A Customized Convolutional Neural Network with Low Model Complexity for JPEG Steganalysis

Junwen Huang
Sun Yat-sen University
Guangzhou, China
huangjw38@mail2.sysu.edu.cn

Jiangqun Ni*
Sun Yat-sen University
Guangzhou, China
issjqni@mail.sysu.edu.cn

Linhong Wan
Sun Yat-sen University
Guangzhou, China
wanlh5@mail2.sysu.edu.cn

Jingwen Yan
University of Shantou
Shantou, China
jwyan@stu.edu.cn

ABSTRACT

Nowadays, convolutional neural network (CNN) is appied to different types of image classification tasks and outperforms almost all traditional methods. However, one may find it difficult to apply CNN to JPEG steganalysis because of the extremely low SNR (embedding messages to image contents) in the task. In this paper, a selection-channel-aware CNN for JPEG steganalysis is proposed by incorporating domain knowledge. Specifically, instead of random strategy, kernels of the first convolutional layer are initialized with hand-crafted filters to suppress the image content. Then, truncated linear unit (TLU), a heuristically-designed activation function, is adopted in the first layer as the activation function to better adapt to the distribution of feature maps. Finally, we use a generalized residual learning block to incorporate the knowledge of selection channel in the proposed CNN to further boost its performance. J-UNIWARD, a state-of-the-art JPEG steganographic scheme, is used to evaluate the performance of the proposed CNN and other competing JPEG steganalysis methods. Experiment results show that the proposed CNN steganalyzer outperforms other feature-based methods and rivals the state-of-the-art CNN-based methods with much reduced model complexity, at different payloads.

CCS CONCEPTS

• **Security and privacy** → *Authentication*; • **Computing methodologies** → *Machine learning*.

KEYWORDS

steganalysis, JPEG, selection channel, CNN

ACM Reference Format:
Junwen Huang, Jiangqun Ni, Linhong Wan, and Jingwen Yan. 2019. A Customized Convolutional Neural Network with Low Model Complexity for JPEG Steganalysis. In *ACM Information Hiding and Multimedia Security Workshop (IH&MMSec '19), July 3–5, 2019, Paris, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3335203.3335734

1 INTRODUCTION

Existing schemes of JPEG steganography tend to embed secret messages in DCT domain by modifying quantized DCT coefficients. To ensure the security of steganography, the state-of-the-art methods adopt content-adaptive property by defining the distortion function to evaluate embedding cost of quantized DCT coefficients. That is, the modification in complex regions which are difficult to model is assigned to a low embedding cost or high embedding probability, and vice versa. In DCT domain, UED [7], UERD [8] and J-UNIWARD [13] exhibit good security performance. And J-UNIWARD is the superior one among them.

On the other hand, steganalysis focuses on detecting the existence of secret messages embedded by a specific or any steganographic scheme in an object. Basically, a universal feature-based image steganalysis method includes three stages. Firstly, extract noise residuals of the given image with diverse well-designed models. Secondly, construct high-dimensional features based on the noise residuals. Thirdly, conduct a binary classification with an ensemble classifier [15], which takes the high-dimensional features as input. For example, SRM [6] and its variants [11] in spatial domain, DCTR [12] and GFR [17] in DCT domain, all adopt such framework and achieve outstanding performance. By incorporating the knowledge of selection channel, one can further improve the performance of above methods [3–5].

In recent years, applications of convolutional neural network (CNN) have achieved great success, especially in the tasks of computer vision. For steganalysis of spatial images, our previously proposed YeNet [20] is the first CNN that outperforms significantly the best hand-crafted feature sets, e.g., SRM [6], in detection performance. When it comes

[*] Corresponding author.

IH&MMSec '19, July 3–5, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6821-6/19/06. . . $15.00
https://doi.org/10.1145/3335203.3335734

to the JPEG steganalysis, although the statistics of JPEG stego images might be well maintained in DCT domain, the associated statistical artifacts in spatial domain (decompressed JPEG images) would be somehow magnified due to the relatively large quantization step size. Keep this in mind, the CNN-based steganalyzer should take the decompressed JPEG images (in spatial domain) as input. In other words, the YeNet could be generalized to JPEG steganalysis with modifications in some key nodes.

In this paper, we propose a selection-channel-aware CNN by generalizing YeNet to DCT domain with several key modifications for JPEG steganalysis. To tackle the issue of low SNR in the task of JPEG steganalysis, domain knowledge is introduced into the proposed CNN in different ways. Firstly, recognize that the first convolutional layer plays the role of residual extractor, kernels of this layer are initialized with 30 basic high-pass filters in SRM [6] to improve the SNR. Secondly, TLU (Truncated Linear Unit) proposed in [20] is adopted as the activation function of the first layer instead of ReLU or TanH, where a universal threshold T is selected to better adapt to the distribution of the residual maps. Finally, we further boost the performance of the proposed CNN-based JPEG steganalyzer by taking advantage of the knowledge of selection channel through an elaborately-designed learning structure. Experiments are carried out to evaluate the performance of the proposed CNN at different payloads, which demonstrates that the proposed CNN-based JPEG steganalyzer outperforms the best hand-crafted feature sets, e.g., GFR, by a clear margin, and rivals the state-of-the-art CNN-based methods, e.g. SRNet and SCA-SRNet, with much reduced model complexity.

The rest of this paper is organized as follows. We provide an overview of the proposed CNN architecture and elaborate the details in Section 2. Experiment results and analysis are given in Section 3. We summarize this paper in Section 4.

2 THE PROPOSED CNN

2.1 Overview

The proposed CNN architecture is illustrated in Fig. 1, which includes two stages. The first stage preprocesses the JPEG image and its selection channel. Feature maps extracted from corresponding branch are known as residual measure and residual distortion measure, which are merged with a simple elementwise summation and activated by ReLU. The two integrated signal channels are then fed into the subsequent deep CNN for feature extraction and classification, which will be discussed later in the following subsections.

The second stage is a deep CNN. The CNN first performs a normal convolution, which is then followed by six cascaded residual building blocks proposed in [10] as shown in Fig. 2. In these blocks, we avoid using 1×1 convolution with a specific stride to reduce the size of feature maps, for it may cause potential performance degradation. Instead, we propose to use 3×3 convolution with stride 2 or 3, or average pooling followed by 1×1 convolution with stride 1. Then, all feature maps are fed into a global average pooling layer to generate a feature vector of the given image. A fully-connected layer and softmax function map the vector to classification probability.

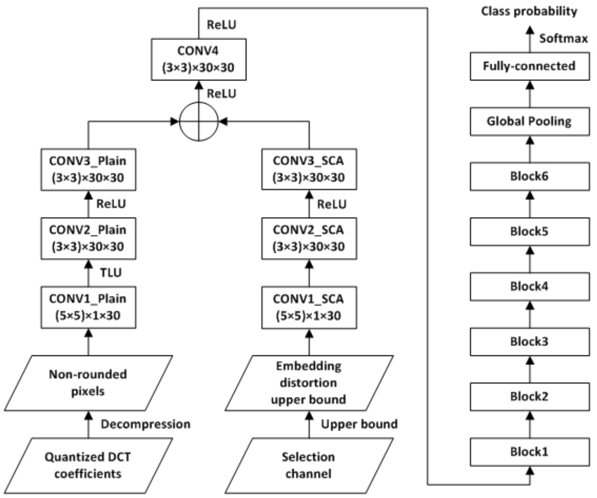

Figure 1: The proposed CNN architecture. Layer types and configurations are inside the boxes. Dimensions of kernels follow: (height × width) × input × output.

Note that we call the proposed CNN with the knowledge of selection channel as SCA-CNN, while it degenerates into a Plain-CNN when the knowledge of selection channel is not available.

2.2 The Preprocessing Layer

Domain knowledge is introduced into the proposed CNN by the heuristically-designed architecture, especially the customized setting of the first convolutional layer for steganalysis. We elaborate details in this subsection.

2.2.1 *Initialization.*

Given a JPEG image with size of $M \times N$ (both M, N are multiples of 8), let $\mathbf{c} = \left(c_{kl}^{(m,n)}\right)$ be the matrix of quantized DCT coefficients, where $c_{kl}^{(m,n)}$ is the (k,l)th element of the (m,n)th 8×8 block, and $1 \leq m \leq M/8$, $1 \leq n \leq N/8$, $0 \leq k, l \leq 7$. The (i,j)th element of the (k,l)th DCT basis, $0 \leq i, j \leq 7$, is computed as

$$f_{ij}^{(k,l)} = \frac{\omega_k \omega_l}{4} \cos \frac{\pi k(2i+1)}{16} \cos \frac{\pi l(2j+1)}{16}, \qquad (1)$$

where $\omega_0 = 1/\sqrt{2}$ and $\omega_k = 1$ for $k > 0$. Then one can decompress the JPEG image and obtain non-rounded pixel values with

$$x_{ij}^{(m,n)} = \sum_{k,l=0}^{7} f_{ij}^{(k,l)} q_{kl} c_{kl}^{(m,n)}, \qquad (2)$$

where $x_{ij}^{(m,n)}$ is the (i,j)th pixel value in the (m,n)th 8×8 block of the decompressed image $\mathbf{x}$, and q_{kl} is the element of corresponding JPEG luminance quantization matrix. As shown in Fig. 1, Plain-CNN takes $\mathbf{x} = \left(x_{ij}^{(m,n)}\right)$ as input.

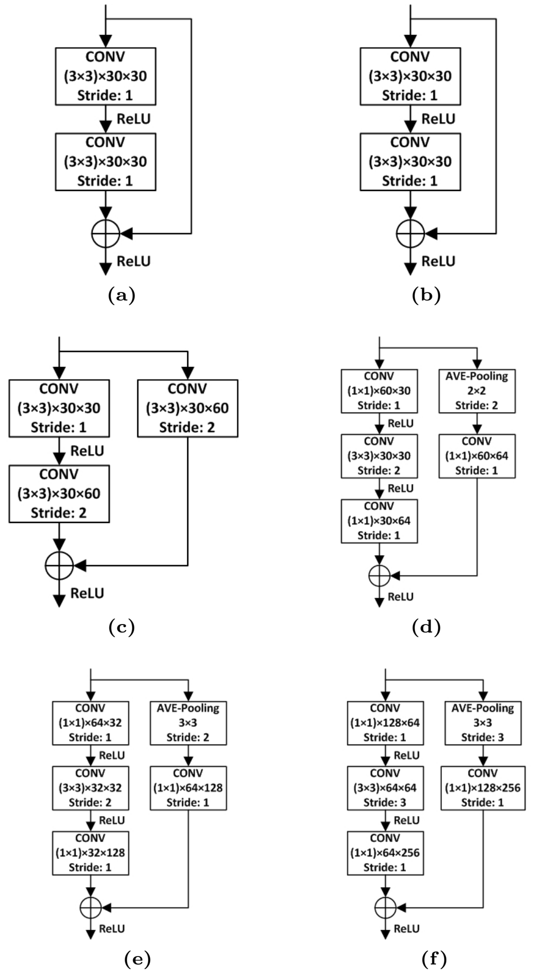

Figure 2: Building blocks used in the proposed CNN. (a)-(f) are block 1-6 respectively. Layer types and configurations are inside the boxes. Dimensions of kernels follow: (height × width) × input × output.

Similar to many existing CNN-based JPEG steganalysis methods, the first convolutional layer of the proposed CNN is expected to act as a residual extractor, which suppresses image content and improves the SNR for steganalysis. For the initialization of kernels in the first layer, a CNN with low model complexity usually fails to learn a good residual extractor on a relatively small dataset if initialized with random numbers. Thus, we propose to initialize the kernels with hand-crafted filters which leads to a much better starting point of training stage. Besides, inspired by the residual computation in feature-based methods, high-pass filter bank, such as basic linear filters in SRM [6], DCT bases in DCTR [12] and Gabor filters in GFR [17], may all serve as the good choice to initialize the parameters. Therefore, to reduce the model complexity of the proposed CNN, 30 basic linear filters in SRM (the "spam" filters and their symmetrical version), are adopted to initialize the kernels of the first layer. Because kernel size of the first layer is set to be 5×5 as shown in Fig. 1, zero-padding is first applied to filters to attain the equal size. Note that kernels of the first convolutional layer could be further updated during the training stage of the proposed CNN.

Table 1: Classification Accuracy of Plain-CNN with Different Values of T at 0.4 bpnzAC for QF=75

Threshold	15	31	63
Accuracy	0.9227	0.9292	0.9252

2.2.2 Activation Function. As verified by extensive experimental results in [20], TLU (truncated linear unit) is a better activation function that adapts to the distribution of noise residuals in steganalysis, where the threshold T is the only hyper-parameter of TLU to be determined. For JPEG steganalysis, however, the residual distributions in spatial domain corresponding to the embedding changes in DCT domain should be evaluated to determine T, which is relevant to the QFs and embedding rates in a way, but it is not distinctive. Therefore, in our work, we suggest selecting a fixed T according to the performance of Plain-CNN, as shown in Table 1. Because the CNN trained at 0.4 bpnzAC for QF=75 is the prime network to seed others in Section 3.2, we select T according to its detection performance. It is shown that most of the key artifacts due to the embedding changes in DCT domain is preserved in the case of $T = 31$, resulting in the best classification accuracy among all candidate values. Therefore, we set the threshold of TLU to be 31. Considering the tradeoff between performance and computation speed, TLU is only adopted in the first layer as in [20].

2.3 Incorporate Selection Channel

2.3.1 Residual Distortion Measure. When the steganalyzer is exactly aware of the scheme associated with the JPEG steganography, the selection channel $\boldsymbol{\beta}$, or the embedding change probabilities of the quantized DCT coefficients $\mathbf{c}$, can be computed according to the steganographic algorithm. To incorporate the knowledge of selection channel, [5] proposed to use the upper bound $\mathbf{t}(\boldsymbol{\beta}) = \left(t_{ij}^{(m,n)}\right)$ of L_1 embedding distortion to characterize the knowledge of selection channel for JPEG steganography (residual distortion measure), which is computed as

$$t_{ij}^{(m,n)} = \sum_{k,l=0}^{7} \left| f_{ij}^{(k,l)} \right| q_{kl} \beta_{kl}^{(m,n)}. \tag{3}$$

A computationally efficient residual distortion measure is also given in [5]:

$$\delta_{uSA}^{1/2}(\boldsymbol{\beta}) = \sqrt{\mathbf{t}(\boldsymbol{\beta}) * |\mathbf{g}|}, \tag{4}$$

where $\mathbf{g}$ is a filter in the high-pass filter bank. By incorporating $\delta_{uSA}^{1/2}(\boldsymbol{\beta})$ into some feature-based methods, these methods become selection-channel-aware and outperform their prior arts. For the proposed CNN-based JPEG steganalyzer, the high-pass filter bank is the basic linear filters in SRM.

2.3.2 Design of SCA-CNN. Given a JPEG image, we decompress quantized DCT coefficients $\mathbf{c}$ to obtain non-rounded pixel values $\mathbf{x}$ according to (2) and compute $\mathbf{t}(\boldsymbol{\beta})$ according to (3), where $\boldsymbol{\beta}$ is selection channel. For SCA-CNN, $\mathbf{x}$ and $\mathbf{t}(\boldsymbol{\beta})$ are two different signals from the same image. The key issue to design SCA-CNN is how to incorporate $\mathbf{t}(\boldsymbol{\beta})$ into Plain-CNN.

If we follow the scheme proposed in [20] to incorporate $\mathbf{t}(\boldsymbol{\beta})$, the first stage of the proposed CNN should be as illustrated in Fig. 3a. Input $\mathbf{t}(\boldsymbol{\beta})$ is convolved by $|\mathbf{g}|$, the absolute value of the kernels shared by the preprocessing layer, then output feature maps are activated by a square root function to obtain $\delta_{uSA}^{1/2}(\boldsymbol{\beta})$ according to (4). In this case, only parameters in left branch ($\mathbf{x}$) are learnable. Then the two signal channels, i.e., truncated residual maps and $\delta_{uSA}^{1/2}(\boldsymbol{\beta})$, are simply integrated by an elementwise summation.

For the proposed SCA-CNN, we make the two signal channels learnable, as shown in Fig. 3b, for the following reasons. Firstly, $|\mathbf{g}|$ do not participate in back propagation explicitly in [20]. Secondly, $\delta_{uSA}^{1/2}(\boldsymbol{\beta})$ may be a too specific formula of residual distortion measure to incorporate into a CNN. These two issues may prelimit to some extent the solution space of residual distortion measure. Obviously, it is wise to design a architecture to learn a suitable residual distortion measure. Therefore, in Fig. 3b, $\mathbf{x}$ and $\mathbf{t}(\boldsymbol{\beta})$ are first fed into the CNN, then they are convolved by three layers respectively. The left branch aims to extract feature maps for residual measure, while the right branch is expected to extract feature maps for residual distortion measure. As elaborated in Section 2.2.1 and 2.2.2, the preprocessing layer is initialized with 30 filters in SRM, followed by TLU activation. In addition, the first layer in the right branch (selection channel) is initialized with the absolute value of 30 filters in SRM, and no activation function is applied. All parameters in both branches are learnable when training the CNN in order to extract discriminate feature maps to merge, especially the learnable residual distortion measure. Elementwise summation is adopted to integrate two signal channels, and then followed by ReLU activation. In fact, our scheme to incorporate the knowledge of selection channel is expected to behave as a generalized residual learning block of ResNet illustrated in Fig. 4.

Compared with YeNet, the first stage of our scheme includes more parameters, which results in more discriminate feature maps of $\mathbf{x}$ and $\mathbf{t}(\boldsymbol{\beta})$. Besides, such architecture makes residual distortion measure learnable, which is expected to further boost performance of the proposed CNN.

2.4 Strategy for Low Payload and High QF Steganalysis

When the payload gets lower, it is difficult for CNNs to converge to a good local optimum from scratch. In [20], the performance gain between YeNet and SCA-YeNet gets smaller when the payload gets higher. Such phenomenon implies that CNN takes advantage of the knowledge of selection channel better if it is trained on the training set generated at a higher payload, because the content-adaptive property of almost all steganographic schemes gets weaker in this case. Therefore, curriculum learning [1] or transfer learning strategy [16] is suitable for the case of low payload steganalysis. In practice, one can train the proposed CNN from scratch at a high payload, e.g. 0.4 bpnzAC, and save the best model according to the performance on validation set. Then the saved model is fine-tuned in the case of low payload, e.g. 0.3 bpnzAC,

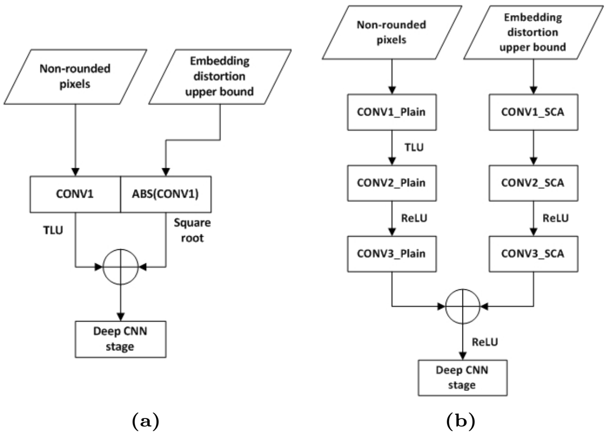

Figure 3: Different schemes to incorporate the knowledge of selection channel into the proposed CNN. (a) Scheme that follows [20]. (b) Our scheme.

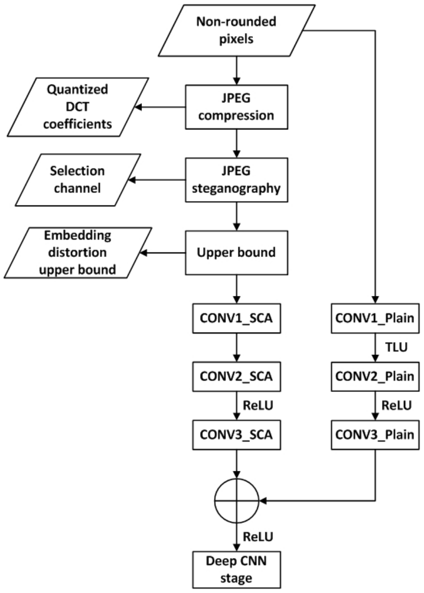

Figure 4: The first stage of the proposed SCA-CNN is designed to behave as a generalized residual learning block of ResNet.

and further fine-tuned in other harder cases, such as lower payloads and higher QFs. According to our experiments, such scheme for low payload steganalysis is time-saving.

3 EXPERIMENT RESULTS

3.1 Experiment Setup

Experiments were carried out to evaluate the performance of the proposed CNN-based JPEG steganalyzer. The whole dataset comes from two image sources, BOSSbase v1.01 and BOWS2. All involved images are resized to the ones of 256×256. The dataset setup is exactly same as [2]. That is, the training set contains 14,000 cover-stego pairs (4,000 from BOSSbase and 10,000 from BOWS2). The validation set containing 1,000 cover-stego pairs is utilized to select hyper-parameters, and final testing results are reported after classification of 5,000 cover-stego pairs in test set. The performance of the proposed CNN is evaluated at payload which ranges from 0.1 to 0.5 bpnzAC for quality factors 75 and 95. On the other hand, it is unnecessary to prepare a validation set for feature-based methods. Thus, the training set for feature-based methods is the union set of training set and validation set for CNN, and the performance will be still evaluated on the same test set.

We implemented the proposed CNN based on Tensorpack [18] and trained it with Adamax [14] method. As stated in Section 2.3.2, kernels of the first layer to convolve $\mathbf{x}$ are initialized with high-pass filters in SRM, and the bias term of this layer is initialized to 0. Kernels to convolve $\mathbf{t}(\boldsymbol{\beta})$ are initialized with the absolute value of corresponding filters. Batch normalization is applied to all convolutional layers except these two layers. Other convolutional layers are all initialized using [9]. Note that "SAME" padding is applied to all convolutional layers of the proposed CNN. Weights of the fully-connected layer are initialized from Gaussian distribution with zero mean and 0.01 standard deviation, and the bias term is initialized to 0. All weights in the proposed CNN except the first stage are regularized with L_2 norm. Cross-entropy and the regularization term whose weight decay is 5×10^{-4} is adopted as loss function to minimize during training stage. During training stage, a mini-batch contains 16 cover-stego pairs, and data augmentation (mirroring, rotation or the combination of both) is randomly applied to each cover-stego pair before a training iteration. Note that the raw non-rounded pixels $\mathbf{x}$ is fed into the CNN without any pre-processing. When the proposed CNN is trained from scratch at 0.4 bpnzAC for QF=75, the learning rate is set to 0.002 first and divided by 10 after the 130th and 230th epoch. When it is fine-tuned in other cases, the learning rate is set to 0.001 initially and divided by 10 after the 50th and 80th epoch. We conducted all of the following experiments on an NVIDIA GeForce GTX TITAN X GPU card.

3.2 Comparison with Other State-of-the-Art Methods

Expressed as classification accuracy, the performance of the proposed CNN as well as other state-of-the-art steganalysis methods is shown in Table 2 at payload which ranges from 0.1 to 0.5 bpnzAC in the case of QF=75. It is observed that both Plain-CNN and SCA-CNN outperform GFR and its selection-channel-aware version, the best hand-crafted feature set for JPEG steganalysis, by a clear margin. Compared with SCA-GFR, Plain-CNN improves the accuracy by 10.11% at 0.4 bpnzAC and 12.82% at 0.2 bpnzAC, respectively. For CNN-based JPEG steganalyzer, we compare the performance with the state-of-the-art schemes, i.e., XuNet [19] and the recently emerged SRNet [2] and its selection-channel-aware version SCA-SRNet, which are summarized in Table 2 below. It is shown that our proposed Plain-CNN and SCA-CNN outperform the XuNet by a clear margin and have a comparable performance with SRNet and SCA-SRNet. It is also observed that the SCA-CNN outperforms consistently Plain-CNN for tested payloads. In specific, the performance of SCA-CNN is boosted by 1.31% at 0.4 bpnzAC and 3.67% at 0.2 bpnzAC respectively, indicating that the SCA-CNN could better take advantage of the knowledge of selection channel to achieve better detection performance and makes it one of the best CNN based methods at present. However, the performance gains between SCA-CNN and Plain-CNN tends to decrease as the embedding rate increases. This is because the adaptivity of the adopted steganographic scheme, J-UNIWARD, declines for large embedding payload, which is also reported in [20]. When the quality factor is increased to 95, all JPEG steganalyzers suffer from performance degradation because it is more difficult to detect embedding messages in higher quality factor. However, Plain-CNN and SCA-CNN still retain the comparable performance with SRNet and SCA-SRNet.

3.3 Model Complexity

As the large-scale deployment of AI applications on thin clients, e.g., mobile terminals, the CNNs with low model complexity are becoming increasingly demanded. We compare the model complexity of the proposed network with other competing CNN-based schemes in terms of the number of model parameters as shown in Table 3. It is noticed that, although the proposed CNN models only show the comparable performance with SRNet and SCA-SRNet, which are the state-of-the-art CNN-based JPEG steganalyzers, they indeed exhibit much lower model complexity, say, lower than one order of magnitude. This is attributed to the customized design of the network structure by incorporating the domain knowledges, e.g., the initialization of the kernels with SRM in first layer and the introduction of the heuristically-designed activation function TLU. With the emergence of techniques for neural network search, the model complexity of the proposed networks is expected to be further reduced in the future.

4 CONCLUSION

How to design a customized CNN for JPEG steganalysis still remains a challenging topic because of the low SNR in the task. To tackle this issue, a customized designed architecture for JPEG steganalysis is proposed by incorporating the

Table 2: Comparison with Other State-of-the-Art Methods in Terms of Classification Accuracy

Method	Payload (bpnzAC), QF=75					Payload (bpnzAC), QF=95				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
GFR	0.5489	0.6290	0.7140	0.7945	0.8623	0.5128	0.5430	0.5861	0.6390	0.7056
SCA-GFR	0.5811	0.6755	0.7554	0.8281	0.8846	0.5197	0.5527	0.5986	0.6522	0.7126
XuNet	0.5763	0.7011	0.8107	0.8839	0.9227	0.5136	0.5376	0.5919	0.6603	0.7578
SRNet	0.6799	0.8111	0.8847	0.9330	0.9615	0.5723	0.6560	0.7484	0.8238	0.8852
SCA-SRNet	0.7310	0.8374	0.9079	0.9422	0.9679	0.6288	0.6757	0.7654	0.8360	0.8904
Plain-CNN	0.6746	0.8037	0.8787	0.9292	0.9605	0.5708	0.6709	0.7530	0.8280	0.8808
SCA-CNN	**0.7307**	**0.8404**	**0.9050**	**0.9423**	**0.9661**	**0.6381**	**0.6873**	**0.7734**	**0.8415**	**0.8875**

Table 3: Model Complexity in Terms of the Number of Model Parameters

Network	Parameter Number
XuNet	5754k
SRNet	4779k
SCA-SRNet	4779k
Plain-CNN	**234k**
SCA-CNN	**252k**

domain knowledges of the task. Firstly, to set a good start point for training CNN, kernels of the first convolutional layer are initialized with the high-pass filter bank from S-RM instead of random numbers. Secondly, to better adapt to the distribution of feature maps, the activation function TLU in our previous work is generalized to the application of JPEG steganalysis with relevant modification. Thirdly, to incorporate the knowledge of selection channel, a generalized residual learning block is introduced into the proposed CNN to further boost the performance. Experiments are carried out to evaluate the proposed CNN model and other competing methods, which shows that our customized CNN-based JPEG steganalyzer achieves the best performance compared with other hand-crafted feature sets, and has the comparable performance with the state-of-the-art CNN-based methods with much reduced model complexity.

ACKNOWLEDGMENTS

This work is supported in part by National Natural Science Foundation of China under Grants U1726315 and 61772573, and in part by the Science and Technology Program of Guangzhou under Grants 201707010029 and 201804010265.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

[2] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. 2018. Deep Residual Network for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security* (2018).

[3] Tomáš Denemark, Jessica Fridrich, and Pedro Comesaña-Alfaro. 2016. Improving selection-channel-aware steganalysis features. *Electronic Imaging* 2016, 8 (2016), 1–8.

[4] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. 2014. Selection-channel-aware rich model for steganalysis of digital images. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on.* IEEE, 48–53.

[5] Tomáš Denemark Denemark, Mehdi Boroumand, and Jessica Fridrich. 2016. Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security* 11, 8 (2016), 1736–1746.

[6] Jessica Fridrich and Jan Kodovsky. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 868–882.

[7] Linjie Guo, Jiangqun Ni, and Yun Qing Shi. 2014. Uniform embedding for efficient JPEG steganography. *IEEE transactions on Information Forensics and Security* 9, 5 (2014), 814–825.

[8] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi. 2015. Using statistical image model for JPEG steganography: uniform embedding revisited. *IEEE Transactions on Information Forensics and Security* 10, 12 (2015), 2669–2680.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Vojtech Holub and Jessica Fridrich. 2013. Random projections of residuals for digital image steganalysis. *IEEE Transactions on Information Forensics and Security* 8, 12 (2013), 1996–2006.

[12] Vojtěch Holub and Jessica Fridrich. 2015. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security* 10, 2 (2015), 219–228.

[13] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *Computer Science* (2014).

[15] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. 2012. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security* 7, 2 (2012), 432–444.

[16] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2016. Learning and transferring representations for image steganalysis using convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on.* IEEE, 2752–2756.

[17] Xiaofeng Song, Fenlin Liu, Chunfang Yang, Xiangyang Luo, and Yi Zhang. 2015. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 15–23.

[18] Yuxin Wu et al. 2016. Tensorpack. https://github.com/tensorpack/.

[19] Guanshuo Xu. 2017. Deep convolutional neural network to detect J-UNIWARD. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 67–73.

[20] Jian Ye, Jiangqun Ni, and Yang Yi. 2017. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2545–2557.