Hindawi
Security and Communication Networks
Volume 2020, Article ID 8847741, 9 pages
https://doi.org/10.1155/2020/8847741

Research Article

DRHNet: A Deep Residual Network Based on Heterogeneous Kernel for Steganalysis

Yang Xu, Zixi Fu, Guiyong Xu, Sicong Zhang, and Xiaoyao Xie

Key Laboratory of Information and Computing Science of Guizhou Province, Guizhou Normal University, Guiyang 550001, China

Correspondence should be addressed to Sicong Zhang; gs.sczhang16@gzu.edu.cn

Received 31 July 2020; Revised 5 November 2020; Accepted 18 November 2020; Published 12 December 2020

Academic Editor: Zhihua Xia

Convolutional neural networks as steganalysis have problems such as poor versatility, long training time, and limited image size. For these problems, we present a heterogeneous kernel residual learning framework called DRHNet—Dual Residual Heterogeneous Network—to save time on the networks during the training phase. Instead of using the image as an input of the network, we extract and merge the images into a feature matrix using the rich model and use the generated feature matrix as the real input of the network. The architecture we proposed has good versatility and can reduce the computation and the number of parameters while still getting higher accuracy. On BOSSbase 1.01, we evaluate the performance of DRHNet in the setting of the spatial domain and frequency domain. The preliminary experimental results show that DRHNet shows excellent steganalysis performance against the state-of-the-art steganographic algorithms.

1. Introduction

As the most commonly used scheme of modern steganography, the least significant bit (LSB) will inevitably change the correlation between adjacent pixels of the image and the correlation of adjacent pixels of the residual image (high-frequency component) of the image [1, 2]. Before the renaissance of the neural network, the mainstream steganographic analysis method extracts the statistic that can describe the correlation of adjacent pixels of the residual image as a steganographic analysis feature and then uses the machine learning tool to train the steganographic analysis classifier [3, 4].

Convolutional neural network (CNN) has been widely used in the field of image classification [5–7]. Since steganalysis can be regarded as a two-class problem for images, the goal is to determine whether an image is embedded with the ciphertext. Steganalysis began to use convolutional neural networks to attack steganography. Qian et al. [8] first proposed the application of convolutional neural networks to steganalysis. They described a neural network steganalyzer with a Gaussian activation function equipped with a fixed preprocessing high-pass KV filter. The high-pass KV filter was used to suppress the image content, thus improving the signal-to-noise ratio (SNR) between the stego signal and the host image. Ye et al. [9] proposed a new network in which rather than a random strategy, the weights in the first layer of the proposed CNN are initialized with the basic high-pass filter set used in the calculation of residual maps in the spatial rich model (SRM), which acts as a regularizer to suppress the image content effectively. To better capture the structure of embedding signals, which usually have extremely low SNR (stego signal to image content), a new activation function called truncated linear unit (TLU) is adopted in their CNN model. Boroumand et al. [10] described a deep residual architecture, SRNet, designed to minimize the use of heuristics and externally enforced elements that are universal in the sense that it provides state-of-the-art detection accuracy for both spatial domain and JPEG steganography. The key part of the proposed architecture is a significantly expanded front part of the detector that "computes noise residuals" in which pooling has been disabled to prevent suppression of the stego signal.

The problem with using neural networks as a steganographic analysis tool is that it is impossible to analyze larger sized images due to limitations in computer resources. And the versatility of such steganographic analysis tools is not good; that is, a network trained with a steganographic algorithm cannot analyze images with another steganographic algorithm. Finally, the training time of the neural network is too long.

In this paper, we propose a heterogeneous kernel residual learning framework to save time on the networks during the training phase. Experimental results show that the DRHNet detection error achieves less than 10% when using S-UNIWARD [11] as the steganography algorithm, and the payload is 0.4 bpp. In summary, we make the following contributions in this paper:

(i) We address the accuracy-computation-time problem by introducing a versatile deep residual learning for steganalysis.

(ii) Instead of using the image as an input to the network, we extract and merge the images into a feature matrix through the rich model and use the generated feature matrix as the real input of the network.

(iii) The heterogeneous kernel is used as the convolution kernel of the network that we proposed. The heterogeneous kernel is adopted to reduce network parameters and reduce computational complexity.

2. Preliminaries

2.1. Feature Selection Method. The spatial rich model (SRM) [12] is a typical image steganographic analysis method. It designs a wide variety of spatial high-pass filters and uses these filters to filter the image to obtain a rich variety of residual images; then, it separately counts the frequency of occurrence of each adjacent residual sample pattern in a residual image. The cooccurrence matrix of the residual image is obtained. Finally, the elements of the cooccurrence matrix are rearranged into vectors as a steganographic analysis feature. The JPEG rich model (JRM) is the image steganographic analysis method that is widely used in the JPEG domain. JRM is similar to SRM. The only difference is that the features of JRM consist of the second-order cooccurrence matrices of the block coefficients of JPEG and their residual. However, the features of SRM are composed of the fourth-order cooccurrence matrices of different kinds of filter residuals. SRM is used in the spatial domain and JRM is used in the JPEG domain. The images of JPEG format are the carriers of the steganographic algorithms in the frequency domain.

Ma et al. [13] proposed a general feature selection method based on decision rough set α-positive region reduction to reduce the dimension of steganalysis features. Their results show that the reduced feature set can obtain the detection ability comparable to the original feature set, which effectively decreases the computation cost.

2.2. The Deep Learning Method. Because of gradient vanishment/explosion, the deep networks are normally difficult to train. He et al. [14] proposed ResNets, which solve the problem of gradient vanishment/explosion in deeper networks. This means the deeper network can show better accuracy rather than degradation. However, better accuracy comes with much more computation and time.

To reduce computational complexity, Singh et al. [15] presented a deep learning architecture in which the convolution operation leverages heterogeneous kernels. They improved the convolution kernel and achieved 3× to 8× FLOPs based improvement in speed while still maintaining (and sometimes improving) the accuracy.

Currently, deep learning is widely applied to improve the steganalysis performance. Hu et al. [16] proposed a new self-seeking steganalysis method based on visual attention and deep reinforcement learning. The visual attention method selects a region from the image and deep reinforcement learning is utilized to yield a summary region. Then, the summary regions are adopted to replace the misclassified training images to improve the steganalysis performance. Their experimental results show that their method can achieve steganalysis performance comparable to the state-of-the-art steganographic detection algorithms.

3. DRHNet

The proposed network architecture is called DRHNet–Dual Residual Heterogeneous Network. The "residual" here has two meanings, one of which means that 34 layers of ResNet are used as the main network structure, and the other meaning is that the residual of the image is treated as the object. Firstly, we explain the method of preprocessing, that is, how to get the feature matrix, and the principle of discriminating embedded images with residuals and then demonstrate the architecture of the network. At last, we describe the details of the experiment.

3.1. Method and Principle. The steganographic embedding process makes subtle changes to the image, which is similar to introducing weak noise (stealth noise) into the image. At the same time, the steganographic embedding process not only changes the adjacent pixel correlation of the natural image but also changes the adjacent pixel correlation of the residual image (noise component) of the natural image. SRM and other residual image based steganographic analysis methods [17, 18] model the residual image instead of directly modeling the image itself, mainly to weaken the interference of the image content on the steganalysis feature.

The residual of the cover image or stego image is extracted with k high-pass filters to form k submodels. Then, quantize, round, and truncate each submodel and extract the cooccurrence matrix in both horizontal and vertical directions. At this time, $2k$ cooccurrence matrices are generated for each picture. Cooccurrence matrices with similar properties are symmetrically merged and all elements are rearranged into feature vectors. At this point, the feature has been obtained, and its form is as follows [12]:

$$\overrightarrow{F_{k_{c_x}}} \leftarrow \text{Range}\left(\text{Merge}\left(M_h\left(c_x\right), M_v\left(c_x\right)\right)\right), \tag{1}$$

where $\overrightarrow{F_{k_{c_x}}}$ is the feature of the x^{th} cover image c_x calculated by using the k^{th} submodel. Merge$(\cdot)$ is to make two matrices merge to be one by combining elements having the same or similar statistical laws in the horizontal cooccurrence matrices $M_h(\cdot)$ and vertical one $M_v(\cdot)$. Range$(\cdot)$ is a function to rearrange the merged matrices into a feature vector. Among them, $x = 1, 2, \ldots, z$, where z is the last image of the train set; the feature of the stego image s_x can also be calculated. $c_x, s_x \in \mathbb{Z}^{n_1 \times n_2}$ are spatial domain images of size $n_1 \times n_2$ in which each value in the matrix is between 0 and 255.

$M_h(\cdot)$ can be obtained by the following formula [12], and $M_v(\cdot)$ can be obtained in the same way:

$$M_h\left(c_{x_{i,j}}\right) \leftarrow \text{Extr}_{h_d}\left(\text{Trunc}_T\left(\text{Round}\left(\frac{\text{HP}_k\left(c_{x_{i,j}}\right)}{q}\right)\right)\right), \tag{2}$$

where the positive vertex q is a quantization factor and the positive integer T is a truncation threshold; there are two important parameters that affect the dimensionality and steganalysis performance of SRM features. d is the order of the symbiotic matrix. If d is too large, sparse features will appear. If d is too small, the statistical diversity is not rich enough. $\text{HP}_k(\cdot)$ demonstrates the residual extracted by the kth high-pass filter; the specific definition is as follows:

$$\text{HP}_k(\cdot) \leftarrow \widehat{x}_{i,j}\left(\mathcal{N}_{i,j}\right) - \partial x_{i,j}, \tag{3}$$

where $x \in \mathbb{Z}^{n_1 \times n_2}$ is the pixel value of the cover or stego image at (i, j), $\text{HP}_k(\cdot) \in \mathbb{R}^{n_1 \times n_2}$ is the residual value of the residual f obtained by using a high-pass filter on the cover or stego image at (i, j), and $\partial \in \mathbb{Z}$ is the coefficient before the pixel value $x_{i,j}$. $\mathcal{N}_{i,j}$ is the pixel value of $x_{i,j}$ neighborhood $x_{i,j} \notin \mathcal{N}_{i,j}$, and $\left\{x_{i,j}, \mathcal{N}_{i,j}\right\}$ is the support set of image residuals. $\widehat{x}_{i,j}(\cdot)$ is the calculation result of the correlation between $x_{i,j}$ and it is the neighborhood $\mathcal{N}_{i,j}$ on different filters. Round$(\cdot)$ means rounding up by element, and $\text{Trunc}_T(\cdot)$ means a truncation operation by element. $\text{Extr}_{h_d}(\cdot)$ extracts the residual as a cooccurrence matrix. The common filters used in SRM are shown in Figure 1.

We choose $q = 0.5, 1, 2$, $T = 2$, and $d = 4$ and use all the merge rules designed by SRM. The obtained SRM feature instance is called the SRMQ3 (SRM feature using 3 kinds of quantization factors). It has 106 features, 17 of which are 338-dimensional features and 89 of which are 325-dimensional features. The dimension of the RMQ3 feature is $338 \times 17 + 325 \times 89 = 34671$. We use 0, 0 as the segmentation between each feature to fill it into a feature matrix of 187×187 , and null values after the last feature in the matrix are filled with 0. It is defined as follows:

$$\text{MF}_{c_x} \leftarrow \left[\overrightarrow{F_{1_{c_x}}}, t0, 0n, q\overrightarrow{F_{2_{c_x}}}\, h_{,0}, 0x, 7 \ldots CF\overrightarrow{_{106_{c_x}}}; , 0, \ldots, 0\right]. \tag{4}$$

By observing the feature vectors, we find that there are no elements with a value of 1 in the vector. We intended to split the feature vectors by "1, 1″, but considering that the maximum pooling is used in the subsequent network design, this will cause the network to train the separator we set as an important parameter, so, finally, we use "0, 0″ as the separator.

After obtaining the feature matrix of the cover image MF_{c_x} and the stego image MF_{s_x}, our goal is to use DRHNet to train a mapping Map$(\cdot)$ based on the difference between them, so that the mapping satisfies the following equation:

$$\begin{cases} \text{Map}\left(\text{MF}_{s_x}\right) = 1, \\ \text{Map}\left(\text{MF}_{c_x}\right) = 0, \end{cases} \tag{5}$$
$$\text{Map}(\cdot) \leftarrow \text{DRHNet}\left(\text{MF}_{c_x}, \text{MF}_{s_x}\right).$$

As discussed in Section 2, the feature extraction procedure of JRM is similar to SRM. The difference is that the features of JRM consist of the second-order cooccurrence matrices of the block coefficients of JPEG and their residual. JRM will double the feature dimension through Cartesian calibration, which produces 22510 features. Finally, a 151×151 feature matrix is generated.

3.2. Dual Residual Heterogeneous Network Architecture

3.2.1. Deep Residual Network. The structure used in this paper is similar to the 34-layer structure of ResNet [14]. We also adopt batch normalization (BN) [19] after each convolution and before ReLU [20]. The difference is that we added the SRM-Extract-Merge (SRMEM) layer between the image and the first convolutional layer of DRHNet for steganographic analysis of the spatial domain; we added the JRM-Extract-Merge (JRMEM) layer between the image and the first convolutional layer of DRHNet for steganographic analysis of the frequency domain. This reduces the data dimension that the network actually handles from 256×256 to 187×187 or 151×151. And the image represented by the feature matrix is no longer the content of the image. On the contrary, it is the statistical feature of the image residual, so it is more abstract. In addition, since Adamx [21] can reach convergence faster than stochastic gradient descent (SGD), we use Adamx as the optimizer to replace SGD.

The structure of the network is shown in Figure 2. The layered structure in the figure is not just a layer of convolution, but a convolution block containing two layers of convolution. The DRHNet network parameter settings are shown in Table 1. Please note that the DRHNet used for the steganographic analysis of the spatial domain is known as S-DRHNet in the following sections and the DRHNet used for the steganographic analysis of the frequency domain is known as J-DRHNet. The S-DRHNet and J-DRHNet share similar network architecture. The only difference is that they own different feature extraction layers. The S-DRHNet adopts the SRMEM as the feature extraction layer and the J-DRHNet uses the JRMEM as the feature extraction layer.

3.2.2. Heterogeneous Kernel. Another difference of DRHNet compared to ResNet is that the HetConv [15] is used as the convolution kernel, instead of the conventional convolution kernel. The channel is filled in the order of a 3×3 and three

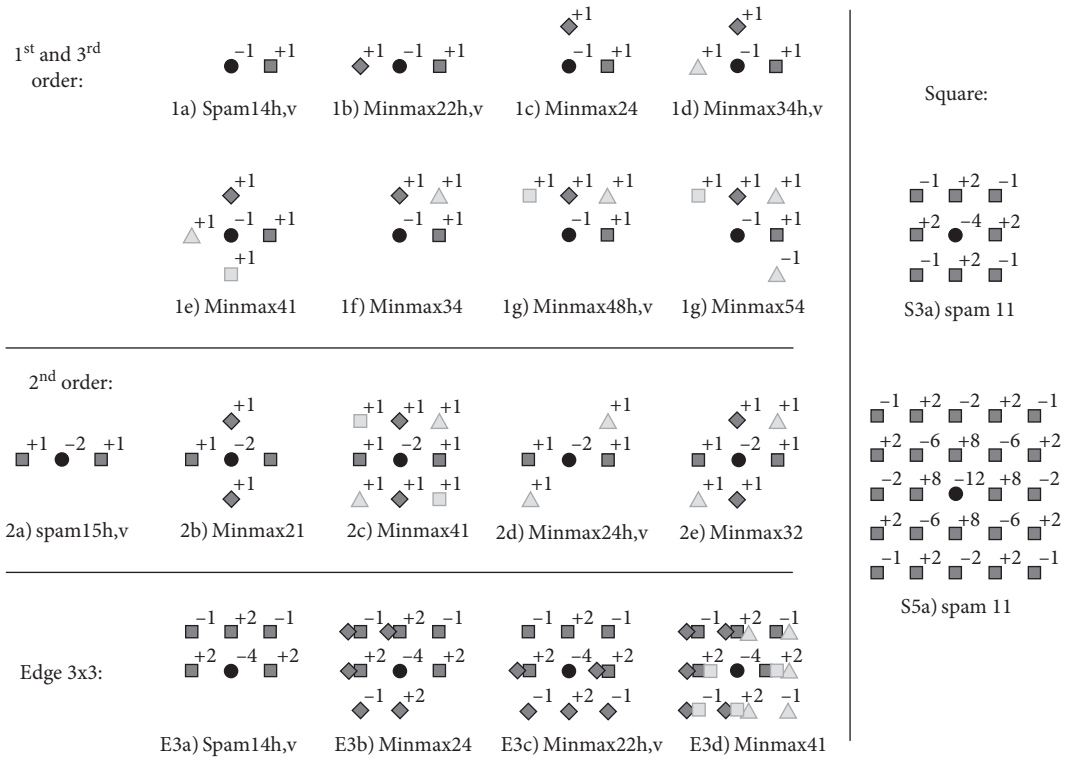

Figure 1: The commonly used filters of SRM.

1×1 convolution kernels. The convolution kernel of the next convolutional layer remains in this order, but the overall arrangement is shifted to the right by a convolution kernel. The structure of the DRHNet convolution kernel is shown in Figure 3.

It can be seen that there are two convolutional layers in the convolution block, each layer consisting of 64 convolution kernels of 3×3 and 1×1 sizes, arranged and offset in the above order. Using HetConv instead of a conventional convolution kernel can reduce network parameters and reduce computational complexity.

4. Experimental Results and Analysis

All experiments in this paper were evaluated and contrasted on BOSSbase 1.01, which contains 10,000 grayscale images with a size of 512×512. The experimental environment for this article is a host with an NVIDIA GeForce 1080 Ti graphics card and an Intel i7-9700 CPU. The more pixels image has, the more information it can embed, as well as the higher the computational complexity in the steganographic analysis process. This does not affect the performance of DRHNet, regardless of which size of the image will be extracted in the preprocessing into a feature matrix of 187×187 or 151×151. To facilitate comparison with the other steganalysis methods, we resized all the images into 256×256. In the setting of the spatial domain, WOW [22], S-UNIWARD [11], and MiPOD [23] are used as steganographic algorithms to embed ciphertext in the image. The SCA-TLU-CNN [9] and the SRNet [10] are utilized as the competing steganographic analysis method to be compared with S-DRHNet. In the setting of the frequency domain, the UED [24] and J-UNIWARD [11] are chosen as the steganographic algorithms. Because SRNet can also be used in the frequency domain, we compare J-DRHNet with SRNet to evaluate the effectiveness of DRHNet in the setting of the frequency domain. The payload of each steganographic algorithm is set from 0.2 to 0.4 bits per pixel (bpp), respectively. For the dataset, 5000 cover-stego image pairs, 10000 images, were randomly selected as the training set. For clarity of expression, the following are all counted in cover-stego image pairs. As the same, 2500 were selected as the validation set, and the remaining 2500 combined with the 2500 randomly selected unembedded image pairs were used as the test set. The experimental epoch of this paper is 100 times, the minibatch size of the training set is 20 cover-stego image pairs, and the validation set is 10. The first 80 epochs of the experiment trained the network at a learning rate of 0.001 and trained it at 0.0001 for the last 20 epochs.

4.1. DRHNet Steganalysis Level Experiment. We adopt the detection error P_{error} as the evaluation criteria. The definition of P_{error} is as follows:

$$P_{\text{error}} = \frac{n_{\text{FP}} + n_{\text{FN}}}{n}, \tag{6}$$

where n is the total number of images in the test set, and n_{FP} and n_{FN} are the number of false positive and false negative errors in the machine learning concept.

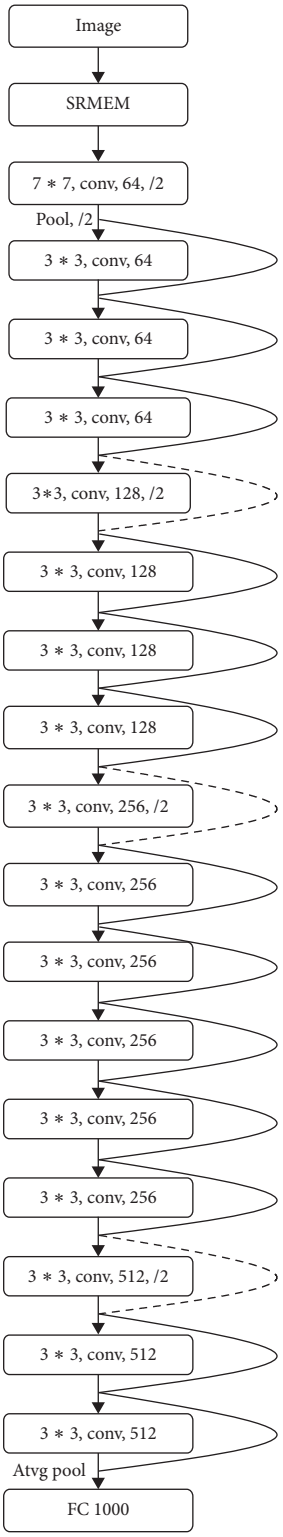

Figure 2: Network architectures for DRHNet.

4.1.1. The Performance of S-DRHNet in the Spatial Domain. The performance of S-DRHNet in the spatial domain is shown in Figures 4–6. It is observed in Figures 4–6 that SRNet's [10] P_{error} is less than 1% lower than our proposed structure when applying the WOW steganography algorithm and with a payload of 0.4 bpp; SCA-TLU-CNN [9] has a detection error rate of 6% lower than the DRHNet when applying the S-UNIWARD steganography algorithm and with a payload of 0.2 bpp. Apart from the above two situations, DRHNet generally has better performance than the other two steganographic analysis networks. And it can be seen that as the payload increases, P_{error} of DRHNet decreases faster than the other two networks.

The ROC curves of SCA-TLU-CNN, SRNet, and S-DRHNet against S-UNIWARD at 0.4 bpp are shown in Figure 7. The AUC of S-DRHNet, SRNet, and SCA-TLU-CNN are 0.97, 0.94, and 0.92. The accuracy of S-DRHNet against S-UNIWARD is higher than SRNet and SCA-TLU-CNN at the high payload.

The training of DRHNet was iterated 100 times in total. The learning rate was set to 0.001 in the first 80 iterations and 0.0001 in the last 20 iterations. Figure 8 shows the changes in detection error during training and validation of S-DRHNet. The data are obtained on S-UNIWARD at payload 0.4 bpp.

Table 1: Parameter s for DRHNet.

Layer name	Output size	Convolution kernel
Conv_1	94×94	$7 \times 7, 64$
Max pool		
Conv_2	47×47	$\left[\begin{pmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{pmatrix} \times 3\right]$
Conv_3	24×24	$\left[\begin{pmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{pmatrix} \times 4\right]$
Conv_4	12×12	$\left[\begin{pmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{pmatrix} \times 6\right]$
Conv_5	6×6	$\left[\begin{pmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{pmatrix} \times 3\right]$
Avg pool		
Fc 1000		
Softmax		

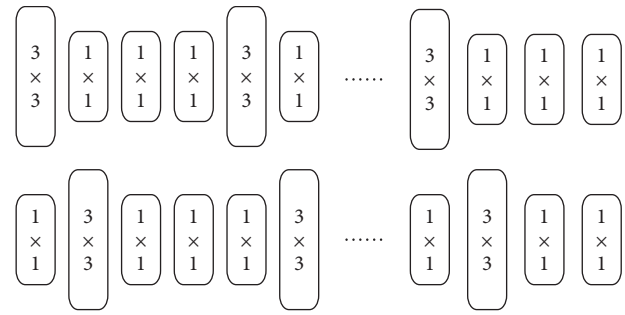

Figure 3: $3 \times 3 \times 64$ convolutional block structure in DRHNet.

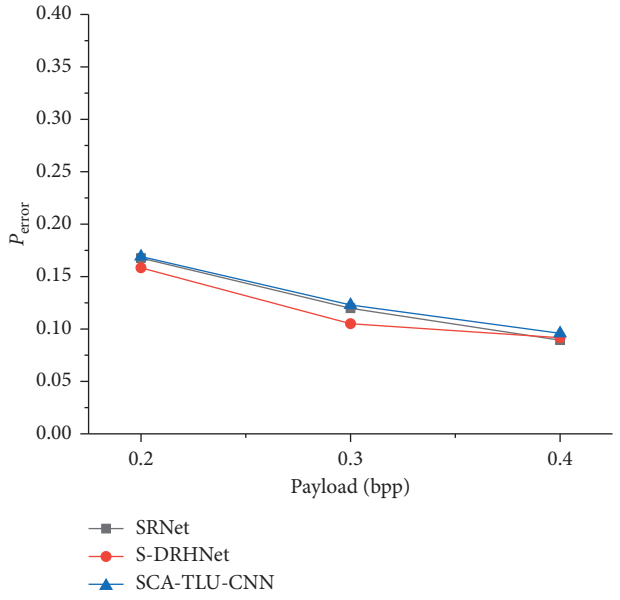

Figure 4: Detection error P_{error} of SCA-TLU-CNN, SRNet, and S-DRHNet against WOW.

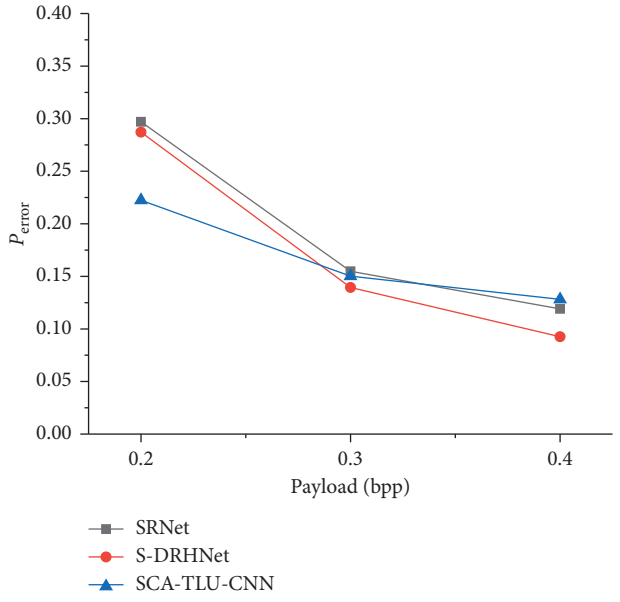

Figure 5: Detection error P_{error} of SCA-TLU-CNN, SRNet, and S-DRHNet against S-UNIWARD.

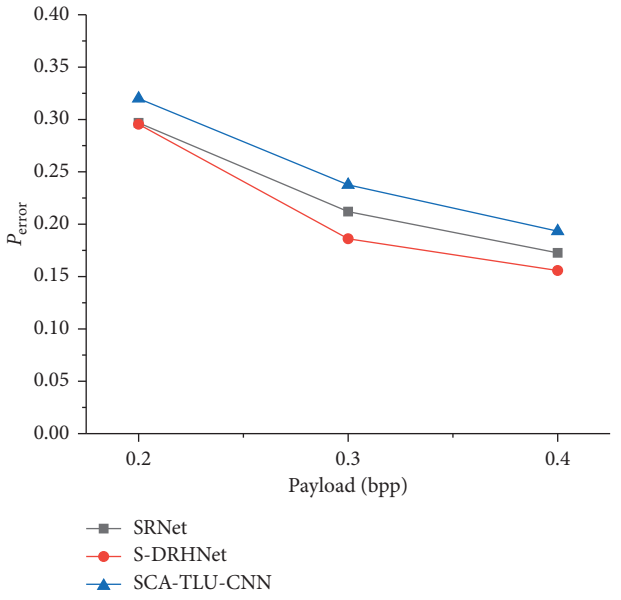

Figure 6: Detection error P_{error} of SCA-TLU-CNN, SRNet, and S-DRHNet against MiPOD.

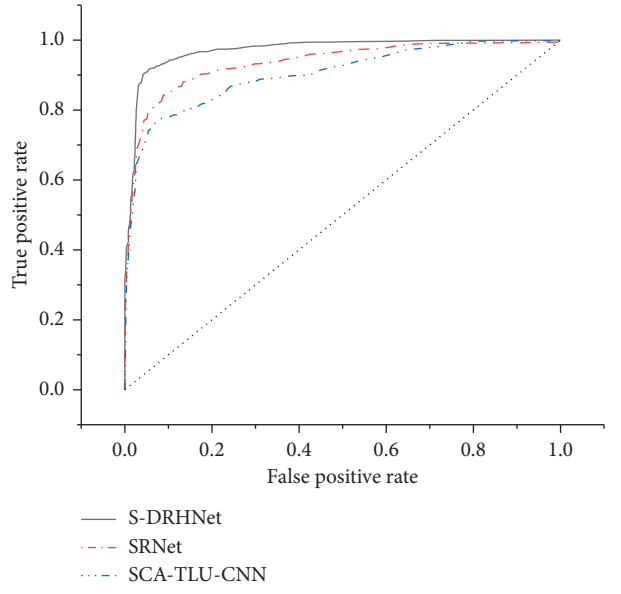

Figure 7: ROC curves of SCA-TLU-CNN, SRNet, and S-DRHNet for S-UNIWARD at 0.4 bpp.

Figure 9 shows the progression of the training and validation loss when training S-DRHNet in the same circumstance. Because the curves of detection error and loss of J-DRHNet during training and validation are similar to S-DRHNet, we only show the corresponding curves of S-DRHNet here.

4.1.2. The Performance of J-DRHNet in the Frequency Domain. The performance of J-DRHNet in the frequency domain is shown in Figures 10 and 11. The J-DRHNet shows better performance than SRNet against J-UNIWARD. When the payload is high, the J-DRHNet shows better performance than SRNet against UED.

The ROC curves of SRNet and J-DRHNet against UED at 0.4 bpp are shown in Figure 10. We can observe from Figure 12 that the accuracy of J-DRHNet against UED is close to that of SRNet.

4.2. DRHNet Steganalysis Generality Experiment. Using the feature matrix extracted by SRM or JRM as the input of the network makes DRHNet have good versatility. To evaluate the versatility of DRHNet, we adopt one steganographic

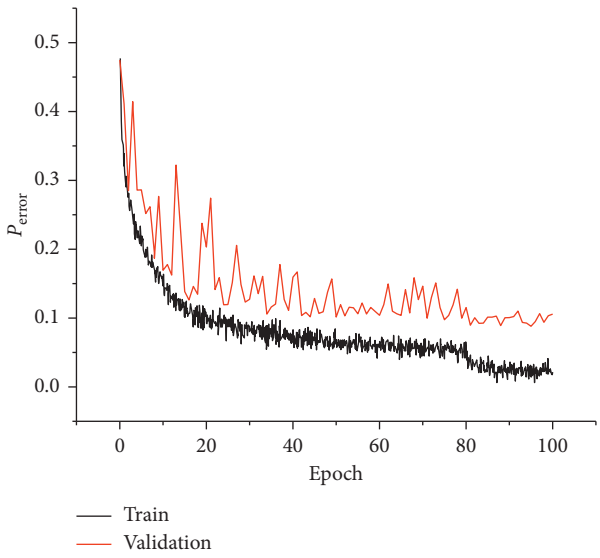

Figure 8: S-DRHNet's training and validation detection error P_{error} for S-UNIWARD at 0.4 bpp.

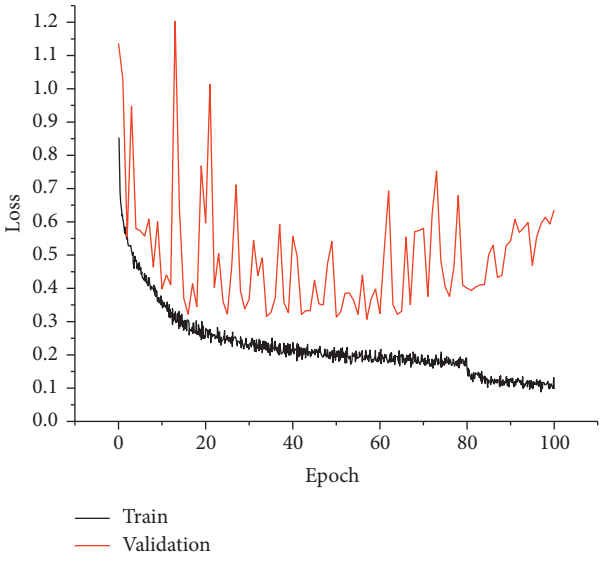

Figure 9: S-DRHNet's training and validation loss for S-UNIWARD at 0.4 bpp.

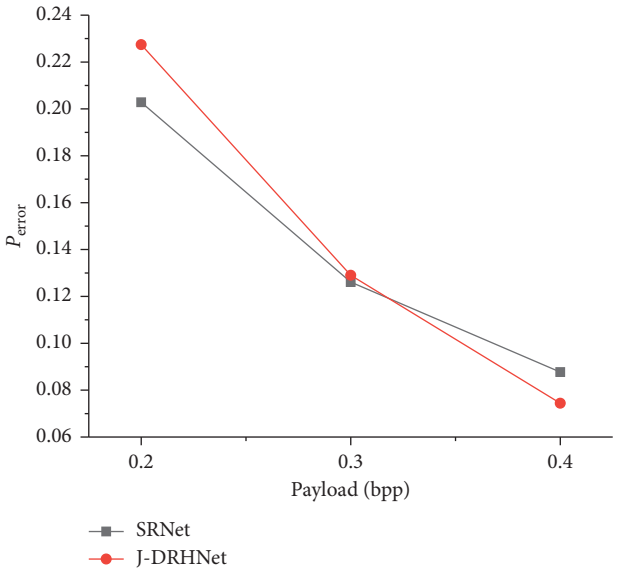

Figure 10: Detection error P_{error} of SRNet and J-DRHNet against UED.

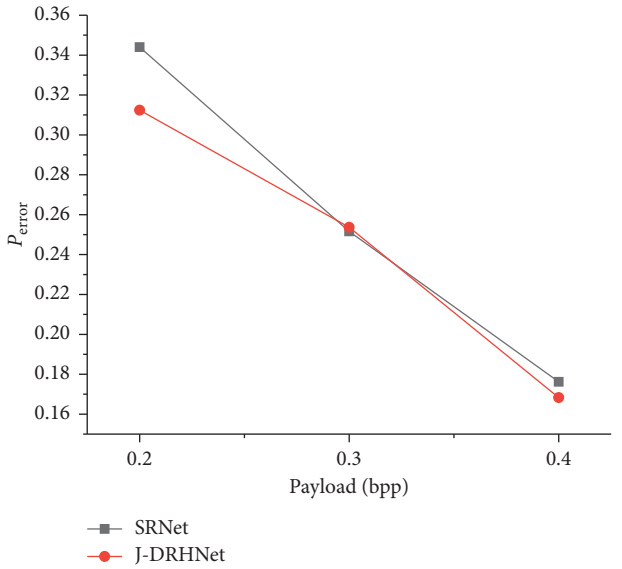

Figure 11: Detection error P_{error} of SRNet and J-DRHNet against J-UNIWARD.

algorithm to generate steganographic images as the training set and validation set. Then, the steganographic images generated by another steganographic algorithm are used as the test set. We observe the detection error of DRHNet in this situation. Because the J-DRHNet and S-DRHNet own similar architecture except for the feature extraction modules, we only show the results of the cross test of S-DRHNet here. The detection error of the cross test of S-DRHNet is shown in Table 2. The detection error obtained by using the same steganography algorithm for the test set and training set is shown in bold in Table 2, and the error rate of the cross test is shown in normal font. The detection error obtained by adopting different steganographic algorithms is generally 0.01–0.03 higher than that of using the same steganography algorithm. From the results in Table 2, we can conclude that the S-DRHNet shows good versatility against different steganographic methods.

4.3. The Time Consumption and Computational Complexity of DRHNet. DRHNet also reduces time consumption while improving accuracy. Table 3 shows the parameters, computational complexity, and time consumption of the above four types of steganalysis networks. The network structure of

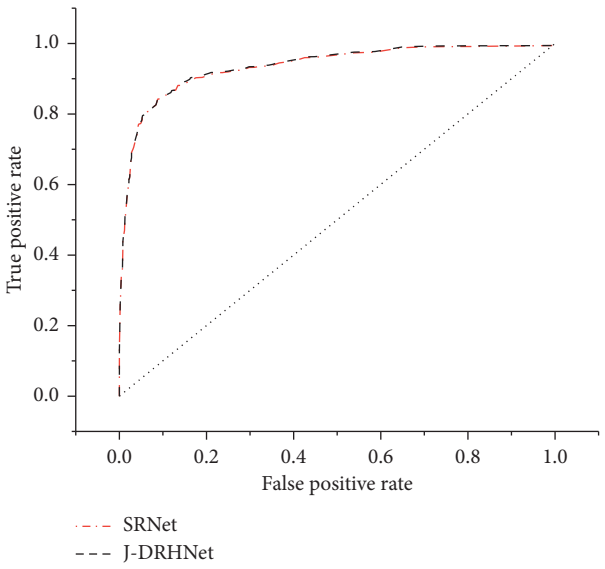

Figure 12: ROC curves of SRNet and J-DRHNet against UED at 0.4 bpp.

Table 2: The detection error of the cross test of S-DRHNet.

Train\test	Payload (bpp)	WOW	S-UNIWARD	MiPOD
WOW	0.2	**0.1584**	0.1807	0.1802
	0.3	**0.1050**	0.1268	0.1341
	0.4	**0.0917**	0.1099	0.1125
S-UNIWARD	0.2	0.2943	**0.2872**	0.2960
	0.3	0.1496	**0.1395**	0.1561
	0.4	0.1083	**0.0926**	0.1069
MiPOD	0.2	0.3182	0.3297	**0.2954**
	0.3	0.2049	0.2088	**0.1861**
	0.4	0.1609	0.1653	**0.1557**

Table 3: The computational complexity of the three networks.

Network	Input resolution	Parameters (M)	FLOPs (G)	Hours
SCA-LTU-CNN	256×256	10.52	1.74	44
SRNet	256×256	12.18	2.09	53
S-DRHNet	$\mathbf{256 \times 256}$	**9.88**	**1.67**	**41**
J-DRHNet	$\mathbf{256 \times 256}$	**9.88**	**1.59**	**37**

SCA-TLU-CNN has ten layers, and the SRNet has twelve layers, so ResNet is higher than SCA-TLU-CNN in the three metrics mentioned in Table 3. Although the DRHNet designed in this paper is a 34-layer network structure, the application of HetCov as a convolution kernel greatly reduces the parameters of the network, resulting in shortened computational complexity and time consumption than the other two networks, while still ensuring considerable accuracy. Because the feature matrix of J-DRHNet is smaller than S-DRHNet, the time consumption of J-DRHNet is lower than that of S-DRHNet.

5. Conclusion

In this paper, a deep neural network with high accuracy and low time consumption is proposed for steganalysis. The SRMEM and JRMEM layers are used to extract features from the original images and combine them into a feature matrix, which provides versatility for the steganographic analysis method while reducing the network dimension. Furthermore, we select HetConv as the convolution kernel of the DRHNet network, which greatly reduces the computational complexity while ensuring accuracy. By combining different feature preprocessing modules, that is, SRMEM and JRMEM, DRHNet can be flexibly applied in both spatial domain and frequency domain. The preliminary experimental results show that the DRHNet shows excellent steganalysis performance in both the spatial domain and frequency domain. The DRHNet outperforms the existing state-of-the-art steganographic analysis algorithms such as SCA-TLU-CNN and SRNet and shows excellent performance against the state-of-the-art steganographic methods such as S-UNIWARD, J-UNIWARD, and WOW. The image embedded in ciphertext may be compressed during transmission.

The cross test of SRNet and J-DRHNet will be further studied and verified in the following research. How to extract the embedded image after compression will be our next research direction.

Data Availability

The software code used to support the findings of this study is available from the corresponding author upon request. The data used to support the findings of this study are available at http://agents.fel.cvut.cz/stegodata/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China under Grant no. U1831131, the Special Funds of Central Government of China for Guiding Local Science and Technology Development under Grant no. [2018]4008, and the Science and Technology Planned Project of Guizhou Province, China, under Grant no. [2020]2Y013.

References

[1] Z. Xia, X. Wang, X. Sun, and B. Wang, "Steganalysis of least significant bit matching using multi-order differences," *Security and Communication Networks*, vol. 7, no. 8, pp. 1283–1291, 2014.

[2] S. Gupta, A. Goyal, and B. Bhushan, "Information hiding using least significant bit steganography and cryptogrammmphy," *International Journal of Modern Education and Computer Science*, vol. 4, no. 6, p. 27, 2012.

[3] C. Chen and Y. Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," in *Proceedings of the*

2008 IEEE International Symposium on Circuits and Systems, pp. 3029–3032, IEEE, Washington, DC, USA, June 2008.

[4] Q. Liu, "Steganalysis of DCT-embedding based adaptive steganography and YASS," in *Proceedings of the Thirteenth ACM Multimedia Workshop On Multimedia and Security*, pp. 77–86, ACM, New York, USA, September 2011.

[5] H. Li, Z. Lin, X. Shen et al., "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5325–5334, Boston, MA, USA, June 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

[7] M. Rastegari, V. Ordonez, J. Redmon et al., "XNOR-Net: Imagenet Classification Using Binary Convolutional Neural networks," in *Proceedings of the European Conference on Computer Vision*, pp. 525–542, Springer, Amsterdam, The Netherlands, October 2016.

[8] Y. Qian, J. Dong, W. Wang et al., "Deep learning for steganalysis via convolutional neural networks," *International Society for Optics and Photonics*, vol. 9409, Article ID 94090J, 2015.

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

[10] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

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

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

[13] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set α-positive region reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 336–350, 2019.

[14] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[15] P. Singh, V. K. Verma, P. Rai et al., "HetConv: heterogeneous kernel-based convolutions for deep CNNs," 2019, http://arxiv.org/abs/1903.04120.

[16] D. Hu, S. Zhou, Q. Shen, S. Zheng, Z. Zhao, and Y. Fan, "Digital image steganalysis based on visual attention and deep reinforcement learning," *IEEE Access*, vol. 7, pp. 25924–25935, 2019.

[17] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[18] K. Jan and J. Fridrich, "Steganalysis in high dimensions: fusing classifiers built on random subspaces," *Proceedings of SPIE-The International Society for Optical Engineering*, vol. 7880, no. 1, pp. 181–97, 2011.

[19] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, http://arxiv.org/abs/1502.03167.

[20] B. Xu, N. Wang, T. Chen et al., "Empirical evaluation of rectified activations in convolutional network," 2015, http://arxiv.org/abs/1505.00853.

[21] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[22] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings 2012 IEEE International workshop on information forensics and security (WIFS)*, pp. 234–239, IEEE, Costa Adeje, Spain, December 2012.

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

[24] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Transactions on In-Formation Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.