IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 4, 2020, accepted October 20, 2020, date of publication October 26, 2020, date of current version November 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033895

StegoPNet: Image Steganography With Generalization Ability Based on Pyramid Pooling Module

XINTAO DUAN[1], WENXIN WANG[1], NAO LIU[1], DONGLI YUE[1], ZIMEI XIE[1], AND CHUAN QIN[2], (Member, IEEE)

[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@htu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672354, and in part by the Key Scientific Research Project of Henan Provincial Higher Education under Grant 19B510005 and Grant 20B413004.

ABSTRACT In terms of payload capacity and visual effects, the existing image steganography technology based on deep neural networks still needs improvement, to solve this problem, this article proposes a new deep convolutional steganography network based on the pyramid pooling module to achieve better image steganography. The deep convolutional neural network itself can extract features efficiently. Based on the combination of up-sampling structure, we added a pyramid pool module, under the premise of ensuring safety, fully integrated the previous important global features, achieved good hiding and extraction effects, fully integrated the previous important global features, and effective it reduces the loss of contextual information between different sub-regions in the feature extraction process and achieves better hiding and extraction effects under the premise of ensuring security. Experiments show that the average peak signal-to-noise ratio (PSNR)/structure similarity (SSIM) and other indicators between the images obtained by this method have achieved good results in the experiment. Also, we have verified through ablation experiments that the pyramid pooling module can enhance the steganography effect of the network model and can further cut down the loss function of the model.

INDEX TERMS Image steganography, deep neural network, pyramid pooling module.

I. INTRODUCTION

IN the era of globalization, Internet communication has become a convenient way of communication between people. With the growing scale of the Internet, the amount of data transmitted by network communication has also increased sharply, which contains valuable and even confidential information of individuals or organizations. Therefore, how to ensure communication security has become an important issue. People usually choose two methods, information hiding or encryption, to protect their communication security to prevent secret information from being leaked. Such technologies usually use images, audio, or video as the carrier of messages. In addition, some of these technologies also require the ability to hide the fact that the communication behavior itself exists, so as to further protect the communication from external attacks.

In the information hiding technology, the two parties to the communication are the sender and the receiver respectively. The sender embeds the secret information in the carrier in an invisible way and transmits it to the receiver of the communication through a common channel. After the receiver receives the secret carrier, it then uses the specified method to extract the secret information. The famous ''Prisoner Problem'' [1] describes the information hiding model very well. Assuming that prisoners Alice and Bob are held in different cells, the two communicate to plan an escape, but the guard Wendy always monitors the communication process between the two, trying to prevent the two prisoners from breaking out. Among them, Alice and Bob can be regarded as the sender and receiver of the information, Wendy can be regarded as the supervisor, with the authority to read, modify and block the communication information between Alice and Bob. From the perspective of carrier selection, the digital image on the Internet is characterized by a huge amount and greater redundancy, which brings huge use of space to information hiding technology. For example, in traditional image steganography technology, typical image Steganography methods include Least Significant Bit Matching (LSBM), in order to minimize the impact of the embedding operation on the carrier,

The associate editor coordinating the review of this manuscript and approving it for publication was Ziyan Wu.

Filler *et al.* [2] designed an adaptive steganographic lattice code quantization method(STC), and based on this, built a minimum embedding distortion steganography algorithm [3]–[5] proposed a HUGO algorithm based on the principle of minimizing embedded distortion, innovatively applying STC adaptive coding to image adaptive steganography. Holub *et al.* [6] propose the WOW(Wavelet Obtained Weights) algorithm that can better resist a rich model than HUGO. On this basis, Pevný *et al.* [7] put forward a distortion that can be applied in multiple domains (spatial domain, JPEG domain, JPEG domain with side information) calculation method (UNIWARD). But no matter what kind of information hiding method will cause the visual effect of the cover image and the change of mathematical statistics due to the embedded secret message, and this change usually depends on the amount of hidden information and the cover image itself. The indicator of the amount of hidden information is usually bits per pixel, which is bpp. The bpp will increase with the increase of hidden messages, and the corresponding cover image will change as well [8]. For the cover image, the embedding area is more likely to be selected in the high-frequency area, that is, the texture area because the visual change of the cover image will be relatively smaller.

In recent years, the application of deep neural networks has spread across many fields, its powerful feature extraction and feature representation capabilities have enabled it to achieve impressive results in various fields. For example, in the field of verification code recognition, Wang *et al.* [9] used the DenseNet model and adopted cross-layer connections to improve the recognition accuracy while reducing the problem of gradient disappearance and reducing the number of parameters; Chen *et al.* [10] based on the deep learning method, through the intermediate layer of the pre-trained deep learning model to output the convolution results, combined with the positive mean vector method to establish a visual feature vector database, to achieve automatic image annotation. At the same time, image hiding based on deep neural networks has also appeared in recent years. Aiming at the huge difference between image information steganography for deep neural networks and traditional information steganography, the original artificially designed embedding algorithm is transformed into embedding networks with different structures and parameters obtained by deep learning. Correspondingly, the original artificially designed extraction algorithm transforms the embedded network of different structures and parameters obtained by deep learning. Artificial intelligence with the theme of deep learning infiltrates and develops image information steganography, and has achieved good results. These advantages benefit from the powerful feature learning and representation capabilities of deep neural networks. For example, Tang *et al.* [11] proposed an automatic steganographic distortion learning framework that uses GAN to calculate the embedding change probability of each pixel in the cover image, and Rehman *et al.* [12] proposed image steganography based on deep learning. The general architecture of the company uses the encoder-decoder idea to hide one image into another image, but only realizes the embedding of grayscale images into color images, which limits the hidden information capacity. Then [13] used a deep neural network to embed the full-size color secret image into the cover image of the same size, but the image quality of this method needs to be improved, and there are still obvious distortions in chroma and image content details. Zhu *et al.* [14] think that neural networks can be awakened by adversarial sample coding through training, so as to obtain the required information. This feature can be used to realize data hiding, and proposed a network framework that includes encoder and decoder, but only achieves the hiding of text messages does not realize the hiding of secret images. Yang *et al.* [15] proposed a safer steganography algorithm, based on adversarial training and including three component modules: generator, embedding simulator, and discriminator, this framework significantly improves security performance. Similarly, Hu *et al.* [16] pointed out that the traditional embedding-based steganography will inevitably leave traces of modification after embedding secret information into a cover image, it is threatened by more and more advanced steganalysis algorithms based on deep learning, but the embedding-free steganography (SWE) without modifying the cover image data seems to be able to overcome such detection problems, therefore, a new image SWE method is proposed, this method is based on a deep convolution to generate an adversarial network, has the characteristics of high accuracy of information extraction and strong anti-detection ability. In order to better resist third-party image decryption and steganalysis, Luo *et al.* [17] proposed a carrier-free information hiding method. This method can select the appropriate carrier according to the needs, and at the same time, combined with DCT to generate a hash sequence to achieve a better robust image steganography method. However, the problem with the above methods is that the amount of embedded secret information is small, and the steganography capacity still needs to be improved. So, we added a pyramid pooling module on the basis of the downsampling-upsampling network structure to achieve large-capacity embedding while ensuring the visual effect of the image and image steganography that is resistant to detection. Among them, the cover image used by our network and the secret image have the same size, both are D×H×W (where D is the number of channels, the color image $D = 3$, and H and W are the size of the image 256). The main contributions of our work are summarized as follows:

1) We propose a new deep convolutional neural network and realize image steganography with a steganography capacity of 1 byte/pixel and good generalization ability.

2) We added a pyramid pooling module to the hiding network and reveal network and achieved better hiding and extraction effects.

3) We further reduced the loss function of the model during the training process by adding the pyramid pooling module.

The rest of this article will be arranged as follows: The second part will introduce the related work of steganography based on a deep convolutional neural network and the part

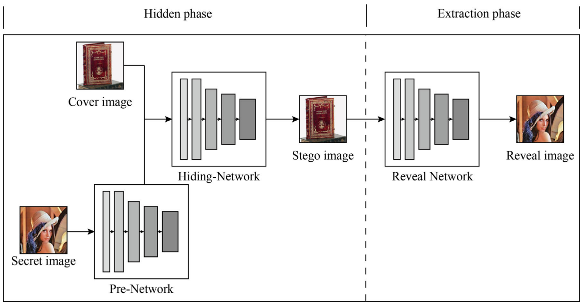


FIGURE 1. The depth image steganography framework proposed in [13].

of PSPNet related to our work. The third part introduces our method in detail. The fourth part is the experimental results and analysis. The fifth section is the conclusion.

II. RELATED WORK

Deep convolutional neural networks have achieved very successful applications in the field of steganography [11]–[16], [18], [19]. In this article, we propose a new deep convolution neural network to achieve image steganography, and in our steganography framework, add a pyramid pool module to the hiding network and the reveal network to achieve better results. Below we will briefly introduce the steganography framework based on deep convolutional neural networks [13] and the pyramid pooling module in PSPNet [20].

A. STEGANOGRAPHY FRAMEWORK

The deep convolutional neural network has become a research hotspot in many fields with its efficient knowledge expression ability [21], and [13] creatively uses the deep convolutional neural network to determine the position of the embedded secret message in the image. The model includes a hidden phase (including Pre-Network and Hiding-Network) and an extraction phase (including Reveal-Network): First of all, Pre-Network is responsible for preprocessing the secret image. The purpose of this step is to convert the original input secret image into higher-level and more valuable features to help the subsequent embedding operation; secondly, Hiding-Network cascades the output feature map of the cover image and the Pre-Network as input, and outputs the stego image containing the secret message, here Hiding-Network acts as an encoder; and finally, the Reveal-Network is used as an encoder to process the stego image to obtain the extracted secret image. The overall steganography network framework is shown in Figure 1.

The entire steganography process of the framework can be expressed as follows:

$$C' = G_{Hiding}(C, G_{Pre}(S)) \tag{1}$$

$$S' = G_{Reveal}(C') \tag{2}$$

where C is the cover image, S is a secret image, C' is the stego image, S' is the extracted secret image, G_{Hiding} represents the hidden network, G_{Pre} represents the preprocessing network of the secret image, and G_{Reveal} represents the revealed network.

B. PYRAMID SCENE PARSING NETWORK(PSPNet)

The purpose of scene parsing based on semantic segmentation in the field of computer vision is to assign a category label to each pixel in the image, so as to achieve a complete understanding of the scene [22], [23]. The Pyramid Scene Parsing Network (PSPNet) [20] uses the pyramid pooling module to aggregate the characteristics of global context information in different regions and successfully achieves diversified scene parsing. This network will be used for the traditional dilated FCN of pixel prediction [24], [25] extend to specially designed global pyramid pooling. The network structure is shown in Figure 2. Given the input image (a) as the input of the network, the feature map of the last convolutional layer is obtained after the convolutional layer (b) is processed, and then the pyramid pooling module (c) is used to obtain different the sub-region representation includes operations such as pooling, convolution, and up-sampling. Then all the feature maps are cascaded and input into the subsequent convolutional layer to finally obtain the predicted value (d) of each pixel.

The size of the receptive field in a deep convolutional neural network has a lot to do with the degree to which the network combines contextual information, and can even be used as an estimate of the degree of combination, although [21] pointed out that the receptive field of ResNet is large enough to contain the entire input image, the empirical acceptance domain in the high-level is not as large as the theoretical domain [22], which causes the network to be insufficient in integrating the previous important global features. To avoid excessive loss of context information, [20] proposed a hierarchical global prior, called a pyramid pooling module, which is composed of channels of different sizes and can fully integrate the change information between different sub-regions. As shown in Figure 2(c), several features of different proportions are merged through channels of different sizes. The red color shows the global pooling used to generate a single bin output. The subsequent pyramid divides the feature map into different sub-regions according to different levels and finally forms a summary representation. If the level size of the pyramid is N, the 1 × 1 dimension represented by the context is used to represent 1/N of the original dimension. Then, perform an up-sampling operation on the status feature map to obtain a feature map of the same size (which is consistent with the original feature image). Finally, different features are cascaded to obtain the final pyramid pooling global feature.

In our work, we used the pyramid pooling module as an intermediate part of our steganography framework and modified the module to meet our needs. Besides, we also used the pyramid pooling module before and after the downsampling, and up-sampling processing and skip connection are implemented.

III. PROPOSED IMAGE STEGANOGRAPHY SCHEME

Next, we will introduce the overall framework and operating principles of StegoPNet in detail, including the details of the

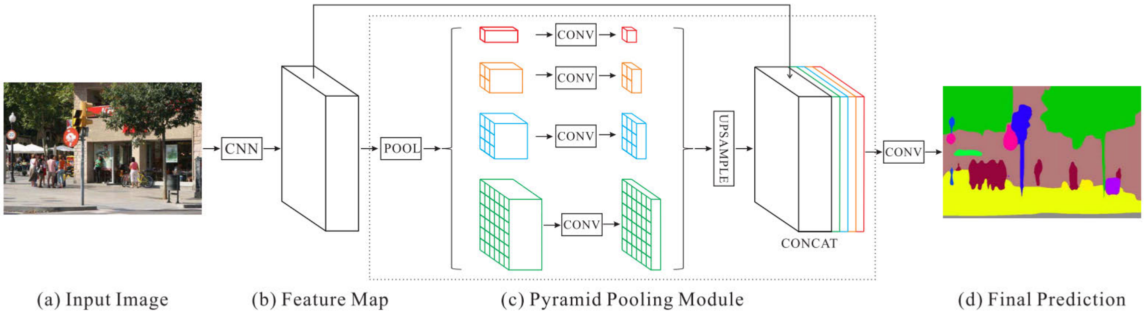


FIGURE 2. Pyramid scene parsing network [20].

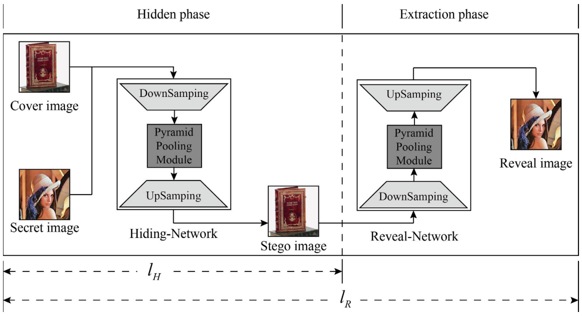


FIGURE 3. This article presents an image steganography framework. Our network model includes three stages: down-sampling, pyramid pooling module, and up-sampling. l_H is the loss between the stego image generated by the Hiding-network and the cover image, and l_R is the loss between the extracted secret image and the stego image.

hidden network and reveal network, as well as the internal structure of the pyramid pooling module we use, and the loss function used to constrain the network model training.

A. THE OVERALL STEGANOGRAPHY FRAMEWORK

As shown in Figure 3, our method adopts the idea of encoder and decoder, which includes a hidden phase and a reveal phase: the hidden network in the hidden phase is used by the sender to embed the secret image in the cover image and generate the stego image. The stego image is transmitted by the sender to the receiver to transfer the secret image; the reveal network in the reveal phase is used by the receiver to extract the secret image after receiving the stego image as the input of the reveal network. Different from the steganography framework shown in Figure 1, our method does not preprocess the secret image separately but directly concatenates the cover image and the secret image into a 6-channel feature map into the steganography framework. Our entire steganography process can be expressed as follows:

$$\begin{aligned} C' &= G_{Hiding}(C, S) \\ &= G_{H-Up}(G_{H-Pyramid}(G_{H-Down}(C, S))) \qquad (3) \\ S' &= G_{Reveal}(C') \\ &= G_{R-Up}(G_{R-Pyramid}(G_{R-Down}(C'))) \qquad (4) \end{aligned}$$

where C represents the cover image, S represents the secret image, C' represents the steganographic image, S' represents the extracted secret image, G_{Hiding} represents the hidden network, G_{Reveal} represents the reveal network, G_{Up} represents the up-sampling phase, $G_{Pyramid}$ represents the pyramid pooling module, G_{Down} represents the down-sampling phase, at the beginning of the G subscript we H and D are used to distinguish between hidden and extracted networks.

What we have to do is to force the stego image to be close to the original cover image and the extracted image to be close to the secret image. Therefore, we use the mean square error (MSE) loss function as a constraint in the network training process and use the backpropagation algorithm [30] to drive the network to continuously adjust the weight to complete the network training. Specifically, the mean square error and the global loss function are expressed as follows:

$$\begin{aligned} l_H &= \frac{1}{n}\sum_{i=1}^{n} \|G_{Hiding}(C, S) - C\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \|C' - C\|^2 \qquad (5) \\ l_R &= \frac{1}{n}\sum_{i=1}^{n} \|G_{Reveal}(C') - S\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \|S' - S\|^2 \qquad (6) \\ l_{Sum} &= l_H + \alpha \cdot l_R \qquad (7) \end{aligned}$$

Among them, C, C', S, S' stands for a cover image, stego image, secret image, and extract image n is the number of samples, L_{Sum} is the total loss of the network model, L_H is the hiding network loss, L_R is the reveal network loss, and α is the hyperparameter.

B. PYRAMID POOLING MODULE

Inspired by the work of [20], we try to use the pyramid pooling module to fully integrate global features in the network training process to reduce the loss of new context in the feature extraction process to achieve better steganographic effects, so the pyramid pooling The module was introduced into our steganographic network, and different from the number and size of levels in the original text, we have modified them to meet the requirements of steganography. Figure 4 is the pyramid pooling module added to the steganographic network after modification, see Table 1 for internal details.

C. HIDING NETWORK AND REVEAL NETWORK

As shown in Figure 5, our network architecture consists of three stages: downsampling-pyramid pooling-upsampling

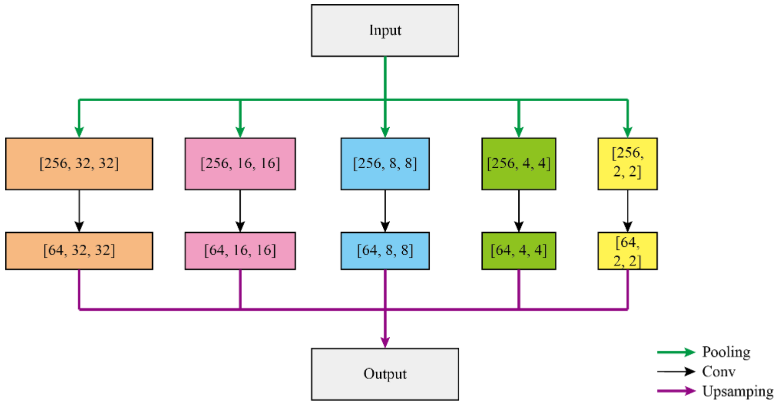


FIGURE 4. Pyramid pooling module. All [D, H, W] in this article respectively represent the number of channels and the size of the feature map. Specifically, we will modify the number of pyramid levels to 5, and the size of each level, that is, the feature size H×W, respectively, to 32 × 32, 16 × 16, 8 × 8, 4 × 4, and 2 × 2.

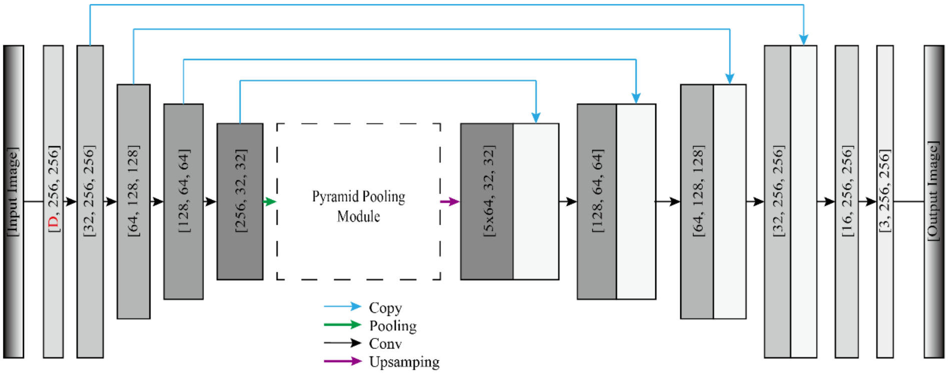


FIGURE 5. The detailed architecture of Hiding-Network and Reveal-Network. In the convolution of the up-sampling and down-sampling stages, the kernel size = 4, stride = 2, and padding = 1, all convolutional layers use ReLU, except the last layer is Sigmoid.

TABLE 1. Internal details of the pyramid pooling module.

			Pyramid Pooling Module		
Input			[256×32×32]		
AvgPool()	(32×32)	(16×16)	(8×8)	(4×4)	(2×2)
Conv()	(256,3×3,64)	(256,3×3,64)	(256,3×3,64)	(256,3×3,64)	(256,3×3,64)
BN+ReLU			BN+ReLU		
Upsample()	(64×64)	(64×64)	(64×64)	(64×64)	(64×64)
Output			[320×32×32]		

and uses skip connections between down-sampling, and up-sampling, which is similar to the overall architecture of U-Net [26], The specific processing process of the three stages of the hidden network is as follows:

1). First, the cover image and the secret image are cascaded to obtain a 6-channel feature map, which will be directly used as the input of the hidding network. After four down-sampling convolutional layer processing, a feature map with a size of 256 × 32 × 32 is obtained. In this stage, Conv-BatchNorm-ReLU (kernel size = 4, stride = 2, padding = 1) is included in the convolutional layer of each down-sample.

2). Then, the feature map obtained in step 1) will be input into the pyramid pooling module, and after 5 channels including Pooling-Conv-Upsamping are processed separately, 5 feature maps of different sizes are obtained. After all the 5 feature maps are cascaded, a feature map with a size of (5 × 64) × 32 × 32 is obtained and output to the up-sampling stage;

3). Finally, input the feature map output by the pyramid pooling module into the up-sampling stage, and each up-sampling includes DeConv-BatchNorm-ReLU (where kernelsize = 4, stride = 2, padding = 1), and the skip connection will be down The feature map of the corresponding size in the sampling stage is copied to the upsampling stage, as shown by the blue line in Figure 5. In this stage, the copied feature map and the output feature map of the previous layer are cascaded as the next up-sampling input. After 4 times of up-sampling, the final output size is 3 × 256×256 stego image after being processed by a conv layer containing Conv-BatchNorm-Sigmoid.

The architecture of our reveal network is roughly the same as that of the hiding network, see Table 2 for internal details.

TABLE 2. Internal details of hiding-network and reveal-network.

	Hiding-Network	Reveal-Network
Input image	Secret, Cover	Stego
Conv()+BN+ReLU	(6, 3×3, 32)	(3, 3×3, 32)
Conv()+BN+ReLU	(32, 3×3, 64)	(32, 3×3, 64)
Conv()+BN+ReLU	(64, 3×3, 128)	(64, 3×3, 128)
Conv()+BN+ReLU	(128, 3×3, 256)	(128, 3×3, 256)
-	Pyramid Pooling Module	
DeConv()+BN+ReLU	(576, 4×4, 128)	(576, 4×4, 128)
DeConv()+BN+ReLU	(256, 4×4, 64)	(256, 4×4, 64)
DeConv()+BN+ReLU	(128, 4×4, 32)	(128, 4×4, 32)
DeConv()+BN+ReLU	(64, 4×4, 16)	(64, 4×4, 16)
Conv()+BN+Sigmoid	(16, 3×3, 3)	(16, 3×3, 3)
Output image	Stego	Extract

The difference is that the input of the hiding network is the feature map with the number of channels D = 6 after the cover image and the secret image are cascaded, while the input of the reveal network is the number of channels D = 3 stego image.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this article, 50,000 images are collected from ImageNet to form a training set to train our network model, to ensure the generalization ability of the model, the cover images and secret images used for model training and testing are randomly selected from the data set. The network model training learning rate is initialized to 0.001, the value of the hyperparameter is initialized to 0.6, and the weight initialization is HeKaiming initialization [27]. The optimizer chooses Adam. Batchsize, that is, the number of images invested in each training is initialized to 16, and the network is trained for 150 iterations. GPU-NVIDIA GeForce 1080, the PyTorch version we use is 1.2.0, and the version of python is 3.6.5.

A. SUBJECTIVE ANALYSIS

In our method, one image is hidden into another image, so we are not simply modifying the cover image LSB. To prove this, we intuitively show the hiding and extraction of images at different stages of model training the effect is shown in Figure 6 and Figure 7, starting from the leftmost column are cover images, stego image, secret image, and extraction image. In Figure 6, due to the large loss function in the early stage of network training, whether it is a stego image or extracting a secret image, the pixel loss has a relatively obvious visual difference, such as brightness and chroma. In contrast to Figure 7, late loss when the function reaches the minimum, whether it is a cover image or a secret image, the visual gap between them is very small and there is almost no visual distortion. Next, we will further analyze the quality of image generation, and randomly select two sets of images in the test set to draw their histograms and compare them. As shown in Figure 8 the high-frequency and low-frequency areas of the histogram have little change, that is, the cover image and secret image have little change after network processing.

B. OBJECTIVE ANALYSIS

PSNR [28] is an objective standard for evaluating image quality. The smaller the compressed distortion, the higher the PSNR value. For image steganography, embedding the secret image into the cover image can be regarded as embedded noise. It is equivalent to reducing the quality of the cover image, therefore, PSNR can be used as a standard to evaluate

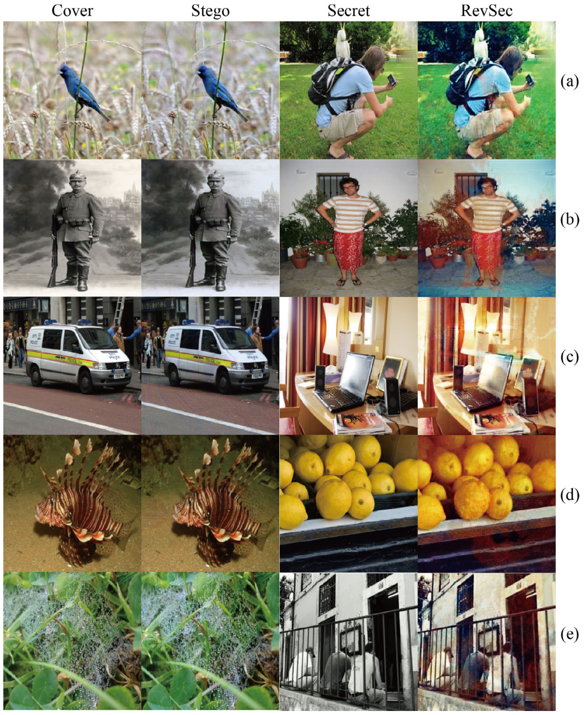


FIGURE 6. In the early stage of model training, the visual effects of steganographic images and extracted images are significantly different from the original images.

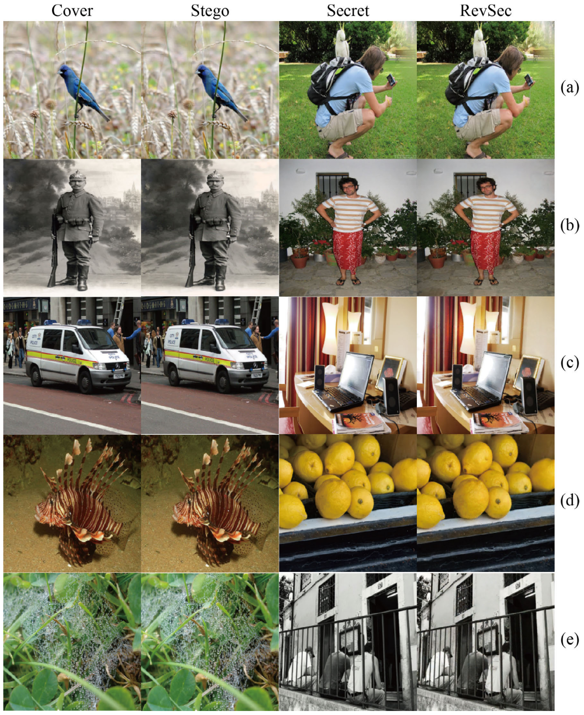


FIGURE 7. In the later stage of model training, when the model parameters almost converge.

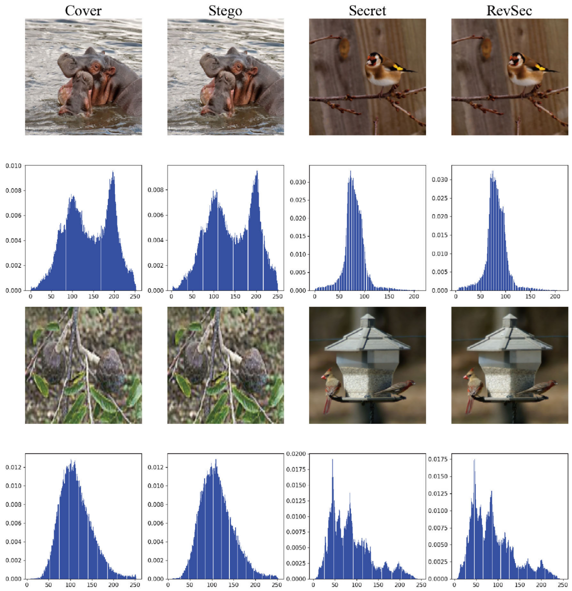


FIGURE 8. The changes in the cover image and the secret image before and after being processed by the steganographic model. Each image corresponds to the histogram below it.

the degree of distortion caused before and after the image is embedded in the secret message. The following is the calculation formula of root mean square MSE:

$$MSE = \frac{1}{WH}\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}\|C_{i,j} - C'_{i,j}\| \tag{8}$$

where, C represents the cover image, and C' represents the stego image, and both have the same size. The formula for calculating the peak signal-to-noise ratio is as follows:

$$PSNR = 10log_{10}(\frac{(2^n - 1)^2}{MSE}) \tag{9}$$

where n is the number of bits per pixel.

Natural images are usually highly structural, that is, there is a high correlation between them, and the human visual system (HSV) mainly obtains structural information from within the visible area, so it can be measured by the structural similarity SSIM [29] indicator Quantify the degree of image distortion. This indicator is measured based on the brightness, contrast, and structure between two samples. The larger the value, the better:

$$SSIM(x, y) = [l_{luminance}(x, y)^{\alpha} \cdot l_{contrast}(x, y)^{\beta} \cdot l_{structure}(x, y)^{\gamma}] \tag{10}$$

here we set both α, β, and γ to 1, and get:

$$\begin{aligned} SSIM(x, y) &= [l_{luminance}(x, y) \cdot l_{contrast}(x, y) \cdot l_{structure}(x, y)] \\ &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \end{aligned} \tag{11}$$

where μ and σ represent the average value and covariance of the variables.

In Table 3, our PSNR value and SSIM value are calculated using the cover image and stego image, secret image, and extracted image respectively. The PSNR and SSIM between the cover image and the stego image obtained by our method using ImageNet can reach 40.89dB/0.9962, and 39.06dB/0.9816 between the secret image and the extracted image. Among them, the test results for 2500 images in the test set showed that the average PSNR/SSIM of the cover image and stego image reached 40.48dB/0.9856 and the distance between the secret image and the extracted image reached 38.97dB/0.9850.

TABLE 3. For the comparison between the cover image and the secret image processed by our method, the indicators used here are PSNR and SSIM. The average values in the table are from a test set consisting of 2500 images randomly selected from ImageNet. The * in the fourth row represents the effect of the network model in the test set after removing the pyramid pooling module.

Image	*Cover-Stego (PSNR, SSIM)*	*Secret-RevSec (PSNR, SSIM)*
Fig.8	40.89, 0.9962	39.06, 0.9816
Fig.9	39.13, 0.9955	38.02, 0.9809
ImageNet Average	**40.48, 0.9856**	**38.97, 0.9850**
ImageNet Average*	39.46, 0.9731	37.53, 0.9703

TABLE 4. Payload capacity calculation result.

Fig 7	*Secret-ReSec(bpp)*
(a)	23.0571
(b)	23.1912
(c)	23.4916
(d)	23.7627
(e)	23.8271
ImageNet Average	23.5927

Besides, we also use formula (12) to calculate the effective load capacity of image information. The number of bytes of secret information contained in each pixel in the image is called the payload capacity. The specific calculation formula is as follows:

$$payload\ capacity = (1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}|S_{i,j} - S'_{i,j}|}{N \times M}) \times 8 \times 3(bpp) \tag{12}$$

where N, M represents the width and height of the image. The calculation results are in Table 4. This value comes from the five sets of secret images and extracted images in Figure 7.

C. CAPACITY ANALYSIS

The method we use is an information steganography method based on a deep neural network. Steganography ability is higher than traditional embedded algorithms because our method is to force the network to find suitable pixels on the cover image to hide the secret image by training the network. Here we use the relative capacity to calculate, the calculation method is as follows:

$$Relative\ capacity = \frac{Absolute\ capacity}{Image\ size} \tag{13}$$

It can be seen from Table 5 that our relative capacity is much larger than other methods, reaching 1 byte/pixel. Of course, this is also one of the main advantages of image steganography realized by deep neural networks.

TABLE 5. Payload capacity calculation result.

Method	*Absolute capacity (bytes/image)*	*Stego-image size*	*Relative capacity (bytes/pixel)*
[15]	26214~104857	512×512	1e-1~4e-1
[30]	1535~4300	1024×1024	1.46e-3~4.10e-3
[31]	18.3~135.4	64×64	1.49e-3~1.10e-2
Ours	256×256	256×256	1

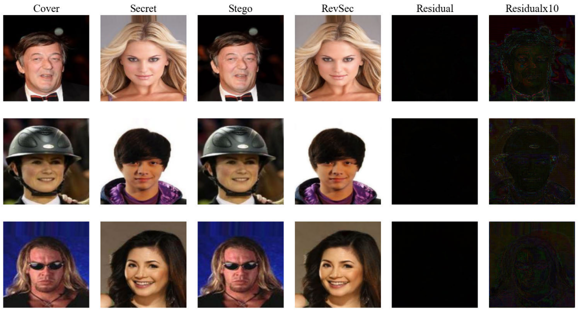


FIGURE 9. CelebA dataset.

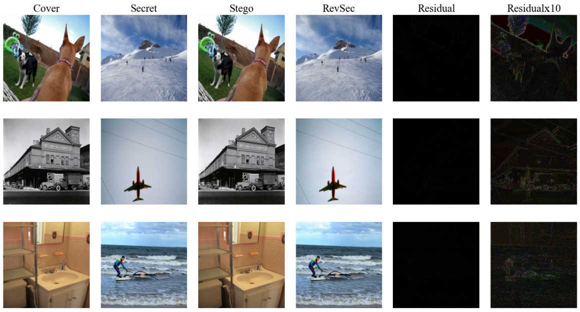


FIGURE 10. COCO dataset.

D. GENERALIZATION ABILITY ANALYSIS

Although our model training performed well in the test set, from the perspective of practical applications, the input of the model may be different types or even special images for a certain purpose in future use. In order to verify that our model can achieve the same good effect on different samples in different scenarios, we will select several image samples from different data sets to test the generalization ability of our model. As shown in Figure 9, Figure 10 shows images randomly selected from the CeleA and COCO data sets. Observing the residual images in the two columns on the right, even if the residual image is magnified 10 times, there is almost no trace of the secret image.

Besides, we used aerial images to test the model. As shown in Figure 11, the test results have good visual effects, proving that our model can be applied to other different scenarios.

E. ABLATION EXPERIMENT

To prove the effectiveness of the pyramid pooling module in improving the steganography effect in the model, we removed the pyramid pooling module and kept other hyperparameters, training data sets, and initialization methods unchanged, and retrained the model, and using the same test set to test the optimal model, the PSNR/SSIM results obtained are shown in the last row of Table 1. It is obvious that after removing the pyramid pooling module, all indicators have decreased.

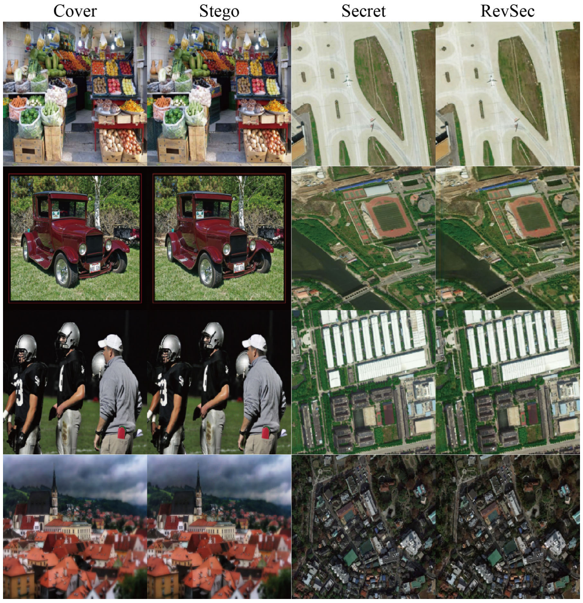


FIGURE 11. Aerial image.

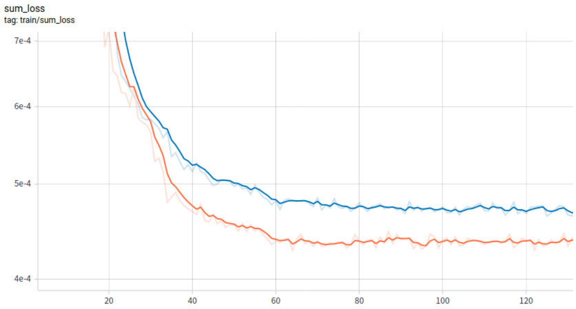


FIGURE 12. Loss change during model training.

TABLE 6. Loss function value of the network model.l_{Sum} is the steganography model proposed in this article, l_{Sum}* is the steganography model after removing the pyramid pooling module.

Epoch	l_{Sum}	l_{Sum}*
20	7.8497e-04	9.2564e-04
40	4.7537e-04	5.1515e-04
60	4.3778e-04	4.8223e-04
80	4.3683e-04	4.7509e-04

As shown in Figure 12, orange represents the model that contains the pyramid pooling module used in this article, and blue represents the model without the pyramid pooling module. During the model training process, the convergence effect of orange is better than that of blue. At the end of the training, the loss of orange is lower than that of blue, and even the optimal model is generated earlier (our optimal model was generated in the 65th epoch, And 104 epoch after removing the pyramid pooling module). The detailed loss function value is shown in Table 6.

F. STEGANALYSIS

StegExpose [32] is a relatively general steganalysis tool at present. It can use proven attack methods and analyze LSB steganography images in a timely and effective manner. We use StegExpose to detect our proposed steganography model, which includes four detection methods, chi-square

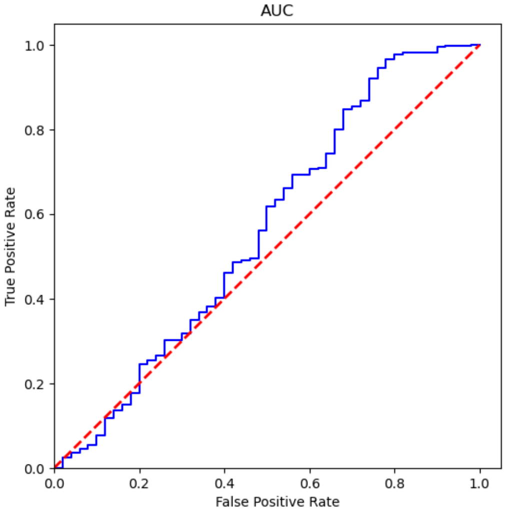


FIGURE 13. ROC Curves: Detect our proposed steganography through StegExpose. The test images include 50 unretouched natural images and 400 invisible images modified by our steganography method.

attack [33], RS analysis [34], sample pair analysis [35], and main set [36]. Figure 13 is the ROC curve, in which the horizontal axis is the false positive rate (representing images without hidden information classified as embedded images), the vertical axis is the true positive rate (representing the correct recognition of embedded images with hidden information), and the red dotted line represents random it is guessed that the blue color is drawn using this analysis tool, which is closer to the red dotted line (close to random guessing), which means that our model can resist StegExpose's steganalysis very well.

V. CONCLUSION

This article proposes an end-to-end network model that implements full-size color image steganography. We have added pyramid pooling modules and jump connections to the network model to fully integrate the important features needed, and further enhance the steganography effect on the premise of ensuring the steganography capacity and security. The experimental results show that our method has a good steganography effect, the maximum value of SSIM between the stego image and the cover image is 0.9962, and the SSIM between secret image and extracted image can reach 0.9816; higher steganographic capacity, relatively hidden capacity is 1 byte/pixel; with excellent generalization capabilities, users can easily use our network model to hide and extract images in other scenes, such as automatic hiding and extraction of military remote sensing images. In short, our method realizes image steganography with higher capacity and safer transmission and can be applied to various scenarios to transmit all kinds of secret messages needed. In future work, we will try to introduce an attention mechanism to guide the network model to better embed secret images during training to achieve safer and more efficient image steganography.

REFERENCES

[1] J. Simmons, "The Prisoners' problem and the subliminal channel," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 1983, pp. 51–67.

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

[3] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE*, vol. 6505, Jan. 2007, Art. no. 650502.

[4] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," in *Proc. Secur., Steganograp., Watermarking Multimedia Contents VII*, Jan. 2005, pp. 328–340.

[5] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Inf. Hiding 12th Int. Conf. (IH)*, Calgary, AB, Canada, Jun. 2010, pp. 161–177.

[6] 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.

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

[8] J. Fridrich and M. Goljan, "Practical steganalysis of digital images: State of the art," in *Proc. Secur. Watermarking Multimedia Contents IV*, San Jose, CA, USA, Jan. 2002, pp. 1–13.

[9] J. Wang, J. Qin, X. Xiang, Y. Tan, and N, Pan, "CAPTCHA recognition based on deep convolutional neural network," *Math. Bioences Eng.*, vol. 16, pp. 5851–5861, Jun. 2019.

[10] Y. Chen, L. Liu, J. Tao, X. Chen, R. Xia, Q. Zhang, J. Xiong, K. Yang, and J. Xie, "The image annotation algorithm using convolutional features from intermediate layer of deep learning," *Multimedia Tools Appl.*, vol. 2020, Sep. 2020, doi: 10.1007/s11042-020-09887-2.

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

[12] A. Rehman, R. Rahim, M. S. Nadeem, and S. Hussain, "End-to-end trained CNN encoder-decoder networks for image steganography," in *Proc. Comput. Vis. ECCV Workshops*, Munich, Germany, vol. 11132, Sep. 2018, pp. 723–729.

[13] S. Baluja, "Hiding Images in Plain Sight: Deep Steganography," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 2069–2079.

[14] J. Zhu, R. Kaplan, J. Johnson, and F. Li, "Hiding images in plain sight: Deep steganography," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 2069–2079.

[15] J. Yang, K. Liu, X. Kang, E. K. Wong, and Y.-Q. Shi, "Spatial image steganography based on generative adversarial network," 2018, *arXiv:1804.07939*. [Online]. Available: http://arxiv.org/abs/1804.07939

[16] 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, Aug. 2018.

[17] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *J. Real Time Image Process.*, vol. 17, no. 1, pp. 38303–38314, Sep. 2019.

[18] Z. Liao, J. Peng, Y. Chen, J. Zhang, and J. Wang, "A fast Q-learning based data storage optimization for low latency in data center networks," *IEEE Access*, vol. 8, pp. 90630–90639, Jun. 2020.

[19] B. Yin, X. Wei, J. Wang, N. Xiong, and K. Gu, "An industrial dynamic skyline based similarity joins for multidimensional big data applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2520–2532, Apr. 2020.

[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu HI, USA, Jul. 2017, pp. 6230–6239.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 7–9.

[23] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[25] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 2169–2178.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Munich, Germany, Oct. 2015, pp. 234–241.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[28] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Istanbul, Turkey, Aug. 2010, pp. 2366–2369.

[29] 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.

[30] K.-C. Wu and W. Chung-Ming, "Steganography using reversible texture synthesis," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 130–139, Jan. 2015.

[31] Z. Zhang, J. Liu, Y. Ke, Y. Lei, J. Li, M. Zhang, and X. Yang, "Generative steganography by sampling," *IEEE Access*, vol. 7, pp. 118586–118597, Mar. 2019.

[32] B. Boehm, "StegExpose—A tool for detecting LSB steganography," 2014, *arXiv:1410.6656*. [Online]. Available: http://arxiv.org/abs/1410.6656

[33] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proc. 3rd Int. Workshop Inf. Hiding*, Dresden, Germany, Oct. 1999, pp. 31–76.

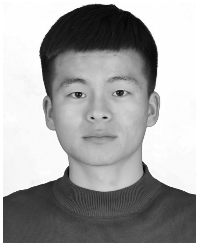
[34] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proc. Workshop Multimedia Secur. New Challenges &Sec*, Ontario, ON, Canada, 2001, pp. 27–30.

[35] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1995–2007, Jul. 2003.

[36] S. Dumitrescu, X. Wu, and N. D. Memon, "On steganalysis of random LSB embedding in continuous-tone images," in *Proc. Int. Conf. Image Process. (ICIP)*, Rochester, NY, USA, Sep. 2002, pp. 641–644.

XINTAO DUAN received the master's degree in computer application technology from Shanghai Normal University, in 2004, and the Ph.D. degree in communication and information systems from Shanghai University, in 2011. He has been teaching and researching with Henan Normal University, since July 2004. His research interests include image encryption, information hiding, image forensics, and deep learning.

WENXIN WANG received the B.S. degree from Nanyang Normal University, China, in 2018. 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.



NAO LIU received the B.S. degree from Henan Agricultural University, China, in 2018. 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.

DONGLI YUE received the M.S. degree from the Chinese Academy of Sciences, Hefei, China, in 2003. She is currently an Associate Professor with Henan Normal University, Xinxiang, China. Her current research interests include areas of image processing and machine learning.

ZIMEI XIE received the M.S. degree from Shantou University, Shantou, China, in 2006. She is currently a Lecturer with Henan Normal University, Xinxiang, China. Her current research interests include areas of image processing and machine learning.

CHUAN QIN (Member, IEEE) 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 School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, as a Faculty Member, 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 articles in these research areas.