Computer Modeling in Engineering & Sciences CMES, vol.124, no.2, pp.721-745, 2020

Image Information Hiding Method Based on Image Compression and Deep Neural Network

Xintao Duan[1, *], Daidou Guo[1] and Chuan Qin[2]

Abstract: Image steganography is a technique that hides secret information into the cover image to protect information security. The current image steganography is mainly to embed a smaller secret image in an area such as a texture of a larger-sized cover image, which will cause the size of the secret image to be much smaller than the cover image. Therefore, the problem of small steganographic capacity needs to be solved urgently. This paper proposes a steganography framework that combines image compression. In this framework, the Vector Quantized Variational AutoEncoder (VQ-VAE) is used to achieve the compression of the secret image. The compressed and reconstructed image is visually indistinguishable from the original image and facilitates more embedded data information later. Finally, the compressed image is transmitted to a SegNet deep neural network that contains a set of encoders and decoders to achieve image hiding and extraction. Experimental results show that the steganographic framework guarantees the quality of steganography while its relative steganographic capacity reaches 1. Besides, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values can reach 42 dB and 0.94, respectively.

Keywords: Image steganography, deep neural network, VQ-VAE, SegNet.

1 Introduction

The rapid development of information technology has made digital multimedia (such as images, audio, video, etc.) an important carrier for military, commercial, and personal to obtain and transfer information. Therefore, it is more and more easy to become the target of third-party eavesdropping and malicious attacks (such as information tampering and copyright infringement) in the transmission of public channels. For this reason, steganography was proposed to make up for the shortcomings of traditional encryption technology that cannot guarantee form security.

[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. Email: duanxintao@htu.edu.cn.
Received: 17 December 2019; Accepted: 23 March 2020.

Image steganography aims to hide the secret image into the cover image and extract it when needed, to cover the secret communication behavior. LSB [Tirkel, Rankin, Van Schyndel et al. (1993); Yang, Weng, Wang et al. (2008)] is the earliest steganography method proposed. It replaces the least significant bit of the image element with a message bit, thereby enabling information steganography. To the influential S-UNIWARD [Holub and Fridrich (2012)], WOW [Holub and Fridrich (2013)], Hill-CMD [Sedighi, Cogranne and Fridrich (2015)], MiPOD [Li, Wang, Li et al. (2015)] and other content-adaptive steganography in recent years, the secrets are artificially designed through embedded algorithms Information is hidden into the spatial domain or transform domain of the cover image, and obtains excellent imperceptibility and security. For example, Luo et al. [Luo, Huang and Huang (2010)] proposed an edge adaptive image steganography method based on LSB matching. Qin et al. [Qin, Zhang, Cao et al. (2018)] proposed an adaptive reversible data hiding scheme suitable for encrypted images. This solution can not only achieve perfect image recovery, but also the embedding capacity is considerable. But on the one hand, the traditional image steganography algorithm's embedded strength and position are often designed in advance and cannot be changed. On the other hand, when embedding information, the content of the cover image may not be fully balanced, such as the high and low-frequency component ratio distribution and the embedded secret image is smaller than the cover image, resulting in unsatisfactory steganographic capacity.

Because deep learning can better reflect the essential characteristics of data [Schmidhuber (2014)] and has made a series of breakthrough progress in the areas of image processing, natural language processing, and speech recognition. Such as Generative Adversarial Network (GAN) [Goodfellow, Pouget-Abadie, Mirza et al. (2014)] and Convolutional Neural Network (CNN) [Lecun, Bottou, Bengio et al. (1998)]. In recent years, it has brought new impetus to the field of image steganography. Volkhonskiy et al. [Volkhonskiy, Nazarov, Borisenko et al. (2017)] proposed a GAN-based steganographic enhancement algorithm that uses traditional algorithms to hide secret messages into the generated image and enhances security. Tang et al. proposed the use of generative adversarial networks for automatic steganographic distortion learning [Tang, Tan, Li et al. (2017)] and CNN-based image steganographic adversarial embeddings [Tang, Li, Tan et al. (2019)]. This method works under the conventional framework of distortion minimization. Hu et al. [Hu, Wang, Jiang et al. (2018)] proposed the use of a deep convolutional generation adversarial network (DCGAN) to hidden the image. El-Emam [El-emam (2008)] and Saleema et al. [Saleema and Amarunnishad (2016)] are dedicated to using neural networks to optimize embedded images generated by traditional steganography methods.

Although steganography based on deep learning gets rid of the process of artificial design, it is still in its infancy. On the research question of continuously improving the three parameters of security, robustness and steganographic capacity in steganographic communication systems, the research of steganography based on deep learning still has a

long way to go. Therefore, we tried a new steganography architecture in the study, in which VQ-VAE-2 [Oord, Vinyals and Kavukcuoglu (2017); Razavi, Oord and Vinyals (2019)] was cited. This structure enables each bit of the image to be fully compressed, and each valid information can be fully retained, and the reconstructed image is good for the Human Visual System (HVS). Also, the SegNet [Badrinarayanan, Kendall and Cipolla (2015)] neural network based on the encoder-decoder structure is used to implement the steganography of the secret image, and finally the CNN is used to implement the secret image extraction. In this network model, there are three network modules: compression network, hiding network, and revealed network. The compression network is used to achieve image compression and maximize the retention of important image content information. The hiding network hides the secret image to be sent into the cover image through a series of operations such as convolution. Then, the steganographic image is sent to the receiver, and the receiver uses the revealed network to extract the secret image.

The organization of the article is as follows: The second part introduces the preliminaries of image compression and image steganography. The third part describes the proposed research methods. The fourth part describes the results and analysis of the implementation. The fifth part is the conclusion.

2 Preliminaries

2.1 VQ-VAE

Vector quantization (VQ) is a method of signal compression. The basic idea is: form a vector of several scalar data groups and then give the overall quantization in the vector space, to achieve compressed data without losing important information. VQ has high compression rate and good visual quality in image processing. However, research on image steganography not only requires high security, but also needs to achieve the effect of confusing human vision in image reconstruction. Oord et al. [Oord, Vinyals and Kavukcuoglu (2017)] proposed the VQ-VAE model, which uses discrete latent variables. Inspired by vector quantization, training is performed in a new way and avoids posterior collapse. The discrete latent variable VAE model is not only similar to the continuous latent variable VAE model, but also has the flexibility of discrete distribution. In VQ-VAE, the posterior and prior distributions are classified, and samples extracted from these distributions can be indexed by embedding. These embeds are then used as input to the decoder network. Its structure is shown in Fig. 1.

First, the original image X passes through the CNN in the Encoder to obtain a continuous encoding vector $Z_e(X)$ of size $L \times W \times D$.

$$Z = encoder(X) \tag{1}$$

Here $\hat{X} = Decoder(Z_q(X))$ is a vector of size D. In addition, VQ-VAE also maintains an Embedding Space (that is, a coding table), which is recorded as

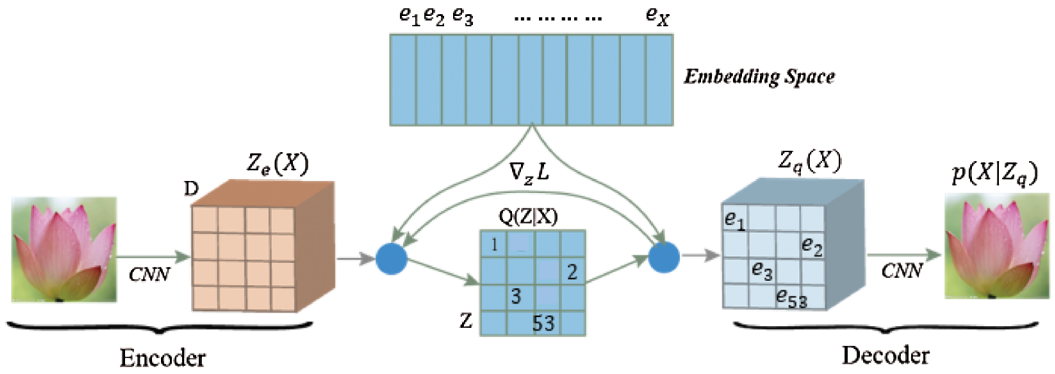

Figure 1: VQ-VAE strcuture

$$E=[e_1, e_2, \ldots, e_X] \tag{2}$$

Each e_i here is a vector of size D. Then, VQ-VAE maps Z to one of these X vectors by nearest neighbor search

$$Z \rightarrow e_x, k=\arg\min_j \left\|Z-e_j\right\|_2 \tag{3}$$

We can record the encoding table vector corresponding to Z as e_{bottom} (the final encoding result). Finally, input $Z_q(X)$ into a Decoder to build the original image $\hat{X}=Decoder(Z_q(X))$.

In brief, the entire process in Fig. 1 mainly implements four processes: color image information is converted into three-dimensional data; three-dimensional data is converted into two-dimensional data; two-dimensional data is converted into three-dimensional data, and three-dimensional data is converted into color images. Of these four processes, the first two processes implement compression, and the last two parts implement reconstruction.

2.2 Image steganography based on deep neural network

Compared with traditional artificially designed steganography algorithms, the deep learning-based steganography method can automatically hide and extract image information. No manual intervention is required during the process. By adjusting parameter information to extract different information features and the strength of information embedding, the efficiency of image steganography is greatly improved.

Steganography based on deep neural networks generally uses neural networks to find embedded location information suitable for images. For example, a deep steganography framework proposed by Baluja [Baluja (2017)] can be used to embed the entire secret image into the cover image. The implementation process is shown in Fig. 2. There are three parts in this network framework: Prep Network, Hiding Network, and Reveal Network. The Prep Network normalizes the secret image and extracts important features

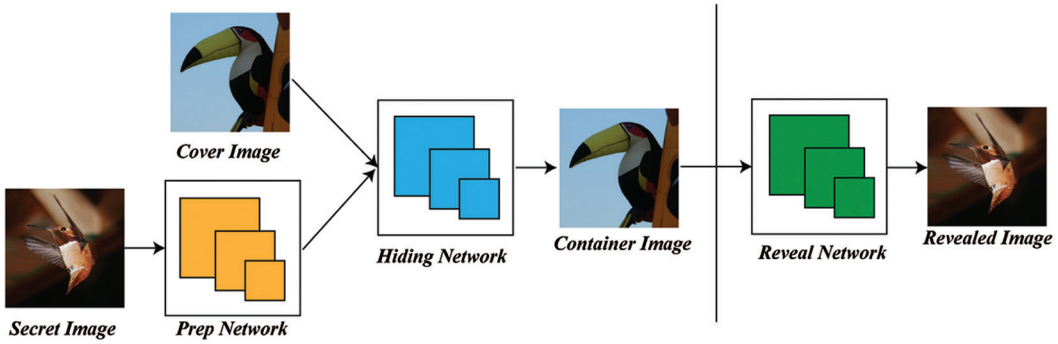

Figure 2: Deep steganography network structure

at the same time. The Hiding Network encodes a secret image and a cover image having the same size to obtain a Container image. At the same time, the model also trains a Reveal Network to extract the secret image. In addition, Wu et al. [Wu, Yang and Li (2018)] proposed a large image steganography method based on convolutional neural networks. This method includes a set of encoders and decoders, and uses HighWay Network [Srivastava, Greff and Schmidhuber (2015)], ResNet [He, Zhang, Ren et al. (2016)] and ResNet [Xie, Girshick, Dollar et al. (2016)] to form the core part of its steganographic network structure. Duan et al. [Duan, Jia, Li et al. (2019)] proposed a reversible image information hiding based on U-Net [Ronneberger, Fischer and Brox (2015)] deep neural network, which finally made the steganographic capacity 1. Liu et al. [Liu and Lee (2019)] proposed an improved reversible image steganography method based on pixel value ordering (PVO) to increase the steganographic capacity. Sort by considering three consecutive or adjacent pixels as a group, where the maximum and minimum values are used for the difference calculation, and the number of differences is recorded. This method effectively increases the steganographic capacity. Yedroudj et al. [Yedroudj, Comby and Chaumont (2019)] proposed a steganography method for 3-player games, which mainly includes three sub-networks. Adversarial learning between three different networks, experiments show that this method has a significant improvement in improving steganography quality. Yang et al. [Yang, Ruan, Huang et al. (2019)] proposed a framework for generating steganography. It has three sub-modules: a generator with a U-Net architecture, a double tangent function that does not require pre-training, and a steganographic analyzer based on a convolutional neural network and multiple high-pass filters as discriminators. This method also has a significant improvement in steganography quality.

2.3 SegNet image segmentation network

SegNet [Badrinarayanan, Kendall and Cipolla (2015)] is a Fully Convolutional Neural Network, which was first used in the field of image segmentation. The main structures include encoder, decoder, and a pixel-level classification layer. The encoder is used to

generate low-resolution features, and the decoder role is to map this coarse feature to the pixel-level classification across the entire input image-level resolution feature map. The most iconic point of SegNet is that the decoder samples its low-resolution input feature map. In a word, it uses a pooled index to achieve nonlinear Upsampling. The pooled index is corresponding to the decoder. The encoder performs the calculation of the maximum pooling operation. This eliminates the need to learn upsampling. The feature map after Upsampling is sparse, so a convolution operation is then performed using a trainable convolution kernel to generate a dense feature map. As shown in Fig. 3, the left side is a convolution extraction feature, which increases the receptive field by pooling, and the picture becomes smaller, which is the encoding process. On the right are deconvolution and Upsampling. The features of the image classification are reproduced by deconvolution, and the Upsampling is restored to the original size of the image, which is a decoding process. Finally, the maximum value of different classifications is output by Softmax, and then, the segmentation map is obtained. SegNet uses max-pooling to remember the position of the maximum value when downsampling, and it can quickly expand the size during upsampling, which means that upsampling does not involve deconvolution, which greatly speeds up training time.

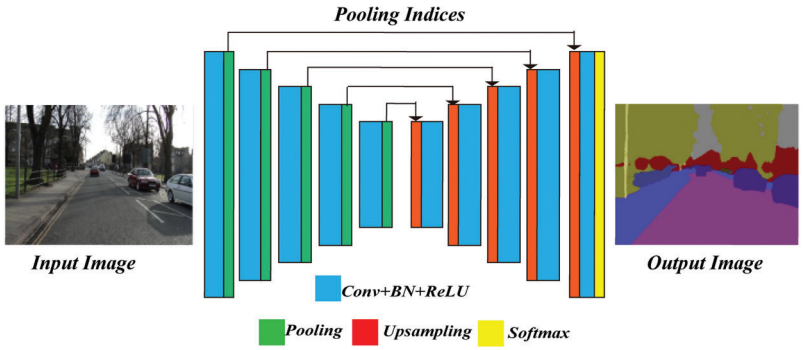

Figure 3: SegNet structure

2.4 Steganalysis

The research of steganography and steganalysis has been promoting and drawing on each other in confrontation. Steganalysis is the reverse detection method of steganography. It belongs to the category of pattern recognition. Its main purpose is to judge whether the secret information is contained in the statistical characteristics of the carrier, to estimate the length of the embedded secret information, and to identify steganography tools, estimate the steganographic key, and finally extract the secret information [Nissar and Mir (2010)]. The key point of blocking hidden communication is to determine whether the carrier contains hidden information. The general steganalysis process is generally divided into two stages: feature construction and classifier training. Since the steganographic embedding operation modifies high-frequency signals, in the feature

construction phase, a high-pass filter is usually used to calculate the residual image, and various statistical models are used to extract the steganographic analysis features. Early common image steganalysis methods include SPAM [Pevny, Bas and Fridrich (2010)], SRM [Fridrich and Kodovsky (2012)], tSRM [Tang, Li, Luo et al. (2014)], and DCTR [Holub and Fridrich (2015)]. Good feature representation plays a crucial role in the detection accuracy of steganalysis. Therefore, the current research on the general steganalysis method mainly focuses on the design and extraction of features. Similarly, the steganalysis method based on deep learning is an important development direction in the future. A method for steganographic analysis of large-scale JPEG images using a hybrid deep learning framework as proposed in Zeng et al. [Zeng, Tan, Li et al. (2018)]. Xu et al. [Xu, Wu and Shi (2016)] proposed convolutional neural networks for steganographic analysis.

3 The proposed image steganography method

In this section, we will describe and explain each component of the proposed steganography framework in detail.

3.1 Process description

As shown in Fig. 4, the steganography method proposed in this paper mainly includes three stages:

- The image preprocessing phase. Before the sender sends the secret image S to the receiver, the original image O is obtained by the compression module to obtain a secret image S that needs to be hidden.

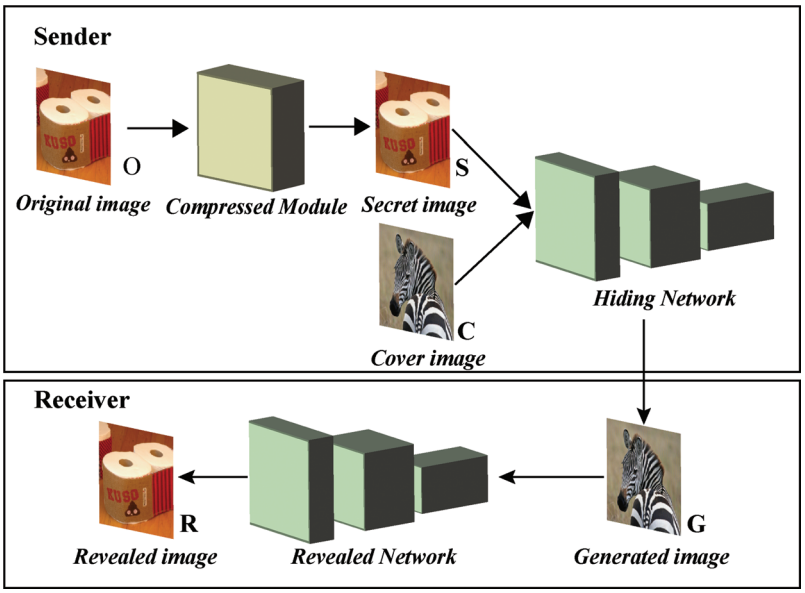

Figure 4: Steganography overview diagram

- Steganography phase of secret image. The sender takes the secret image S and a cover image C obtained in the previous step as inputs to the hidden network, and conceals the secret image S into the cover image C through operations such as convolution, and finally generate a generated image G very similar to the cover image C.
- The extraction stage of the secret image. The generated image G obtained in the previous step is input to the extraction network, and finally the required revealed image R is obtained.

Also, in our image steganography framework, the hiding network and the revealed network are trained models in advance and to ensure that the difference between the Cover image and the Generated image, and the Secret image and the Revealed image are minimized, these two subnetworks are Simultaneous training by adjusting hyperparameters.

3.2 Image compression

The image compression module used in this article is the VQ-VAE-2 [Razavi, Oord and Vinyals (2019)] model. Compared to the first-generation VQ-VAE model, the original VQ-VAE encoding has only one layer, and VQ-VAE-2 introduced the Hierarchical coding process. As shown in Fig. 5, the encoding of the model is divided into two levels: Top Level and Bottom Level. The Bottom Level has a large potential space of 64×64 (Global feature). This layer encodes the captured image's local information such as texture information. Quantize to get the quantized dictionary vector

$$e_{top} \leftarrow Quantize(E_{top}(x)) \tag{4}$$

Using this dictionary as a condition, together with the input x, compute the quantized form of the underlying latent space. The potential space of the Top Level is small, 32×32 (Local feature), which represents the global information such as object shape and geometry.

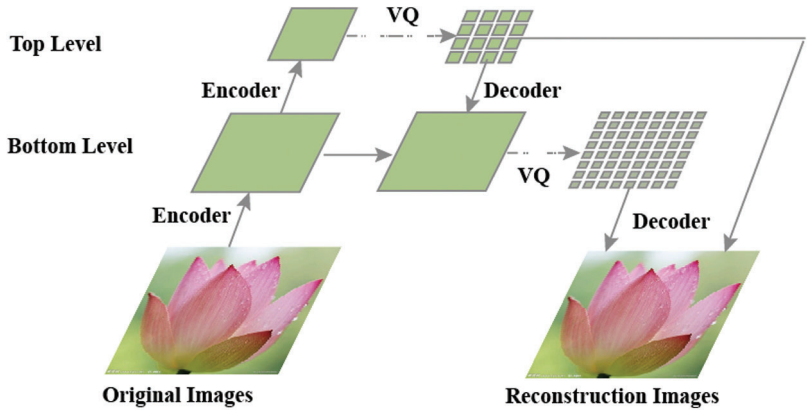

Figure 5: VQ-VAE-2 strcuture

Finally, the upper and lower quantized dictionary vectors e_{top} and e_{bottom} are simultaneously input to the decoder, the previous loss function is calculated, and the encoding and decoding network and dictionary weights are updated. Its loss function is shown in the following formula.

$$L(X, D(e)) = \|X - D(e)\|_2^2 + \|sg[E(X)] - e\|_2^2 + \beta\|sg[e] - E(X)\|_2^2 \tag{5}$$

where X is the input image, D is the decoder, E is the encoder, and sg means that the gradient is not calculated and the error is not passed to this corresponding variable. β represents a hyperparameter. The loss function is divided into three parts: $\|X - D(e)\|_2^2$ is the reconstruction error, $\|sg[E(X)] - e\|_2^2$ calculates the distance between the latent vector and the dictionary vector obtained by the encoder, and uses itas the auxiliary error term. The encoder and decoder are not updated. $\beta\|sg[e] - E(X)\|_2^2$ calculate the distance between the latent vector and the dictionary vector.

For the Bottom Level and Top Level layers, one obtains global features and the other obtains local features. Among them, there are residual links. This layered structure enables the encoder to extract more image features. Thereby reducing errors during reconstruction.

3.3 Hiding network structure

In our proposed steganography method, a SegNet network based on the encoder-decoder network structure is also used to implement steganography of secret images. As shown in Fig. 6. At the left end of the figure, first input two images of size $m \times n$ (m=n), and generate a 6-channel feature tensor (RGB images) or 2-channel feature tensor (Grayscale images) by concatenation convolution operation. Each encoder generates a series of feature maps through a set of convolutions (convolution kernel size is 4×4), followed by

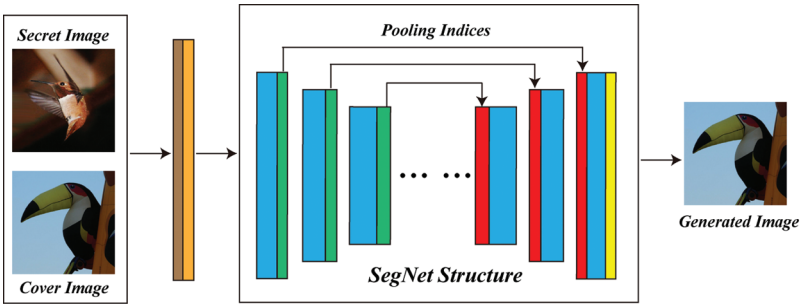

Figure 6: Image hiding network based on SegNet. The left half of the SegNet structure is the encoding stage, in which the operations of convolution and max-pooling are mainly performed, and the max-pooling index value is saved. The right half is the decoding stage. In this stage, the max-pooling index value saved in the encoding stage is used to perform the upsampling and convolution operations. Using the max-pooling index in the decoding process can improve the distribution of the boundaries and reduce the parameters of network training. Finally, softmax is used to classify the pixels

batches normalization, ReLU activation function, and max-pooling layer (2×2, *stride*=2). The original SegNet network uses the same convolution, which achieves the same size as the original image after the volume and operation. Max-pooling layer is used to achieving spatial invariance on small space movements, and there is a larger receptive field in feature mapping. But the use of Max-pooling causes a loss in resolution. This loss has a negative impact on the boundary definition, so the encoder network must focus on capturing and saving the boundary information before performing downsampling. And the 2×2 pooling window can be implemented with 2 bit, which makes the efficiency higher.

The decoder uses the max index stored in the corresponding encoder feature map to upsample the input feature map. The sparse feature maps generated are followed by a series of trainable convolution kernels to output dense feature maps, followed by batches normalization for normalization regularization reduces overfitting, and the decoder corresponding to the input generates a multi-channel feature map. In this process, the size of the convolution kernel does not change. The high-dimensional feature representation output by the decoder is sent to a trainable soft-max multi-classifier, which classifies each pixel individually. The network structure is described in Tab. 1.

Table 1: Brief description of hiding network structure

Layer	***Input size***	***Channel***	***Operation***	***Output size***
Concatente	256×256	6	Concatente Layer	128×128
Layer 1	128×128	64	Conv+BN+ReLU+Pooling	32×32
Layer 2	32×32	256	Conv+BN+ReLU+Pooling	8×8
Layer 3	8×8	512	Conv+BN+ReLU+Pooling	2×2
Layer 4	2×2	512	Conv+BN+ReLU+Pooling	4×4
Layer 5	4×4	512	Upsampling+Conv+BN+ReLU	16×16
Layer 6	16×16	1024	Upsampling+Conv+BN+ReLU	64×64
Layer 7	64×64	256	Upsampling+Conv+BN+ReLU	256×256
Layer 8	256×256	64	Upsampling+Conv+BN+ReLU +Softmax	256×256
Layer 9	256×256	3	Output	256×256

Conv means convolution and BN means batch normalization.

Since the network is based on the structure of the encoder-decoder, the container image of the intermediate representation is required to be as similar as possible to the cover image, and can be expressed by the following formula.

$$L(C,S,G,R)=\|C-S\|+\beta\|G-R\| \tag{6}$$

where C, S, G, R respectively represent cover image, secret image, generated image, and revealed image. β is a hyper-parameter used to measure reconstruction errors. $\|C-S\|$ does not apply to the weight of the extraction network that accepts the Container image and extracts the Secret image, that is, its weight is not shared with the extraction network. All networks accept the error signal $\beta\|G-R\|$, so that the two networks will continuously adjust the error loss of the secret image and the cover image through training to ensure that the Secret image can be completely encoded into the Cover image.

In addition, the cross-entropy cost function is mainly used:

$$L=-\sum_{c=1}^{M} y_c \log(p_c) \tag{7}$$

where M is the number of samples.

3.4 Revealed network structure

The Secret image extraction process refers to the network model of Duan et al. [Duan, Jia, Li et al. (2019)], as shown in Fig. 7. In this network structure, the Secret image is accurately extracted by 6 convolutional layers. In CNN, the Dropout operation is used. The activation functions and the pooling layer enhances the nonlinear learning ability of the network. The purpose of CNN is to use nonlinear features to learn the fitting parameters. Learn the weighting parameters in each layer of the network to accommodate the mapping between input and output 3×3. In this network, the filter size of each convolutional layer is designed to be 3×3, and each convolution layer is followed by a ReLU activation function and a batch normalization operation. In the left and right blocks of the network, the feature vectors of 64 components are mapped to the required number of categories using a convolution of 3×3, and the Secret image and the Covert image are calculated using the Sigmoid activation function. In the process of extracting the secret image, to

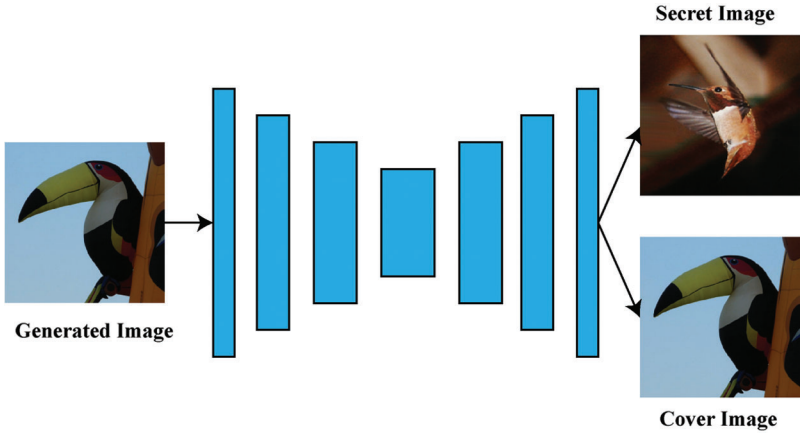

Figure 7: Image revealed network structure

Table 2: Brief description of revealed network structure

Layer	*Input size*	*Channel*	*Operation*	*Output size*
Layer 1	256×256	3	Conv+ReLU+BN	256×256
Layer 2	256×256	64	Conv+ReLU+BN	256×256
Layer 3	256×256	128	Conv+ReLU+BN	256×256
Layer 4	256×256	256	Conv+ReLU+BN	256×256
Layer 5	256×256	128	Conv+ReLU+BN	256×256
Layer 6	256×256	64	Conv+ReLU+BN	256×256
Layer 7	256×256	3	Sigmoid	256×256

Conv means convolution and BN means batch normalization.

keep the size of the image unchanged, we set Stride to 1 and Padding to 1. The network structure is described in Tab. 2.

4 Experiment

4.1 Experimental environment and dataset

In the third section, we make a theoretical description of the proposed steganography method. In this section, we perform experimental simulations for further performance evaluation. The experimental environment is Python 3.6 programming languages and the Pytorch framework under the Ubuntu operating system. The experimental device has dual NVIDIA 1080 Ti GPU and 16 GB RAM, 1 TB HDD and 256 GB SSD.

In the selection of the data set, we selected 45,000 images from the ImageNet dataset as the training set and 5000 images as the test set.

4.2 Experimental results of image compression and reconstruction

We use the VQ-VAE-2 network to achieve image compression and reconstruction. According to VQ-VAE, the original picture is first compressed into a discrete coding space, so that the amount of information will be reduced accordingly, and the decoder can reconstruct the picture from this space. The compressed image mainly uses the pixelCNN algorithm. Therefore, the reconstructed image after sampling can still maintain good quality. VQ-VAE-2 is an improvement on the basis of VQ-VAE. In simple terms, it is to divide the encoding and decoding of the original layer into two layers, one is Top Level and the other is Bottom Level. In our experiments, the original VQ-VAE-2 network structure and its parameters were used. We used the 256×256 ImageNet dataset for training. The training process will automatically compress it to the bottom quantized latent layer of 64×64 (Shrink 4 times) and the top quantized latent layer of 32×32 (Shrink 8 times). Fig. 8 is the process of image reconstruction through the VQ-VAE-2 network. The 0.25 of the hyperparameter β is the optimal value that we continuously adjust. In addition, we set the batch size of the network to 64 and the number of trainings

Figure 8: Image reconstruction process. This image is an image reconstructed from two latent layers of VQ-VAE-2. The first image is a reconstructed image of the Top Level layer, the second image is a reconstructed image containing the Top Level and Bottom Level, and the third image is the original image. It can be seen from the image that for each latent layer, additional details are added during the reconstruction process

Table 3: VQ-VAE-2 network structure

	Input size	*Operation*	*Filter Size*	*Output Size*
Encoder	256×256	Conv1+ReLU+Residual1	3×3	64×64
	64×64	Conv2+ReLU+Residual2	3×3	32×32
Decoder	32×32	Residual1+Residual2+ReLU+Cat +Conv1	4×4	256×256

Residual refers to the residual module used in the network, Conv refers to the convolution operation, and Cat refers to the concatenation. Padding is set to 1, Stride is set to 2.

to 30,000. The effect of the final compression and reconstruction is shown in Fig. 9. The first and second rows represent the original image and the reconstructed image, respectively. Tab. 3 is a brief description of the structure of VQ-VAE-2.

4.3 Subjective steganographic results

We stew the compressed and reconstructed image as a secret image. We set the parameters of the hidden network as follows: the batch size is 8, the learning rate is set to 0.001, and the hyperparameter β is set to 0.75. In addition, the Adam optimization algorithm is used to automatically adjust the learning rate so that the network parameters can be smoothly learned. Fig. 10 shows the effect after training is stable. The first to fourth lines in the figure respectively represent the cover image, the generated image, the secret image, and the revealed image. Through the gradual and stable training of the neural network, we can intuitively find that the final result is good for the HVS, and people can't see the difference visually. Also, we performed some other experimental analyses of its hidden

Figure 9: VQ-VAE-2 compression reconstruction image

Figure 10: Intuitive diagram after network training is stable

effects. Fig. 11 shows four sets of data extracted for analysis, and an intuitive display of the errors between the cover image and the generated image, the secret image and the steganographic image. Considered the samples under more common sence, we randomly downloaded some texture images and noise images from the Internet for experiments. The experimental results are shown in Fig. 12.

As shown in Fig. 13. We performed a histogram analysis of the data. Looking at the histogram of the two sets of extracted data, the carrier image and the generated image, the secret image and the extracted image changed little on the three-channel decomposition maps of R, G, and B. The changes of each channel were basically Be consistent, not miscellaneous. It is verified that the modified model follows the principle of complementary pixels when embedding secret information instead of random

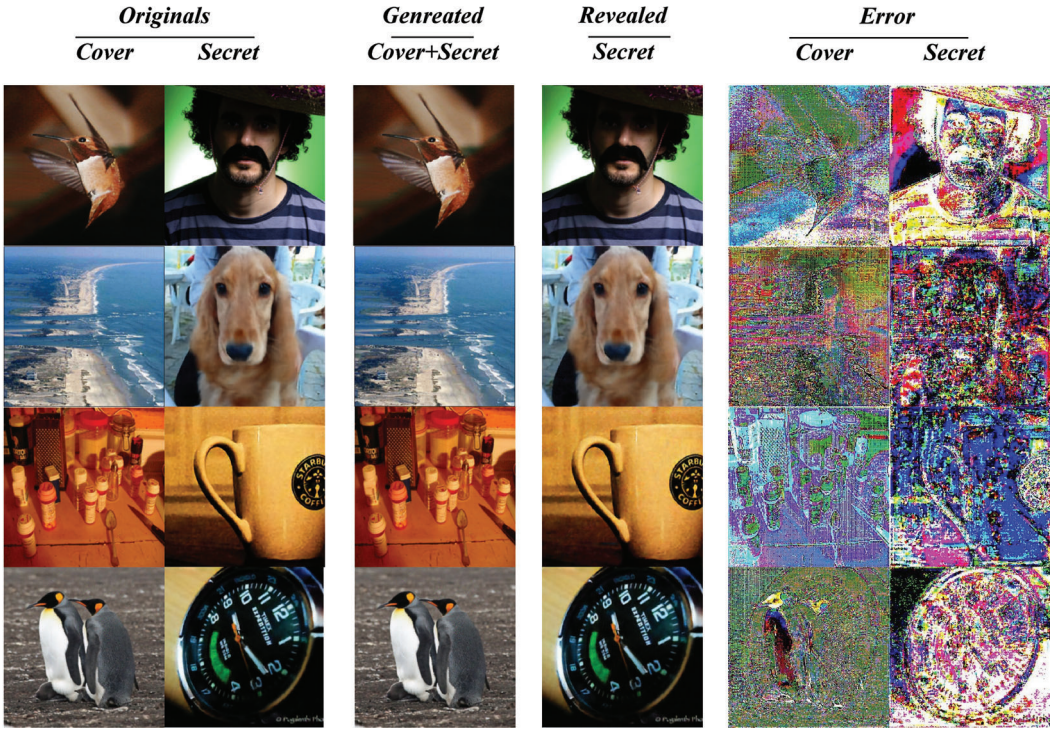

Figure 11: Sample from full-image hiding system. The error results of the last two columns of the figure are obtained by subtracting each corresponding pixel value between the two images. This will slightly detect if it contains secret information

out-of-order embedding. In addition, we also analyzed its PSNR value, SSIM value and steganographic capacity parameters, see Sections 4.5, 4.6 and 4.7 in detail.

4.4 Calculation of the revealed rate, the cover changing rate, and the payload capacity

Revealed Rate: this refers to the probability that a secret image can be correctly extracted.

Cover Changing Rate: this refers to the change rate between the cover image and the generated image.

Payload Capacity: this refers to the number of bits of information contained in each pixel.

$$Revealed\ Rate = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left|S_{i,j} - R_{i,j}\right|}{N \times M} \tag{8}$$

$$Cover\ Changing\ Rate = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left|C_{i,j} - G_{i,j}\right|}{N \times M} \tag{9}$$

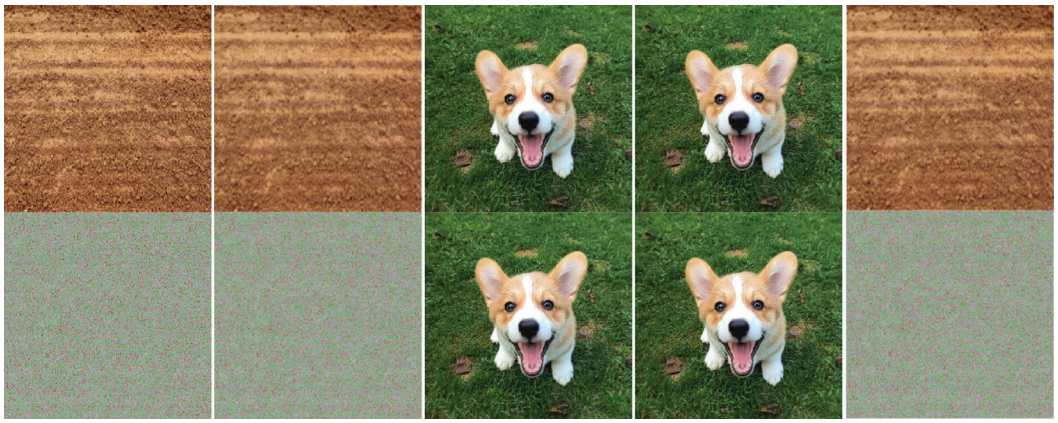

Figure 12: Graph of random test results. In the figure, the first column is the original image, the second column is the secret image obtained by compression, the third column is the cover image, the fourth column is the generated image, and the fifth column is the revealed image

$$Payload\ capacity = Revealed\ Rate \times 8 \times 3 \quad (bpp) \tag{10}$$

Here, S, G, C, R respectively represent Secret image, Genreated image, Cover Image, and Revealed image. "8, 3" represents 8 bits and 3 channels each. We perform a brief calculation on the four groups of images in Fig. 11, and the calculation results are shown in Tab. 3. Fig. 14 is the calculation of the payload capacity of the compressed and uncompressed images by formula (10).

From Tab. 4, it can be seen that, while our method has good hiding ability and extraction ability, the quality of the generated image also performs well. For example, the value of the cover image change rate in the second column of the table remains below 1%.

4.5 Stegangraphic results peak signal noise ratio analysis (PSNR)

PSNR provides an objective standard for measuring image distortion or noise level. It is often used for objective evaluation of image degradation before and after compression in areas such as image compression. The evaluation result is expressed in dB (decibel). The larger the PSNR value between the two images, the more there is no degradation. When the degradation degree is large, the PSNR value tends to 0 dB. PSNR is an index used to measure image quality, such as in the fields of image compression and super-resolution reconstruction of images.

Here two main values are defined. One is the mean squared MSE and the other is the peak signal to noise ratio (PSNR). The formula is as follows.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||C(i,j) - S(i,j)||^2 \tag{11}$$

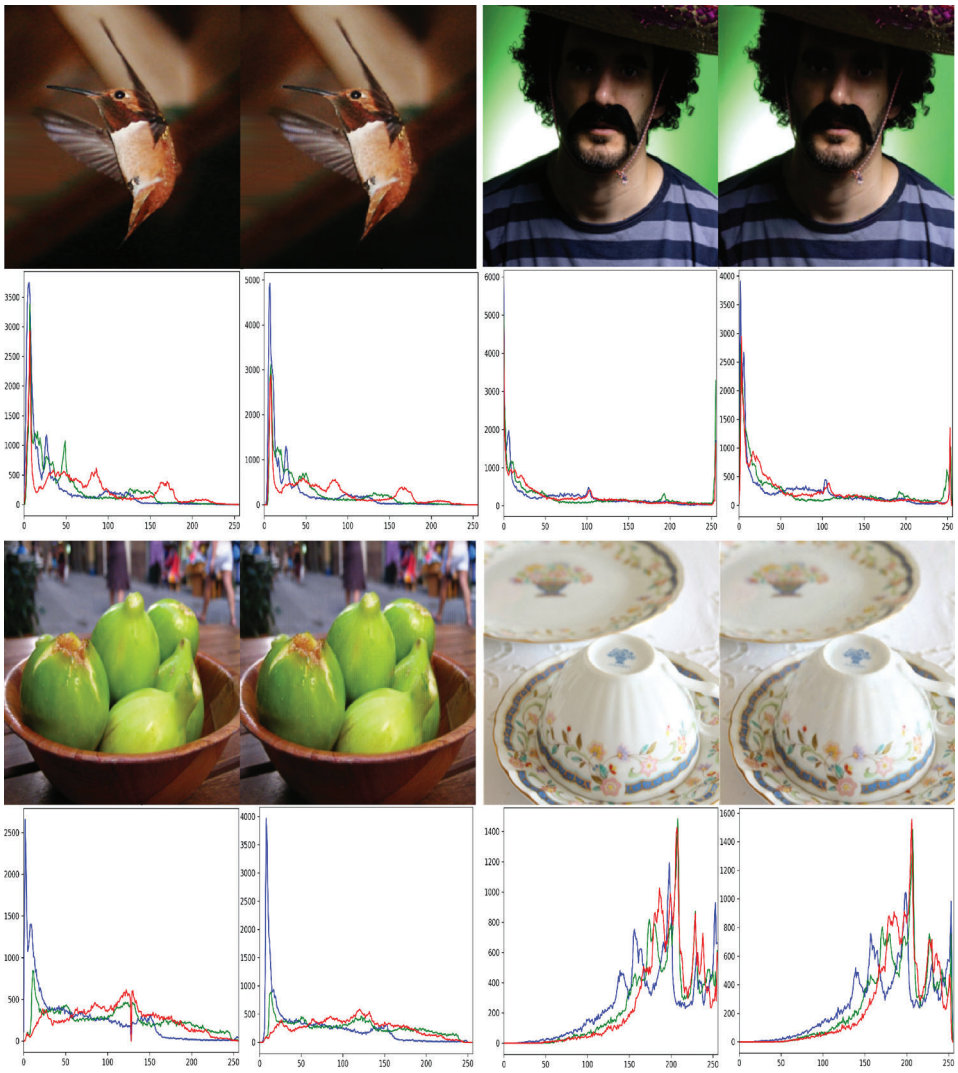

Figure 13: Sample effect histogram after training stabilization

Among them, C and S represent the host image and the secret image, respectively, and their sizes are set to. MSE indicates the mean variance of the host image compared with the steganographic image.

$$PSNR=10\log_{10}(\frac{MAX_I^2}{MSE}) \tag{12}$$

where MAX_I represents the maximum pixel value of the image. In other words, MAX_I is equal to 2^b-1 and b represents the number of bits per pixel. For grayscale images, the

Table 4: Calculate the above three values

Number	*Revealed Rate (%)*	*Cover Changing Rate (%)*	*Payload Capacity (bpp)*
1	99.5%	0.99%	23.76 bpp
2	99.4%	0.96%	23.86 bpp
3	98.3%	0.78%	23.60 bpp
4	97.0%	0.88%	23.28 bpp

This representation is calculated for formulas 8, 9, and 10. It can be found from the results that our method has achieved satisfactory results on all three values.

Figure 14: Payload analysis. In this experiment, a set of pictures were randomly selected. The picture on the left is uncompressed and the picture on the right is compressed. It can be proved through experiments that after the compressed image is hidden, the payload value will be higher

maximum pixel value is 255. For RGB images, for RGB images (each pixel has three color parameters of R, G, and B), the PSNR is defined in a similar manner. Tab. 5 is the PSNR value calculated by our method.

4.6 Steganographic results structural similarity index analysis (SSIM)

The SSIM value is a new index for measuring the similarity of two images. The larger the value, the better. The maximum value is 1, which is often used in image processing. Structural similarity theory believes that natural image signals are highly structured, that is, there is a strong correlation between pixels, especially the closest pixels in the airspace. This correlation contains important information about the structure of objects in the visual scene. The structural similarity index defines the structural information from

Table 5: PSNR value

Category	PSNR value
Bird (Fig. 11)	PSNR=47.1319
Man (Fig. 11)	PSNR=43.4185
Fruits (Fig. 11)	PSNR=46.3026
Cup (Fig. 11)	PSNR=44.0718
Image Net	PSNR (Average)=42.2739

The calculated value in the last row of the table is the result obtained by randomly sampling 100 images in the experiment and averaging.

the perspective of image composition as independent of brightness and contrast, reflects the properties of the object structure in the scene, and models distortion as a combination of three different factors: brightness, contrast, and structure. The mean is used as an estimate of brightness, the standard deviation is used as an estimate of contrast, and the covariance is used as a measure of structural similarity.

$$SSIM(C,S)=\frac{(2\mu_C\mu_S+C_1)(2\sigma_C\sigma_S+C_2)}{(\mu_C^2+\mu_S^2+C_1)(\sigma_C^2+\sigma_S^2+C_2)} \tag{13}$$

where C_1 and C_2 are two variables to stabilize the division with weak denominator. Moreover, μ and σ present the average and covariance of the variables. Tab. 6 is the SSIM value calculated by our method.

Table 6: SSIM value

Category	PSNR value
Bird (Fig. 11)	SSIM=0.9778
Man (Fig.11)	SSIM=0.9531
Fruits (Fig. 11)	SSIM=0.9875
Cup (Fig. 11)	SSIM=0.9783
Image Net	SSIM (Average)=0.9438

The calculated value in the last row of the table is the result obtained by randomly sampling 100 images in the experiment and averaging.

It is worth mentioning that we also calculated the PSNR and SSIM values for the results of these unnatural images such as noise images and texture images, and the results are consistent with the calculated values generated by the normal natural images described above.

Table 7: Steganographic capacity comparison result

Method	*Absolute capacity (bytes/image)*	*Image size*	*Relative capacity (bytes/image)*
Tang, Li, Tan et al. (2019)	≥37.5	64×64	9.16*e*-3
Zhou, Cao and Sun (2016)	3.72	≥512×512	1.42*e*-5
Zhou, Sun, Harit et al. (2015)	1.125	512×512	4.29*e*-6
Zheng, Liang, Ling et al. (2017)	2.25	512×512	8.58*e*-6
Xu, Mao, Jin et al. (2014)	64×64	800×800	6.40*e*-3
Wu and Wang (2014)	1535-4300	1024×1024	1.46*e*-3
			4.10*e*-3
Liu and Lee (2019)	≈32620	512×512	1.24*e*-1
Ours	256×256	256×256	1

This value is obtained by the relative capacity calculation formula. Since this method can achieve steganography of a fullsize image, the value can reach 1.

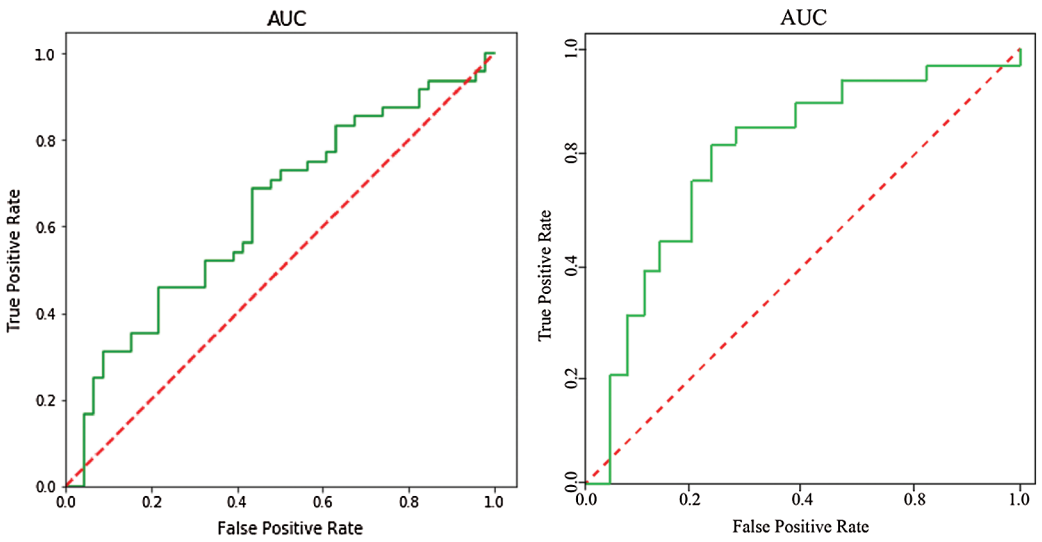

Figure 15: ROC curve. The left is our proposed method, and the right is LSB-based image steganography. The test data of these two graphs are the same and are compressed by the VQ-VAE-2 network. According to the figure, we can see that our proposed method has better resistance to steganographic analysis. The data source comes from ImageNet (which includes both steganographic and original images)

4.7 Capacity analysis

The steganographic method using the deep network-based information steganography method is higher than the traditional artificially designed embedded algorithm. Tab. 7 is a comparison of the steganography capacity of some current mainstream steganography methods and our proposed method. The formula is as follows.

$$Relative\ capacity = \frac{Absolute\ capacity}{The\ size\ of\ the\ image} \tag{14}$$

4.8 Statistical analysis

StegExpose [Boehm (2014)] is a steganalysis tool for detecting LSB (least significant bit) steganography in lossless images such as PNG and BMP. This article was tested with StegExpose. Four detection methods are included in the tool: sample pair analysis [Dumitrescu, Wu and Wang (2003)], RS analysis [Fridrich, Goljan and Du (2002)], chi-square attack [Westfeld and Pfitzmann (2000)] and primary sets [Dumitrescu, Wu and Memon (2002)]. The detection threshold is its hyperparameter, which is used to balance the true positive rate and false positive rate of StegExpose results. Fig. 15 is an ROC curve. Among them, "True positive" represents an embedded image that is correctly identified as having hidden data inside, and "False positive" represents a clean graphic that is incorrectly classified as an embedded image. The graph is drawn with a green polyline, indicating that StegExpose can only be a little better than random guessing (red lines). In other words, the proposed steganography method can better resist StegExpose attacks.

5 Conclusion and future work

On the one hand, with the continuous penetration and influence of deep learning on various aspects, and on the other hand, based on traditional artificially designed image steganography algorithms, compared with the research of deep learning in this field has certain advantages. Therefore, we propose a method of image steganography based on deep neural networks in the research. First, the image is compressed and reconstructed, retaining important image information, and the visual quality is high. Later, based on the deep neural network for image steganography and extraction, it was proved by experiments that our proposed method can effectively improve the steganographic capacity, while its PSNR and SSIM values can reach 42 dB and above 0.94, respectively. All aspects of the parameters have a better performance. At present, our research is limited to hiding one image into another image and does not achieve the hiding of multiple images.

The next work will consider hiding two images into one image, so as to achieve more efficient image steganography.

Acknowledgement: The paper was supported by the National Natural Science Foundation of China (61672354), the key scientific research project of Henan Provincial Higher Education (Nos. 19B510005 and 20B413004). The authors would like to thank the anonymous reviewers for their valuable suggestions.

Funding Statement: The paper was supported by the National Natural Science Foundation of China (61672354), the key scientific research project of Henan Provincial Higher Education (Nos. 19B510005 and 20B413004).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Badrinarayanan, V.; Kendall, A.; Cipolla, R. (2015): Segnet: a deep convolutional encoder-decoder architecture for image segmentation.

Baluja, S. (2017): Hiding images in plain sight: deep steganography. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 30,* pp. 2069-2079, Curran Associates, Inc., Red Hook, USA.

Boehm, B. (2014): *Stegexpose—A Tool for Detecting LSB Steganography.* arXiv e-prints, arXiv:1410.6656

Duan, X.; Jia, K.; Li, B.; Guo, D.; Zhang, E. et al. (2019): Reversible image steganography scheme based on a U-Net structure. *IEEE Access*, vol. 7, pp. 9314-9323. DOI 10.1109/ACCESS.2019.2891247.

Dumitrescu, S.; Wu, X.; Memon, N. (2002): On steganalysis of random LSB embedding in continuous-tone images. *Proceedings of International Conference on Image Processing*, vol. 3, pp. 641-644.

Dumitrescu, S.; Wu, X.; Wang, Z. (2003): Detection of LSB steganography via sample pair analysis. *IEEE Transactions Signal Processing*, vol. 51, no. 7, pp. 1995-2007. DOI 10.1109/TSP.2003.812753.

El-emam, N. (2008): Embedding a large amount of information using high secure neural based steganography algorithm. *International Journal of Computer and Communication Engineering*, vol. 4, pp. 223-232.

Fridrich, J.; Goljan, M.; Du, R. (2002). Reliable detection of LSB steganography in color and grayscale images. *Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, pp. 27-30.

Fridrich, J.; Kodovsky, J. (2012): Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882. DOI 10.1109/TIFS.2012.2190402.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D. et al. (2014). Generative adversarial networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2672-2680.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, Las Vegas, NV. DOI 10.1109/CVPR.2016.90.

Holub, V.; Fridrich, J. (2012): Designing steganographic distortion using directional filters. *IEEE International Workshop on Information Forensics and Security*, pp. 234-239.

Holub, V.; Fridrich, J. (2013): Digital image steganography using universal distortion. *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, pp. 59-68, France, Montpellier. DOI 10.1145/2482513.2482514.

Holub, V.; Fridrich, J. (2015): Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219-228. DOI 10.1109/TIFS.2014.2364918.

Hu, D.; Wang, L.; Jiang, W.; Zheng, S.; Li, B. (2018): A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, vol. 6, pp. 38303-38314.

Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. (1998): Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324. DOI 10.1109/5.726791.

Li, B.; Wang, M.; Li, X.; Tan, S.; Huang, J. (2015): A strategy of clustering modification directions in spatial image steganography. *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 1-2. DOI 10.1109/TIFS.2014.2377671.

Liu, H. H.; Lee, C. M. (2019): High-capacity reversible image steganography based on pixel value ordering. *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1062. DOI 10.1186/s13640-019-0458-z.

Luo, W.; Huang, F.; Huang, J. (2010): Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 201-214. DOI 10.1109/TIFS.2010.2041812.

Nissar, A.; Mir, A. (2010): Classification of steganalysis techniques: a study. *Digital Signal Processing*, vol. 20, no. 6, pp. 1758-1770. DOI 10.1016/j.dsp.2010.02.003.

Oord, A.; Vinyals, O.; Kavukcuoglu, K. (2017): Neural discrete representation learning. arXiv preprint arXiv: 1711.00937.

Pevny, T.; Bas, P.; Fridrich, J. (2010): Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215-224. DOI 10.1109/TIFS.2010.2045842.

Qin, C.; Zhang, W.; Cao, F.; Zhang, X.; Chang, C. C. (2018): Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Processing*, vol. 153, pp. 109-122. DOI 10.1016/j.sigpro.2018.07.008.

Razavi, A.; Oord, A.; Vinyals, O. (2019). Generating diverse high-fidelity images with VQ-VAE-2. *Annual Conference on Neural Information Processing Systems*. abs/1906.00446. pp. 14837-14847.

Ronneberger, O.; Fischer, P.; Brox, T. (2015): U-net: convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, Springer.

Saleema, A.; Amarunnishad, T. (2016): A new steganography algorithm using hybrid fuzzy neural networks. *Procedia Technology*, vol. 24, pp. 1566-1574. DOI 10.1016/j.protcy.2016.05.139.

Schmidhuber, J. (2014): Deep learning in neural networks: an overview. *Neural Networks*, vol. 61, pp. 85-117. DOI 10.1016/j.neunet.2014.09.003.

Sedighi, V.; Cogranne, R.; Fridrich, J. (2015): Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 221-234.

Srivastava, R. K.; Greff, K.; Schmidhuber, J. (2015): Highway networks. arXiv preprint arXiv: 1505.00387.

Tang, W.; Li, B.; Tan, S.; Barni, M.; Huang, J. (2019): CNN-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074-2087. DOI 10.1109/TIFS.2019.2931817.

Tang, W.; Li, H.; Luo, W.; Huang, J. (2014): Adaptive steganalysis against wow embedding algorithm. *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, pp. 11-13, ACM Press, New York, USA. DOI 10.1145/2600918.2600935.

Tang, W.; Tan, S.; Li, B.; Huang, J. (2017): Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547-1551. DOI 10.1109/LSP.2017.2745572.

Tirkel, A. Z.; Rankin, G.; Van Schyndel, R.; Ho, W.; Mee, N. et al. (1993): Electronic watermark. *Digital Image Computing, Technology and Applications*, pp. 666-673, Macquarier University, Sydney.

Volkhonskiy, D.; Nazarov, I.; Borisenko, B.; Burnaev, E. (2017): Steganographic generative adversarial networks. CoRR.arxiv:abs/1703.05502.

Westfeld, A.; Pfitzmann, A. (2000): Attacks on steganographic systems. In Pfitzmann, A. (ed.), *Information Hiding*, pp. 61-76, Springer Berlin Heidelberg, Berlin, Heidelberg.

Wu, K. C.; Wang, C. M. (2014): Steganography using reversible texture synthesis. *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 130-139.

Wu, P.; Yang, Y.; Li, X. (2018): Stegnet: mega image steganography capacity with deep convolutional network. *Future Internet*, vol. 10, no. 6, pp. 54. DOI 10.3390/fi10060054.

Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. (2016): Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, pp. 5987-5995. DOI: 10.1109/CVPR.2017.634.

Xu, G.; Wu, H. Z.; Shi, Y. Q. (2016): Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708-712. DOI 10.1109/LSP.2016.2548421.

Xu, J.; Mao, X.; Jin, X.; Jaffer, A.; Lu, S. et al. (2014): Hidden message in a deformation-based texture. *Visual Computer*, vol. 31, no. 12, pp. 1653-1669. DOI 10.1007/s00371-014-1045-z.

Yang, C. H.; Weng, C. Y.; Wang, S. J.; Sun, H. M. (2008): Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 488-497. DOI 10.1109/TIFS.2008.926097.

Yang, J.; Ruan, D.; Huang, J.; Kang, X.; Shi, Y. Q. (2019): An embedding cost learning framework using GAN. *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 839-851. DOI 10.1109/TIFS.2019.2922229.

Yedroudj, M.; Comby, F.; Chaumont, M. (2019): Steganography using a 3-player game. arXiv preprint arXiv: 1907.06956.

Zeng, J.; Tan, S.; Li, B.; Huang, J. (2018): Large-scale JPEG steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1200-1214. DOI 10.1109/TIFS.2017.2779446.

Zheng, S.; Liang, W.; Ling, B.; Hu, D. (2017): Coverless information hiding based on robust image hashing. *International Conference on Intelligent Computing*, vol. 10363, pp. 536-547.

Zhou, Z.; Sun, H.; Harit, R.; Chen, X.; Sun, X. (2015): Coverless image steganography without embedding. *International Conference on Cloud Computing and Security*, vol. 9483, pp. 123-132.

Zhou, Z. L.; Cao, Y.; Sun, X. M. (2016): Coverless information hiding based on bag-of-words model of image. *Journal of Applied Sciences*, vol. 34, pp. 527-536.