IAS-CNN: Image adaptive steganalysis via convolutional neural network combined with selection channel

Zhujun Jin¹,², Yu Yang¹,², Yuling Chen¹ and Yuwei Chen²

Abstract
Steganography is conducive to communication security, but the abuse of steganography brings many potential dangers. And then, steganalysis plays an important role in preventing the abuse of steganography. Nowadays, steganalysis based on deep learning generally has a large number of parameters, and its pertinence to adaptive steganography algorithms is weak. In this article, we propose a lightweight convolutional neural network named IAS-CNN which targets to image adaptive steganalysis. To solve the limitation of manually designing residual extraction filters, we adopt the method of self-learning filter in the network. That is, a high-pass filter in spatial rich model is applied to initialize the weights of the first layer and then these weights are updated through the backpropagation of the network. In addition, the knowledge of selection channel is incorporated into IAS-CNN to enhance residuals in regions that have a high probability for steganography by inputting embedding probability maps into IAS-CNN. Also, IAS-CNN is designed as a lightweight network to reduce the consumption of resources and improve the speed of processing. Experimental results show that IAS-CNN performs well in steganalysis. IAS-CNN not only has similar performance with YedroudjNet in S-UNIWARD steganalysis but also has fewer parameters and convolutional computations.

Keywords
Steganalysis, convolutional neural networks, selection channel, adaptive steganalysis, lightweight network, deep learning

Date received: 11 October 2019; accepted: 11 February 2020

Handling Editor: Yunpeng Li

Introduction
Image steganography is a technique that embeds secret information into the cover image and modifies the image content and statistical features as little as possible.¹ The embedding of secret information can be accomplished in two domains: spatial domain and frequency domain. Steganography based on the spatial domain is characterized by slightly modifying the pixel values to achieve similar visual quality between cover image and steganographic image. Steganography based on the frequency domain is generally applied to JPEG images and accomplished by changing discrete cosine transform (DCT) coefficients. Least significant bits (LSB)² is an early spatial-domain steganography algorithm which embeds secret information into the lowest significant bit of the pixel value of the cover image. The algorithm is simple but changes the statistical features of the image. Nowadays, many adaptive steganography algorithms have been proposed, such as HUGO,³ WOW,¹ and S-UNIWARD⁴ in the spatial domain.

¹State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China
²Information Security Center, Beijing University of Posts and Telecommunications, Beijing, China

Corresponding author:
Yuling Chen, State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, Guizhou, China.
Email: 61997525@qq.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
domain, in which texture-rich regions of images are selected to embed secret information. The main approach of adaptive steganography algorithms is to define distortion function and calculate the cost of pixel changes to estimate whether the pixel is suitable to be modified. The algorithm makes steganography trace difficult to detect and maintains higher order statistical features of images, which brings great challenges to steganalysis.

Steganalysis is a technique to detect the trace of steganography. Early steganalysis is mainly based on simple statistical features in lower dimensions. In order to detect adaptive steganography algorithms, steganalysis algorithms using higher order statistical features have been proposed, such as spatial rich model (SRM)\(^5\) and several models\(^6,7\) based on it. SRM generates a rich model of the noise component using a variety of high-pass filters and combines it with the support vector machine (SVM) or ensemble classifiers. However, to improve the performance of the model, the dimension of features becomes larger, which increases the computational complexity. In addition, the design of features in SRM depends on experience, and it is difficult to be optimized by a large margin.

Due to the development of convolutional neural networks (CNNs), a variety of steganalysis algorithms based on CNN were proposed to enhance the efficiency of steganalysis. Qian et al.\(^8\) proposed a network called gaussian-neuron convolutional neural network (GNCNN) which uses high-pass filtering (HPF) layer to enhance the steganographic noise and uses Gaussian function as the activation function. Xu et al.\(^9\) proposed a new network in which an absolute activation (ABS)\(^5\) layer is added after the first convolutional layer to improve statistical modeling in the following layers. In addition, a TanH\(^10\) activation function is used to avoid overfitting. Batch normalization (BN)\(^11\) is also used in the network to prevent the training of network from falling into poor local minima and optimize scales and biases for feature maps. The CNN proposed by Ye et al.\(^12\) utilizes a series of high-pass filters used to calculate residual maps in SRM to initialize the weights of the first convolutional layer. Furthermore, it incorporates the knowledge of selection channel\(^13\) and uses a new activation function called truncated linear unit (TLU) to improve the performance of CNN. Yadroudj et al.\(^14\) proposed a new network using TLU activation function and BN layer inspired by the work in XuNet\(^9\) and YeNet.\(^12\) Zhang et al.\(^15\) proposed a CNN utilizing separable convolutions\(^16\) and spatial pyramid pooling (SPP).\(^17\) Separable convolutions are used to achieve group convolution of residuals generated by high-pass filters. SPP enables the network to steganalyze arbitrary size images. However, the parameters of manually designed high-pass filters are fixed which cannot be adjusted with the learning of the network. Furthermore, state-of-the-art networks generally contain a large number of convolutional layers and convolutional kernels which will impact the efficiency of networks, and in most networks, all residuals are input to the network with the same importance.

In IAS-CNN, to enhance the steganographic noise and reduce the impact of image content, a high-pass filter in SRM is used to calculate residual maps. Considering that the manually designed filter is not necessarily optimal, the parameters of the filter are added to the learning of the network. In adaptive steganography, especially in the low steganography rate, the changes of images brought by steganography are slight. Inspired by the work in YeNet,\(^12\) to further enhance the signal of steganographic noise, the selection channel is combined into the network which aims to strengthen residuals in regions that have a high probability of being embedded with information and promote the network to learn key features. Embedding probability maps of images are computed and incorporated into the residuals before feature extraction in the proposed network. The depth of the network and the number of the parameters can influence the efficiency of the network. To improve the speed of network processing, the network is designed as a lightweight network.

The rest of the article is organized as follows. In section “Proposed CNN,” we introduce the overall architecture of our network and the details of each layer. In section “Experiments,” we present the experimental results and analysis. Finally, the conclusion and future research are summarized in section “Conclusion.”

**Proposed CNN**

**Overall architecture**

Figure 1 shows the overall architecture of the proposed network. The network contains pre-processing layer, feature extraction layer, and classification layer. In preprocessing layer, one of the filters of SRM is used to extract residual features of the image and then the features are combined with the knowledge of selection channel as the output. Feature extraction layer is composed of five convolutional layers and five average pooling layers. The final classification layer consists of two fully connected layers, two dropout layers, and a two-way softmax. In our network, the two-way softmax is implemented by a fully connected layer and softmax function. Some parameters of the proposed network have been shown in the figure. The specific processing of each layer will be described below.

**Pre-processing layer filter**

Steganography can be viewed as adding a slight amplitude noise to the cover image. We can hardly see the
difference between steganographic image and cover image from the image content. However, the noise changes the dependencies between neighboring pixels. Thus, the dependencies can be used to detect the steganographic noise which can be applied to the steganalyzers. High-pass filters designed in the SRM can be used to calculate residual maps and capture the noise of steganography. Therefore, we add a pre-processing layer for residuals calculation prior to feature extraction in the proposed network. At the same time, we also utilize the knowledge of selection channel which will be described in section “Selection channel.” In CNN, the operation of residuals information extraction can be accomplished by convolution.

Filters from SRM are usually used to simulate the extraction of residuals. To reduce the number of training parameters and improve the speed of processing, we chose a filter from SRM as the convolution kernel in the convolutional computation. The number of residual maps generated from each image in the pre-processing layer is related to the number of filters. If \( N \) filters are used, \( N \) residual extractions are performed for each image in the pre-processing layer, and the number of weights in the second convolutional layer can be expressed as \( 16 \times (3 \times 3) \times N \) which is shown in Figure 2. As the number of filters \( N \) increases, the number of parameters and the number of convolutional computations increase linearly in the first two layers of the proposed network, and then the consumption of training time also increases. When only one filter is used, only one residual map is generated from each image. Then in the second convolutional layer, the number of weights is significantly reduced, and the number of convolutional computations is also reduced, which can reduce training time and improve training efficiency.

Generally, filters of size \( 3 \times 3 \) or \( 5 \times 5 \) are chosen as the convolutional kernels of the pre-processing layer. Five filters selected from classes “First,” “Second,” “Third,” “SQUARE 3 × 3,” and “SQUARE 5 × 5” are used to initialize the first layer convolutional kernel, respectively. Filters in classes “First,” “Second,” and “SQUARE 3 × 3” are used to initialize convolutional kernels of size \( 3 \times 3 \). The remaining two filters are used to initialize convolutional kernels of size \( 5 \times 5 \). Since the manually designed convolutional kernel is not necessarily the optimal convolutional kernel, the learning of convolutional kernel is usually added to the CNN. In our network, the convolutional kernel of the

---

**Figure 1.** Overall architecture of the proposed network.

**Figure 2.** The changing trend of the number of weights in the second convolutional layer as the number of filters increases.
pre-processing layer is continuously adjusted as it learns. We normalize the selected filter and preserve the form of residual extraction before initializing the convolutional kernel of the pre-processing. Taking the filter of SQUARE $3 \times 3$ as an example, in equation (1), we use multiplication to change the center element to $-1$ while keeping the sum of the values of the convolutional kernel to 0. Our experimental results showed that the first layer convolutional kernel changed slightly in the training process, so it is no longer constrained during the network training process

$$W_1 = \frac{1}{4} \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad (1)$$

For the steganography algorithm WOW at the payload of 0.4, different filters are used to initialize the first layer of the network, and the experimental results are shown in Table 1.

Table 1. The comparison of network performance using different filters.

| Filter class | First | Second | Third | SQUARE 3 × 3 | SQUARE 5 × 5 |
|--------------|-------|--------|-------|--------------|--------------|
| Accuracy     | 0.7370 | 0.7670 | 0.7790 | 0.7895       | 0.7745       |

where $k_{rc}$ denotes the value of element in row $r$ and column $c$ of convolutional kernel, and $\theta_{ij} \in \{-1, 0, 1\}$ denotes the difference between the pixel value of steganographic image and that of cover image. Then, we use $p_{ij}$ to represent the probability that the pixel $x_{ij}$ is modified into $x_{ij} + 1$, as well as the probability that the pixel $x_{ij}$ is modified into $x_{ij} - 1$. These two ways of modification have the same probability in most of the existing steganography algorithms. As a result, the probability of modifying and not modifying pixel can be expressed as $2p_{ij}$ and $1 - 2p_{ij}$, and equation (3) can be verified. Furthermore, equation (4) can be derived

$$E[\theta_{ij}] = 2p_{ij} \quad (3)$$

$$\Psi(p_{ij}) = \sup E[|\delta_{ij}|] = 2 \sum_{r,c} |k_{rc}| \cdot p_{ij}^{rc} \quad (4)$$

We use $A = (2p_{ij})$ to represent embedding probability map and then $\Psi$ can finally be described as given in equation (5), in which $K$ denotes the convolutional kernel of the pre-processing layer

$$\Psi(A) = A \ast |K| \quad (5)$$

We calculate the costs of pixel modification and then estimate the embedding probability maps of images. $\Psi(A)$ is computed by convolving the $A$ with the absolute value of the pre-processing layer filter. In order to apply the knowledge of selection channel to the network, $\Psi(A)$ can be incorporated into the output of pre-processing through the method of elementwise multiplication or elementwise summation. From the perspective of multiplication, the output of the first layer can be described as equation (6)

$$Y_1 = R \times a\Psi(A) \quad (6)$$

where $R$ denotes the results of residual extraction, and $a$ is used to adjust the influence of selection channel on residual maps. We carried out experiments under the conditions of $a = 3$ and $a = 2$, respectively, and the experimental results are shown in Tables 2 and 3.

Tables 2 and 3 show that, in the case of $a = 3$, there is a big gap between training accuracy and validation accuracy on the epoch of 500. The accuracy of validation set is significantly lower than that of training set.
The network ran into the problem of overfitting, and the learning of selection channel concealed the learning of steganographic noise. In the case of \( a = 2 \), the performance of the network improved, which can also be seen from Table 4. Thus, in order to integrate the knowledge of selection channel into the network without excessive impact, we choose \( a = 1 \).

The selection channel can also be combined into CNN through elementwise summation. We estimated the performance of the method of summation and multiplication \((a = 1)\) by observing the training loss. Figures 3 and 4 show the training loss change of IAS-CNN using different methods to apply the knowledge of selection channel. We chose WOW as the steganography algorithm and 0.4 as the payload.

Comparing the two curves, we can easily find that the decline rate of training loss using elementwise summation method is significantly higher than that using elementwise multiplication method. Thus, we choose the method of elementwise summation in our network to use the selection channel.

From Ye et al. research, it can be indicated that when the activation function of each convolutional layer is ReLU, the propagation and contribution of \( \Psi(A) \) in the network can be expressed as equations (7) and (8)

\[
Y_2 = (R + \Psi(A)) \ast W_2 + B_2
\]

\[
= R \ast W_2 + B_2 + \Psi(A) \ast W_2
\]

Equation (7) represents the output of the second convolutional layer, where \( W_2 \) denotes the weights of the second convolutional layer and \( B_2 \) denotes the biases of the second convolutional layer. In addition, the output contains features extracted from the images and the selection channel of the images. Equation (8) represents the output of the \( n \)th convolutional layer. The pooling process is not represented in the equation, but the input of the fully connected softmax classifier still contains the above two parts through convolutional layers and pooling layers.

In summary, in order to better propagate the knowledge of selection channel, ReLUs are used as the non-linear activation functions from the second convolutional layer to the sixth convolutional layer. Table 5 shows the effect of selection channel on the accuracy of steganography detection. The steganography algorithm

---

Table 2. The performance of the network on WOW at the payload of 0.4 \((a = 3)\).

| Epoch | 100  | 300  | 500  |
|-------|------|------|------|
| Training accuracy | 0.6039 | 0.6754 | 0.7111 |
| Validation accuracy | 0.6090 | 0.6800 | 0.6675 |

Table 3. The performance of the network on WOW at the payload of 0.4 \((a = 2)\).

| Epoch | 100  | 300  | 500  |
|-------|------|------|------|
| Training accuracy | 0.6153 | 0.6959 | 0.7191 |
| Validation accuracy | 0.6030 | 0.6700 | 0.7070 |

Table 4. The comparison of network performance when \( a = 3 \) and \( a = 2 \) (on WOW at the payload of 0.4).

| \( a \) | 3  | 2  |
|-------|----|----|
| Test accuracy | 0.7240 | 0.7345 |

Figure 3. The training loss change of the elementwise summation method.

Figure 4. The training loss change of the elementwise multiplication method.
WOW and three payloads of 0.2, 0.4, and 1.0 were used to test.

**Feature extraction layer**

After the pre-processing layer, our network generates an output that contains the residuals of image and the selection channel of the same image. Then, the network needs to further extract features before inputting the features into the classifier. IAS-CNN uses five convolutional layers to extract features. The first four convolutional layers used 16 convolutional kernels of size $3 \times 3$, while the remaining one uses 16 convolutional kernels of size $5 \times 5$. At the same time, each convolutional layer is followed by an average pooling layer.

**Classification layer**

The network obtains the features extracted from the image after passing through the layers described above. As referred to the section “Selection channel,” the features are composed of two parts, one is extracted from the residual of the image and the other from the selection channel. Then, the features need to be integrated and divided into cover and stego, which are implemented in the classification layer. In order to achieve the above functions, the classification layer mainly consists of two components: fully connected layer and softmax layer. In summary, the classification layer takes the extracted features as input and the classification result as output.

Generally, most of the learning parameters of CNN exist in the fully connected layer. A large number of parameters in the fully connected layer can reduce the training efficiency of the network and make the network run into a problem called overfitting. To reduce the number of the parameters, we set the stride of the pooling layer to be 2, which can reduce the size of each feature map, and set the number of convolutional kernels of the last convolutional layer to be 16. Most of the existing steganalyzers, such as YedroudjNet and ZhuNet, have more convolutional kernels in the last convolutional layer. At the same time, we used the dropout proposed by Hinton et al.\textsuperscript{18} to solve the same problem. The dropout method can be described as follows: when the network propagates forward, the activation value of a neuron stop working with a certain probability. In our proposed network, we use two 128-D feature fully connected layers and added a dropout after each fully connected layer. The parameter of each dropout was set to be 0.5.

Furthermore, the problem of overfitting also can arise due to insufficient data, and several exciting ways can be used to improve the generalization ability of the network. In addition to the method of adding dropout layer, we mainly mention two methods. One is to use regularization, that is, to add a penalty term to the loss function. The second is to use validation set to judge whether overfitting has occurred by comparing training loss and validation loss. Considering the computational efficiency, we only used the validation set to avoid overfitting.

In order to output the result of classification, we apply the two-way softmax at the end of IAS-CNN.

**Experiments**

**Data set**

The data set used in this article is BOSSBase v1.0\textsuperscript{19} containing 10,000 $512 \times 512$ gray-level cover images. And, we scaled the images to the size of $256 \times 256$ pixels in all experiments. In addition, we generated steganographic images of different algorithms and also generated embedding probability maps of cover images and steganographic images according to different steganography algorithms.

**Parameters**

During each experiment, we divided the data set into training, validation, and test three sets. Training set contained 8000 cover images and 8000 steganographic images. Validation set contained 1000 cover images and 1000 steganographic images. Test set contained the remaining 1000 cover images and 1000 steganographic images. The images used in the training set, validation set, and test set did not coincide with each other. Each image has its corresponding embedding probability map.

Custom initial value of the convolutional kernel was used in the pre-processing layer. Xavier\textsuperscript{20} initializer was used to initialize the convolutional kernels of the following five convolutional layers. And the initial biases from the second and sixth layers were set to be zero. IAS-CNN includes two dropout layers, and the parameter was set to be 0.5 in both layers. When the payload is 0.2, 0.4, and 1.0, the learning rate of the proposed network was set to be 0.01, 0.03, and 0.04, respectively, using the ADADELTA\textsuperscript{21} gradient descent algorithm. The parameters based on ADADELTA were described as follows: the decay rate was set to be

| Accuracy | Payload |
|----------|---------|
|          | 0.2     | 0.4     | 1.0     |
| No use of selection channel | 0.5720  | 0.7540  | 0.8790  |
| Use selection channel        | 0.6500  | 0.7895  | 0.9335  |
0.95; the fuzz factor epsilon was set to be $1 \times 10^{-6}$; the mini-batch size is 100, which contained 50 cover images and 50 steganographic images. Based on the above settings, the proposed network was trained to minimize the cross-entropy loss.

**Results and analysis**

Table 6 shows the detection accuracy of IAS-CNN. Three steganography algorithms of the spatial domain, such as HUGO, WOW, and S-UNIWARD were used to evaluate the performance of the network. And, three payloads of 0.2, 0.4, and 1.0 were applied to each algorithm. In the steganalysis of payload 0.2 and payload 0.4, we used parameters trained by relatively higher embedding rate data sets to initialize the network and then used the corresponding data sets to adjust the network which is the difference from the experiments of sections “Pre-processing layer filter” and “Selection channel.”

It can be indicated from Table 6 that the accuracy of IAS-CNN increased with the increase in payload. Comparing the experimental results of Tables 5 and 6, it can be concluded that the method of initializing the network with parameters obtained from relatively higher payload training has better performance in low payload steganalysis. The embedding probability map is used to incorporate the selection channel into the network. The calculation of the embedding probability map is related to the payload, which is usually unknown. In order to evaluate the detection ability of IAS-CNN using the mismatched selection channel and to assess the generalization ability of the network, we chose WOW as a steganography algorithm and then train the network with embedding probability maps with payloads of 0.2, 0.4, and 0.6, respectively, and then carry out steganalysis with payloads of 0.2, 0.4, and 1.0. During training, the selection channel and steganography payload are matched. Experimental results are shown in Table 7.

We use the probability map payload to represent the payload corresponding to the embedding probability map and use the steganalysis payload to represent the payload of the test steganographic image. There are two cases of probability maps mismatch: the first is that the probability map payload is lower than the steganalysis payload and the second is the opposite. In Table 7, it can be observed that when performing the steganalysis with the payload of 0.4, the detection accuracy of IAS-CNN trained with embedding probability maps of 0.2 payload is 73.7%, while that of IAS-CNN trained with embedding probability maps of 0.6 payload is 67.2%. The detection performance of the network in the first case is better than that in the second case. At the same time, Table 7 shows that in the steganalysis with the payload of 1.0, as the probability map payload increases, the detection accuracy of the network gradually improves. As mentioned above, in the first case of probability maps mismatch, the closer the two payloads are, the more beneficial the channel selection is to enhance residuals in regions with high embedding probability. In the second case, the detection performance of the network is weakened, because the selection channel enhances the residuals of regions where the secret information is not embedded.

In order to further evaluate the performance of the proposed network, we compared it with the existing steganalyzers. Figure 5 shows the comparison of IAS-CNN with SRM and GNCNN (on the payload of 0.4).

**Table 6.** The performance of IAS-CNN.

| Algorithm | Payload 0.2 | Payload 0.4 | Payload 1.0 |
|-----------|-------------|-------------|-------------|
| HUGO      | 0.7045      | 0.6815      | 0.6240      |
| WOW       | 0.7835      | 0.8075      | 0.7505      |
| S-UNIWARD | 0.9265      | 0.9335      | 0.9325      |

**Table 7.** The performance of IAS-CNN when a certain payload selection channel is used for training and other steganography payloads are used for steganalysis.

| Accuracy | Testing |
|----------|---------|
|          | 0.2     | 0.4     | 1.0     |
| Training | 0.2     | 0.6815  | 0.7370  | 0.7885  |
|          | 0.4     | 0.5735  | 0.8075  | 0.8545  |
|          | 0.6     | 0.5155  | 0.6720  | 0.8915  |

**Figure 5.** The comparison of IAS-CNN with SRM and GNCNN (on the payload of 0.4).
GNCNN based on deep learning. In Figure 5, it can be observed that the detection accuracy of IAS-CNN is higher than that of SRM and GNCNN. In Table 8, the comparison of IAS-CNN with other steganalysis models based on deep learning shows that IAS-CNN outperforms NSC-YeNet, YeNet using selection channel, and YeNet without selection channel. In Table 9, the comparison with maxSRMd2 and SC-DA-YeNet on the payload of 0.4 indicates that IAS-CNN performs similarly to maxSRMd2 but uses less data for training.

### Conclusion

Hand-crafted high-pass filters are used in SRM, a mature steganalysis model, to extract a variety of image residuals. However, with the increasing complexity of steganography, manual extraction of features becomes more complex and difficult. As a result, deep learning has been applied to steganalysis, especially CNNs. In CNN, self-learning image features replace manual design features. At the same time, to improve the efficiency of CNN, an HPF layer is generally used to generate residual maps before feature extraction. However, hand-crafted filter used in HPF layer is fixed, that is to say, it will not change during the training process. As a result, we select a manual design filter to initialize the pre-processing layer and add it to the learning of the network. In addition, we integrate the knowledge of selection channel into the image pre-processing to enhance crucial residuals and initialize the network with parameters trained with a high payload rate data set to improve the performance of the network. Furthermore, IAS-CNN has fewer residual extractions and convolutional computations. When computational capability and storage space are limited, IAS-CNN is more suitable for steganalysis. In our proposed network, a type of filter is used in the pre-processing layer. In the future, we will consider increasing the diversity of feature extraction, for example, using multiple filters to obtain residuals and then generating minimum residual map and maximum residual map for further feature extraction.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.
Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China “Theory and Key Technologies of Distributed Autonomous Security Supervision for Data Opening and Sharing” (61962009), Major Scientific and Technological Special Project of Guizhou Province (20183001), and Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ014, 2018BDKFJJ019, and 2018BDKFJJ022).

ORCID iD
Zhujun Jin https://orcid.org/0000-0003-0430-942X

References
1. Holub V and Fridrich J. Designing steganographic distortion using directional filters. In: Proceedings of the IEEE international workshop on information forensics and security (WIFS), Tenerife, 2–5 December 2012, pp.234–239. New York: IEEE.
2. Fridrich J, Goljan M and Du R. Detecting LSB steganography in color, and gray-scale images. IEEE Multimedia Mag 2001; 8(4): 22–28.
3. Pevný T, Filler T and Bas P. Using high-dimensional image models to perform highly undetectable steganography. In: Proceedings of the international workshop on information hiding, Calgary, AB, Canada, 28–30 June 2010, pp.161–177. New York: Springer.
4. Holub V, Fridrich J and Denemark T. Universal distortion function for steganography in an arbitrary domain. EURASIP J Inform Secur 2014; 2014(1): 113.
5. Fridrich J and Kodovský J. Rich models for steganalysis of digital images. IEEE T Inform Forensic Secur 2012; 7(3): 868–882.
6. Holub V and Fridrich J. Random projections of residuals for digital image steganalysis. IEEE T Inform Forensic Secur 2013; 8(12): 1996–2006.
7. Denemark T, Sedighi V, Holub V, et al. Selection-channel-aware rich model for steganalysis of digital images. In: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Atlanta, GA, 3–5 December 2014, pp.48–53. New York: IEEE.
8. Qian Y, Dong J, Wang W, et al. Deep learning for steganalysis via convolutional neural networks. Proc SPIE 2015; 94: 94090J.
9. Xu G, Wu H-Z and Shi Y-Q. Structural design of convolutional neural networks for steganalysis. IEEE Signal Pr Lett 2016; 23(5): 708–712.
10. Karlík B and Vehbi A. Performance analysis of various activation functions in generalized MLP architectures of neural networks. Int J Artif Intell Expert Syst 2015; 1(4): 111–122.
11. Ioffe S and Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on international conference on machine learning, vol. 37, Lille, 7–9 July 2015, pp.448–456. New York: ACM.
12. Ye J, Ni J and Yi Y. Deep learning hierarchical representations for image steganalysis. IEEE T Inform Forensic Secur 2017; 12(11): 2545–2557.
13. Denemark T, Fridrich J and Comesana-Alfaro P. Improving selection-channel-aware steganalysis features. Electron Imag 2016; 2016(8): 1–8.
14. Yedroudj M, Comby F and Chaumont M. Yedroudj-net: an efficient CNN for spatial steganalysis. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, ICASSP'2018, Calgary, AB, Canada, 15–20 April 2018, http://arxiv.org/abs/1803.00407
15. Zhang R, Zhu F, Liu J, et al. Efficient feature learning and multi-size image steganalysis based on CNN, 2018, http://arxiv.org/abs/1807.11428
16. Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, Hi, 21–26 July 2017, pp.1800–1807. New York: IEEE.
17. He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 2015; 37(9): 1904–1916.
18. Hinton G, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors. Comput Sci 2012; 3(4): 212–223.
19. Bas P, Filler T and Pevný T. Break our steganographic system: the ins and outs of organizing BOSS. In: Proceedings of the international workshop on information hiding, Prague, 18–20 May 2011, pp.59–70. New York: Springer.
20. Gilorot X and Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th international conference on artificial intelligence and statistics (AISTATS), Sardinia, 2010, pp.249–256, http://proceedings.mlr.press/v9/gilorot10a/gilorot10a.pdf
21. Zeiler MD. ADADELTA: an adaptive learning rate method, 2012, https://arxiv.org/abs/1212.5701