Multimedia Tools and Applications (2019) 78:22711–22725
https://doi.org/10.1007/s11042-019-7601-9

RestegNet: a residual steganalytic network

Weike You[1,2] · Xianfeng Zhao[1,2] · Sai Ma[1,2] · Yaqi Liu[1,2]

Received: 4 April 2018 / Revised: 22 November 2018 / Accepted: 3 April 2019 /
Published online: 23 April 2019

Abstract

Deep convolutional networks bring new energy to image steganography. It is an opportunity for steganalysis research. However, the operations to widen the gap between covers and stegos are only in the preprocessing layers for most existing networks. In this paper, a residual steganalytic network (RestegNet) is proposed to overcome this limitation. We design a novel building block group, which consists of two alternating building blocks: 1) A sharpening block based on residual connections (ShRC), which makes the noise of steganography overwhelm the image content, and aims to enhance steganographic signal detectability. 2) A smoothing block based on residual connections (SmRC), which seeks to downsample the feature maps to boil them down to useful data. First, we use the same preprocessing layers as previous methods to ensure minimum performance. Then, we use these building block groups to exaggerate the traces of steganography further and make the difference between covers and stegos in the feature extraction layers. Contrastive experiments with previous methods conducted on the BOSSbase 1.01 demonstrate the effectiveness and the superior performance of the proposed network.

Keywords Convolutional neural network · Deep learning · Image steganography · Steganalysis · Residual connections

✉ Xianfeng Zhao
zhaoxianfeng@iie.ac.cn

Weike You
youweike@iie.ac.cn

Sai Ma
masai@iie.ac.cn

Yaqi Liu
liuyaqi@iie.ac.cn

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China

1 Introduction

Image steganography aims to conceal messages in images to covert communication. The goal of steganalysis is to detect it. Steganalysis is a challenging task because it needs to recognize the change in pixels while it is entirely invisible to the naked eye.

In the past, the main steganalyzers are based on well-designed handcrafted features, trying to cope with the content-adaptive steganographic schemes better. For instance, the Spatial Rich Model (SRM) [4] and its variants tSRM [12], maxSRM [3]. Recently, the steganalysis results have been improved over a short period. In large part, these advances have been driven by Convolutional Neural Network (CNN). CNN has obtained spectacular results in image recognition and segmentation tasks. It also brings new air to image forensics, which is an opportunity for steganalysis. For instance, Qian [9], XuNet [14, 15], TLU-CNN [16], SCA-TLU-CNN [16], Yedroudj-Net [17], Zhu-Net [19], ReST-Net [8], JPEG-phase-aware XuNet [2], deep network targets to J-UNIWARD [13]. All these methods have comparable or even better performances than traditional handcrafted feature-based methods. The typical steganalysis networks combine two elements from the traditional computer vision tasks. The first one is image preprocessing, where the goal is to extract noise residuals. The second one is the binary classification, where the goal is to classify the data into two groups, namely covers and stegos. However, most of these schemes only focus on increasing the difference between covers and stegos in the preprocessing layer, *e.g.*, high-pass filtering layers for transforming original images to noise residuals, activation layers to capture the hidden signals better. The other layers are merely deepened, widened, or use various convolution kernels.

To break the limitations as mentioned above, our goal in this work is to widen the gap between covers and stegos in the feature extraction layers and make the detection of embedding traces more accurate. We propose a residual deep learning network for image steganalysis, which is called *RestegNet*. The feature extraction layers of the proposed network are composed of two building blocks: A sharpening block based on residual connections (*ShRC*), and a smoothing block based on residual connections (*SmRC*). ShRC first performs parallel filtering and batch normalization operations on the same input, then it concatenates them to one single output. Finally, it sums the feature maps with the initial input in the end, which supplements the information by adding residuals. SmRC first performs parallel filtering and batch normalization operations on the same input, and downsamples them. Then, it concatenates them to one single output. In the end, it sums the feature maps with the downsampled initial input, which distills the information by downsampling. ShRC and SmRC are alternately used to supplement and distill the information after preprocessing layers to overcome the existing limitations. The preprocessing layers of previous networks are preserved, which allows us to enhance the performance without breaking their initial behavior. Fig. 1 illustrates the above process in a simplified sequence flow diagram.

In summary, the main contributions of our work are two-fold:

(1) A novel steganalysis network is proposed, and two building blocks are designed to emphasize traces of steganography to be further exaggerated:

  (a) ShRC is intended to supplement information, aims to highlight small, faint evidence of embedding to be greatly exaggerated.

  (b) SmRC is designed to distill information, aims to compress redundant of the useless image content of covers and stegos.

(2) The proposed network brings substantial improvements and achieves superior performance on both detection accuracy and convergence speed. It surpasses the results of

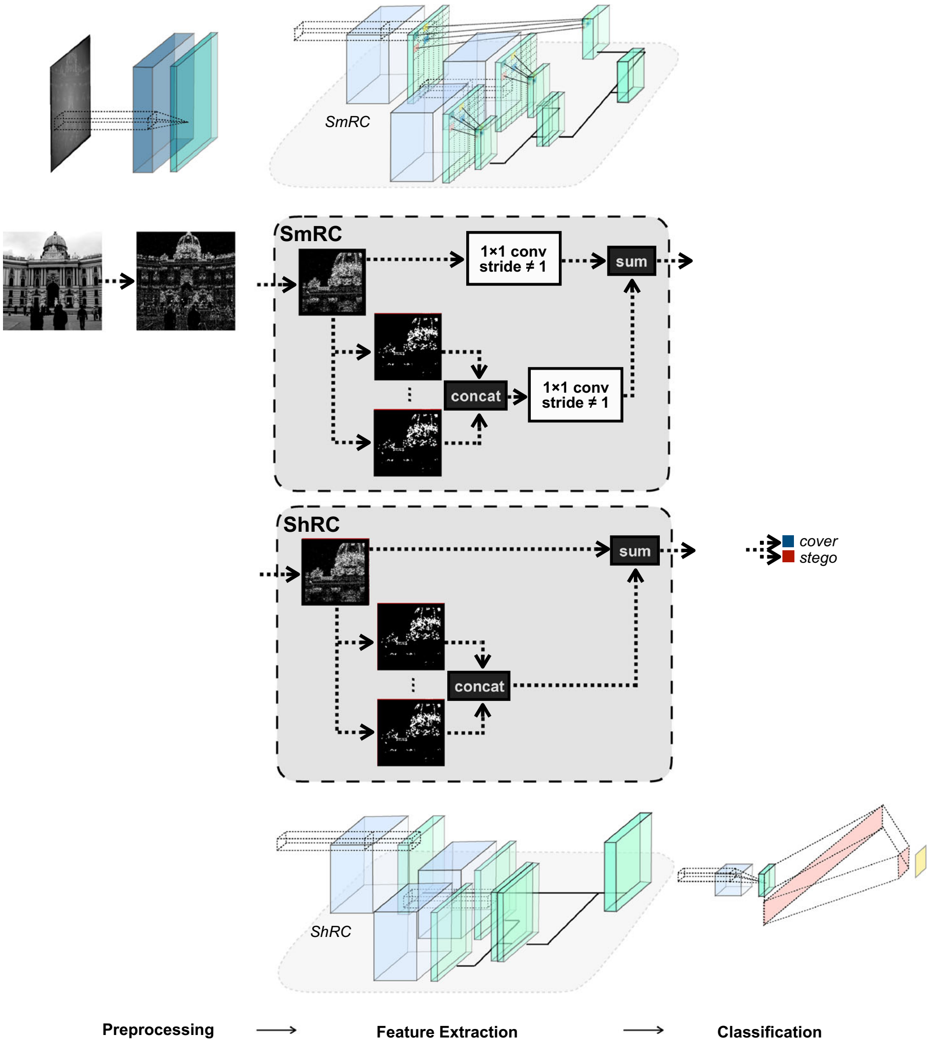


Fig. 1 The flow diagram for **RestegNet**

XuNet (the detection accuracy is increased by **3.0787%** to **7.4514%**) and TLU-CNN (The convergence speed is significantly improved by more than **8 times**).

The rest of this paper is organized as follows. In Section 2, we review the related works. Section 3 presents the proposed network architecture. Related experiments and discussions are presented in Section 4. The conclusion is drawn in Section 5.

2 Related work

In this section, we briefly review the previous achievements and existing problems. Tan et al. first utilized a CNN for steganalysis [11]. Their tests appear that it should be comparable or even better than handcrafted feature-based methods. Qian et al. [9] proposed a steganalytic demonstrate utilizing CNN can automatically learn feature representations with convolutional layers, and it achieves comparable performance with SRM [4]. Xu et al. [14, 15] introduced a CNN structure with batch normalization (BN), global average pooling and the absolute activation function, exceed the performance of SRM for the first time. Ye et al. [16] proposed a CNN structure with 30 high-pass filters for preprocessing, and a new activation function called truncated linear unit (TLU) is used. By incorporating the selection channel knowledge, their network obtains significant performance improvements than previous networks. Li et al. [8] proposed a different network architecture with parallel sub-nets using diverse activation functions for preprocessing to learn in more pipelines and have a better performance.

It is shown that the usual ways to improve performance include [19]: using high pass filters and proper activation functions, incorporating the selection channel knowledge, and using deeper or broader networks with various types of convolutional kernels as ResNet [5], DenseNet [7], and GoogleNet [10]. However, all the operations tailored for steganalysis are in the preprocessing layers, and the other layers are not designed for widening the gap between covers and stegos. They are intended only to extract information on different scales or depths. Residual connections can solve this problem. The classic building block of residual connections is described in ResNet [5]. The output layer is adding in, element-wise, the input layer to the residual layer. Concurrent with our work, the paper of [13] presents residual connections to achieve the strength of modeling, since it allows the gradient to pass backward (and the data to move forward) directly. We find it has other uses. Since the residual connections can be regarded as a filter to learn something new, and when the stride of the convolution layer is not equal to 1, it is equivalent to a downsampling operation. By alternately adding residuals (sharpening) and downsampling (smoothing), we can make the difference between covers and stegos through the feature extraction layers. It inspires us to design two building blocks to solve the problem, which will be presented in the next section.

3 RestegNet

In this section, we elaborate on the proposed steganalysis network. The overall process flowchart is shown in Fig. 1, and the step-by-step instructions are provided in the following parts.

RestegNet accepts an input image and outputs a two class labels (cover and stego). First, as stated earlier, it retains the preprocessing part of the existing network to ensure minimum performance. Then, it is alternately constructed of two building blocks for feature extraction: ShRC and SmRC. Finally, a pooling layer and a fully connected layer followed by a softmax are at the end of the RestegNet to normalized probabilities for classification.

The two building blocks mentioned above are described as follows:

ShRC As illustrated in Fig. 2, the complete process is given as follows:

Input: The original signal x, the stacked convolutional layers F_n, the layers of batch normalization BN_n, a concatenation layer $//$, a summation layer $+$.

Output: The sharpened signal of previous layers.

Step 1: Multiple convolution filters first simultaneously filter the original signal x (with the stride size is one), that extract the additional components $F_n(x)$. The batch normalized version of the outputs are then concatenated. This allows the model to get multi-level feature extraction.

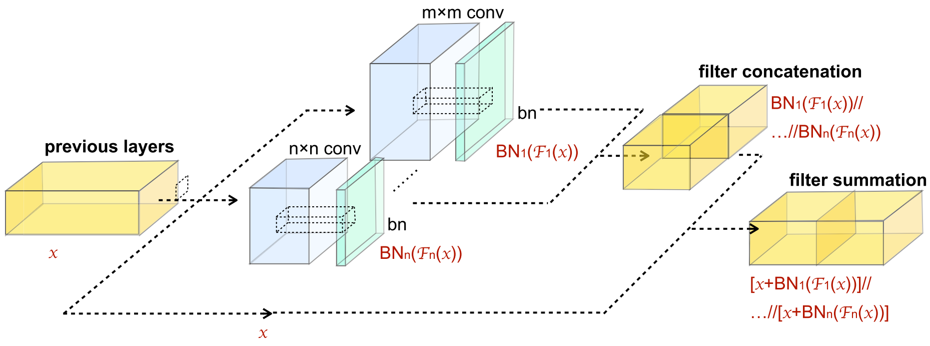


Fig. 2 a building block of **ShRC**

Step 2: The output of $Step1$ is added to the original signal x, thus producing a sharpened signal of the original:

$$[x + BN_1(F_1(x))]//[x + BN_2(F_2(x))]//...//[x + BN_n(F_n(x))]. \tag{1}$$

We visualize the input and the output to give an intuitive understanding of it in Fig. 3. As can be seen, ShRC creates a much richer signal.

SmRC As illustrated in Fig. 4, the complete process is given as follows:

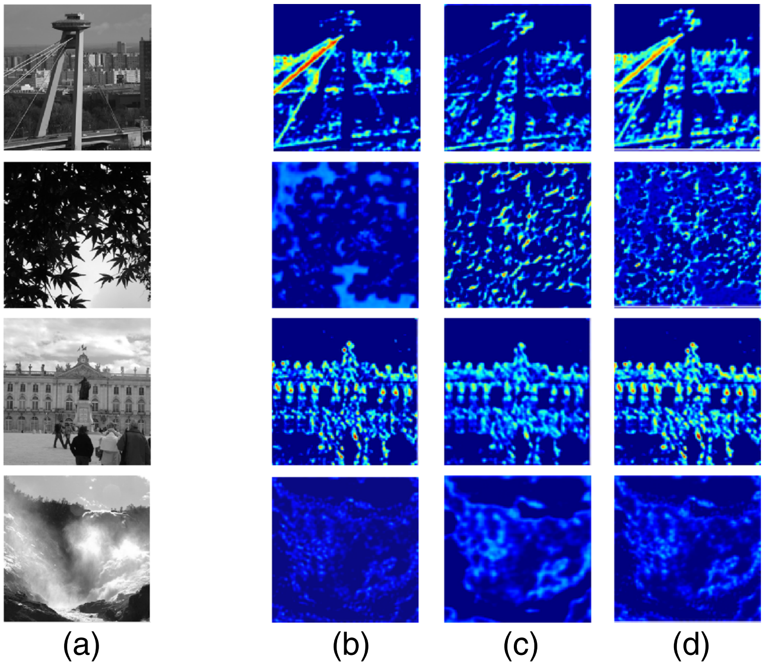


Fig. 3 An intuitive understanding of **ShRC**. **a** The original image. **b** The signal before a ShRC. **c** The signal extracted by the branch of a ShRC. **d** The signal after ShRC

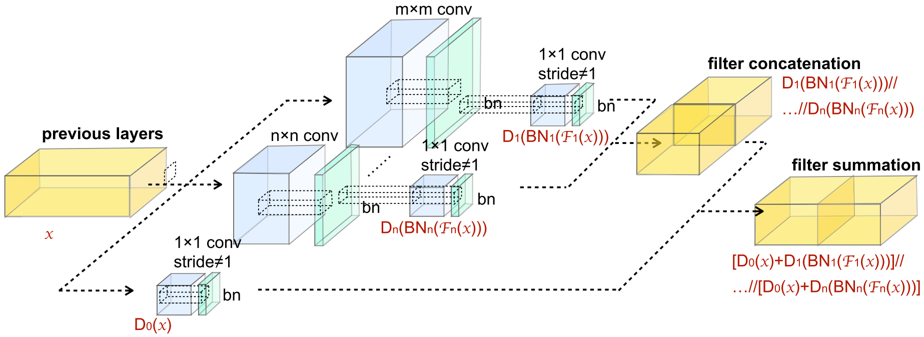


Fig. 4 a building block of **SmRC**

Input: The original signal x, the stacked convolutional layers F_n, the layers of batch normalization BN_n, the stacked layers for downsampling D_n (1×1 convolutional layers with a stride of n ($n \neq 1$)), a concatenation layer $//$, a summation layer $+$.

Output: The smoothed signal of previous layers.

Step 1: The original signal x is first filtered by multiple convolution filters ($m \times m$ ($m > 1$), and the stride size is one). Next, the outputs are batch normalized and filtered by the stacked layers for downsampling (1×1 convolutional layers with the stride

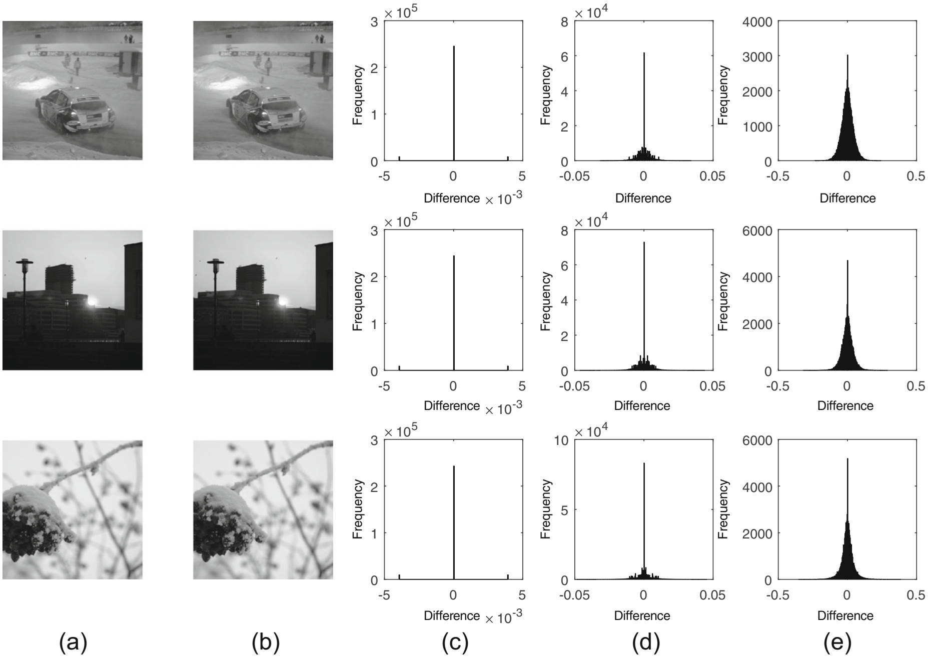


Fig. 5 The effect of alternately using **ShRC** and **SmRC**. **a** The original cover image. **b** The original stego image. **c** The original pixel value difference histogram. **d** The pixel value difference histogram after a ShRC. **e** The pixel value difference histogram after alternately using a ShRC and a SmRC

of n ($n \neq 1$)). The results $D_n(BN_n(F_n(x)))$ are then concatenated. This allows the model to get downsampled multi-level feature extraction.

Step 2: A 1×1 convolutional layer filters the original signal with the stride of n ($n \neq 1$) that downsamples the input signal directly $D_0(x)$.

Step 3: Sum them up to produce the smoothed signal of the original:

$$[D_0(x) + D_1(BN_1(F_1(x)))]//[D_0(x) + D_2(BN_2(F_2(x)))]// \ldots//[D_0(x)+D_n(BN_n(F_n(x)))].(2)$$

The reason we use this building block group is that we notice it effectively widens the gap between covers and stegos, which is shown in Fig. 5.

It looks like the inception module used in ReST-Net [8] as proposed in GoogLeNet [10]. The difference is that ReST-Net concatenates the outputs in the depth dimensions to construct subnets, while RestegNet sums up the results of parallel operations to get the enhanced signals.

Instantiated RestegNet is instantiated with preprocessing layers of XuNet [14, 15] and TLU-CNN [16]. See Figs. 6 and 7. The contrastive experiments will be presented in the next section.

It is designed with computational efficiency and practicality in mind so that inference can be run on individual devices including even those with limited computational resources.

4 Experiments

The environments We use a well-known content-adaptive steganographic methods S-UNIWARD [6] by Matlab implementations with random embedding key. Our proposed network is compared with two popular networks: XuNet [14, 15], TLU-CNN (YeNet) [16]. All the experiments were ran on two Nvidia GTX 1080 GPU cards.

Datasets In this paper, we use standard datasets to test the performance of the proposed networks. The dataset is 10,000 grey-level images of the BOSSbase 1.01 [1] at different payloads(0.1, 0.2, 0.3, 0.4 bit per pixel(bpp)). Due to our GPU computing power and time limitation, images are cropped such that their scale (all edges) is 256 pixels with a ratio of $1 : 1$ of positive to negatives and $2 : 1 : 1$ of training to validation to test.

Hyper-parameters We apply an adaptive learning rate method (AdaDelta) to train all the networks. The momentum and the weight decay of networks are set to 0.95 and 0.0005 respectively. Each mini-batch has 16 images. The base learning rate is 0.4. The learning rate policy is poly, which the effective learning rate follows a polynomial decay, and the power is 0.5. The maximun number of iterations is 60k. The code[1] is available.

Complexity analyses The trainable parameters number of RestegNet with preprocessing layers of XuNet is 9,370,747, and it has a speed of 5.13915 iter/s, 19.4585s/100 iters. Although it is about 4 times slower than XuNet (21.1308 iter/s, 4.73242s/100 iters), 60k iterations of training only takes about 3.5 hours. The trainable parameters number of RestegNet with preprocessing layers of TLU-CNN is 9,393,876, and the speed is around 2.62887 iter/s,

[1] http://github.com/spadeke/RestegNet

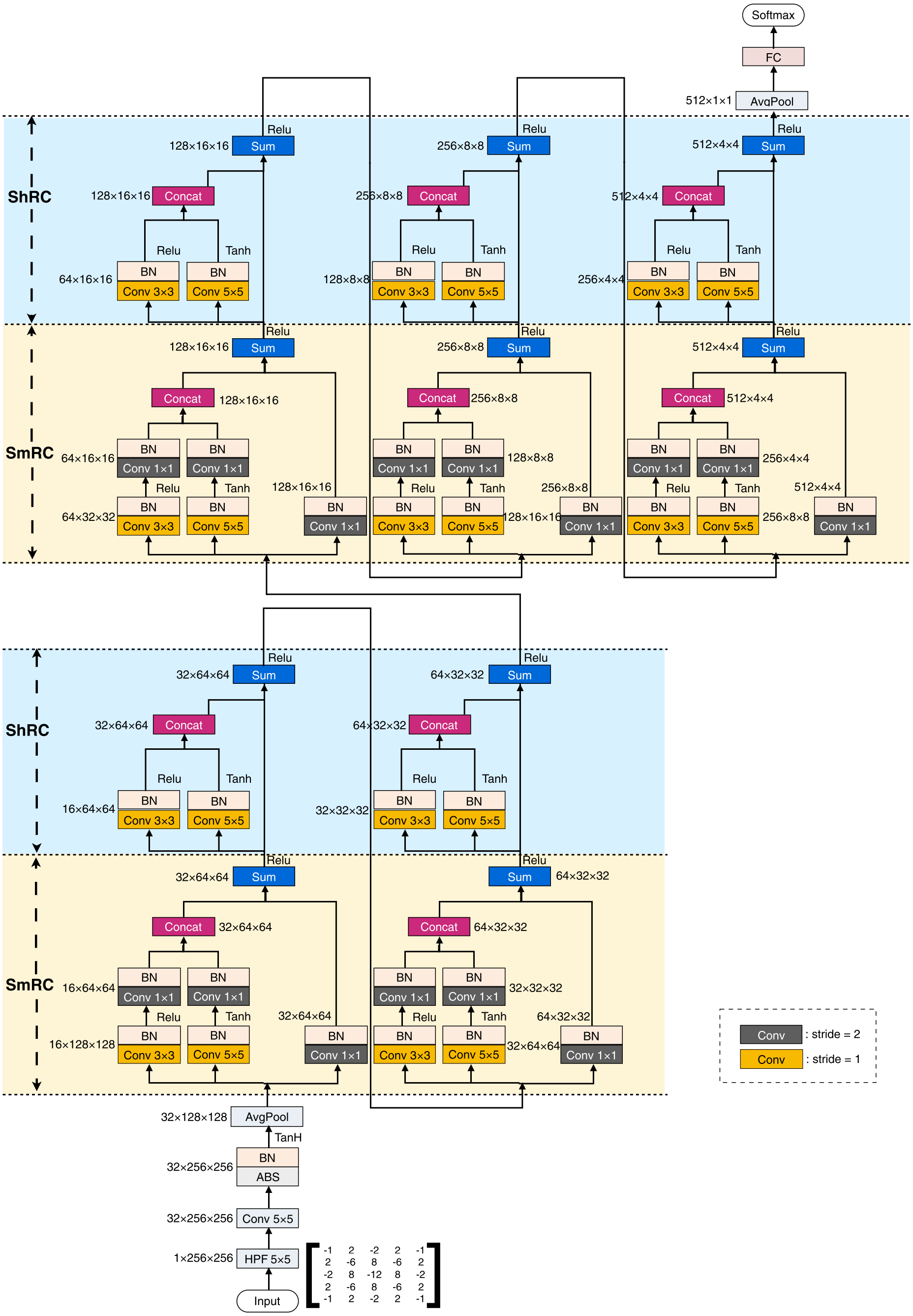
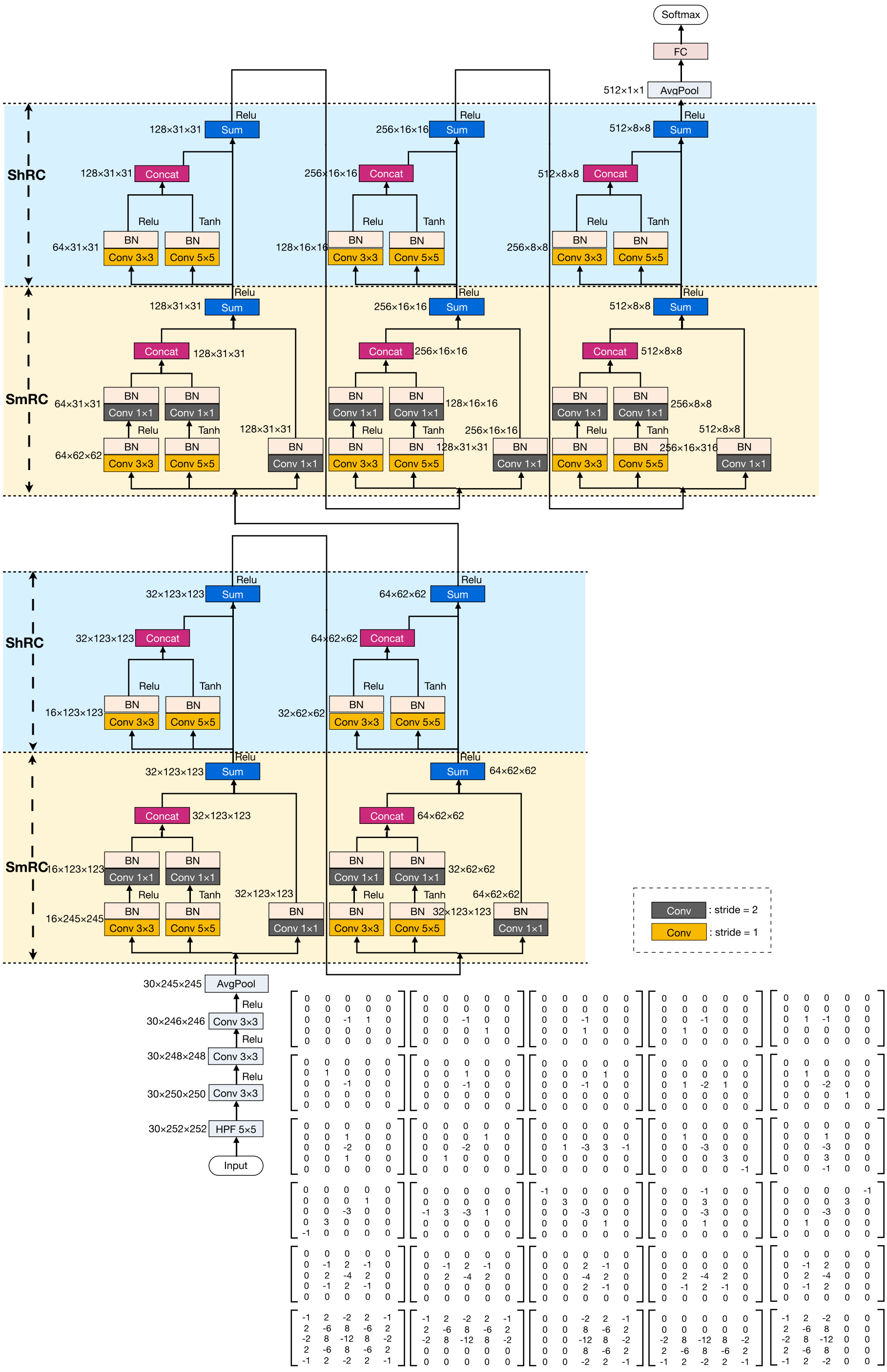


Fig. 6 The instantiated **RestegNet with preprocessing layers of XuNet** (RestegNet_XuVer). Data sizes following (number of channels) × (height) × (width) are displayed on the both sides.



$$\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&-1&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&-1&0&0\\0&0&0&1&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&-1&0&0\\0&0&1&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&-1&0&0\\0&1&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-1&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

$$\begin{bmatrix}0&0&0&0&0\\0&1&0&0&0\\0&0&-1&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&-1&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&0&-1&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-2&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&1&0&0&0\\0&0&-2&0&0\\0&0&0&1&0\\0&0&0&0&0\end{bmatrix}$$

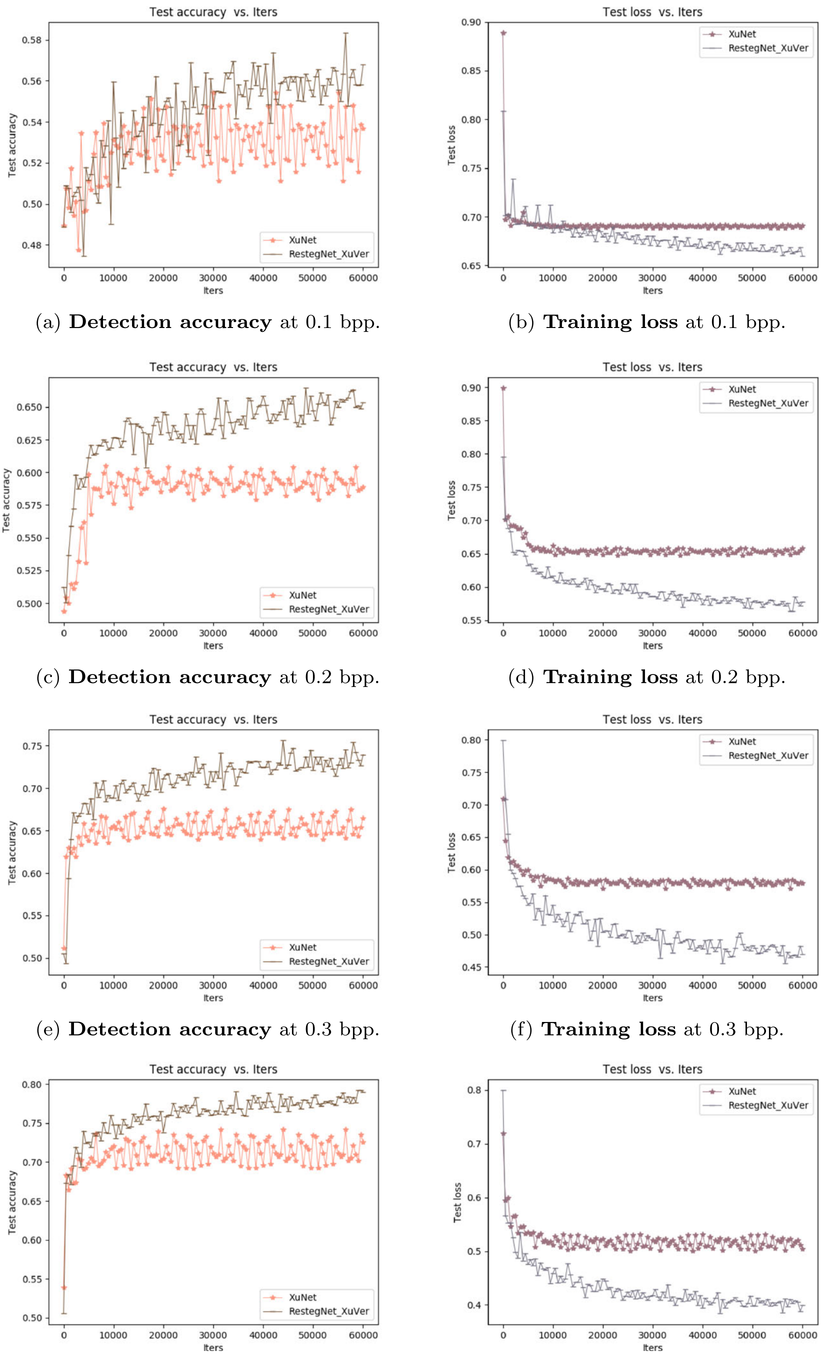
$$\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&-2&0&0\\0&0&1&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&0&-2&0&0\\0&1&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-3&3&-1\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&1&0&0&0\\0&0&-3&0&0\\0&0&0&3&0\\0&0&0&0&-1\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&1&0&0\\0&0&-3&0&0\\0&0&3&0&0\\0&0&-1&0&0\end{bmatrix}$$

$$\begin{bmatrix}0&0&0&0&0\\0&0&0&1&0\\0&0&-3&0&0\\0&3&0&0&0\\-1&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\-1&3&-3&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}-1&0&0&0&0\\0&3&0&0&0\\0&0&-3&0&0\\0&0&0&1&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&-1&0&0\\0&0&3&0&0\\0&0&-3&0&0\\0&0&1&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&-1\\0&0&0&3&0\\0&0&-3&0&0\\0&1&0&0&0\\0&0&0&0&0\end{bmatrix}$$

$$\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&2&-1&0\\0&0&-4&2&0\\0&0&2&-1&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&-1&2&0&0\\0&2&-4&0&0\\0&-1&2&0&0\\0&0&0&0&0\end{bmatrix}$$

$$\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&-2&2&-1\\0&0&8&-6&2\\0&0&-12&8&-2\\0&0&8&-6&2\\0&0&-2&2&-1\end{bmatrix}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\begin{bmatrix}-1&2&-2&0&0\\2&-6&8&0&0\\-2&8&-12&0&0\\2&-6&8&0&0\\-1&2&-2&0&0\end{bmatrix}$$

Fig. 7 The instantiated **RestegNet with preprocessing layers of TLU-CNN** (RestegNet_TLUVer). Data sizes following (number of channels) × (height) × (width) are displayed on both sides



(a) **Detection accuracy** at 0.1 bpp.

(b) **Training loss** at 0.1 bpp.

(c) **Detection accuracy** at 0.2 bpp.

(d) **Training loss** at 0.2 bpp.

(e) **Detection accuracy** at 0.3 bpp.

(f) **Training loss** at 0.3 bpp.

(g) **Detection accuracy** at 0.4 bpp.

(h) **Training loss** at 0.4 bpp.

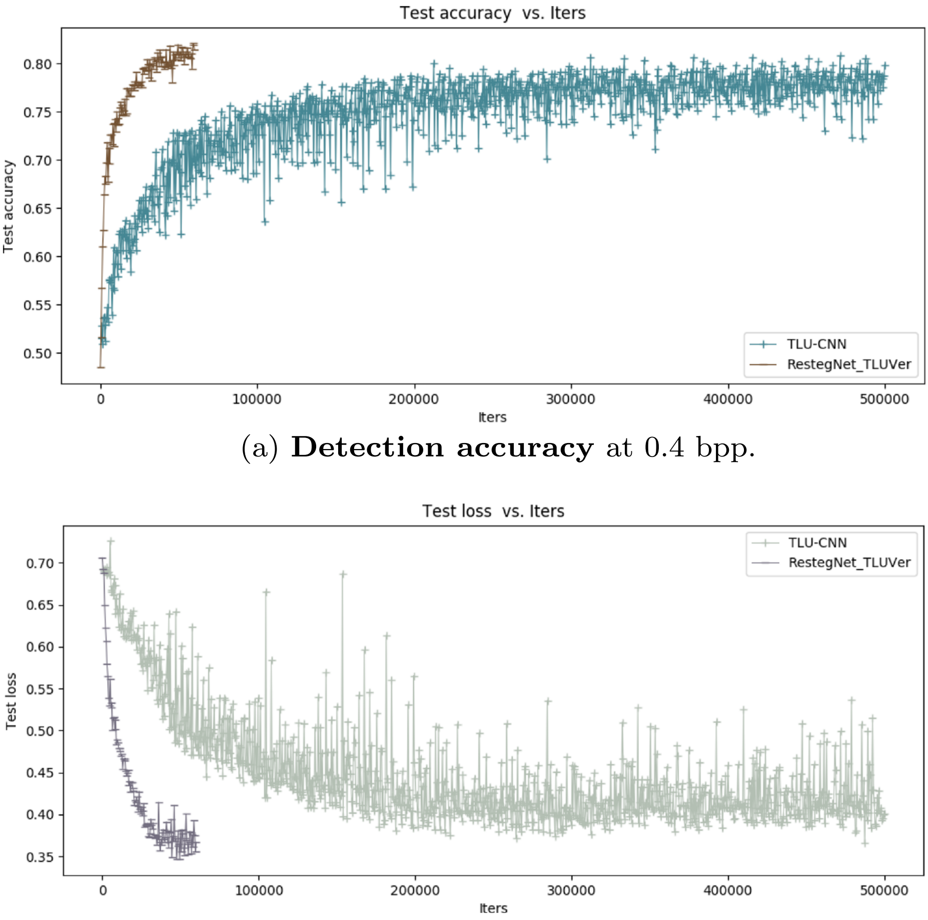
Fig. 8 **Performance comparison** of the detection accuracy (*Left*) and training loss (*Right*) of XuNet and RestegNet_XuVer for S-UNIWARD at different payloads(0.1, 0.2, 0.3, 0.4 bit per pixel(bpp)) on cropped images

Table 1 Detection Accuracy (%) of XuNet and RestegNet_XuVer

Network	Embedding Payload (bpp)			
Architecture	0.1	0.2	0.3	0.4
XuNet	52.6250	58.4375	66.8750	73.3750
RestegNet_XuVer	55.7037	65.8889	73.8518	79.3333

38.0391s/100 iters. It is about 2 times slower than TLUNet (5.64659 iter/s, 17.7098s/100 iters), 60k iterations of training takes about 6.5 hours.

Results The comparison result of test accuracy and loss for XuNet [14, 15] and the instantiated RestegNet with preprocessing layers of XuNet (shown in Fig. 6, named



(a) **Detection accuracy** at 0.4 bpp.

(b) **Training loss** at 0.4 bpp.

Fig. 9 **Performance comparison** of the detection accuracy (*Up*) and training loss (*Down*) of TLU-CNN and RestegNet_TLUVer for S-UNIWARD at 0.4 bit per pixel(bpp) payload

as RestegNet_XuVer) for 60k iterations is shown in Fig. 8 and Table. 1. The involved staganographic methods is S-UNIWARD. The crop size is 256 × 256.

As Fig. 8, RestegNet has significantly better performance than XuNet at any payload, especially at the payload of 0.2, 0.3 and 0.4 bpp. At the beginning of training, XuNet quickly reaches the highest accuracy, and the convergence stops at the same time, while the accuracy of RestegNet continues to increase, and the loss continues to decrease. After 60k iterations, the proposed network has reduced error rate by 3.0787% to 7.4514%.

We also compare the detection accuracy and training loss of TLU-CNN [16] and the instantiated RestegNet with preprocessing layers of it (shown in Fig. 7, named as RestegNet_TLUVer). Since training TLU-CNN is time-consuming, we only experiment S-UNIWARD at 0.4 bpp payload with our limited hardware resources. The results can be seen in Fig. 9, which can be shown that even at more than eight times the iterations (since the training of RestegNet_TLUVer is about 2 times slower than TLUNet as mentioned before, 4 times the training time), the performance of TLU-CNN (the detection accuracy is 79.85% after 500k iterations) still worse than RestegNet_TLUVer (the detection accuracy is **81.5625%** after 60k iterations).

Results show that it effectively improves detection accuracy, dramatically accelerates the convergence speed, and keeps the computational budget affordable. For S-UNIWARD with different payloads on cropped images, the proposed network is distinctly better than XuNet. Besides, it tremendously accelerates the convergence speed of TLU-CNN. Briefly, the experiment results indicate that even with the same preprocessing layers, RestegNet dramatically improves the performance of previous networks due to the vast differences in the architecture of the feature extraction layers.

5 Conclusion

RestegNet yields substantial evidence that constructing a network with ShRC and SmRC is a viable method for improving convolutional neural networks for steganalysis. The main advantage of this method is a significant quality gain with a modest increase in computational requirements compared to previous networks. RestegNet is one incarnation of these guidelines used in our assessment. We hope our effective approach will help ease future research in image steganalysis.

It is expected that RestegNet can achieve a better quality of result with better preprocessing layers (*i.e.,* ReST-Net [8], etc.). It is still an open question how to comprehensively construct building blocks to better distinguish covers and stegos. Although the strategy of our work has been proved to be effective, a more systematic method is desirable as future work.

Acknowledgments This work was supported by NSFC under 61802393, U1736214, U1636102 and 61872356, National Key Technology R&D Program under 2016YFB0801003 and 2016QY15Z2500, and Project of Beijing Municipal Science & Technology Commission under Z181100002718001.

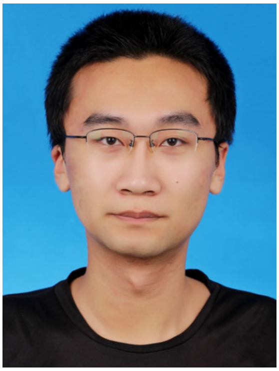
References

1. Bas P, Filler T, Pevný T (2011) “break our steganographic system”: the ins and outs of organizing BOSS. In: Information hiding - 13th international conference, IH 2011, prague, Czech Republic, May 18-20, 2011, revised selected papers. Springer, pp 59–70

2. Chen M, Sedighi V, Boroumand M, Fridrich JJ (2017) Jpeg-phase-aware convolutional neural network for steganalysis of JPEG images. In: Proceedings of the 5th ACM workshop on information hiding and multimedia security, IH&MMSec 2017, Philadelphia, PA, USA, June 20-22, 2017. ACM, pp 75–84
3. Denemark T, Sedighi V, Holub V, Cogranne R, Fridrich JJ (2014) Selection-channel-aware rich model for steganalysis of digital images. In: 2014 IEEE International workshop on information forensics and security, WIFS 2014, Atlanta, GA, USA, December 3-5, 2014, pp 48–53
4. Fridrich JJ, Kodovský J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensic Secur 7(3):868–882
5. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp 770–778
6. Holub V, Fridrich JJ, Denemark T (2014) Universal distortion function for steganography in an arbitrary domain. EURASIP J Inf Secur 2014(1):1
7. Huang G, Liu Z, Weinberger KQ (2016) Densely connected convolutional networks. CoRR arXiv:1608.06993
8. Li B, Wei W, Ferreira A, Tan S (2018) Rest-net: Diverse activation modules and parallel sub-nets based cnn for spatial image steganalysis. IEEE Signal Process Lett PP(99):1–1
9. Qian Y, Dong J, Wang W, Tan T (2015) Deep learning for steganalysis via convolutional neural networks. In: Proceedings of the international society for optics and photonics, media watermarking, security, and forensics 2015, San Francisco, CA, USA, February 9-11, 2015, p 94090J
10. Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp 1–9
11. Tan S, Li B (2014) Stacked convolutional auto-encoders for steganalysis of digital images. In: Asia-pacific signal and information processing association annual summit and conference, APSIPA 2014, Chiang Mai, Thailand, December 9-12, 2014. IEEE, pp 1–4
12. Tang W, Li H, Luo W, Huang J (2014) Adaptive steganalysis against WOW embedding algorithm. In: ACM information hiding and multimedia security workshop, IH & MMSec ’14, Salzburg, Austria, June 11-13, 2014, pp 91–96
13. Xu G (2017) Deep convolutional neural network to detect J-UNIWARD. In: Proceedings of the 5th ACM workshop on information hiding and multimedia security, IH&MMSec 2017, Philadelphia, PA, USA, June 20-22, 2017. ACM, pp 67–73
14. Xu G, Wu H, Shi Y (2016) Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett 23(5):708–712
15. Xu G, Wu H, Shi Y (2016) Ensemble of CNNs for steganalysis: an empirical study. In: Proceedings of the 4th ACM workshop on information hiding and multimedia security, IH&MMSec 2016, Vigo, Galicia, Spain, June 20-22, 2016. ACM, pp 103–107
16. Ye J, Ni J, Yi Y (2017) Deep learning hierarchical representations for image steganalysis. IEEE Trans Inf Forensic Secur 12(11):2545–2557
17. Yedroudj M, Comby F, Chaumont M (2018) Yedrouj-net: an efficient CNN for spatial steganalysis. CoRR arXiv:1803.00407
18. Yu C, Li J, Li X, Ren X, Gupta BB (2018) Four-image encryption scheme based on quaternion fresnel transform, chaos and computer generated hologram. Multimedia Tools Appl 77(4):4585–4608. https://doi.org/10.1007/s11042-017-4637-6
19. Zhang R, Zhu F, Liu J, Liu G (2018) Efficient feature learning and multi-size image steganalysis based on CNN. CoRR arXiv:1807.11428

Weike You received her B.S. degree in network engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2015. She is currently pursuing her Ph.D. degree in cyberspace security at the State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences (CAS), Beijing. Her current research interests include steganography and steganalysis in digital media.

Xianfeng Zhao received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2003. From 2003 to 2005, he was a postdoctoral fellow with the Data Assurance and Communication Security Center, Chinese Academy of Sciences (CAS), Beijing. From 2006 to 2011, he was an associate professor with the State Key Laboratory of Information Security (SKLOIS), Institute of Software, CAS, Beijing. Since 2012, he has been a professor with SKLOIS, which was moved to Institute of Information Engineering, CAS, Beijing, in 2012. Prof. Zhao's research interests are information hiding and multimedia security, including watermarking, steganography, and steganalysis. In these fields, he has published more than 100 papers, owned and applied more than 20 patents. He serves as PC members of many conferences such as IWDW, IWDCF, and CSS, and from 2014, he has been an associate editor of International Journal of Digital Crime and Forensics.



Sai Ma received his B.S. degree in communication engineering from Sichuan University, Chengdu, China, in 2015. He is currently pursuing his Ph.D. degree in cyberspace security at the State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences (CAS), Beijing. His current research interests include steganography and steganalysis in digital media.

Yaqi Liu received the B.S. degree from Beijing Technology and Business University, Beijing, China, in 2013 and the M.S. degree from Beijing Technology and Business University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree for information security in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include image processing and multimedia security.