AG-Net: an Advanced General CNN model for Steganalysis

Han Zhang 1,2, Fuxian Liu 2, Zhihua Song 1, Xiaofeng Zhang 1, Yongmei Zhao 1,3

1 College of Equipment Management and UAV Engineering, Air Force Engineering University, Xi’an 710038, China
2 Air and Missile Defense College, Air Force Engineering University, Xi’an 710051, P.R.China
3 School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an, PR China

Corresponding author: Zhihua Song (e-mail: ola_alo@126.com).

This work was supported in part by the National Natural Science Foundation of China under Grant 71571190, 62002381.

ABSTRACT Steganalysis has made great progress over the past few years due to the advancement of deep convolutional neural networks (DCNNs), which have been successfully used to multi-domains. Correspondingly, the performance of steganalysis models inevitably encounters a bottleneck since the CNN based steganography models perform better. In this paper, we propose an Advanced General convolutional neural Network for steganalysis (AG-Net) to address this problem. We firstly design a confrontation module to extract and compare features of cover and stego images, which are captured from an unknown steganography network. Then, we construct the association between two adjacent confrontation modules according to the feature comparison of the previous module, to accumulate the differences of mid-and high-level features between the cover and stego images. Thirdly, we deliver the loss of the last confrontation module to a softmax layer after batch normalization and scalarization, to classify and detect stego images. Extensive experiments and evaluations demonstrate that the proposed AG-Net can achieve promising performance in response to different challenging steganographic algorithms.

INDEX TERMS general steganalysis; convolutional neural network; confrontation module

I. INTRODUCTION

Steganalysis is a method to judge whether the object of analysis contains secret information when the stego is known to use the carrier but the stego method is unknown. It is used to distinguish the objects of cover and stego, namely, the binary classification problem. At present, the research focuses on digital images as the carrier of hidden information, mainly with respect to the steganalysis method and technology of digital images [1].

Steganalysis belongs to the category of pattern recognition. Due to the successful application of deep learning in the field of pattern recognition, steganalysis technology has been developed. Steganalysis is divided into two stages: feature construction and classifier. Since the steganalysis operation modifies the high frequency signal of the image, in the feature construction stage, the high pass filter is often used to calculate the residual image, and then various statistical models are used to extract the features of steganalysis. Good features play a key role in the detection accuracy of steganalysis [2]. However, the analytical characteristics of traditional pairs are mainly based on artificial experience and heuristic design, which requires knowledge in specific fields [3]. The feature construction and classifier training of steganalysis are carried out independently, so it is difficult to achieve synchronous optimization between feature design and classifier. Aiming at the above problems of traditional steganalysis [4], this paper combines deep learning with steganalysis and uses the deep learning model to obtain the simulation complex representation of the model to achieve the purpose of automatic learning for effective feature expression. At the same time, using the end-to-end learning process of deep learning, the feature construction and classifier are completed synchronously in a structure. (If the optimization cannot be synchronized, it means that the classification guidance cannot be used to capture useful information in the feature extraction step.

Among many deep learning models, convolutional neural networks are the most representative and the most commonly used network model in steganalysis [5]. At present, the steganalysis method based on convolutional neural networks mainly uses two convolution layers, three convolution layers or five convolution layers to design the network. In 2014, Tan et al. first proposed a steganalysis method based on deep learning, and demonstrated the potential of steganalysis in the steganalysis field for the first time [6]. The Xu-Net network improves the feature learning ability. The Ynnynet network introduces channel selection in deep learning for the first time. Chen, Xu et al. introduced steganalysis of JPEG images in 17 years. Zeng proposed a hybrid deep learning steganalysis model for steganalysis of jpeg images. These are steganography methods for spatial
domain and JPEG images. In 2014, Goodflow et al., proposed a kind of generative confrontation network (GAN), that aims to promote the evolution of all parties by using game confrontation between two sides in the information hiding field, making the generative confrontation network popular [7].

With the continuous development of deep learning, convolutional neural network has achieved good results in steganalysis, but these results are achieved under the condition of a high embedding rate, and the detection effect is not good. Although the detection effects of some results under the condition of a low embedding rate are improved [8], this is not suitable for steganalysis of JPEG images. Follow-up research will change the network structure, number of layers, activation function and addition of filters are studied in order to improve the non-universality of steganalysis and improve the detection rate.

Therefore, in the framework of deep learning, we propose an improved general convolution neural network model, called AG-Net, for steganalysis. The main contributions and works are summarized as follows:

1. We firstly design a confrontation module to extract and compare the features extracted from the cover part and the stego part, which were originally intercepted from any kind of steganography network.

2. We then construct the association between two adjacent confrontation modules according to the feature comparison of the previous module, to accumulate the differences of mid-and high-level features between the cover and stego parts of the previous module.

3. We thirdly deliver the loss of the last confrontation module to a softmax layer after batch normalization and scalarization, to classify and detect stego images.

A large number of experiments show that the proposed algorithm achieves better performance in challenging steganography, and it outperforms existing techniques in terms of error probability in different steganography. It can improve the generality and accuracy of steganalysis to a certain extent. The rest of the paper is arranged as follows. After a brief review of CNN, in Section II, we describe the proposed AG-Net steganalysis model. In Section III, the experimental results are presented. Finally, Section V discusses the conclusion and future work.

II. RELATED WORKS

The structure of the convolutional neural network model is very similar to the traditional steganalysis process. The first step of steganalysis is residual acquisition, which is composed of several high-pass filters to obtain the residual image (the noise signal of the image). The first layer of the convolution neural network is a convolution layer, which is similar to a low-pass filter to obtain a feature map (the key content of the response image). The second and third steps of steganalysis are truncation and quantization, respectively, where the purpose is to reduce the residual range and retain the main residual information. The convolution layer of the convolution neural network is the activation function and pooling layer, whose functions are nonlinear transformation and dimensionality reduction of the feature map, respectively. After residual processing, steganalysis must summarize and reduce the residuals by certain statistical methods to obtain the features, while the convolutional neural network, at the end of the feature representation, obtains the merged feature graph through several fully connected layers. At the end of the steganalysis and convolution network, a classifier is connected to output the classification results. In the second part, we will review the existing steganalysis methods based on deep learning and focus on their advantages and disadvantages.

A. Steganalysis for spatial domain steganalysis

Spatial steganography is a method of hiding secret information within an image by processing the different pixel values of the covered image, thus affecting the statistical characteristics of the image. Next, we focus on the main contribution of the deep learning method in spatial steganalysis and compare different machine learning methods [9]. In 2014, Tan and Li first proposed a network architecture for steganalysis based on deep learning. Tan & Li et al. [10] proposed the first CNN structure for the steganalysis of digital images in the spatial domain. The similarities between SRM and CNN are comparable to spam. The network has a four-layer convolutional neural network structure. In addition, its average pool is better than its maximum pool. However, the FC layer is slow because it is too large. The network uses three models (three training modes) to detect the Hogo steganalysis algorithm; its embedding rate is 0.4 bpp, and the error rate is 48%. Spam and ram reflect the potential of steganalysis in the steganalysis field. Although the proposed model performs better than SPAM [24], it remains inferior to SRM [11]. Moreover, it reduces the dimensionality of features and becomes a useful tool in steganalysis. The generated network is shown in Table 1:

| Xu-net | Improvement measures | Implementation steps |
|--------|----------------------|----------------------|
|        | add absolute value layer | after the first convolution |
|        | Tanh Activation function | used in the first two layers of network |
|        | 1 * 1 convolution kernel | in layer 3-5 network |

In 2015, Qian [12] et al. first proposed a network model combining steganalysis with deep learning. Its advantages are restoration B/W steganalysis and GCNN. This model is better than spam and introduces a Gaussian activation function and HPF layer. The hybrid set function is used to improve the feature learning ability of the network front end, and the activation function ReLU is used to reduce the difficulty of back-propagation. The global average pool and 1 * 1 convolution kernel are used in the back end of the network to...
prevent overfitting and information loss. However, the performance of this network is inferior to that of SRM, and their networks will not converge without a high pass filter. BN is increased to improve the convergence rate and avoid falling into local minima. The performance of this CNN network for steganalysis is comparable to that of the spatial rich model [13]. This paper not only emphasizes the role of a high pass filter in network convergence but also proves that transfer learning is conducive to detecting a low embedding rate steganography framework. In 2016, Pibre et al. [14] designed a CNN-based steganalysis network with poor performance but higher detection accuracy.

In the same year, Xu-Net [15] V1 et al. added many new network structures on the basis of Qian-Net. This network uses an absolute ABS activation layer, BN (batch normalization) and pool layer. Its primary advantage is that it not only ensures more effective learning characteristics but also effectively avoids overfitting and other problems and improves the accuracy of SRM [16]. The absolute ABS activation layer is used to generate the feature map generated by the first convolution layer. It can learn more efficient features that may help avoid overfitting problems.

Zeng [17] et al. proposed a hybrid deep learning steganalysis model for detecting steganalysis methods of JPEG images, but its network performance cannot be compared with PHARM. Xu [18] et al. proposed a 20-layer deep learning neural network for JPEG steganalysis: that is, a fast connection technique. This network framework uses a convolution layer instead of a pooling layer, which improves the detection accuracy to a certain extent. Compared with Zeng et al., the error rate is reduced by 35%. Chen then proposed an improved deep learning framework based on the JPEG compression algorithm based on Xu-Net and achieved good network performance. Wu [19] et al. proposed a deep residual network framework for detecting adaptive JPEG steganography. The layer number of the network architecture is deeper than before. Although batch normalization is not effectively used in the network, the layer number of the network exerts a very obvious effect on capturing the statistical characteristics of digital images. Huang proposed a ResDet with a high embedding rate. In 2018, Yedrouj-Net [20] proposed a CNN that outperforms the state-of-the-art methods in terms of error probability. In 2020, Eslam M discards the pre-processing step, to reduces the computational cost [31].

B. Based on JPEG image steganalysis

Other networks were proposed in 2017 for JPEG image steganalysis. In [21], the authors proposed a hybrid deep learning steganalysis model for detecting steganalysis methods in JPEG images. At present, JPEG based on deep learning is mainly realized by modifying the coefficient of variation. After inserting the secret information, the digital image is saved and transmitted through the network. The use of this file format is the most extensive and prominent. When compressed to one-eighth of the original image size, the image of this format has high visibility. However, compared with other weak signals, hidden information is difficult to find in JPEG compressed images.

Steganalysis researchers have applied deep learning technology to networks and achieved many successful results. We can further improve the accuracy of the deep learning framework by analyzing and comparing different network designs, fusion and learning strategies, and deep learning skills and introducing prior domain knowledge into the CNN architecture.

Convolutional neural networks have been continuously functioning in many directions and have achieved breakthroughs in speech recognition, face recognition, general object recognition, motion analysis, natural language processing and even analysis. For steganalysis, many studies have shown that convolutional neural networks have achieved very successful results. However, there are still two problems to be solved: the practicability of steganalysis and the accuracy of steganalysis are discussed.

III. Proposed AG-Net Steganalysis Framework

In this section, we propose a novel general CNN structure named “AG-Net” to improve the accuracy of spatial domain steganalysis. The core idea of the proposed AG-Net is to extract the features of cover and stego layer by layer and compare the differences. Each time, the comparison model level is enlarged for classification and identification. We design an improved frontal confrontation module to extract and compare the cover and stego images in the location steganography network and then guide the next group of modules through the output of the first module to construct the association between the two modules. Finally, after batch standardization and scalarization, steganography is used to classify and recognize steganalysis images. The core idea of the proposed AG-Net is to extract the features of cover and stego layer by layer and compare the differences. Each time, the comparison model level is enlarged for classification and identification. The architecture of the proposed AG-Net is shown in Figure 1.
FIGURE 1. Architecture of an Advanced General CNN model for Steganalysis. On the left side of the image is a general steganographic network, the process is unknown, and it is applicable to any framework. All we know is cover image and stego image. Therefore, the input objects of steganalysis network designed by us are only these two kinds of images, and there is no need to pay attention to other parameters from steganography network. On the right side is the stego analysis framework we designed. The input cover image and stego image are classified and identified by using softmax through the hierarchical progression of multiple comparison modules.

A. Pre-processing module

The pre-processing module is designed to process the input cover and stego images with convolutional blocks to filter residual noise components of images. The convolutional block is composed of a predefined high-pass convolutional layer with kernel, stride 1 and padding 2. By initial convolution, the pre-processing module can narrow dynamic ranges to suppress input images and promote the signal-noise ratio, which is computed by the weak stego signal and the strong image signal. We designed the pre-processing module according to previous studies [22], which certified that networks without pre-processing blocks converge more slowly and networks with pre-processing modules can express signals in a more compact and robust way.

As shown in Figure 1, the inputs of pre-processed module are the cover and stego images, for the cover and stego processing branches respectively, and the outputs of the module are convolutional images, feeding the remainder network. Since images are resized to 256×256, the output is 30×256×256 for convolutional block of each branch.

B. Confrontation module

The confrontation module is composed of two compound blocks (for the cover and stego branches) and a confrontation block, as shown in Figure 1. Designed for extracting deeper features, the compound block contains a convolutional layer, followed by a batch normalization layer, a nonlinear activation layer, and a pooling layer. The confrontation is designed to compare the features of cover and stego branches. The difference between the two branches is delivered to the next module to inspire learning of more contrastive features. As shown in Figure 1, the inputs of the confrontation module are the features learned by the cover and stego block of the previous module and the difference extracted by the confrontation block of the previous module (for the first confrontation module, no difference is delivered).

Considering the size of input images, we design four sequential connected confrontation modules, with different sizes for compound blocks of the four modules. The architectures of different compound blocks are shown in Figure 2. The parameters of different compound blocks are shown in Table 2.
Table 2: the parameters of different compound blocks.

| Block | input | convolution | stride | pad | output |
|-------|-------|-------------|--------|-----|--------|
| 1     | 256×256 | 30@ (5×5×30) | 1      | 2   | 128×128 |
| 2     | 128×128 | 30@ (3×3×32) | 1      | 1   | 64×64   |
| 3     | 64×64   | 32@ (3×3×64) | 1      | 1   | 32×32   |
| 4     | 32×32   | 64@ (3×3×128) | 1      | 1   | 1×1     |

As shown in Figure 2, all four blocks are composed of a convolutional layer, batch normalization layer, scale layer, activation layer and pooling layer. The function and parameter of every layer of the compound block are analyzed as follows:

1. A convolution layer. We set the size of convolutional layers according to Xu-Net [23], with kernel size 5×5 for Block 1 and 3×3 for Blocks 2, 3 and 4. The convolutional layers have no bias terms since they are moved to the scale layer.

2. A batch normalization (BN). The BN layer is used to normalize the distribution of each feature as described in reference [24]. The normalized features have a zero mean and a unit variance. By using a BN layer, the block can desensitize the training parameters for initialization [25], allowing us to speed up the learning with a larger learning rate and improving the classification accuracy.

3. A scale layer. The scale layer is a bias term separated from the convolutional layer as an independent part. Similar to ResNet [26], the scale layer attempts to learn more efficient translation parameters. Separating the scale function with the convolutional layer can enable learning of more typical features.

4. A nonlinear activation layer. The nonlinear activation layer is used to transform the inseparable features to a separable space. We employ the truncation function (Trunc) for Block 1 to narrow the range of data values, which are sparse and not statistically significant, to prevent deeper layers from modeling large values. The formula of Trunc is given in Eq. 1.

\[
Trunc(x) = \begin{cases} 
T, & x > T \\
 x, & -T \leq x \leq T \\
-T, & x < -T 
\end{cases} \tag{1}
\]

The nonlinear activation function of the other three blocks is ReLU. The ReLU can avoid vanishing and exploding gradients since it is sparse. The formula of ReLU is given in Eq. 2.

\[
ReLU(x) = \begin{cases} 
x, & x \geq 0 \\
0, & x < 0 
\end{cases} \tag{2}
\]

5. An average pooling layer. We use average pooling in Block 1, 2 and 3, and global average pooling in Block 4. The benefit of average pooling is down-sampling features, thus to reduce dimensionality. Compared with general average pooling, global average pooling can transport feature maps of any size to one by one element, to avoid grasping the embedded pixel information [27].

As shown in Figure 2 and Table 2, the parameters of different blocks differ according to their positions and functions. The input size of the first compound block of cover and stego branches is the output of the corresponding preprocessing module. In the process of graphics processing, a fixed high pass filter is used for image preprocessing to obtain the corresponding residual image. When using different convolution cores to process images, we must ensure that the output residual images obtained by filtering are acceptable. Among the four convolution layers, the number of kernels in each layer is set to 30, 30, 32 and 64, and the kernel size is set for Block 1 and for the other three blocks. The stride size is 1, and the padding size is 2 for Block 1 and 1 for the other three blocks. The stride and padding size are set to ensure that the size of the output is half that of the input.

The confrontation block is designed to learn the difference between the cover and stego branches. Therefore, the confrontation block is a D-value computation for the convolutional kernels in the cover and stego blocks of the same confrontation module. Obviously, the input of the confrontation block is the convolutional kernels of the cover and stego blocks, and the output is the D-value between the kernels of the two branches.

The confrontation module is the core of the entire network. The compound block can learn deeper features by its convolutional layer and achieve efficiency and robustness by the batch normalization layer, scale layer and pooling layer. The confrontation block can learn the difference between the cover branch and stego branch. The difference can also be accumulated by delivery to the next confrontation module, such that the learned features would be more representative.

### C. Connection between confrontation modules

The output of a previous confrontation module contains the features learned by the two compound blocks and the difference learned by the confrontation block. The features of compound blocks are delivered to the corresponding compound block of the following confrontation module. The difference of the previous module is delivered to the compound block of stego branch only, to initialized the feature maps of the convolutional layer. The difference delivered to the following stego compound block is used to guide the previous module lost to the following confrontation module, and further emphasize the difference between the middle and higher-level features of the cover branch and stego branch.

It is worth mentioning that the difference of the previous module has the same size with the convolutional kernel of the
compound block. When the difference is used to initialize the convolutional kernel of the following stego compound branch, there may have a problem that the sizes are not match. For example, the size of the difference of Block 1 is $5 \times 5$, and the size of the stego compound branch of Block 2 is $3 \times 3$. When it comes to this problem, we select the center part of the previous one to match the convolutional kernel of the following stego compound branch. So as to construct the association between two adjacent confrontation modules.

**D Classification module**

The main idea of this module is to classify the features of the fourth confrontation module. The classification module has three layers of full connection. The first two layers of full connection contain 128 nodes, and the activation function is ReLU. The last layer of full connection contains two nodes, and the activation function is softmax. Since many papers have described this function in detail, the readers are referred to the original papers [32] for more information.

**IV. SIMULATION AND TRAINING ENVIRONMENT**

We conduct several experiments to evaluate the effectiveness of the proposed AG-Net in this section. First, we introduce the dataset and different steganography methods used in experiments. Then, we present the parameters used in the training and test process. Third, the experimental results are displayed and analyzed in detail.

**A. Dataset and steganography methods**

In this research, experiments are conducted on a commonly used publicly available source BOSSBase. BOSSBase is a standard database in steganalysis with 10000 images. All of the images are selected from seven types of cameras. The cover image is gray and formed by zooming, clipping and other operations in the format of PGM. The size of images is $512 \times 512$. We selected some categories of BOSSBase and show them in Figure 3. Figure 3 contains different scenes such as architecture, and scenery. In terms of texture characteristics, when the image is magnified 200 times, more changes of the image will be focused on the position with complex texture.

![Figure 3: Samples of several categories in BOSSBase dataset. Images have different pattern when magnified to different times. The first column is the original image. The second column is a dense (noisy) image. The third column is the difference value, which is the interpolation image caused by the difference value that has not been magnified, which is completely black. The fourth column is the image magnified 30 times.

We use three well-known content adaptive steganography algorithms, HILL [28], WOW and SCUNIWARD, for the embedding of secret images in the spatial domain. All three steganography algorithms are carried out with an STC simulator using publicly available codes. It is worth mentioning that we use MATLAB as our platform instead of C++ to create a random embedding key, which can prevent overfitting in the training set and gain a better generalization capability. In contrast, deep learning-based steganography algorithms could achieve extraordinary classification performance with a fixed embedding key created on the C++ platform. This similar phenomenon was proven in [29], where the authors created stego images with different embedding keys.

Our network was trained on a PC with an Intel Xeon (R) bronze 3104 CPU, operating speed of 1.7 GHz, Titan x GPU card, 32 GB DDR4 memory, NVIDIA GeForce RTX2080Ti graphics processing unit (GPU), and memory of 11 GB. To further verify the feasibility of this method, our configuration is based on the common PyTorch toolkit experimental environment [30]. The open-source operating system of this paper is based on the deep learning framework of the software.

**B. Parameters in training and test process**

Since the input size of the proposed AG-Net is $256 \times 256$, we resampled all the $512 \times 512$ images of BOSSBase to $256 \times 256$. For training and testing, we split the resized BOSSBase into two sets, 50% for cover and 50% for stego. Both for cover and stego sets, 4000 out of the 5000 images are randomly selected for training, and the remaining are set for validation. We mainly evaluate the performance of all methods under three kinds of payloads: 0.3, 0.4 and 0.5 bit / pixel (BPP). Note that AG-Net is trained with different embedding schemes at different payloads, for tasks of varying difficulties, which accumulates and guide the decay of the learning rate.

**C. Experimental results and analysis**

We conducted several experiments with state-of-the-art deep learning based steganalysis algorithms for performance comparison and show the simulation results in Table 3.
It is obvious that the proposed AG-Net performs better than other deep learning-based methods. This result is possibly due to the following reasons: (1) the confrontation module can capture the most represented features; (2) the difference accumulates through adjacent confrontation modules.

IV. CONCLUSION

In this paper, we proposed an advanced general convolutional neural network for steganalysis, which is shown to perform better for general steganography methods. The experiments showed that AG-Net performs better than other state-of-the-art algorithms. Compared with other algorithms, AG-Net includes a confrontation module to compare and accumulate the differences between the cover and stego branches. Interesting avenues for future work include balancing efficiency and effectiveness.

REFERENCES

[1] Sajedi, H., & Jamzad, M. (2008). A Steganalysis Method Based on Contourlet Transform Coefficients. International Conference on Intelligent Information Hiding & Multimedia Signal Processing. IEEE Computer Society.

[2] Lionel, P., P Jérôme, Dino, I., & Marc, C. (2016). Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source mismatch. (Vol.2016, pp.1-11).

[3] Zeng, H. J., He, Q., Liu, G., Chen, Z., Zhang, B., & Ma, W. Y. (2006). Query-based snippet clustering for search result grouping. US, 276.

[4] Jalan, J. (2009). Feature selection, statistical modeling and its applications to universal jpeg steganalyzer. Dissertations & Theses Gradworks.

[5] Xiu, H. (2020). Research on Activation Function in Deep Convolutional Neural Network.

[6] Qian, Y., Ding, D., Wei, W., & Tan, T. (2016). Learning and transferring representations for image steganalysis using convolutional neural network. IEEE International Conference on Image Processing. IEEE.

[7] Wang, K., et al. "Generative Adversarial Networks: Introduction and Outlook." ACTA AUTOMATICA SINICA 44(2017):588-598.

[8] Padhye, J. Stepping-stone network attack kit (sneak) for evading timing-based detection methods under the cloak of constant rate multimedia streams. Dissertations & Theses Gradworks.

[9] Hussain, I., Zeng, J., & Tan, S. (2020). A survey on deep convolutional neural networks for image steganography and steganalysis. KSII Transactions on Internet and Information Systems, 14(3), 1228-1248.

[10] Tan, S., & Li, B. (2014). Stacked convolutional auto-encoders for steganalysis of digital images. IEEE.

[11] Fridrich, J. (2012). Rich models for steganalysis of digital images. IEEE transactions on information forensics and security.

[12] Qian, Y., D Jing, Wei, W., & Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. Proceedings of SPIE - The International Society for Optical Engineering, 9409, 94090J-94090J-10.

[13] Lizhe, Y., Bo, X., & Xianjie, W. (2010). Bp network model optimized by adaptive genetic algorithms and the application on quality evaluation for class teaching. IEEE, 3, V3-273 - V3-276.

[14] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(6), 1137-1149.

[15] Holub, V., & Fridrich, J. (2015). Low-complexity features for jpeg steganalysis using undecimated dct. IEEE Transactions on Information Forensics & Security, 10(2), 219-228.

[16] Fridrich, J. (2012). Rich models for steganalysis of digital images. IEEE transactions on information forensics and security.

[17] Zeng, J., Tan, S., Li, B., & Huang, J. (2017). Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis. Electronic Imaging.

[18] Xu, & Guanshuo. (2017). Deep Convolutional Neural Network to Detect J-UNIWARD. Acm Workshop on Information Hiding & Multimedia Security. ACM.

[19] Wu, S., Zhong, S., & Liu, Y. (2017). Deep residual learning for image steganalysis. Multimedia Tools and Applications.

[20] Wu, S., Zhong, S., & Liu, Y. (2017). Deep residual learning for image steganalysis. Multimedia Tools and Applications.

[21] Jishen, Zeng, Shunquan, Tan, Bin, & Li, et al. (2017). Pre-training via fitting deep neural network to rich-model features extraction procedure and its effect on deep learning for steganalysis. Electronic Imaging, 2017(7).

[22] Qian, Y., D Jing, Wei, W., & Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. Proceedings of SPIE - The International Society for Optical Engineering, 9409, 94090J-94090J-10.

[23] Jia, Y., Shellhammer, E., Donahue, J., Karayev, S., Long, J., & Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. ACM.

[24] Min, L., Ming, L. X., Xue, Y., Yu, Y., & Wang, M. (2016). Improvement of universal steganalysis based on spam and feature optimization. Springer International Publishing.

[25] Wu, C. Y., Singhal, N., & P K’Ahenbu Hl Supplementary Material: Video Compression through Image Interpolation.

[26] Fridrich, J., & Soukal, D. (2006). Matrix embedding for large payloads. IEEE Transactions on Information Forensics and Security, 1(3), 390-395.

[27] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. Computer Science.

[28] Jian, Y., Ni, J., & Yang, Y. (2017). Deep learning hierarchical representations for image steganalysis. IEEE Transactions on Information Forensics and Security, 12(11), 2545-2557.

[29] Bas, P., Filler, T., & Tomás Pevn. (2011). "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. Information Hiding - 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers. Springer-Verlag.

[30] Yang, J., Kai, L., Kang, X., Wong, E., & Shi, Y. (2017). Steganalysis Based on Awareness of Selection-Channel and Deep Learning. International Workshop on Digital Watermarking. Springer, Cham.

[31] Mustafa, E. M., Fouad, M. M., & Elshafer, M. A. (2020). Enhancing cnn-based image steganalysis on gpus. Journal of Information Hiding and Multimedia Signal Processing, 11(3), 138-150.

[32] Zhang, R., Zhu, F., Liu, J., & Liu, G. (2018). Efficient feature learning and multi-size image steganalysis based on cnn.
[33] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., D Erhan, & Goodfellow, I., et al. (2013). Intriguing properties of neural networks. Computer Science.