Inf. Hiding*, in Lecture Notes in Computer Science, Calgary, AB, Canada, vol. 6387, R. Böhme and R. Safavi-Naini, Eds. New York, NY, USA: Springer-Verlag, Jun. 2010, pp. 161–177.

[12] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2013, pp. 234–239.

[13] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[14] J. J. K. O. Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proc. Int. Conf. Image Process.*, 1996, pp. 239–242.

[15] I. J. Cox, J. Kilian, and F. T. Leighton, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 2010.

[16] W.-H. Lin, S.-J. Horng, T.-W. Kao, P. Fan, C.-L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantiZation," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 746–757, Aug. 2008.

[17] T.-S. Nguyen, C.-C. Chang, and T.-H. Shih, "Effective reversible image steganography based on rhombus prediction and local complexity," *Multimedia Tools, Appl.*, vol. 77, no. 14, pp. 26449–26467, 2018.

[18] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Proc. SPIE*, vol. 9409, p. 94090J, Mar. 2015.

[19] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set α-positive region reduction," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[20] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont. (2015). "Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch." [Online]. Available: https://arxiv.org/abs/1511.04855

[21] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.

[22] S. Husien and H. Badi, "Artificial neural network for steganography," *Neural Comput. Appl.*, vol. 26, no. 1, pp. 111–116, 2015.

[23] I. Khan, B. Verma, V. K. Chaudhari, and I. Khan, "Neural network based steganography algorithm for still images," in *Proc. Int. Conf. Emerg. Trends Robot. Commun. Technol. (INTERACT)*, 2010, pp. 46–51.

[24] V. Kavitha and K. S. Easwarakumar, "Neural based steganography," in *PRICAI: Trends in Artificial Intelligence* (Lecture Notes in Computer Science), vol 3157, C. Zhang, H. W. Guesgen, W. K. Yeap, Eds. Berlin, Germany: Springer, 2004.

[25] A. S. Brandao and D. C. Jorge, "Artificial neural networks applied to image steganography," *IEEE Latin Amer. Trans.*, vol. 14, no. 3, pp. 1361–1366, Mar. 2016.

[26] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. NIPS*, 2017, pp. 2069–2079.

[27] I. J. Goodfellow *et al.*, "Generative adversarial networks," *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 2672–2680, 2014.

[28] K.-F. Wang, C. Gou, Y.-J. Duan, Y.-L. Lin, X.-H. Zheng, and F.-Y. Wang, "Generative adversarial networks: The state of the art and beyond," *Acta Automatica Sinica*, vol. 43, no. 3, pp. 321–332, 2017.

[29] D. Berthelot, T. Schumm, and L. Metz. (2017). "BEGAN: Boundary equilibrium generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1703.10717

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. 2015, pp. 234–241.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[33] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proc. Int. Conf. (DBLP)*, 2008, pp. 536–543.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[35] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.

[36] Y. Ye, J. Shan, L. Bruzzone, and L. Shen, "Robust registration of multimodal remote sensing images based on structural similarity," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2941–2958, May 2017.

[37] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," in *Proc. Int. Conf. Cloud Comput. Secur.*, 2015, pp. 123–132.

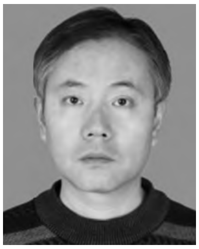
[38] Z.-L. Zhou, Y. Cao, and X.-M. Sun, "Coverless information hiding based on bag-of-words model of image," *J. Appl. Sci.*, vol. 34, no. 5, pp. 527–536, 2016.

[39] S. Zheng, L. Wang, B. Ling, and D. Hu, "Coverless information hiding based on robust image hashing," in *Intelligent Computing Methodologies*. Cham, Switzerland: Springer, 2017, pp. 536–547, doi: 10.1007/978-3-319-63315-2_47.

[40] H. Otori and S. Kuriyama, "Texture synthesis for mobile data communications," *IEEE Comput. Graph. Appl.*, vol. 29, no. 6, pp. 74–81, Nov./Dec. 2009.

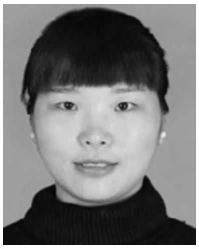
[41] J. Xu *et al.*, "Hidden message in a deformation-based texture," *Vis. Comput. Int. J. Comput. Graph.*, vol. 31, no. 12, pp. 1653–1669, 2015.

[42] K. C. Wu and C. M. Wang, "Steganography using reversible texture synthesis," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 130–139, Jan. 2015.

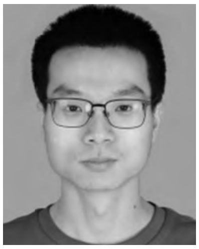


XINTAO DUAN received the Ph.D. degree from Shanghai University, Shanghai, China, in 2011. He is currently an Associate Professor with the College of Computer and Information Engineering, Henan Normal University. His major research interests include image processing, deep learning, and information security.

KAI JIA received the B.S. degree from Pingdingshan University, China, in 2016. He is currently pursuing the M.S. degree with the College of Computer and Information Engineering, Henan Normal University. His research interests include image processing, deep learning, and image steganography.



BAOXIA LI received the B.S. degree from the College of Computer and Information Engineering, Henan Normal University, China, in 2017, where she is currently pursuing the M.S. degree. Her research interest includes coverless information hiding.



DAIDOU GUO received the B.S. degree from the Henan Institute of Science and Technology, China, in 2017. He is currently pursuing the M.S. degree with the College of Computer and Information Engineering, Henan Normal University. His research interest includes coverless information hiding.

EN ZHANG received the Ph.D. degree from the Beijing University of Technology. He held a Postdoctoral position with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with the College of Computer and Information Engineering, Henan Normal University, China. His research interests include outsourcing computation, secure multiparty computation, and rational cryptography.

CHUAN QIN received the B.S. degree in electronic engineering and the M.S. degree in signal and information processing from the Hefei University of Technology, Anhui, China, in 2002 and 2005, respectively, and the Ph.D. degree in signal and information processing from Shanghai University, Shanghai, China, in 2008. Since 2008, he has been with the Faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where he is currently a Professor. He was with Feng Chia University, Taiwan, as a Postdoctoral Researcher, from 2010 to 2012. His research interests include image processing and multimedia security. He has published more than 110 papers in these research areas.