

Research Article

Adaptive Steganalysis Based on Selection Region and Combined Convolutional Neural Networks

Donghui Hu,¹ Qiang Shen,¹ Shengnan Zhou,¹ Xueliang Liu,¹ Yuqi Fan,¹ and Lina Wang²

¹School of Computer and Information, Hefei University of Technology, Hefei 230009, China

²School of Computer and Information, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Donghui Hu; hudh@hfut.edu.cn

Received 19 September 2017; Accepted 23 October 2017; Published 12 November 2017

Academic Editor: Zhenxing Qian

Copyright © 2017 Donghui Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Digital image steganalysis is the art of detecting the presence of information hiding in carrier images. When detecting recently developed adaptive image steganography methods, state-of-art steganalysis methods cannot achieve satisfactory detection accuracy, because the adaptive steganography methods can adaptively embed information into regions with rich textures via the guidance of distortion function and thus make the effective steganalysis features hard to be extracted. Inspired by the promising success which convolutional neural network (CNN) has achieved in the fields of digital image analysis, increasing researchers are devoted to designing CNN based steganalysis methods. But as for detecting adaptive steganography methods, the results achieved by CNN based methods are still far from expected. In this paper, we propose a hybrid approach by designing a region selection method and a new CNN framework. In order to make the CNN focus on the regions with complex textures, we design a region selection method by finding a region with the maximal sum of the embedding probabilities. To evolve more diverse and effective steganalysis features, we design a new CNN framework consisting of three separate subnets with independent structure and configuration parameters and then merge and split the three subnets repeatedly. Experimental results indicate that our approach can lead to performance improvement in detecting adaptive steganography.

1. Introduction

Steganography is a technique for embedding confidential information into multimedia data, which can be used for concealed transmission or copyright protection. Steganalysis is an opposite art to detect the existence of steganography. In past years, researchers have developed a variety of information steganography techniques without affecting image quality. LSB (Least Significant Bit) [1] is a nonadaptive method which does not take into account the contribution of each pixel within an image when embedding information, so that it is proved to be defective with the development of the detection technology.

Some adaptive steganography techniques have been proposed to improve the antidetection ability by adjusting the embedding locations on the basis of the embedding costs. Many currently available adaptive steganography algorithms, such as HUGO BD (Highly Undetectable Stego Bounding

Distortion) [2], WOW (Wavelet Obtained Weights) [3], S-UNIWARD (Spatial-Universal Wavelet Relative Distortion) [4], and HILL (High-pass, Low-pass, and Low-pass) [5], have a high antidetection capability, and most of them are designed under the framework of minimizing a distortion function, in which each pixel of an image suitable for embedding information is firstly assigned a high embedding cost. With the cost we can calculate the value of distortion function and obtain the stego via minimizing the distortion function using some coding techniques, such as STCs (Syndrome-Trellis Codes) [6]. To detect content-adaptive schemes, some researchers handcraft various high-dimensional features such as the spatial rich model (SRM) [7–9] and selection channel aware maxSRM [10] and maxSRMd2 [10]. Some other work focuses on designing efficient convolutional neural network (CNN) architectures [11] to extract features directly from the input images. Qian et al. [12] proposed a CNN based steganalysis method using Gaussian activation function and

in [13] they transfer features generated from a pretrained model to regularize CNN model. In literatures [14, 15], Xu et al. proposed a new-type network structure, including absolute value layer and batch normalization layer. In [16, 17], hybrid networks composing many subnetworks are designed to fit the rich-model features set. However, these CNN based steganalysis methods do not consider the characteristics of adaptive steganography and hence have limitation in evolving diverse and effective steganalysis features due to the inflexible CNN frameworks. Recently there are some other articles on steganalysis based on deep learning. Ni et al. [18] proposed a CNN which has a quite different structure from the ones used in conventional computer vision tasks. Besides, a new activation function called truncated linear unit (TLU) is adopted in the CNN model and it incorporates the selection channel inside the network architecture. In [19], Yang et al. proposed another approach towards using selection channel inside the network architecture which can improve the detection accuracy. Compared to the existing CNN model for steganalysis, our model can evolve effective steganalysis features for the detection of adaptive steganography by merging and separating subnets of CNNs and focusing on a most effective region (MER), which can be the reference in designing the new CNN for steganalysis.

The contribution of this paper is as follows: (1) we design a region selection method to find the most effective region (MER) by calculating and comparing the sum of the embedding probabilities of each pixel in a region. The selected region is used as an input image of CNNs. (2) We propose a network consisting of three separate subnets, with each subnet possessing independent structures and diverse parameters, and the three subnets can be merged and separated repeatedly because some studies have shown that widening the network can significantly improve the performance [20]. Experimental results indicate that both the region selection method and the proposed CNN framework can lead to performance improvement in detecting adaptive steganography in some cases.

2. Proposed Method

Most of the state-of-the-art adaptive steganography methods first assign the distortion value for each pixel via a distortion function based on the embedding cost before embedding information, and then some advanced coding techniques, such as STCs, will be applied to minimize the expected distortion value for all pixels in texture areas. Obviously, since the degree of correlation between each pixel and the surrounding pixels is different, according to the adaptive steganography distortion function, pixels with different texture complexity will get diverse loss values. Hence some pixels, which may not be suitable for modification, in texture areas, are assigned with low costs and some other pixels of complex texture are assigned with high costs. We can also clearly study from [8] that the regions with high probability are substantially consistent with the embedding pixels. By large number of experiments (which will be described in Section 3.2.1), we also found that the embedding probability map for each adaptive steganography algorithm is approximately the same.

Through this observation, we intend to roughly estimate the position of the modified pixels by embedding probability maps without knowing the specific modification point. Moreover, it is imprecise to use the embedding probability immediately to represent whether a pixel is embedded. Due to the fact that the optimal embedding region obtained by an embedding algorithm may be applicable to other embedding algorithms, in this paper, we use the most effective region instead of embedding probability of a pixel to steganalyze an image.

2.1. The Region Selection Method. In this section we first propose a method to predict embedded pixels using the embedding probability maps to find the most effective region. Our main idea is to first calculate the embedding probability of each pixel and then calculate and compare the sum of the embedding probability of all pixels in different regions to find the maximum one.

In the adaptive steganography, to determine whether a pixel is suitable for modification, a distortion metric [3] $D(X, Y)$ is designed to measure the embedding impact as follows:

$$D(X, Y) = \sum_{i=1, j=1}^{n, m} \rho_{i, j} (x_{i, j}, y_{i, j}), \quad (1)$$

where $\rho_{i, j}$ is the cost of pixel changes (from $x_{i, j}$ to $y_{i, j}$). Using the distortion function one can easily evaluate the expected distortion and compute the probabilities map. The probability of modification described in [2, 21] can be calculated by

$$\pi_{\lambda}(Y) = \prod_{i=1, j=1}^{n, m} \frac{\exp(-\lambda \rho_{i, j}(y_{i, j}))}{\sum_{y_{i, j} \in \mathcal{S}_{i, j}} \exp(-\lambda \rho_{i, j}(y_{i, j}))}. \quad (2)$$

where λ is used to satisfy the following distortion constraints:

$$D_{\epsilon} = \sum_{i=1, j=1}^{n, m} \sum_{y_{i, j} \in \mathcal{S}_{i, j}} \pi_{\lambda}(y_{i, j}) \rho_{i, j}(y_{i, j}), \quad (3)$$

or the payload constraint:

$$s = - \sum_{i=1, j=1}^{n, m} \sum_{y_{i, j} \in \mathcal{S}_{i, j}} \pi_{\lambda}(y_{i, j}) \log \pi_{\lambda}(y_{i, j}). \quad (4)$$

As shown in Algorithm 1, we estimate the embedding probability of each pixel in an image, enumerate all possible regions (with constant sizes), and calculate the sum of the probabilities of each region to find the region with the maximal value. This region is the most effective region for steganalysis. Figure 1 gives an experimental result of the method. The method is constrained by the selection channel [22] elements of embedding algorithms and embedding rates.

Due to most of the adaptive steganography methods having similar embedding positions, the inaccuracies in the selection channel do not have an impact on the position of MERs. The main reason is probably the following few points. When calculating the embedding probabilities of an image

Input: input parameters: an input image I , the width W and the height H of the region.
Output: output result: the MER O , save as PGM format.

- (1) Initialize the probability map (matrix) P with random weights θ ;
- (2) //Select a pixel from the input image and calculate the probability.
- (3) **for** $i = 1$, Rows **do**
- (4) **for** $j = 1$, Column **do**
- (5) Calculate change costs $D(X, Y)$ using Eq. (1);
- (6) Compute λ using Eq. (4);
- (7) Set the probability $\pi_\lambda(X, Y) = f(D(X, Y), \lambda)$ using Eq. (2);
- (8) Store $\pi_\lambda(X, Y)$ in P ;
- (9) Initialize the MER O with 0;
- (10) //Select the upper left corner coordinates of the area with size of $W \times H$.
- (11) **for** $i = 1$, Rows- W **do**
- (12) **for** $j = 1$, Column- H **do**
- (13) Calculate the sum of the probability in an matrix ($W \times H$) with top-left corner of (i, j) ;
- (14) Statistics out all of the sum and its corresponding i, j ;
- (15) Select the maximal value of those sums;
- (16) Cut the $W \times H$ area in an input image according to i and j ;
- (17) Save this area O as a PGM format image;

ALGORITHM 1: Finding the most effective region ($W \times H$) for steganalysis.

TABLE 1: The ratios of the distance between two MERs extracted by different algorithms.

Algorithm	Distance (the unit for location is pixel).				
	$D = 0$	$0 < D \leq 30$	$30 < D \leq 70$	$70 < D \leq 100$	$D > 100$
WOW 0.2 & HILL 0.2	0.2575	0.5025	0.1182	0.0472	0.0746
WOW 0.2 & HUGO 0.2	0.2544	0.4900	0.1271	0.0489	0.0796
HILL 0.2 & HUGO 0.2	0.2284	0.4137	0.1553	0.0680	0.1346
WOW 0.1 & 0.2	0.4544	0.4534	0.0525	0.0169	0.0228
HILL 0.1 & 0.2	0.4879	0.4421	0.0372	0.0127	0.0201
HUGO 0.1 & 0.2	0.4642	0.4282	0.0603	0.0200	0.0273

by our region selection method as well as in the extraction process of maxSRM or maxSRMd2, we may need to know the embedding algorithm and embedding rate in advance. Having a priori knowledge can obviously make the model have some limitations when we test the unknown image sets. So we test the robustness of our proposed method to reduce the impact of prior knowledge and we introduce the distance D to denote the distance between the two MERs of images embedding with different algorithms or rates and use the coordinates of the points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ in the upper left corner of different MERs to mark their position in the original images. D is calculated as follows:

$$|D| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (5)$$

Our experiments given in Section 3 also show that when using different embedding algorithms and embedding rates to estimate the embedding probability, the positions of MERs selected by Algorithm 1 are very close. We would like to use Table 1 to explain that our method is robust and universality. One can even use some kind of embedding algorithm (HILL) to roughly estimate other algorithms (WOW, HUGO BD, S-UNIWARD, etc.), which does not affect the performance. Finally, the selected MERs are used as the input of the CNN

model in the following subsection so that the CNN model can focus on the regions with high embedding probabilities.

2.2. The Combined Network

2.2.1. The Overall Structure. Figure 2 shows the network model designed in this paper, including preprocessing module, the convolution and downsampling module, and classification module. Our method broadens the network proposed in [14] with three separate subnets and lengthens the network with “depthconcat” layers, fully connected layers, and dropout layers. In the preprocessing module, the high-pass filter layer uses the KV filter kernel [7, 12], “SQUARE5x5,” to obtain the image residuals. After kernel filtering, the data flow into the combined network comprising three independent subnets where the kernel sizes of each convolution layer in each subnet are 1, 3, and 5, respectively. After several convolution and downsampling layers, the data of the three subnets are merged together through the concat layer *DepthConcat* and flow into a new combined network comprising three subnets, where kernel sizes of the first convolution layer of the three subnets are 1, 3, and 1, respectively. Finally the features are merged for the second time. After two merging operations, the network becomes a single one, including two

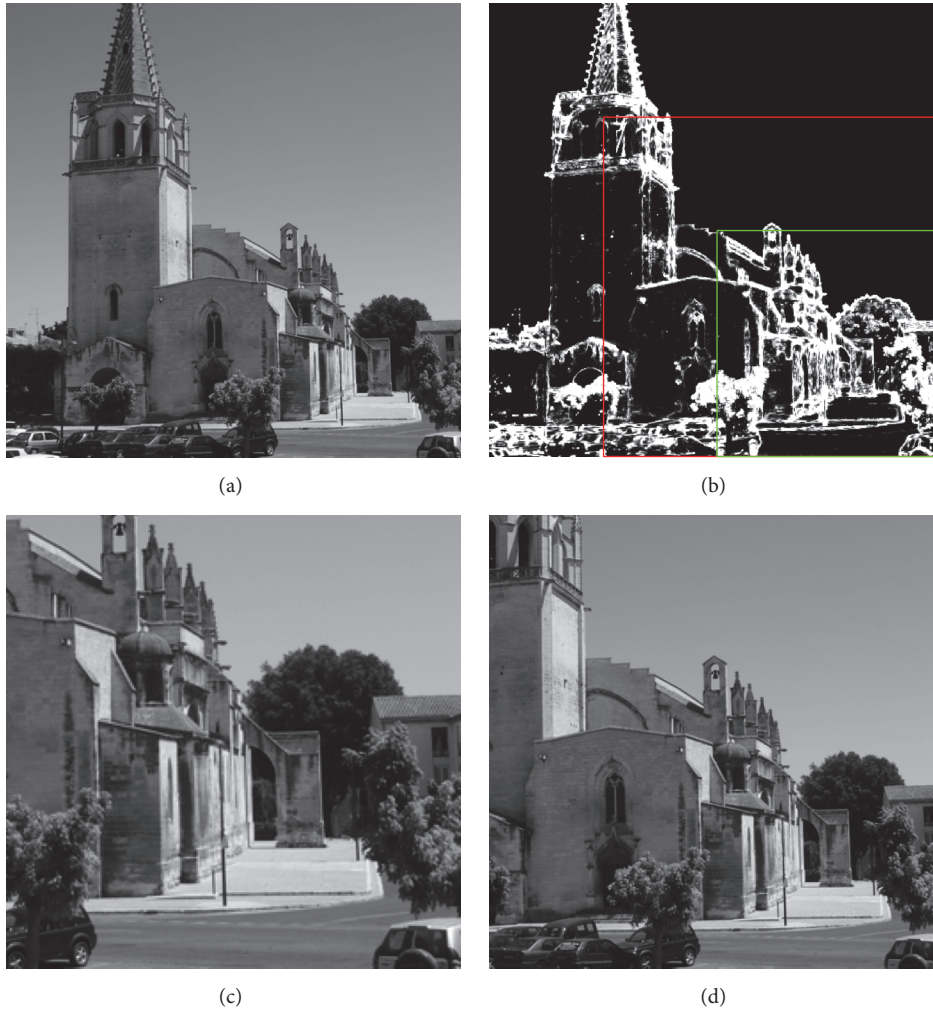


FIGURE 1: The MER for images embedded by HUGO with payload of 0.2 bits per pixel. (a) Cover image. (b) The embedding probability map. (c) The MER with size 256×256 pixels (the region with green line in (b)). (d) The MER with size 384×384 pixels (the region with red line in (b)).

convolution layers, two pooling layers, three full connection layers, two dropout layers, and four activation layers, and so forth. The configuration parameters of each layer are given in detail in Figure 2. Taking the first convolution layer as an example, the value 8 outside the parentheses is the number of the kernels, 1×1 is the size of the kernel, 1(S) is the stride, and the rest of the configuration is entirely the same. It should be noticed that the kernel size of the last pooling layer is not fixed. When training the areas with different sizes, the dimensions are different in the pooling layer so that we need to dynamically modify the kernel size of the last pooling layer to ensure that the data size is 1×1 before feeding the data to the fully connected layer.

2.2.2. Learning Features from the Network. It is widely recognized that when most of the neural networks are employed for generating the final features, original images are trained by the convolution layer and downsampling layer, and so forth. The purpose of convolution process is to model a correlation between a pixel and its surrounding pixels. For instance, if you

set the convolution kernel size to 3, it means that you want to associate a pixel with the 8 points around it (to generate steganalysis feature). Through the establishment of this link, after layers of dimensionality reduction, the final classification can be achieved. At present, the kernel size of each layer is fixed in the design of most neural networks. The large size of the convolution kernel may cause information redundancy; that is, a pixel may establish connections with less relevant pixels. The small size of the convolution kernel may lead to missing some significant information (pixels). Many neural network framework needs to set up a configuration file. Once these parameters have been set, they cannot be arbitrarily modified after training begins. Accordingly, considering the limitations of the single model, we employ three subnets with independent structures to model the feature map with different filter sizes. As described in Section 2.2.1, the two “separating-merging” procedures make the learned features further diversified. In this way, features with different grain sizes can be generated. At the bottom of the structure we add three fully connected layers, among these layers, and we

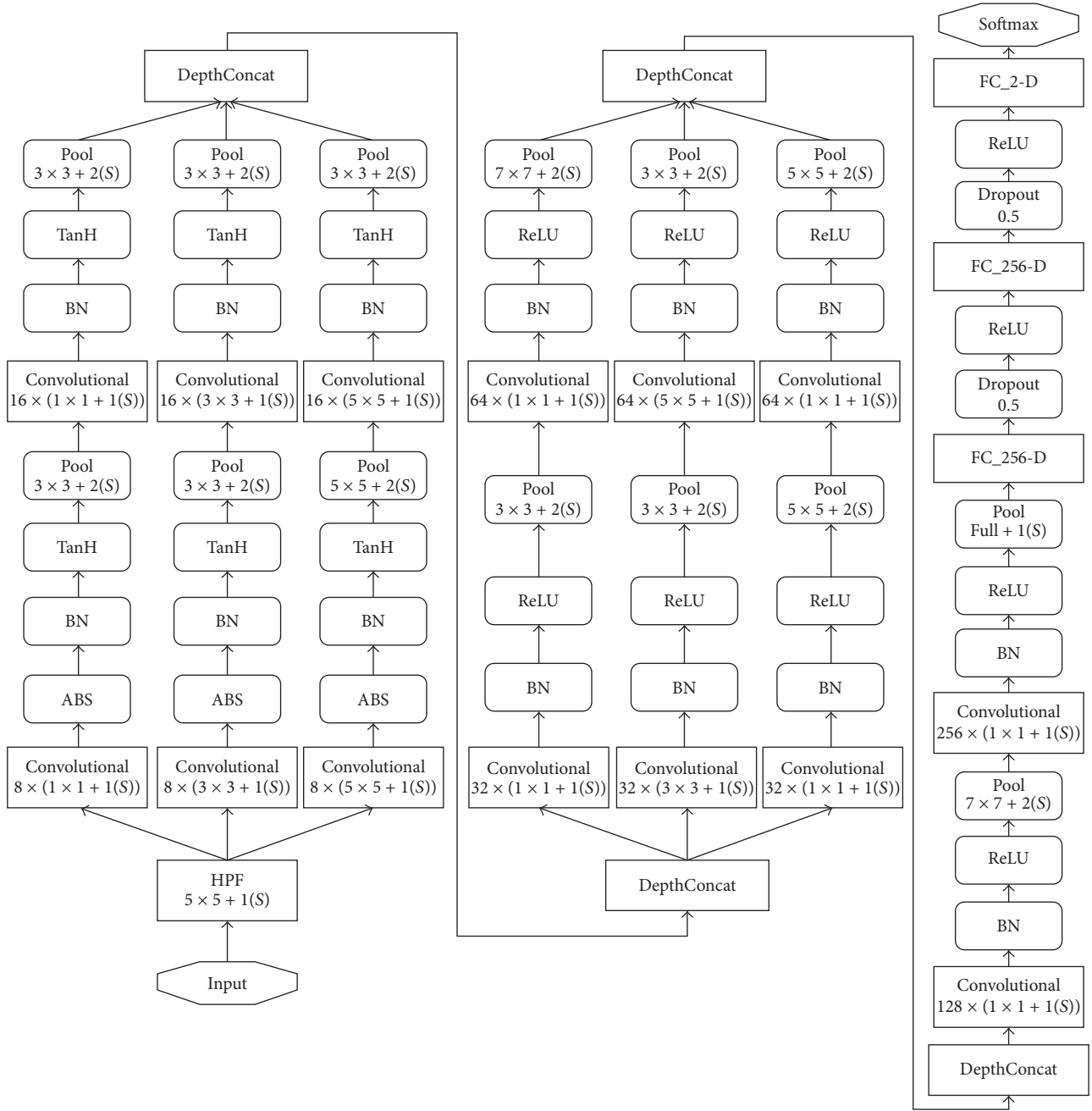


FIGURE 2: An improved model structure consisting of three separate subnets.

use the dropout layer [23] to sparse the coefficients of the network. “Dropout” means that the weights of some random nodes of the hidden layer are not allowed to work when the model is trained but may work in the next iteration. The mathematical formula is shown as follows:

$$Y = \begin{cases} W|_p * X, & \text{training} \\ W * pX, & \text{testing,} \end{cases} \quad (6)$$

where X , Y are the input and output, respectively, parameter p is the retaining probability of the dropout layer, and $W|_p$ is the

subset of weight parameters directly sampled by probability p . During the training phase and testing phase, the output is slightly different as shown in (6).

In training, we calculate the loss and gradient according to the existing label and use the gradient to update the network parameters. In this paper, we select the stochastic gradient descent (SGD) to optimize the parameters of the proposed network architecture, which has several advantages such as high efficiency, fast speed, and simplifying the problem complexity. Deep learning has a limitation that we need to do the parameters initialization manually, which is crucial

to the final results, such as the learning rate, weight decay, the initialization strategy of the parameters, and the ratio of the dropout layer, so that adjusting these parameters is time-consuming. The batch normalization layers [24] are adopted in this model to solve this problem. The normalization process is performed via (7) and (8):

$$\hat{x}_i = \frac{x_i - E[X]}{\sqrt{\sigma_X^2 + \epsilon}}, \quad (7)$$

where values x_i is an input over a minibatch of $X = \{x_1, x_2, \dots, x_m\}$. $E[X]$ and σ_X^2 are the mean and variance of X , respectively. The output values are calculated by scaling and shifting the normalized x_i as shown in

$$\hat{y}_i = \gamma \hat{x}_i + \beta, \quad (8)$$

where γ and β are learned by the network. Finally the last softmax layer is used for classification.

3. Experiments

3.1. Dataset and Settings. The dataset used in this paper is BOSSbase v1.01 [25] containing 10,000 grayscale images with size of 512×512 . We test four adaptive steganographic algorithms: HOGU BD [2], S-UNIWARD [4], WOW [3], and HILL [5] with default parameters during embedding information. We also use these algorithms to get the probability maps. For comparison, we use SRM [7] and maxSRMd2 [10] to extract 34,671-dimension feature and classify them with an ensemble classifier [26]. We use the open source framework Caffe [27] to implement the proposed model. In our approach, among which 5000 pairs of images are selected for training, 2500 pairs are selected as the validation set used to optimize the parameters of the model during the training process, and the remaining 2500 pairs are used as test set to evaluate the efficiencies of classification. We use two high performance graphics cards, NVIDIA Geforce GTX TITAN X and NVIDIA Quadro K5200, to speed up the computation and optimization, because the GPU platform is primarily based on the CUDA framework [28] of NVIDIA company which is widely used in scientific computing and can extremely improve the speed of operation at present effectively.

When training the model, considering the sensitivity of ReLU activation function to the learning rate, we set the initial learning rate to 0.001. For every 50,000 iterations, the learning rate is modified as one-tenth of the current learning rate. The maximum number of iterations is set to 150,000. The momentum used in gradient descent and the weight decay used to prevent overfitting are set to 0.9 and 0.0005, respectively. Note that when the high-pass filter layer in the model is used to obtain the image residuals, it does not need backpropagation process. Consequently, both the learning rate and the weight decay of this layer are set as 0. Except the fully connected layers, the weight decay functions of the other layers are prohibited. For the sake of maximizing the retention of the original image information, we adopt the mean-pooling strategy in all pooling layers. The ratio of

dropout layer between each fully connected layer is set to 0.5. Apart from this exception, the learning rate *lr_mult* (a parameter in Caffe) of weight in all layers is set to 1 (the real learning rate is *lr_mult* multiplied by initial learning rate 0.001) and the learning rate *decay_mult* of bias is twice of the above value *lr_mult*.

3.2. Results and Analysis. Table 1 shows the ratios of approximation of the MERs extracted from 10,000 grayscale images with the size of 512×512 , and the size of the MERs is 384×384 . We compute distance D using (5). Taking the value 0.2575 in the upper left corner as an example, it means that 25.75% of the regions extracted by WOW (with embedding rate 0.2 bpp) and HILL (with embedding rate 0.2 bpp) are the same (D is 0). The first three rows are the results when using different steganographic algorithms with the same embedding rates, and the last three rows are the results when using the same embedding algorithms with different embedding rates. Table 2 shows the performance comparison of multiple methods based on the testing error performance. We directly use the output value of the accuracy layer as the detection rates of all experimental results. We test the four adaptive steganography algorithms described in Section 3.1 with embedding rates of 0.1 to 0.4 bpp. Three kinds of MERs with sizes of 256×256 , 384×384 , and 512×512 are used. The experimental results of the three different sizes of images are shown in columns 3–5. We compare our method with the single network proposed in [14], SRM [7], and maxSRMd2 [10], and the corresponding experimental results are shown in columns 6–8. The best detection result of our proposed method in each row is shown in bold font.

3.2.1. The Effectiveness of Selection of MER. First, we conduct experiments to show that when using different embedding algorithms and embedding rates to estimate the embedding probability for the same image, the MER method and Algorithm 1 we proposed are robustness. From Table 1, we can find most of the distances D are within the range of $[0, 30]$. When D is greater than 70, the ratio is very small which means that the embedding probability maps calculated by different adaptive steganographic algorithms and different rates are approximately the same, and the positions of MERs selected by Algorithm 1 using different steganographic algorithms and embedding rates are very close. Secondly, from Table 2, we can clearly find that the region selection strategy might be more applicable in the case of low embedding rates for the adaptive, algorithms including HUGO BD [2], WOW [3], and HILL [5], but plays a trivial role in detecting S-UNIWARD [4], because the changed pixels will not be limited only to the textured regions of the image as the payload increases. As the changed pixels start to spread all over the image, the MER based method will become less effective and may cause serious information loss in detecting S-UNIWARD due to lack of considering all of the differences between covers and stegos. In addition, the MERs with size of 384×384 are more competitive for detection than those of 256×256 . We further compare the detection performance on regions with the same size (384×384) selected by our method and randomly selected method and show the comparison

TABLE 2: Performance comparison of multiple methods based on the testing error.

Algorithm	bpp	Proposed method				Method [14]		SRM	maxSRM _{id2}
		256	384	512	512	512	512		
WOW	0.1	0.4144	0.4349	0.4577	0.4249	0.4056	0.3001		
	0.2	0.3235	0.3150	0.2917	0.3625	0.3204	0.2353		
	0.3	0.2375	0.2202	0.2160	0.2473	0.2501	0.1905		
	0.4	0.1900	0.1814	0.1847	0.2072	0.2092	0.1593		
HUGO	0.1	0.4083	0.3688	0.3711	0.3957	0.3601	0.3103		
	0.2	0.2979	0.2731	0.2816	0.3070	0.2834	0.2396		
	0.3	0.2533	0.2304	0.2194	0.2316	0.2259	0.1990		
	0.4	0.1899	0.1741	0.1517	0.1746	0.1852	0.1641		
HILL	0.1	0.4422	0.4403	0.4646	0.4500	0.4354	0.3776		
	0.2	0.3464	0.3344	0.3494	0.3642	0.3628	0.3151		
	0.3	0.2653	0.2581	0.2491	0.2683	0.3017	0.2655		
	0.4	0.2107	0.1937	0.1822	0.2083	0.2503	0.2192		
S-UNTIWARD	0.1	0.4680	0.4566	0.4511	0.4525	0.4075	0.3661		
	0.2	0.3624	0.3952	0.3319	0.3364	0.3247	0.2912		
	0.3	0.2781	0.2601	0.2511	0.2675	0.2584	0.2383		
	0.4	0.2348	0.2027	0.1845	0.2074	0.2052	0.1959		

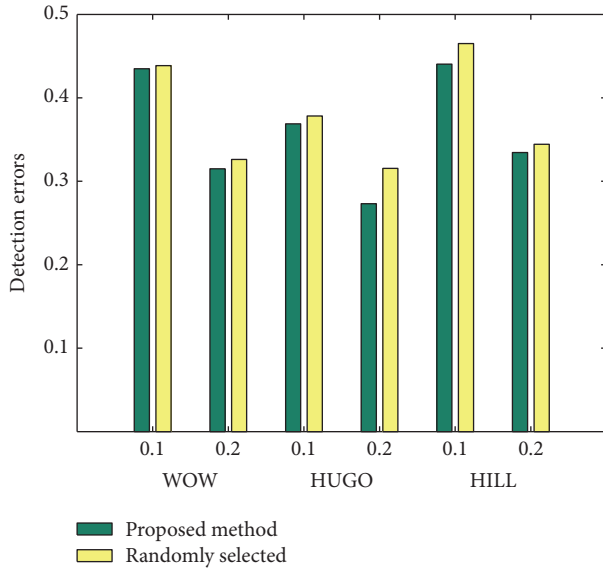


FIGURE 3: The detection errors of different ways to obtain the MER.

results in Figure 3. The histograms in yellow are the detection results on randomly selected regions and the green ones are produced by our method. We can conclude that, for the three kinds of adaptive steganography algorithms with low embedding rates of 0.1bpp and 0.2bpp, our method effectively outperforms by 1%~4% in terms of the detection accuracy compared with the random selection method. More specifically, our proposed method, the region selection, may not be as good as some state-of-the-art methods such as maxSRMd2 [10] because our method is not strictly following the idea of selection channel awareness. At present some of the commonly Selection-Channel-Aware method first estimates the embedding probability of an image and then selects points suitable for feature extraction. As for our method, we estimate the embedding probability to select an area suitable for embedding. However, we treat the points in the MER without distinction when extracting features. In Table 2, MERs with improper size may not be suitable for steganalysis of adaptive steganography algorithms, especially for HUGO and HILL. The total amount of pixels of size 384×384 is about half of the original picture and 256×256 is about a quarter of the original. In the case of appropriate size of the region, the MER selection method achieves better results.

3.2.2. The Effectiveness of the Proposed Network. We also evaluate the effectiveness of the network when it is used without considering the impact of selection of MERs. In Table 2, the 5th column shows the experimental results of our proposed network with original images, and the 6th–8th columns show the performance of all of the compared methods. We can find that in general our network achieves better performance than the method in [14] and SRM [7]. When the embedding rates are high, the detection accuracy is higher than that of the state-of-the-art method maxSRMd2 [10] except for detecting WOW [3]. MaxSRMd2 [10] must

first estimate the embedding probability when extracting features. However, a potential disadvantage of maxSRMd2 is that the point where the embedding probability is high does not necessarily correspond to the real embedded point, which may cause a certain amount of information loss. Our proposed method relies on all pixel points when using the combined networks to extract features from 512×512 images. Therefore, our method is more suitable to detect 512×512 images when the embedding rate is high.

4. Conclusion

In this paper, a new region selection method is proposed to find the effective region for CNN to detect adaptive steganographic methods. We also design a combined network consisting of three separate subnets with independent structures. By repeatedly separating and merging the independent subnets with different configuration parameters, the extracted features are more diverse and effective. Experimental results show that our approach has advantages and disadvantages in different situations. At relatively high embedding rates, the proposed combined CNN model outperforms the state-of-the-art steganalysis methods maxSRMd2 except WOW. The region selection strategy might be more applicable in the case of low embedding rates for several adaptive algorithms including HUGO BD, WOW, and HILL. In the future, we will consider to design new method which can choose the MER more accurately and propose new network structures that can evolve more diverse and effective features for steganalysis.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants nos. U1536204 and 61672386 and the Natural Science Research Project of Colleges and Universities in Anhui Province under Grant no. KJ2017A734.

References

- [1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.
- [2] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics Security*, vol. 5, no. 4, pp. 705–720, 2010.
- [3] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS 2012*, pp. 234–239, December 2012.
- [4] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, article 1, 2014.

- [5] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 4206–4210, 2014.
- [6] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics Security*, vol. 6, no. 3, pp. 920–935, 2011.
- [7] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics Security*, vol. 7, no. 3, pp. 868–882, 2012.
- [8] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis against WOW embedding algorithm," in *Proceedings of the 2nd ACM workshop*, pp. 91–96, Salzburg, Austria, June 2014.
- [9] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis based on embedding probabilities of pixels," *IEEE Transactions on Information Forensics Security*, vol. 11, no. 4, pp. 734–745, 2016.
- [10] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for Steganalysis of digital images," in *Proceedings of the 2014 IEEE International Workshop on Information Forensics and Security, WIFS 2014*, pp. 48–53, December 2014.
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [12] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Proceedings of the IS&T/SPIE Electronic Imaging*, p. 94090J, San Francisco, Calif, USA.
- [13] Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 2752–2756, September 2016.
- [14] G. Xu, H. Wu, and Y. Q. Shi, "Ensemble of CNNs for Steganalysis: An empirical study," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 103–107, ACM, June 2016.
- [15] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.
- [16] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale jpeg steganalysis using hybrid deep-learning framework," 2017.
- [17] J. Zeng, S. Tan, and B. Li, "Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis," in *Proceedings of the Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI '17)*, Burlingame, Calif, USA, January-February 2017.
- [18] J. Ni, J. Ye, and Y. I. Yang, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics Security*, 2017.
- [19] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi, "Steganalysis Based on Awareness of Selection-Channel and Deep Learning," in *Digital Forensics and Watermarking*, vol. 10431 of *Lecture Notes in Computer Science*, pp. 263–272, Springer International Publishing, 2017.
- [20] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.
- [21] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proceedings of the Electronic Imaging*, p. 650502, San Jose, Calif, USA, 2007.
- [22] V. Sedighi and J. Fridrich, "Effect of imprecise knowledge of the selection channel on steganalysis," in *Proceedings of the 3rd ACM Information Hiding and Multimedia Security Workshop, IH and MMSec 2015*, pp. 33–42, June 2015.
- [23] G. Hinton E, N. Srivastava, and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.
- [24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," <https://arxiv.org/abs/1502.03167>.
- [25] P. Bas, T. Filler, and T. Pevný, "'Break Our Steganographic System': The Ins and Outs of Organizing BOSS," in *Information Hiding*, vol. 6958 of *Lecture Notes in Computer Science*, pp. 59–70, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [26] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics Security*, vol. 7, no. 2, pp. 432–444, 2012.
- [27] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.
- [28] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 2010, Addison-Wesley Professional.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

