Steganalysis via Deep Residual Network

Songtao Wu[1,2], Sheng-hua Zhong[1,*], and Yan Liu[2]

[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Department of Computing, The Hong Kong Polytechnic University
cssongtaowu@gmail.com, csshzhong@szu.edu.cn, csyliu@comp.polyu.edu.hk

***Abstract*—Recent studies have demonstrated that a well designed deep convolutional neural network (CNN) model achieves competitive performances on detecting the presence of secret message in digital images, compared with the classical rich model based steganalysis. In this paper, we propose to investigate a category of very deep CNN model−the deep residual network (DRN), for steganalysis. DRN is suitable for steganalysis from two aspects. For the first, the DRN model usually contains a large number of network layers, which proves to be effective to capture the complex statistics of digital images. For the second, DRN's residual learning (ResL) method actively strengthens the signal coming from secret messages, which is extremely beneficial for the discrimination between cover images and stego images. Comprehensive experiments on standard dataset show that the DRN model achieves very low detection error rates for the state of arts steganographic algorithms. It also outperforms the classical rich model method and several recently proposed CNN based methods.**

***Index Terms*—Steganalysis, convolutional neural network, deep residual network, residual learning, steganography**

I. Introduction

Steganalysis is the art of revealing the presence of secret messages embedded in cover signals such as digital images [1]. Although this technique has developed a lot in the past decades [2-5], it is still challenging to detect modern steganographic algorithms accurately.

Most of methods formulate steganalysis as a binary classification problem. Among them, the rich model based steganalysis [5] achieves the best detection accuracy to most of steganographic algorithms. In the training stage, the method first extracts various handcrafted features, i.e. co-occurrence matrices, from the filtered digital images. Then, an ensemble classifier [6] is trained to discriminate cover images and their stego versions. In the testing stage, this trained classifier is used to determine whether a new input image contains secret message. The rich model method proves to be effective for the uniform embedding steganography, for example, the least significant bit (LSB) steganography or the LSB matching steganography. However, it is hard to attack the content adaptive steganography, especially for several state of the art algorithms such as the Highly Undetectable steGOnography (HUGO) [7], the Spatial UNIversal WAvelet Relative Distortion stegnography (S-UNIWARD) [8], the HIgh-pass Low-pass Low-pass steganography (HILL) [9] and the Minimizing the Power of Optimal Detector steganography (MiPOD) [10].

Several pioneering works have been proposed to use deep CNN to attack content adaptive steganography. Unlike the rich model method that utilizes handcrafted features, CNN based methods directly learn effective features from input images to classify covers and stegos. In [11], Tan and Li presented a stacked convolutional auto-encoder to detect the presence of secret message. In this network, three processing units extract features from input images and a three-layer fully connected neural network maps the extracted features into their labels. For each processing unit, it contains a convolutional layer, a maximum pooling layer and a sigmoid activation layer. The network shows better performance than the traditional subtractive pixel adjacency matrix steganalysis [4], but it is worse than the rich model method. Qian *et al.* in [12] proposed a different CNN architecture consisting of five convolutional layers, in which each layer is followed by an average pooling layer and a nonlinear activation layer. To better distinguish cover images and stego images, the paper proposed to use Gaussian rather than sigmoid as the activation function. Even though Qian's network is inferior to the rich model method, the performance gap between CNN and the rich model has been narrowed from 14% (Tan and Li's network) to $2\% - 5\%$. To further improve the accuracy of CNN for steganalysis, Xu *et al.* [13] designed a new CNN model incorporating the domain knowledge of steganography and steganalysis. By taking absolute values to outputs of the first convolutional layer and applying the *tanh* activation function to the first two convolutional layers, the network improves the modeling ability to input images and prevents overfitting. Because of these modifications, Xu's network achieves competitive performances with the rich model method on S-UNIWARD and HILL. After trying numerous experiments for CNN with different structures, Pibre *et al.* [14] found a CNN model that first surpasses the rich model method on S-UNIWARD at 0.4 bit-per-pixel (bpp). Pibre's network has two convolutional layers but no pooling layers. This feature makes the model able to preserve the information generated by message embedding when the data goes through the whole network. The reported detection error rate to 0.4 bpp S-UNIWARD is 7.4%, which is greatly smaller than rich model's 20%. In summary, these pioneering works indicate that the performance of CNN model for steganalysis depends heavily on their architectures.

Deep neural network models are able to approximate highly complex functions more efficiently than the shallow ones [15-17]. This ability indicates that very deep neural network can capture complex statistical properties of natural images, which may be beneficial for image classification. Recent works [18-

1521-9097/16 $31.00 © 2016 IEEE
DOI 10.1109/ICPADS.2016.165

IEEE computer society

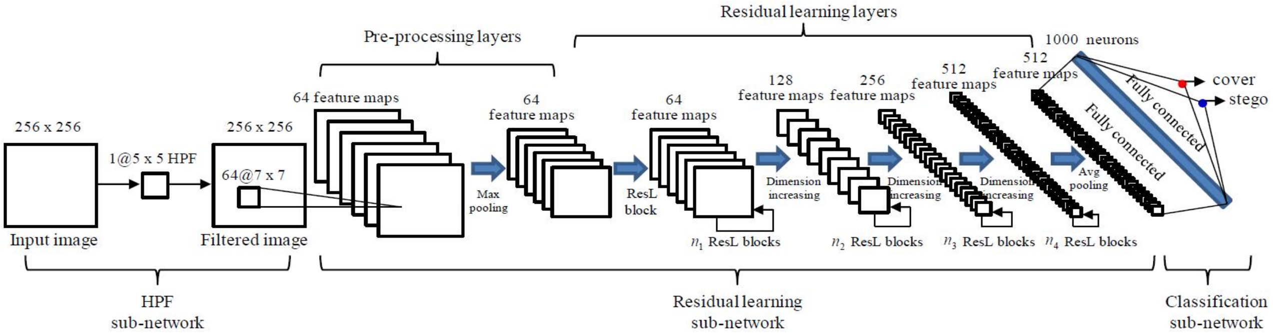

Fig. 1: DRN for steganalysis. In the HPF sub-network, a 5×5 KV kernel filters the input cover/stego image to get the noise residual image. In the residual learning sub-network, a building block for dimension increasing doubles the number of feature maps to the input signal, which is further processed by several building blocks for residual learning (ResL). n_1, n_2, n_3, or n_4 denotes that there are n_1, n_2, n_3, or n_4 ResL blocks following the current layer. The classification sub-network maps features into labels. In the figure, $64@7 \times 7$ denotes that there are 64 filters with the size of 7×7.

20] also verify that very deep CNN models achieve much better performances than previous CNN models for the large scale image recognition task. Even though great success has been achieved for very deep CNN models in image recognition, the research of it for steganalysis is still blank.

Designing effective CNN model for steganalysis requires the domain knowledge of steganography and steganalysis. Actually, steganalysis can be viewed as a special case of binary classification: classifying objects (covers) and objects added with weak signals (stegos). To better discriminate covers and stegos, a CNN model is necessary to preserve or strength weak stego signals generated by message embedding. Unfortunately, this domain knowledge has not yet been considered in recently proposed CNN models.

Recently, He *et al.* [18] has proposed a very deep CNN model $-$ the deep residual network for image classification. The network has successfully overcome the performance degradation problem when a neural network's depth is large. Because of its great success in image recognition, this paper aims to apply the DRN for steganalysis. Two appealing characteristics of DRN make it suitable for steganalysis. For the first, the depth of DRN is large, providing the network with powerful ability to capture useful statistical properties of input covers and stegos. For the second, instead of learning an underlying function directly, DRN explicitly fits a residual mapping, which enforces the network to emphasize the weak signal generated by message embedding. We present comprehensive experiments on the standard BOSSbase [21] dataset for five state of the art steganographic algorithms. Experimental results show that DRN is not only better than the classical rich model method, but also outperforms several recently proposed CNN models for steganalysis.

II. Deep Residual Network for Steganalysis

In this section, we introduce the DRN model for steganalysis. First, the overall structure of DRN is presented. Then, the parameter learning to DRN is described. At the end, we explain rationality of DRN's residual learning for steganlysis.

A. Network Structure

Fig.1 illustrates the architecture of DRN in this paper. The network contains three sub-networks, i.e. the high-pass filtering (HPF) sub-network, the deep residual learning sub-network and the classification sub-network. These sub-networks have their own roles in processing the information in the overall model, which are described as follows.

The HPF sub-network is to extract noise residuals from input cover/stego images. Previous studies indicate that extracting residual signals instead of pixels can largely suppress image content, leading to a narrow dynamic range and a large signal-to-noise ratio (SNR) between the weak stego signal and the image signal. As a result, statistical descriptions to the filtered image become more compact and robust [5]. Mathematically, the residual image $\mathbf{x}$ is the convolution between the input image $\mathbf{I}$ and the HPF kernel $\mathbf{k}$:

$$\mathbf{x} = \mathbf{I} * \mathbf{k} \tag{1}$$

where $*$ denotes convolution operator. We follow the qian's setting and choose the $\mathbf{k}$ as the KV kernel [12].

The residual learning sub-network is to extract effective features for classifying covers and stegos. This sub-network contains two categories of layers, the pre-processing layer and the residual learning layer. The pre-processing layer consists of a convolutional layer with 64 convolutional filters (the size is 7×7), a batch normalization layer, a ReLU activation layer and a maximum pooling layer. The pre-processing layer is to capture many different types of dependencies among elements in the residual image. Its purpose is to make the network extract enough statistical properties to detect the secret message more accurately. For the residual learning layer, it is constituted by two kinds of building blocks: a small block for DRN with small depth and a bottleneck block for DRN with large depth. Details about the structure of two building blocks are introduced in [18]. Each convolutional layer in a building block is followed by a batch normalization layer and a relu activation layer. For ordinary residual learning, both the input and the output have the same size of feature maps. For dimension increasing, the output has double size of feature maps than the input. To enforce each block having the same complexity, the feature map is down-sampled by factor 2 for the dimension increasing block. In our DRN model, there are four stages of processing, which improves the number of feature maps from 64 to 512.

The final classification sub-network consists of fully connected neural network model, mapping features extracted from the residual learning sub-network into binary labels. To ensure the modeling ability of this sub-network, we set the number of neurons to 1000.

B. Network Training

Parameters of the residual learning sub-network and the classification sub-network are learned by minimizing the softmax function:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} log\left(\frac{e^{o_{ik}}}{\sum_{k}^{K} e^{o_{ik}}}\right) \tag{2}$$

where N is the number of training samples, K is the number of labels ($K = 2$), o_{ik} denotes the output of the network. The weight matrix and bias vector for each layer is updated by the mini-batch stochastic gradient descending (SGD) [22].

C. Rationality of Residual Learning for Steganalysis

Residual learning is initially proposed to address the degradation problem for very deep neural networks. Instead of approximating an underlying function $\mathcal{H}(\mathbf{x})$ directly, it turns to fit its residual mapping $F(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. As indicated by He *et al.* in [18], it is easier to fit the residual mapping $F(\mathbf{x})$ than the original mapping $\mathcal{H}(\mathbf{x})$, especially when $\mathcal{H}(\mathbf{x})$ is an identity or a near identity mapping.

Actually, to detect the presence of secret message, steganalysis should correctly classify an input image $\mathbf{y}$ as:

$$\mathbf{y} = \begin{cases} \mathbf{x} + \mathbf{0}, & \text{cover} \\ \mathbf{x} + \mathbf{s}, & \text{stego} \end{cases} \tag{3}$$

where $\mathbf{0}$ is zero signal and $\mathbf{s}$ denotes the weak stego signal generated by message embedding. By feeding $\mathbf{y}$ into a residual learning block, the identity mapping of the network puts forward $\mathbf{x}$ to the output of the block, while the residual mapping $F(\mathbf{x})$ fits $\mathbf{0}$ or $\mathbf{s}$. Since both $\mathbf{0}$ and $\mathbf{s}$ are small signals, they can be effectively modeled by the residual learning network $F(\mathbf{x})$. Consequently, $\mathbf{s}$ is effectively captured by the residual mapping network. Therefore, the weak stego signal is expected to be preserved and emphasized through the whole network.

III. Experiments

A. Experimental Settings

The dataset used for performance evaluation is the BOSSbase 1.01 version [23]. The BOSSbase is a standard dataset for evaluating steganalysis and steganography. It contains 10,000 grayscale natural image with the size of 512×512. Following Qian and Pibre's setting, we crop the original 10,000 BOSSbase images into 40,000 non-overlapped images with the size 256×256. Without decreasing the difficulty of steganalysis, the cropped version leads to two advantages. For the first, the number of training samples of new dataset is larger than the original BOSSbase, which may prevent overfiting to a large extent. For the second, the computational complexity is greatly decreased due to the smaller size of input image.

For the DRN model, we initialize its weight matrices and bias vectors by a zero-mean Gaussian distribution with the fixed standard derivation of 0.01. The learning rate, momentum and weight decay of the model are set to 0.001, 0.9 and 0.0001 respectively. The size of mini-batch for SGD is set to 10. All experiments for the DRN are conducted on Nvidia's Tesla K80 platform.

B. Relationship between the Detection Accuracy and the Depth of DRN

This experiment is conducted to investigate how the depth of DRN affects the performance of steganalysis. 30,000 cover images randomly selected from the cropped BOSSbase, and their stegos which are generated by S-UNIWARD steganography [8] at 0.4 bpp, are used as training set. The rest 10,000 covers and stegos are used for testing. We select DRN models with 10, 20, 30, 40, 50 and 80 convolutional layers for evaluation. These DRN models are configured as TABLE I.

TABLE I: Configurations for DRN models. $[n_1, n_2, n_3, n_4]$ represents the number of blocks for ordinary residual learning, which is illustrated in Fig.1.

# conv.	Block Type	$[n_1, n_2, n_3, n_4]$
10	Small block	[0, 0, 0, 0]
20	Small block	[1, 2, 1, 1]
30	Small block	[2, 3, 3, 2]
40	Small block	[2, 5, 5, 3]
50	Bottleneck block	[2, 3, 5, 2]
80	Bottleneck block	[2, 3, 15, 2]

Fig.2 reports detection error rates of DRN models with different number of convolutional layers. When the number is smaller than 80, detection error rates decreases as the number increases. The result indicates that deeper DRN model can capture more reliable statistical properties of natural images than the shallow one for accurate steganalysis. However, when the number is 80, the overfiting phenomenon arises and results in the increase of the detection error rate. For this reason, we set the number of convolutional layer to 50 in the following experiment.

C. Performance Comparisons

To demonstrate the effectiveness of the DRN for steganalysis, we compare its performances with the rich model method on five states of the art steganographic algorithms, i.e. HUGO-BD (an improved version of the HUGO steganography) [23], the Wavelet Obtained Weights steganography (WOW) [24], S-UNIWARD [8], HILL [9] and MiPOD [10]. Same to the setting in section B, 30,000 randomly selected cover images and their corresponding stegos are used for training CNN

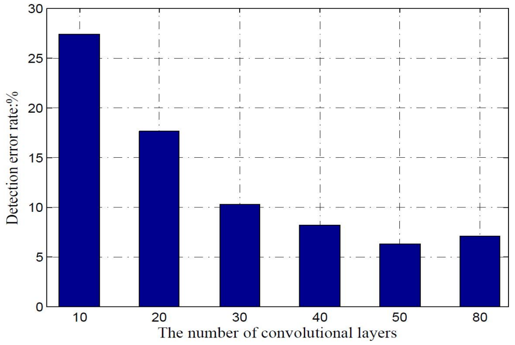

Fig. 2: Detection error rates for DRN with different number of convolutional layers.

models, the rest 10,000 cover images and their stegos are for testing. The number of training epoch is set to 50.

TABLE II: Detection error rates for five steganographic algorithms at payload 0.4 bpp.

Steganography	Rich Model	DRN (# conv. 50)
HUGO-BD	19.4%	**4.1%**
WOW	20.1%	**4.3%**
S-UNIWARD	20.3%	**6.3%**
HILL	24.2%	**10.4%**
MiPOD	22.1%	**4.9%**

TABLE II gives performance comparisons of DRN against to rich model. We can find that DRN is better than the rich model across all five steganographic algorithms. We compare DRN with three representative CNN models, including Qian's network [12], Xu's network [13] and Pibre's network [14]. Results in TABLE III demonstrate that the DRN also outperforms these CNN models for steganalysis.

TABLE III: Detection error rates for CNN models on five steganographic algorithms at 0.4 bpp. '\' denotes that the result is not reported in the paper.

Steganography	Qian [12]	Xu [13]	Pibre [14]	DRN
HUGO-BD	28.9%	\	\	**4.1%**
WOW	29.3%	\	\	**4.3%**
S-UNIWARD	30.9%	19.7%	7.4%	**6.3%**
HILL	\	20.7%	\	**10.4%**
MiPOD	\	\	\	**4.9%**

IV. Conclusion

This paper has investigated a category of very deep convolutional neural network model—the deep residual network—for steganalysis. Because of its large depth and new residual learning method, the deep residual network is naturally suitable for discriminating cover images and stego images. Extensive experiments on several challenging steganographic algorithms validate that the deep residual network achieves significantly better performances than the classical rich model method and other CNN based methods. Our future work will focuse on incorporating more domain knowledge of steganalysis in the deep residual network, aiming to detect content adaptive steganography with higher accuracy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61502311, No. 61373122), Natural Science Foundation of Guangdong Province (No. 2016A030310053), Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), the Science and Technology Innovation Commission of Shenzhen under Grant (No. JCYJ20150324141711640), and Shenzhen University research funding (201535).

References

[1] H. Wang and S. Wang. Cyber warfare: steganography vs. steganalysis. *Communications of the ACM*, 47(10): 76-82, 2004

[2] N. Provos and P. Honeyman. Detecting steganographic content on the internet, in *NDSS*, 2002

[3] A. D. Ker. Steganalysis of LSB matching in grayscale images, *IEEE SPL*, 12(6):441-444, 2005

[4] T. Pevny, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix, *IEEE TIFS*, 5(2):215-224, 2010

[5] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images, *IEEE TIFS*, 7(3):868-882, 2012

[6] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media, *IEEE TIFS*, 7(2):432-444, 2012

[7] T. Pevny, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography, in *IH*, pp. 161-177, 2010

[8] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain, *EURASIP Journal on Information Security*, 1(1):1-13, 2014

[9] B. Li, M. Wang, J. Huang, and X. Li. A new cost function for spatial image steganography, in *ICIP*, pp.4206-4210, 2014

[10] V. Sedighi and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability, *IEEE TIFS*, 11(2):221-234, 2016

[11] S. Tan and B. Li. Stacked convolutional auto-encoders for steganalysis of digital images, in *APSIPA*, 2014

[12] Y. Qian, J. Dong, W. Wang and T. Tan. Deep learning for steganalysis via convolutional neural networks, in *SPIE Media Watermarking, Security, and Forensics*, 2015

[13] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural network for steganalysis, *IEEE SPL*, 23(5):708-712, 2016

[14] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont. Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch, in *SPIE Media Watermarking, Security, and Forensics*, 2016

[15] M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: a comparison between shallow and deep architectures, *IEEE TNNLS*, 25(8):1553-1565 2014

[16] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks, in *NIPS*, pp.2924-2932, 2014

[17] S. Sun, W. Chan, L. Wang, X. Liu, and T. Y. Liu. On the depth of deep neural networks: a theoretical view, in *AAAI*, 2016

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, in *CVPR*, 2016

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large scale image recognition, in *ICLR*, 2015

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, in *CVPR*, 2015

[21] P. Bas, T. Filler and T. Pevny. BOSS (break our steganography system), $http : //boss : gipsa - lab : grenoble - inp : fr$, 2009

[22] M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization, in *ACM KDD*, 2014

[23] T. Filler and J.Fridrich. Gibbs construction in steganography, *IEEE TIFS*, 5(4):705-720, 2010

[24] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters, in *WIFS*, 2012