Deep Learning with Feature Reuse for JPEG Image Steganalysis

Jianhua Yang[1], Xiangui Kang[1*], Edward K.Wong[2] and Yun-Qing Shi[3]
[1] Guangdong Key lab of Information Security, School of Data and Computer Science,
Sun Yat-Sen University, Guangzhou, China
[*]E-mail: isskxg@mail.sysu.edu.cn
[2] Computer Science and Engineering, New York University, NY, USA
E-mail: ewong@nyu.edu
[3] Department of ECE, New Jersey Institute of Technology, NJ, USA
E-mail: shi@njit.edu

Abstract— **It is challenging to detect weak hidden information in a JPEG compressed image. In this paper, we propose a 32-layer convolutional neural networks (CNNs) with feature reuse by concatenating all features from previous layers. The proposed method can improve the flow of gradient and information, and the shared features and bottleneck layers in the proposed CNN model further reduce the number of parameters dramatically. The experimental results shown that the proposed method significantly reduce the detection error rate compared with the existing JPEG steganalysis methods, *e.g.* state-of-the-art XuNet method and the conventional SCA-GFR method. Compared with XuNet method and conventional method SCA-GFR in detecting J-UNIWARD at 0.1 bpnzAC (bit per non-zero AC DCT coefficient), the proposed method can reduce detection error rate by 4.33% and 6.55% respectively.**

I. INTRODUCTION

JPEG image steganography is a technology to hide secret messages into DCT coefficients for covert communication. Current steganography methods have become more and more sophisticated by designing a security distortion function so that the stego images are statistically undetectable from cover images. For instance, uniform embedding distortion (UED) [1] and uniform embedding revisited distortion (UERD) [2] design the distortion functions from DCT domain by allowing the embedding modifications to be proportional to the coefficient of variation of DCT coefficients, thus avoid statistical detect and keep the low computational complexity. Different from UED and UERD, other JPEG steganography method, e.g., JPEG universal wavelet relative distortion (J-UNIWARD) [3], combine the DCT domain and spatial domain to design the distortion functions. So far, J-UNIWARD have achieved the best security performance.

With the rapid development of steganography, steganalysis has also made substantial progress to detect hidden messages in a suspicious image. Conventional steganalysis feature sets, such as the union of JPEG Rich Model and Spatial Rich Model (J+SRM) [4], Discrete Cosine Transform Residual (DCTR) [5], Gabor filtering residual method (GFR) [6] and PHase Aware pRojection Model (PHARM) [7], extract noise residuals by convolving the decompressed (non-rounded) JPEG image with high pass filters. In work [8], based on decision rough set α-positive region reduction, Ma *et al* reduced feature dimensions of J+SRM and GFR. To further enhance the steganalysis feature sets, Denemark *et al* [9] proposed a method to incorporate the prior probabilistic knowledge (selection channel awareness) by accumulating in the histograms a quantity that bounds the expected absolute distortion of the residual. The experimental results have shown that selection channel awareness (SCA) based method, e.g., SCA-GFR can provide a substantial detection gain. After finished the feature extraction, the ensemble classifiers [10] are applied to the feature set for classification task.

Recently, image steganalysis has made significant progress by using convolutional neural networks (CNNs) in the spatial domain [11-16]. However, there are only a few works in the JPEG domain [17-19].

In [17], Zeng *et al.* proposed a hybrid deep-learning structure based on the large-scale ImageNet dataset. In the preprocessing phase, they used hand-crafted convolutional layers with 5 × 5 DCT patterns from DCTR [5] for the decompressed images, and performed quantization and truncation, which was similar to conventional steganalysis methods. The experimental results showed that the integration of quantization and truncation can boost the detection accuracy.

In [18], Chen *et al*. proposed a phase-split module by splitting the feature maps into 64 parallel channels to make the architecture be aware of JPEG phase. To suppress the image content and increase the high frequency stego signal, four 5 × 5 high-pass filters, which include a "KV filter", a "point filter", and 2 Gabor filters had been used. Experimental results showed that different kinds of high pass filters can complement each other and the detection performance will be improved by using JPEG phase awareness.

In [19], Xu proposed a 20-layer CNN structure for JPEG steganalysis. In the preprocess step, the fixed DCT high pass filters were used to suppress the image content effectively. The residuals calculated from high pass filters are processed by an absolute layer, then truncated with a slightly tuned global threshold value of 8. All the pooling layers were performed by 3 × 3 convolutional with a stride of 2, and the shortcut

This work was supported by NSFC (Grant Nos. U1536204, 61772571, 61702429), and the special funding for basic scientific research of Sun Yat-Sen University (6177060230).

TABLE I
THE PROPOSED CNN ARCHITECTURE

Group	Output data	Process	/
Group 1	16 × (256 × 256)	High pass filtering	×1
Group 2	16 × (256 × 256)	Truncation	×1
Group 3	64 × (128 × 128)	32 × (3 × 3) conv (stride 1) 64 × (3 × 3) conv (stride 2)	×1
Group 4	96 × (64 × 64)	96 × (1 × 1) conv (stride 1) 32 × (3 × 3) conv (stride 1)	×2
		128 × (1 × 1) conv (stride 1) 96 × (3 × 3) conv (**stride 2**)	×1
Group 5	96 × (32 × 32)	96 × (1 × 1) conv (stride 1) 32 × (3 × 3) conv (stride 1)	×2
		128 × (1 × 1) conv (stride 1) 96 × (3 × 3) conv (**stride 2**)	×1
Group 6	96 × (16 × 16)	96 × (1 × 1) conv (stride 1) 32 × (3 × 3) conv (stride 1)	×2
		128 × (1 × 1) conv (stride 1) 96 × (3 × 3) conv (**stride 2**)	×1
Group 7	96 × (8 × 8)	96 × (1 × 1) conv (stride 1) 32 × (3 × 3) conv (stride 1)	×2
		128 × (1 × 1) conv (stride 1) 96 × (3 × 3) conv (**stride 2**)	×1
Group 8	160 × (1 × 1)	96 × (1 × 1) conv (stride 1) 32 × (3 × 3) conv (stride 1)	×2
		8 × 8 global average pooling	×1
Group 9	2 × (1 × 1)	2D fully-connected Softmax	×1

connection [20] has been incorporated to achieve a deep architecture. Experimental results have shown that this method can obtain state-of-the-art performance. We refer to it as XuNet [19] in the rest of this paper. Due to the limited computational resource, we only compare the CNN-based method XuNet [19] and conventional method SCA-GFR in this work.

In this paper, we proposed a feature-reuse CNN model for JPEG image deep steganalysis similar to the Dense Convolutional Networks (DenseNet) [21]. The features are reused by concatenating them from all the previous layers within a group.

The rest of this paper is organized as follows. Details of the proposed CNN model is described in Section II. Experiments and analysis are given in Section III. The conclusion and future works are described in Section IV.

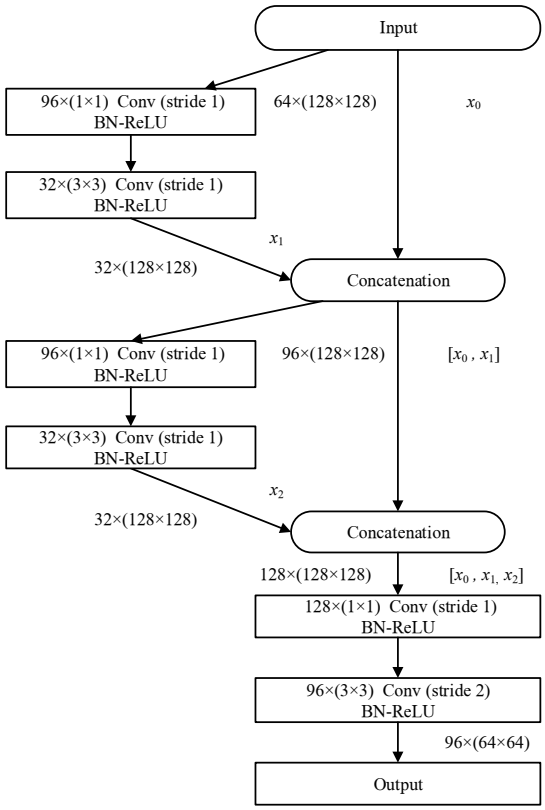

Fig. 1 The architecture of Group 4.

II. THE PROPOSED CNN METHOD

The whole architecture of the CNN model is shown in Table I. It contains the high-pass filters in Group 1, the truncation layer in Group 2, "conv" layers in Group 3, feature reuse layers in Groups 4 ~ 8 and classifier layers in Group 9. Note that in Table I, each convolutional layer is followed by a batch normalization and a ReLU layer, so that each "conv" layer shown in Table I corresponds to a sequence of layers: convolution-batch normalization-ReLU. When we mention that a "conv" layer is used, it means that the sequence of layers is used. The size of the convolutional kernels is equal to number of kernels × height × width. Feature reuse is achieved by concatenating the feature maps with the same size within a group. The feature reuse in Group 4 is shown in Fig. 1. The feature reuse within groups 5 to 8 is similar to that in Group 4. The classifier layers in Group 9 contain a fully-connected layer and a Softmax layer.

A. High pass filtering

The embedded signal, *i.e.*, hidden information, is regarded as the embedding noise, it is very weak when compared with image content. Therefore, steganalysis with deep learning is learning in a very low signal-to-noise ratio (SNR) case. In order

to enhance SNR, a JPEG compressed image was first decompressed to the spatial domain without rounding off the pixel values to integers so as to avoid any loss of information, then the decompressed image is high-pass filtered via the convolution with sixteen 4×4 DCT high-pass filters $B^{(k,l)}$ [5] [19]:

$$B^{(k,l)} = (B^{(k,l)}_{(m,n)}), \quad 1 \le k, l \le 4; 1 \le m, n \le 4 \tag{1}$$

where (k, l) is the index of DCT high-pass filters, (m, n) is the index of a position in 4×4 window.

$$B^{(k,l)}_{mn} = \frac{w_k w_l}{4} \cos\frac{\pi k(2m+1)}{16} \cos\frac{\pi l(2n+1)}{16} \tag{2}$$

and

$$w_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0, \\ 1, & k > 0 \end{cases} \qquad w_l = \begin{cases} \frac{1}{\sqrt{2}}, & l = 0 \\ 1, & l > 0 \end{cases}$$

Given an $M \times N$ decompressed image $X \in R^{M \times N}$, the residual can be calculated from the convolution with DCT high-pass filters.

$$R(X) = X * B^{(k,l)}, \ 1 \le k, \ l \le 4 \tag{3}$$

where '*' denotes a convolution. The high-pass filters are initialized to DCT filter kernels first, then the parameters of high-pass filters are optimized together with other parameters in the architecture during training stage. For easier exposition, we use padding process to make the size of feature maps be a multiple of 8.

B. Truncation Layer

In conventional method, the residuals generated from high pass filters are performed quantization and truncated for statistical calculation. In this work, the value of the feature maps x generated by high-pass filter is first truncated with a threshold value T, and T is chosen to be 8 to stay the same as XuNet. Truncated linear unit (TLU) [16] activation function is applied as follows:

$$f(x) = \begin{cases} -T, & x < -T \\ x, & -T \le x \le -T \\ T, & x > T \end{cases} \tag{4}$$

It is also observed in our work that the training process with TLU function will converge faster than with Tanh or ReLU activation function instead.

C. Feature reuse

Motivated by the deep learning work of DenseNet [21] in computer vision, in this low SNR deep learning work, the features obtained from previous layers are reused by concatenation within a group. In Fig. 1, we show the concatenation of feature maps in Group 4, "input" and "output" mean the input feature maps with size of 128 × 128 and the output feature maps with size of 64 × 64, respectively. As shown in Fig.1, the feature-maps from the previous layers can be used by subsequent layers in the same group, thus the flow of information and gradients throughout the network are improved and will make the proposed architecture easy to train. The process in layer ℓ was shown in Eq. (5), where H_ℓ represents the process with Convolution, Batch Normalization [22] and ReLU (Conv-BN-ReLU) functions:

$$x_\ell = H_\ell([x_0, x_1, ..., x_{\ell-1}]) \tag{5}$$

where $[x_0, x_1, .., x_{\ell-1}]$ refers to the concatenation of the feature maps produced in layers 0,1 ..., $\ell - 1$.

Note that in this low SNR steganalysis task, we cannot apply the DenseNet [21] directly. The main differences between the proposed architecture and the DenseNet [21] are: 1) The pre-activation mode used in DenseNet does not improve the performance of steganalysis, therefore, in our work, we use post-activation, *i.e.*, each convolution layer is followed by a Batch Normalization (BN) and ReLU function; 2) Average pooling layer in the DenseNet is replaced with a convolution layer with stride 2 in order to enhance the propagation of the weak signal. As shown in Fig. 1, we use a convolution layer with stride 2 before "output" instead of average pooling layer.

III. Experiments And Analysis

A. Software Platform and Hyperparameters

Caffe toolbox [23] is selected to implement the proposed CNN architecture. Parameters are updated by the stochastic gradient descent (SGD). A mini-batch of 32 images with 16 cover-stego pairs is used as the input for each training iteration. The momentum is set to 0.9 and the learning rate is initialized with a value 0.001, and then divided by 5 every 30,000 iterations. The convolutional kernels are initialized using a zero-mean Gaussian distribution with a standard deviation 0.01, except that the fully connected layers are initialized using "Xavier" initialization. The biases are initialized to 0.2 in the 3 × 3 convolutions and disabled in the 1 × 1 convolutions.

B. Datasets

BOSSbase. BOSSbase dataset v1.01 [24] contains 10,000 grayscale images. All 10,000 images are first resampled to the size of 256 × 256 by using "imresize" function in MATLAB with default parameter setting, then JPEG compressed with a quality factor 75. The corresponding stego images were generated through data embedding into the compressed cover images. We select 6,000 images for training and the remaining 4,000 images for validation. In training stage, a training image is randomly flipped, and rotated by a multiple of 90 degrees for data augmentation.

ImageNet. In order to test the performance on large scale dataset, 500,000 images were randomly selected from ImageNet ILSVRC 2013 classification dataset [25]. For each image, a 256 × 256 image block is cropped from its left-top region, then converted to a grayscale one and compressed with JPEG quality factor 75. The corresponding stego images were generated through data embedding into the compressed cover images. We use 80% images for training, 10% for validation and 10% for testing.

C. Classification Results on BOSSbase

We compare our method with XuNet [19] and the conventional steganalysis methods SCA-GFR [9] for J-UNIWARD with payloads ranging from 0.1 to 0.5 bpnzAC (bit per non-zero AC DCT coefficient). The performance of the

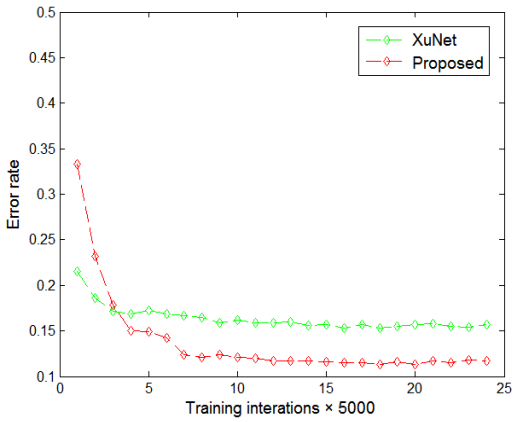

Fig. 2 Comparison of validation errors versus training iterations between the proposed CNN and the XuNet for J-UNIWARD at 0.4 bpnzAC on BOSSbase.

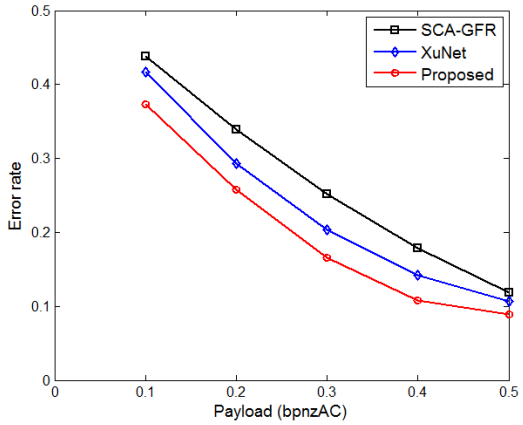

Fig. 3 Detection error for proposed CNN, XuNet and SCA-GFR for J-UNIWARD on Bossbase.

TABLE II
CLASSIFICATION ERROR RATES FOR J-UNIWARD ON BOSSBASE

Payload	Proposed CNN	XuNet	SCA-GFR
0.1 bpnzAC	**0.3730**	0.4163	0.4385
0.2 bpnzAC	**0.2572**	0.2927	0.3391
0.3 bpnzAC	**0.1661**	0.2030	0.2522
0.4 bpnzAC	**0.1084**	0.1416	0.1781
0.5 bpnzAC	**0.0890**	0.1072	0.119

steganalyzer is evaluated using an average classification (detection) error rate:

$$P_E = \frac{1}{2}(P_{FA} + P_{MD}) \tag{6}$$

where P_{FA} and P_{MD} are the false-alarm and missed-detection probabilities.

The validation errors of the proposed CNN and XuNet after 120,000 iterations on BOSSbase for J-UNIWARD at 0.4 bpnzAC are shown in Fig. 2. It shows that the validation error rate of the proposed CNN becomes lower than XuNet after 20,000 iterations and after 100,000 iterations, the performance of two methods becomes steady. We train the proposed CNN model and XuNet with 120,000 iterations, for three times separately. In each time, the trained model is saved every 5,000 iterations, the last three saved models are selected. After training, we obtain nine trained CNN models, and report the average detection error rate for J-UNIWARD in Table II. In Fig. 3, we further illustrate the classification error rates with payloads ranging from 0.1 to 0.5 bpnzAC. It is observed from Table II that the proposed CNN method can reduce the detection errors significantly than XuNet method [19] and SCA-GFR method [9]. For example, at 0.1 bpnzAC on J-UNIWARD, the proposed CNN method achieves 4.33% advantage over XuNet method and 6.55% advantage over SCA-GFR method. Furthermore, the number of parameters in the proposed CNN model is only 16% of that used in XuNet [19], *i.e.*, the numbers of parameters are 924,080 and 5,750,836 respectively.

TABLE III
CLASSIFICATION ERROR RATES FOR DIFFERENT VARIANTS ON THE BOSSBASE FOR J-UNIWARD AT 0.4 BPNZAC

Proposed CNN	#1	#2	#3	#4
0.1151	0.1383	0.1221	0.1546	0.1454

In order to investigate the influence of different parts of the proposed CNN model, we test the following variants of the proposed CNN method.

Variant #1: Replace all 1 × 1 convolutional layers with 3 × 3 convolution layers.

Variant #2: Fix the parameters of the sixteen DCT high pass filters in the training.

Variant #3: Replace all convolutional pooling with average pooling.

Variant #4: Add an absolute layer after the high pass filtering in Group 1.

The classification error rates in the validation of each variants on BOSSbase for J-UNIWARD at 0.4 bpnzAC are shown in Fig. 4. The lowest error rates during 80,000 iterations for comparison have been shown in Table III. It is observed that the proposed compact CNN architecture with the feature concatenation, 1 × 1 convolutional layer and convolutional pooling layers, obtain better performance than all variant methods.

In XuNet, all convolutional kernels were 3 × 3 convolutional layers, thus the parameters also increased. Form variant #1, replacing the 1 × 1 convolutional layers by 3 × 3 convolutional layers makes the detection accuracy decrease, this result may be caused by that the 1 × 1 convolutional layers can project the input channel onto a new channel space, thus improve the cross-channel correlations. What's more, replacing the 1 × 1 convolutional layers by 3 × 3 convolutional layers leads to dramatical increment of training parameters, thus it easier to overfit.

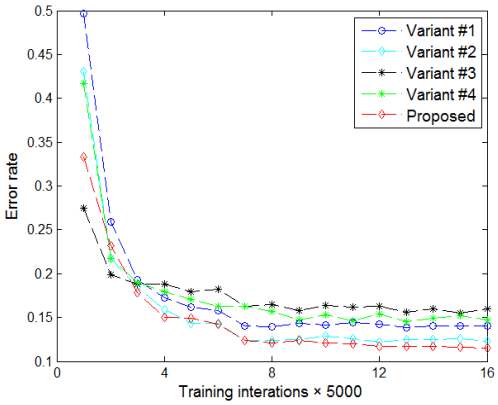

Fig. 4 Comparison of validation errors versus training iterations between the proposed method and different variants for J-UNIWARD at 0.4 bpnzAC on BOSSbase.

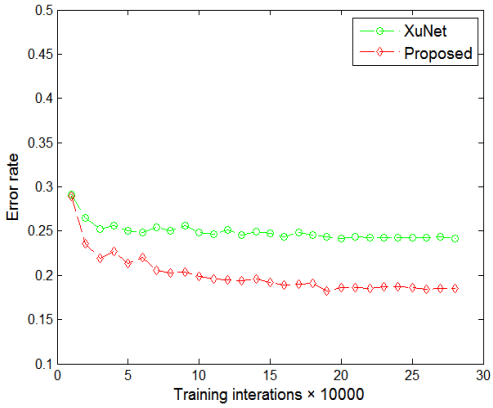

Fig. 5 Comparison of validation errors versus training iterations between the proposed CNN method and XuNet for J-UNIWARD 0.4 bpnzAC on Imagenet.

TABLE IV
CLASSIFICATION ERROR RATES IN DETECTING J-UNIWARD ON IMAGENET

Algorithm	0.2 bpnzAC	0.4 bpnzAC
Proposed CNN	**0.3304**	**0.1534**
XuNet	0.3868	0.2228

Note in XuNet, all high pass filters are fixed during the training stage, thus variant #2 shows one advantage of the proposed method than XuNet. From variant #3, the result demonstrates the advantage of the convolutional pooling layers.

XuNet has an absolute operation layer after the pre-processing of high pass filtering. It is observed from the result of variant #4 that the performance increases for JPEG steganalysis without the absolute operation layer in our proposed CNN model.

D. *Classification Results on ImageNet*

In this experiment on attacking for J-UNIWARD at 0.4 bpnzAC, we train the CNN model in 280,000 iterations for 0.4 bpnzAC on the training images from ImageNet dataset. After every 10,000 training iterations, we perform validation on the validation images from ImageNet dataset, and obtain a validation error rate. The validation error rates versus the numbers of training iterations are shown in Fig. 5. In Fig. 5, we also compare the proposed CNN with the XuNet method [19]. On attacking J-UNIWARD at 0.2 bpnzAC, the CNN model is trained and fine-tuned from the trained CNN model for 0.4 bpnzAC.

After the CNN model is trained, we conduct tests on the 50,000 cover-stego pairs test images from ImageNet using the saved trained model with the best validation result. Test results are shown in Table IV. We compare the proposed CNN method with the XuNet method using the same mini-batch of 40 images with 20 cover-stego pairs. It is observed from Table IV that the proposed method can also achieve lower error rates on large scale dataset. For example, for J-UNIWARD at 0.4 bpnzAC, compared with the XuNet method, our method achieves 6.94% lower in terms of test error rates.

IV. CONCLUSIONS

In this paper, we propose a 32-layer convolutional neural networks (CNNs) with feature reuse by concatenating all features from the previous layers. The proposed method can improve the flow of gradient and information, the shared features and bottleneck layers further reduce the number of parameters in the proposed CNN model dramatically, *i.e.*, the number of parameters is only 16% of that used in the CNN model of the existing XuNet method. Experimental results show that the proposed method can further lower detection error and achieve the state-of-the-art performance.

For the future work, we will incorporate the selection channel aware (SCA) into the CNN architecture, which is similar to conventional method for JPEG steganalysis.

REFERENCES

[1] L. Guo, J. Ni and Y. Q. Shi, "Uniform Embedding for Efficient JPEG Steganography," in IEEE Transactions on Information Forensics and Security, vol. 9, no. 5, pp. 814-825, May 2014.
[2] L. Guo, J. Ni, W. Su, C. Tang and Y. Shi, "Using Statistical Image Model for JPEG Steganography: Uniform Embedding Revisited," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2669-2680, Dec. 2015.
[3] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," EURASIP Journal on Information Security, vol. 2014, no. 1, p. 1, 2014.
[4] J. Kodovsky and J. Fridrich. "Steganalysis of JPEG images using Rich Models," in Proc. SPIE Int. Society for Optics and Photonics Engineering, Burlingame, vol. 8303A, pp. 1-13, Feb.2012.
[5] V. Holub and J. Fridrich, "Low-Complexity Features for JPEG Steganalysis Using Undecimated DCT," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 2, pp. 219-228, Feb. 2015.
[6] X. Song, F. Liu. C. Yang, X. Luo, Yi Zhang, "Steganalysis of adaptive jpeg steganography using 2d gabor filters," in Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security, pp. 15–23, Jun. 2015.
[7] V. Holub and J. J. Fridrich, "Phase-aware projection model for

steganalysis of jpeg images." in Media Watermarking, Security, and Forensics, p. 94090T, 2015.
[8] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang. "Selection of Rich Model Steganalysis Features Based on Decision Rough Set α-Positive Region Reduction, " in IEEE Transactions on Circuits and Systems for Video Technology, 2018.
[9] T. D. Denemark, M. Boroumand and J. Fridrich, "Steganalysis Features for Content-Adaptive JPEG Steganography," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 8, pp. 1736-1746, Aug. 2016.
[10] J. Kodovsky, J. Fridrich and V. Holub, "Ensemble Classifiers for Steganalysis of Digital Media," in IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 432-444, April 2012.
[11] Y. Qian, J. Dong, W. Wang and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, pp. 2752-2756, Sept. 2016.
[12] G. Xu, H. Wu and Y. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis, " in IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708-712, May 2016.
[13] G. Xu, H.-Z. Wu, and Y.-Q. S., "Ensemble of CNNs for steganalysis: an empirical study," in Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security. pp. 103–107, Jun. 2016.
[14] K. Liu, J. Yang and X. Kang, "Ensemble of CNN and rich model for steganalysis, " 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), Poznan, pp. 1-5, May. 2017.
[15] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi, "Steganalysis based on awareness of selection-channel and deep learning," in International Workshop on Digital Watermarking. Springer, pp. 263–272, Aug. 2017.
[16] J. Ye, J. Ni and Y. Yi, "Deep Learning Hierarchical Representations for Image Steganalysis," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2545-2557, Nov. 2017.
[17] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale jpeg steganalysis using hybrid deep-learning framework," arXiv preprint arXiv:1611.03233, 2016.
[18] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "Jpeg-phase-aware convolutional neural network for steganalysis of jpeg images," in 5th ACM Workshop Inf. Hiding Multimedia Security. (IH&MMSec), Jun. 2017.
[19] G. Xu, "Deep convolutional neural network to detect J-UNIWARD," in 5th ACM Workshop Inf. Hiding Multimedia Security. (IH&MMSec), Jun. 2017.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, Jun. 2016.
[21] G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks, " 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 2261-2269, Jul. 2017.
[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, pp. 448–456, July. 2015.
[23] Y. Jia, E. Shelhamer *et al*., "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia. ACM, pp. 675–678, Nov. 2014.
[24] P. Bas, T. Filler, and T. Pevny`, "Break our steganographic system. the ins and outs of organizing boss," in Information Hiding. Springer, pp. 59–70, May. 2011.
[25] O. Russakovsky, J. Deng, H. Su *et al*., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, Dec. 2015.