Multimed Tools Appl
DOI 10.1007/s11042-017-4440-4

Deep residual learning for image steganalysis

Songtao Wu[1] · **Shenghua Zhong**[1] · **Yan Liu**[2]

Received: 13 December 2016 / Revised: 13 January 2017 / Accepted: 23 January 2017

Abstract Image steganalysis is to discriminate innocent images and those suspected images with hidden messages. This task is very challenging for modern adaptive steganography, since modifications due to message hiding are extremely small. Recent studies show that Convolutional Neural Networks (CNN) have demonstrated superior performances than traditional steganalytic methods. Following this idea, we propose a novel CNN model for image steganalysis based on residual learning. The proposed Deep Residual learning based Network (DRN) shows two attractive properties than existing CNN based methods. First, the model usually contains a large number of network layers, which proves to be effective to capture the complex statistics of digital images. Second, the residual learning in DRN preserves the stego signal coming from secret messages, which is extremely beneficial for the discrimination of cover images and stego images. Comprehensive experiments on standard dataset show that the DRN model can detect the state of arts steganographic algorithms at a high accuracy. It also outperforms the classical rich model method and several recently proposed CNN based methods.

Keywords Image steganalysis · Convolutional neural networks · Residual learning

✉ Shenghua Zhong
csshzhong@szu.edu.cn

Songtao Wu
csstwu@szu.edu.cn; csstwu@comp.polyu.edu.hk

Yan Liu
csyliu@comp.polyu.edu.hk

[1] College of Computer Science and Software Engineering, Shenzhen University, Guangdong Sheng, China

[2] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Published online: 15 February 2017

1 Introduction

The development of network technology provides users a great convenience for data communication. A key problem of data communication on the Internet is to transmit data from a sender to its receiver safely, without being eavesdropped, illegally accessed or tampered. Steganography, which is the art or science that hides secret message in an appropriate multimedia carrier including text, image, audio, or video [3], provides an effective solution. In contrast to steganography, steganalysis is to reveal the presence of secret messages embedded in digital medias [33]. These two techniques are widely used in many important fields, such as the commercial communication and the military communication [7, 25].

Image steganography and image steganalysis have attracted great interests in recent years [15, 16, 34]. Early studies on image steganography were to hide secret messages in image regions that are insensitive to human's visual system, indicating that salient regions in digital images [27, 28] are avoided for message hiding. Recent researches have extended image steganography and steganalysis into a more general case, which is illustrated in Fig. 1. For image steganography, the sender hides the message $\mathbf{m}$ in the cover image X. By applying the message embedding algorithm $Emb(X, \mathbf{m}, k)$ and the key k on X, the stego image Y is generated and then passed to the receiver. By applying the message extraction algorithm $Ext(Y, k)$ and key k on Y, the receiver can recover the secret message $\mathbf{m}$. During the communication, the sender and the receiver should pledge that any intended observer in the channel cannot differentiate Y from X. For image steganalysis, however, it represents some observers in the communication channel that attempt to discriminate the stego image Y against the cover image X.

Most of methods formulate image steganalysis as a binary classification problem. This formulation is also called universal steganalysis [18], attracting increasing attentions in recent years. In the training phase, universal methods first extract handcrafted features from input images. Then, a binary classifier such as support vector machine [21] or ensemble classifier [15], is trained based on extracted features to discriminate cover images and stego images. In the testing stage, this trained classifier is used to determine whether a new input image is a cover or a stego. For universal methods, designing features that are sensitive

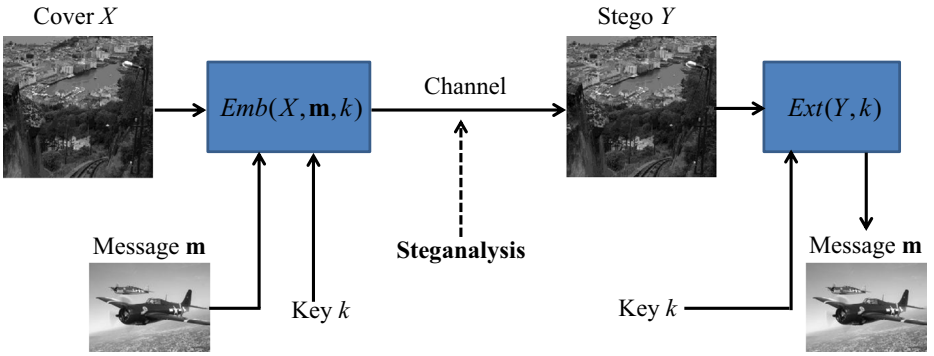


Fig. 1 Schematic illustration of steganography and steganalysis

to message embedding is the key. Subtractive Pixel Adjacent Matrix (SPAM) [24] extracts second order Markov features of adjacent pixels to reliably detect the Least Significant Bit Matching steganography (LSBM). Spatial Rich Model (SRM) [8] combines many diverse co-occurrence matrices to form a large feature vector for message detection. Projection Spatial Rich Model (PSRM) [12] projects noise components into many predefined directions to capture various histogram features.

However, designing effective handcrafted features is difficult. This is a challenging task which needs strong domain knowledge of steganography and steganalysis. In addition, handcrafted features are often high-dimensional in order to capture various statistical properties of input images, making the feature extraction and model training computationally intensive. In order to address these difficulties, many interesting works have proposed to use convolutional neural network for image steganalysis. Compared with handcrafted features, CNN can automatically learn effective features to classify cover images and stego images. Tan and Li [32] presented a stacked convolutional auto-encoder to detect the presence of secret message. Qian et al. in [26] proposed a CNN model by using the Gaussian activation function and average pooling. Xu et al. [36] designed a novel CNN architecture to detect adaptive steganography by incorporating the domain knowledge of steganalysis. Xu et al. [37] also proposed to use an ensemble of CNN models to improve the detection accuracy. Pibre et al. [23] proposed a shallow but wide model. Couchot et al. [5] presented a CNN model with very large convolutional kernels.

Deep neural network models are able to approximate highly complex functions more efficiently than the shallow ones [2]. This ability indicates that very deep neural network can capture complex statistical properties of natural images, which may be beneficial for image classification. Recent works also verify that deep neural networks achieve much better performances than previous methods in many applications [4, 20, 22, 35, 38–40]. Even though great success has been achieved for very deep neural networks in image recognition, existing networks for image steganalysis are still shallow ones.

Recently, He et al. [10] has proposed a very deep CNN model – the deep residual network for image classification. The network has successfully overcome the performance degradation problem when a neural network's depth is large. Because of its great success in image recognition, this paper aims to propose a novel CNN model based on residual learning for image steganalysis. Two appealing characteristics of the proposed DRN make it suitable for image steganalysis. First, the depth of DRN is large, providing the network with powerful ability to capture useful statistical properties of input covers and stegos. Second, instead of learning an underlying function directly, DRN explicitly approximates a residual mapping, which forces the network to preserve the weak signal generated by message embedding. We present comprehensive experiments on the standard BOSSbase [1] dataset and five state of the art steganographic algorithms. Experimental results show that, DRN is not only better than the classical rich model method, but also outperforms several recently proposed CNN models.

The rest of this paper is organized as follows. In Section 2, we introduce basic knowledge about CNN and review modifications to CNN models for image steganalysis. In Section 3, we briefly describe residual learning and explain its rationality for steganalysis. In Section 4, we introduce the proposed model in details. In Section 5, we compare the proposed network with the rich model steganalysis and other CNN based methods on state of the art steganographic algorithms. The paper is finally concluded in Section 6.

2 Convolutional neural networks for steganalysis

2.1 Brief introduction to convolutional neural network

CNN has achieved a great success for many image related tasks [10, 30, 31], indicating its superior capacity to capture the structure of natural images. In general, a typical CNN contains four basic building layers:

- **Convolution layer.** This layer is to use one or several filters (the size is usually set to 3×3, 5×5 or 7×7) to convolve the input images, generating different feature maps for subsequent processing:

$$F_j^{l+1} = \sum_i W_{ij}^l F_i^l + b_j^l \tag{1}$$

  where F_j^l denotes the j-th feature map of the l-th layer, W_{ij}^l represents the convolutional kernel and b_j^l is the bias. The filters W_{ij}^l and bias b_j^l at each layer of a CNN model are not fixed but can be automatically learned by the back-propagation algorithm. Thus, well learned filters can extract different structures in natural images for accurate modeling.
- **Nonlinear mapping layer/activation layer.** This layer is to transform the input feature map through nonlinear mapping:

$$F_j^{l+1} = f\left(F_j^l\right) \tag{2}$$

  $f(\cdot)$ is a point-wise nonlinear function, such as sigmoid, tanh, ReLU [9], leaky ReLU [11], etc. The nonlinear mapping layer is important for CNN, since a neural network with any number of layers is equal to the network with just one layer if there is no nonlinear mapping. In addition, nonlinear mapping makes the CNN extract more complex correlations in natural images.
- **Pooling layer.** This layer is to reduce dimensionality of input feature maps, making the extracted features compact:

$$F_j^{l+1} = pool\left(F_j^l\right) \tag{3}$$

  where $pool(\cdot)$ denotes the pooling function. Generally, there are two kinds of pooling function in existing CNN models: maximum pooling and average pooling. Maximum pooling is to select the maximum value in a local region as the output, while average pooling is to calculate the average value of a local region as the output. The main role of pooling is to aggregate input feature maps into a compact representation. In addition, large distance correlations in natural images can be captured by pooling the feature map into a small size.
- **Batch normalization layer.** This layer is to normalize each data item x_i in a training batch $\mathcal{B}$ into y_i:

$$y_i = \gamma \hat{x}_i + \beta \tag{4}$$

  where γ and β are parameters of batch normalization. $\hat{x}_i$ denotes:

$$\hat{x}_i = \frac{x_i - E_{\mathcal{B}}(x_i)}{\sqrt{Var_{\mathcal{B}}(x_i)}} \tag{5}$$

  In (5), $E_{\mathcal{B}}(x_i)$ and $Var_{\mathcal{B}}(x_i)$ represent the mean and the variance of x_i in terms of the batch $\mathcal{B}$. The main function of batch normalization is to enforce the data far away from

the saturation regions. Due to this advantage, a neural network with batch normalization is relatively insensitive to the parameter initialization and converges in a fast speed than a network without batch normalization.

Structural advantages of CNNs make them suitable to capture complex statistics of natural images. This is the main reason that CNN models achieve superior performances on many image related tasks. In the following part, we review various modifications to CNNs for image steganalysis.

2.2 Convolutional neural network for image steganalysis

Although CNN models have achieved a great success in image classification, directly applying them for image steganalysis may not work. Steganalysis is different from image classification that it aims to discriminate covers and covers added with weak signals (stegos). In order to adapt to this feature, existing methods have made a series of modifications to the basic operations in CNN models:

- **Modifications to Convolution.** Most CNN based steganalytic methods [26, 32, 36, 37] use small size convolutional filters to convolve input images. The advantage of using small size convolutional filters is that they can capture various local correlations among image pixels and thus extract effective features for image steganalysis. Instead of using convolutional kernels with small size, several CNN based steganalysis prefer to use large convolutional kernels to detect steganography. This design can be found at in Pibre's network [23] and Couchot's network [5]. For steganography with a fixed modification pattern, a large kernel can sum enough weak stego signals into a more strong stego signal for accurate detection. However, most of modern steganographic algorithms embed secret messages adaptive to the content of cover images, which would make this method disable.
- **Modifications to Activation Functions.** The activation function is important for CNN based image steganalysis. The design or selection of activation function should enable the CNN to extract discriminative feature for the classification of covers and stegos. Qian et al. [26] in their network demonstrated a Gaussian activation function is better than a classical sigmoidal function. In order to improve the statistical modeling to the noise components of input images, Xu et al. [36] inserted an absolute layer behind the first convolutional layer. Unlike a ReLU activation function that simply discards the signal smaller than zero, the absolute layer in this network can preserve the discriminative information in the negative region.
- **Modifications to Pooling.** For a traditional CNN model, pooling aims to reduce the dimensionality of feature maps and obtain a compact representation to input data. Most CNN based steganalysis [26, 32, 36, 37] use pooling to extract compact features for classifying cover images and stego images. However, Pibre [23] and Couchot [5] believed that pooling is a information losing process and the pooling is discarded in their networks.
- **Modifications to Batch Normalization.** There are few works that use batch normalization for image steganalysis. Xu et al. [36] firstly combined the batch normalization and tanh activation function in their network. The batch normalization makes the data fall into regions far from saturation regions, making their network be learned effectively.

Although many modifications have been made to adapt CNN models for image steganalysis, there still exist no effective learning mechanism to preserve the weak setgo signal

in existing networks. In the following section, we introduce a new learning call residual learning to address this problem and explain its rationality for image steganalysis.

3 Residual learning

3.1 Basic idea of residual learning

In [10], He et al. proposed a novel CNN model called deep residual network for image classification. The main difference between a residual network and a typical CNN is that they have different network architectures, which are shown as Fig. 2. For a typical CNN model, it organizes the architecture by combining basic units such as convolution, nonlinear mapping, pooling or batch normalization in a cascade manner. But for a residual network, it has a shortcut pathway directly connecting the input and the output in a building block. Mathematically, instead of approximating an underlying function $H(\mathbf{x})$ directly, residual learning turns to fit its residual mapping $F(\mathbf{x})$, where:

$$F(\mathbf{x}) := H(\mathbf{x}) - \mathbf{x} \tag{6}$$

The final mapping of a residual learning block is $F(\mathbf{x}) + \mathbf{x}$, which is equal to the output of a typical CNN, that is $H(\mathbf{x})$. However, as indicated by He et al. in [10], it is easier to fit a residual mapping $F(\mathbf{x})$ than the original mapping $H(\mathbf{x})$, especially when $H(\mathbf{x})$ is an identity or a near identity mapping. This property enables that the depth of residual network can be increased to be very large, without degrading the network's classification accuracy.

3.2 Rationality of residual learning for steganalysis

Actually, to detect the presence of secret message, steganalysis should correctly classify an input image $\mathbf{y}$ as:

$$\mathbf{x} = \begin{cases} \mathbf{c} + \mathbf{0}, & \text{cover} \\ \mathbf{c} + \mathbf{m}, & \text{stego} \end{cases} \tag{7}$$

where $\mathbf{c}$ represents the innocent cover image, $\mathbf{0}$ is zero signal, and $\mathbf{m}$ denotes the weak stego signal generated by message embedding. By feeding $\mathbf{x}$ into a residual learning block, the identity mapping of the network puts forward $\mathbf{c}$ to the output of the block, while the residual mapping $F(\mathbf{x})$ fits $\mathbf{0}$ or $\mathbf{m}$. Since both $\mathbf{0}$ and $\mathbf{c}$ are small signals, they can be effectively modeled by the residual learning network $F(\mathbf{x})$. Consequently, $\mathbf{m}$ is effectively captured by the residual mapping network. Therefore, the weak stego signal is expected to be preserved and

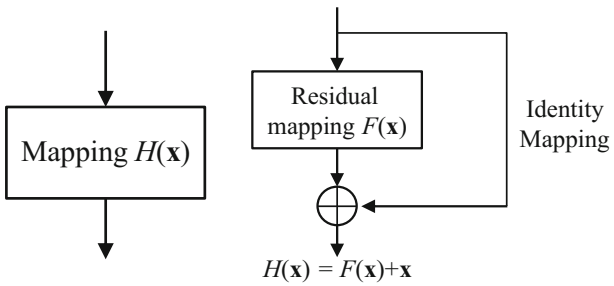


Fig. 2 Basic building blocks in different CNN models. *Left*: a basic building block in a typical CNN model. *Right*: a basic building block in a residual network

emphasized through the whole network. This advantage makes residual learning extremely suitable for image steganalysis.

4 Deep residual network for steganalysis

In this section, we introduce the proposed DRN model for image steganalysis. Firstly, we present the overall architecture of DRN in details. Then, we describe the parameter learning to the DRN model.

4.1 Network architecture

Figure 3 illustrates the architecture of DRN in this paper. The network contains three sub-networks, i.e. the high-pass filtering (HPF) sub-network, the deep residual learning sub-network and the classification sub-network. These sub-networks have their own roles in processing the information in the overall model, which are described as follows.

The HPF sub-network is to extract the noise components from input cover/stego images. Previous studies indicate that preprocessing input images with HPF can largely suppress their contents, leading to a narrow dynamic range and a large signal-to-noise ratio (SNR) between the weak stego signal and the image signal. As a result, statistical descriptions to the filtered image become more compact and robust [8]. For this reason, we do not directly feed original images into the network but input their noise components. Mathematically, the noise component of an image $\mathbf{n}$ is the convolution between the image $\mathbf{I}$ and a HPF kernel $\mathbf{k}$:

$$\mathbf{n} = \mathbf{I} * \mathbf{k} \tag{8}$$

where $*$ denotes convolution operator. We follow the general setting and choose the $\mathbf{k}$ as the KV kernel [26, 36]:

$$KV = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \tag{9}$$

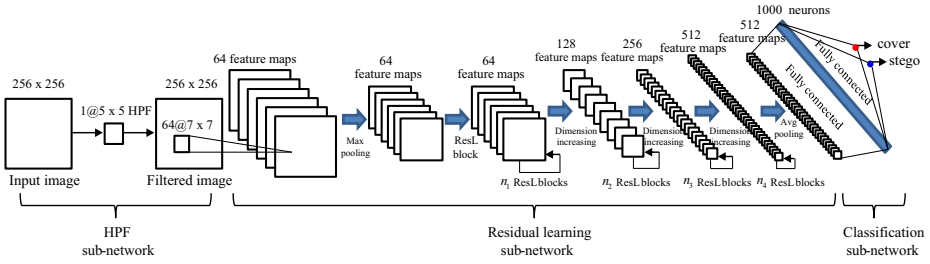


Fig. 3 DRN for steganalysis. In the HPF sub-network, a 5×5 KV kernel preprocesses input cover/stego images to get their noise components. In the residual learning sub-network, there are two kinds of building blocks: the residual learning block (ResL) and the dimension increasing block. n_1, n_2, n_3, or n_4 denotes that there are n_1, n_2, n_3, or n_4 ResL blocks following the current layer. The classification sub-network finally maps features into labels. In this figure, $p@q \times q$ denotes that there are p filters with the size of $q \times q$. The ReLU activation layer, the maximum pooling layer, and the batch normalization layer are not shown in the figure

The residual learning sub-network is to extract effective features for discriminating cover images and stego images. The sub-network firstly use 64 convolutional filters (the size is 7×7) to convolve input images, generating many feature maps for subsequent processing. Following the convolutional layer, there are a ReLU activation layer, a maximum pooling layer and a batch normalization layer. This processing is to capture many different types of dependencies among pixles in the noise component images. Its purpose is to make the network extract enough statistical properties to detect the secret message accurately. For the residual learning layer, it is constituted by two kinds of building blocks: the non-bottleneck block and the bottleneck block, which are shown as Fig. 4. For a non-bottleneck block, it has two convolutional layers with the size of 3×3. Each convolutional layer is followed by a ReLU activation layer, a maximum pooling layer and a batch normalization layer. For a bottleneck block, the number of convolutional layer is three. Furthermore, two sizes of convolutional filters are used in the block: 1×1 and 3×3. In practice, a bottleneck block is more economical for building CNN models with large depths. For ordinary residual learning, both the input and the output of two building blocks have the same sizes. For dimension increasing, the output has double size of feature maps than the input. To force each block having the same complexity, the feature map is down-sampled by factor 2 for the dimension increasing block. In our DRN model, there are four stages of processing, which increases the number of feature maps from 64 to 512.

The final classification sub-network consists of fully connected neural network model, mapping features extracted from the residual learning sub-network into binary labels. To ensure the modeling ability of this sub-network, we set the number of neurons to 1000.

4.2 Network training

Parameters of the residual learning sub-network and the classification sub-network are learned by minimizing the softmax function:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\delta(y_i = k)\cdot log\left(\frac{e^{o_{ik}(\mathbf{x}_i,\theta)}}{\sum_k^K e^{o_{ik}(\mathbf{x}_i,\theta)}}\right) \tag{10}$$

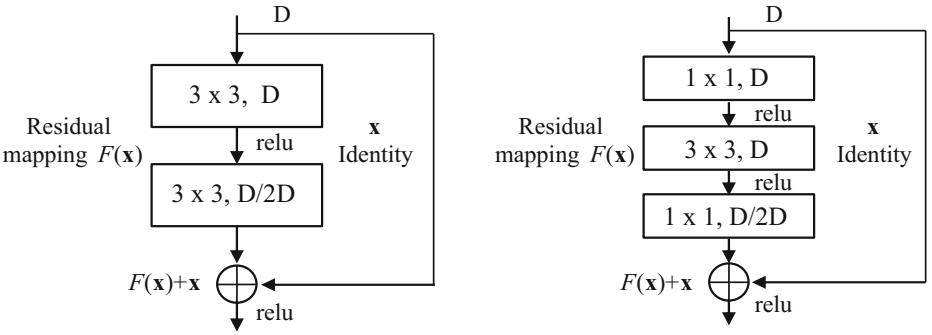


Fig. 4 The non-bottleneck building block and the bottleneck building block in a residual network. The output can have same dimension or double dimension of the input, corresponding the residual learning and the dimension increasing respectively

where y_i denotes the label of the sample $\mathbf{x}_i$, $\delta(\cdot)$ represents the delta function, N is the number of training samples, K is the number of labels ($K = 2$). $o_{ik}(\mathbf{x}_i, \theta)$ denotes the output for the i-th sample $\mathbf{x}_i$ at the k-th label. θ is the parameter of the network. For a neural network model, θ generally represents the weight matrix $\mathbf{W}$ or the bias vector $\mathbf{b}$. The weight matrix and bias vector for each layer is updated by the gradient descent:

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \alpha \frac{\partial L}{\partial \mathbf{W}} \tag{11}$$

$$\mathbf{b}(t+1) = \mathbf{b}(t) - \alpha \frac{\partial L}{\partial \mathbf{b}} \tag{12}$$

where α is the learning rate. Be aware that, in (11) and (12), all training samples are involved in the computation of total loss L. In order to reduce the computation, we follow the learning method in [10] and use the mini-batch Stochastic Gradient Descend (SGD) to optimize the network.

The whole process to train the proposed DRN model is illustrated as the following algorithm:

Algorithm 1 Learning the proposed DRN model

Input : cover images $\mathbf{c}$, steganographic algorithm $\mathcal{A}$, the KV kernel $\mathbf{k}$, the untrained DRN model $\mathcal{D}$, parameter learning rate α.

1 Use steganographic algorithm $\mathcal{A}$ to generate stego images $\mathbf{s}$

$$\mathbf{s} = \mathcal{A}(\mathbf{c})$$

2 Preprocess cover images $\mathbf{c}$ and stego images $\mathbf{s}$ with the KV kernel $\mathbf{k}$ in the HPF subnetwork:

$$\mathbf{n} = \mathbf{x} * \mathbf{k}, \ \mathbf{x} \in \{\mathbf{c}, \mathbf{s}\}$$

3 **while** $\mathcal{D}$ *is not converged* **do**

4 Calculate the softmax loss L with Eq.(10);

5 Update the parameter of $\mathcal{D}$ by gradient descend:

$$\theta(t+1) = \theta(t) - \alpha \frac{\partial L(\mathbf{x})}{\partial \theta}, \ \theta \in \{\mathbf{W}, \mathbf{b}\}$$

6 Check whether the network $\mathcal{D}$ is converged.

7 **end**

Output: a well trained DRN model $\mathcal{D}$.

5 Experiments

In this section, we conduct several experiments to demonstrate effectiveness of proposed DRN for image steganalysis. We first give experimental settings to the DRN model, including the evaluation dataset and the parameter setting of learning the model. Then, we use an experiment to determine the best network architecture for image steganalysis. Based on the best network architecture, we demonstrate the effectiveness of the feature learned by the proposed method. We finally compare the proposed DRN with traditional rich model based methods and the state of the art CNN based methods.

5.1 Experimental settings

The dataset used for performance evaluation is the BOSSbase 1.01 version [1]. The BOSSbase is a standard dataset for evaluating steganalysis and steganography. It contains 10,000 grayscale natural images with the size of 512 × 512. Following general settings in recent CNN based steganalysis [12, 15, 32], we crop the original 10,000 BOSSbase images into 40,000 non-overlapped images with the size 256 × 256. Figure 5 shows several sample images in the cropped BOSSbase dataset.

For the DRN model, we initialize its weight matrices **W** by a zero-mean Gaussian distribution with the fixed standard derivation of 0.01. The bias vector **b** is initialized to be zero. The learning rate α, momentum and weight decay of the model are set to 0.001, 0.9 and 0.0001 respectively. The size of mini-batch for SGD is set to 10. All experiments for the DRN are conducted on Nvidia's Tesla K80 platform.

5.2 Relationship between the detection accuracy and the number of convolutional layers in DRN

This experiment is conducted to investigate how the number of convolutional layers in DRN affects the performance of image steganalysis. We randomly select 30,000 cover images from the cropped BOSSbase, and their stegos which are generated by Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD) steganography [13] at payload 0.4 bit-per-pixel (bpp), are used for training the DRN model. The rest 10,000 covers and stegos are used for testing. The performance is evaluated by the average detection error rate P_E:

$$P_E = \frac{1}{2}(P_{FA} + P_{MD}) \tag{13}$$

where P_{FA} denotes the false alert rate and P_{MD} represents the miss detection rate. For the configuration of DRN, we select DRN models with 10, 20, 30, 40, 50 and 60 convolutional layers for evaluation. These DRN models are configured as Table 1.

Figure 6 reports detection error rates of DRN models with different number of convolutional layers. When the number is smaller than 50, detection error rates decreases as the number increases. The result indicates that deeper DRN model can capture more reliable statistical properties of natural images than the shallow one for accurate steganalysis. However, when the number is 60, the overfiting phenomenon arises and results in the increase of the detection error rate. For this reason, we set the number of convolutional layer to 50 in the following experiment.

Fig. 5 Sample images in the cropped BOSSbase dataset

Table 1 Configurations for DRN models

# conv.	Block Type	$[n_1, n_2, n_3, n_4]$
10	Non-bottleneck block	[0, 0, 0, 0]
20	Non-bottleneck block	[1, 2, 1, 1]
30	Non-bottleneck block	[2, 3, 3, 2]
40	Non-bottleneck block	[2, 5, 5, 3]
50	Bottleneck block	[2, 3, 5, 2]
60	Non-bottleneck block	[2, 8, 8, 7]

$[n_1, n_2, n_3, n_4]$ represents the number of blocks for ordinary residual learning, which is illustrated in Fig. 1

5.3 Effectiveness of the feature learned by DRN

This experiment is to demonstrate the effectiveness of the feature automatically learned by the proposed DRN. Same to the first experiment, we select the S-UNIWARD steganography at 0.4 bpp for evaluation. The last feature map before the output node in DRN model is selected as the automatically learned feature. We choose the classical Spatial Rich Model (SRM) feature [8] for performance comparison. SRM is a classical steganalytic method for detecting modern steganographic algorithm. It consists of many high order co-occurrence matrices to make the model sensitive enough to various operations of data embedding. In order to observe the distribution of cover images and stego images, we use the Linear Discriminant Analysis (LDA) to reduce the dimension of DRN features and SRM feature into 2-dimension.

Figure 7 shows 2D distributions of DRN features and SRM features for cover images and stego images. It is obvious that cover images and stego images of SRM features are completely mixed with each other, while they can be easily separated by the DRN features.

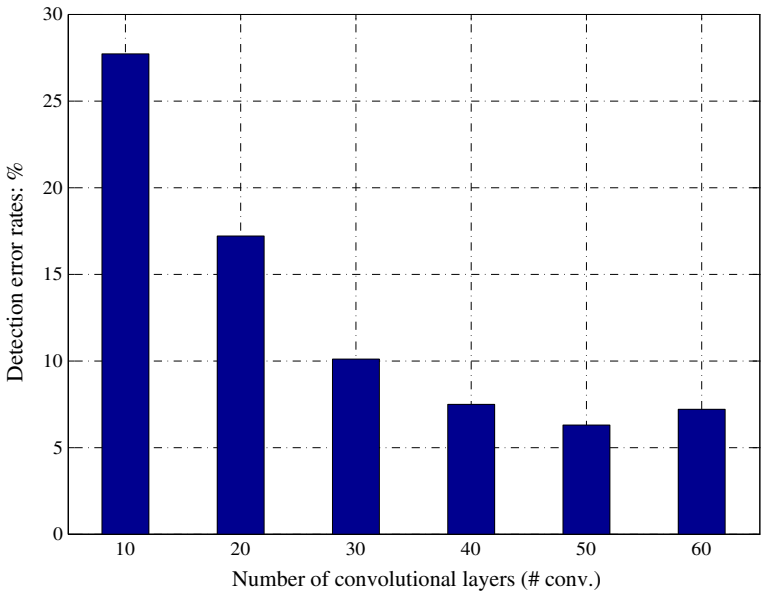


Fig. 6 Detection error rates for DRN with different number of convolutional layers

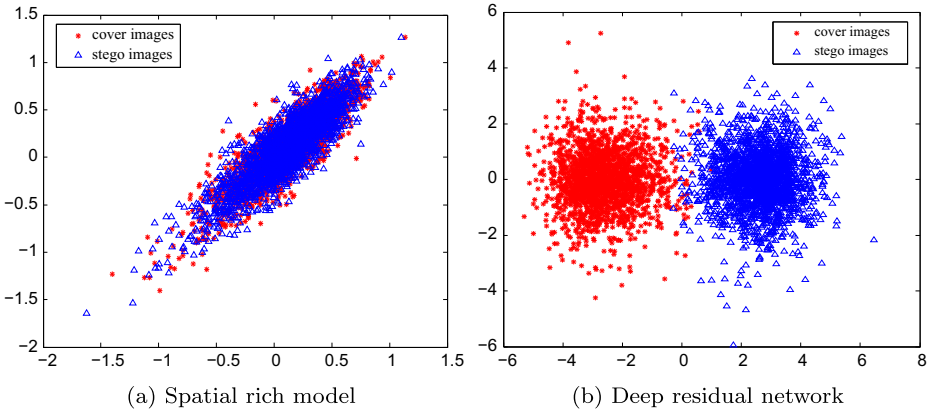


Fig. 7 2D distributions of extracted features for cover images and stego images

This result demonstrates that the proposed DRN can learn more effective features than the classical SRM for image steganalysis.

5.4 Performance comparisons with prior arts

To demonstrate the effectiveness of the DRN for image steganalysis, we compare its performances with the SRM [8] and maxSRMd2 [6]. maxSRMd2 is an improved version of SRM, which is especially designed for adaptive steganography. Unlike SRM that extracts features with an equal weight to each pixel, maxSRMd2 focuses more on the pixels with

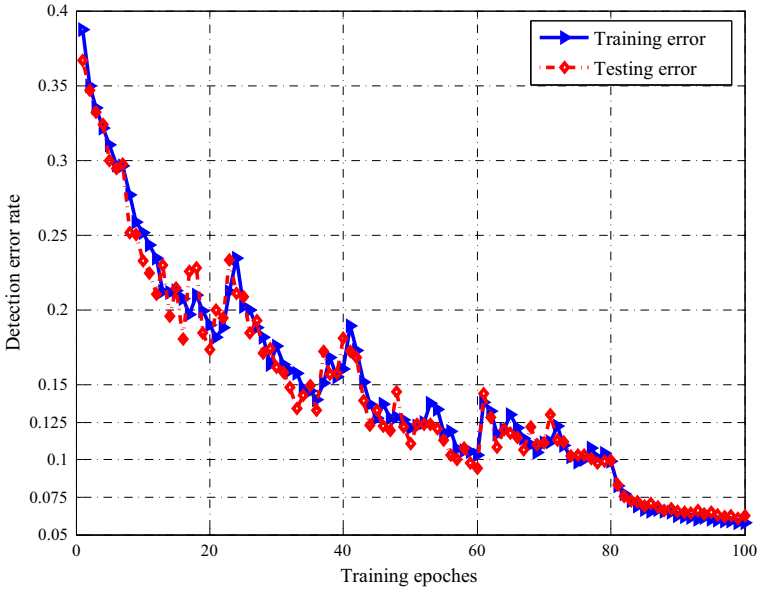


Fig. 8 Training error and detection error of DRN on S-UNIWARD steganography at 0.4 bpp. The performance jump at 50-th epoch is because of the learning rate change

Table 2 Detection error rates for four states of the art steganographic algorithms at payload 0.4 bpp

Steganography	SRM	maxSRMd2	DRN
WOW	20.1 %	15.2 %	4.3 %
S-UNIWARD	20.3 %	18.8 %	6.3 %
HILL	24.2 %	21.6 %	10.4 %
MiPOD	22.1 %	20.4 %	4.9 %

high embedding probability. Both the SRM based steganalysis and the maxSRMd2 based steganalysis use the ensemble learning [15] to train the classifier.

Four states of the art steganographic algorithms, including the Wavelet Obtained Weights steganography (WOW) [14], S-UNIWARD [13], HILL [19] and MiPOD [29], are used for evaluating the effectiveness of steganalytic algorithms. All these algorithms embed secret message adaptive to the content of an input image. They tend to hide messages into pixels of

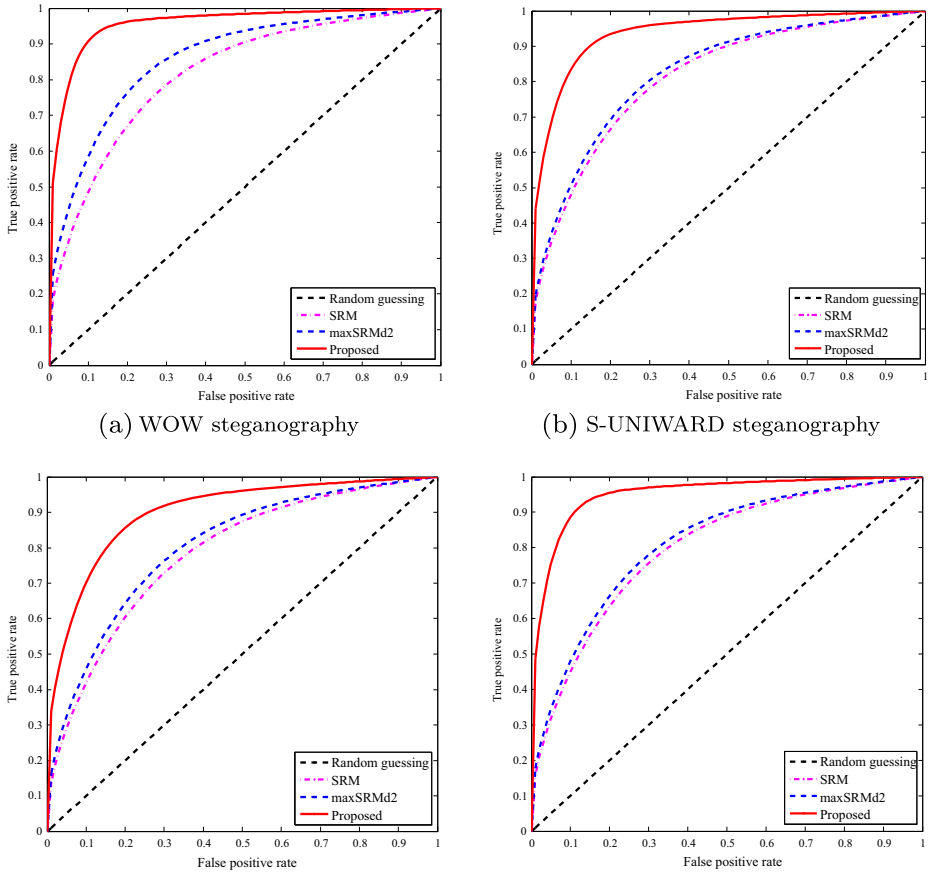


Fig. 9 ROC curves for SRM, maxSRMd2 and the proposed network on four steganographic algorithms

Table 3 Detection error rates for CNN models on three steganographic algorithms at 0.4 bpp. '\' denotes that the result is not reported in the paper

Steganalytic methods	WOW	S-UNIWARD	HILL
Qian's network [26]	29.3 %	30.9 %	\
Xu's network [36]	\	19.7 %	20.7 %
Proposed DRN model	4.3 %	6.3 %	10.4 %

complex regions, i.e. pixels with low embedding distortions. The only difference between these algorithms is that they use different distortion functions for message hiding.

Same to the setting in the first experiment, 30,000 randomly selected cover images and their corresponding stegos are used for training CNN models, the rest 10,000 cover images and their stegos are for testing. The average detection error rate P_E is used as the evaluation criterion.

Figure 8 shows the training error curve and detection error curve of DRN. It can easily find the detection error is very close to the training error, which demonstrates the superior generalization ability of DRN for image steganalysis. Table 2 gives the overall performance comparisons of DRN against to SRM and maxSRMd2. Figure 9 shows Receiver Operating Characteristic (ROC) curves for three methods on four steganographic algorithms. We can find that DRN is better than the rich model based methods across all steganographic algorithms. Meanwhile, among all steganographic algorithms, the detection performances of DRN indicate that HILL is most hard to be detected while WOW is the easiest to be detected. This is consistent with the results of SRM and maxSRMd2. We also compare DRN with three representative CNN models, including Qian's network [26] and Xu's network [36]. Results in Table 3 demonstrate that the DRN also outperforms these CNN models for steganalysis.

6 Conclusion

This paper introduced a novel convolutional neural network model for image steganalysis. The proposed model has two obvious differences with existing works. First, the proposed network has a relatively larger depth than current CNN based models. Second, a novel learning mechanism called residual learning is used to actively preserve the weak stego signal. Experiments on standard dataset have demonstrated that the proposed network has following contributions:

- CNN with large depth shows a superior ability to model natural images. It can extract complex statistical features for classifying cover images and stego images.
- Residual learning proves to be effective to preserve the weak stego signal, make the proposed model capture the difference between cover images and stego images. In addition, features automatically learned by proposed network are more easily classified than classical rich model based features.

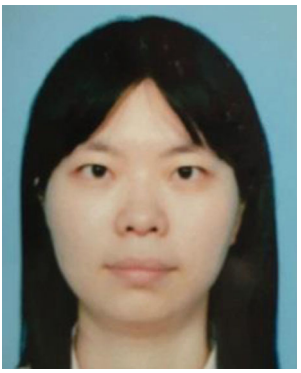
Current work demonstrates that a deep network with residual learning can detect spatial domain steganography effectively. We will extend this work to detect compressed domain steganographic algorithms. Furthermore, like existing CNN models that are computationally expensive, the proposed model also needs enough computational resources to support its efficiency. We will also focus on improving its training efficiency in future.

References

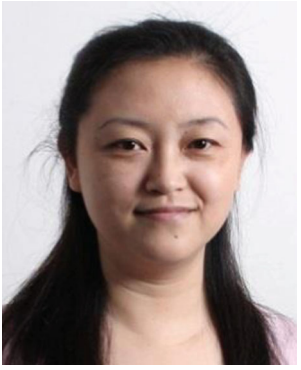
1. Bas P, Filler T, Pevny T (2011) Break our steganographic system: The ins and outs of organizing BOSS. Information Hiding pp 59–70
2. Bianchini M, Scarselli F (2014) On the complexity of neural network classifiers: a comparison between shallow and deep architectures. IEEE Trans Neural Netw Learn Syst 25(8):1553–1565
3. Cheddad A, Condell J, Curran K, Kevitt PM (2010) Digital image steganography: Survey and analysis of current methods. Signal Process 90(3):727–752
4. Chen H, Ni D, Qin J, Li S, Yang X, Wang T, Heng PA (2015) Standard plane localization in fetal ultrasound via domain transferred deep neural networks. IEEE J Biomed Health Inf 19(5):1627–1636
5. Couchot JF, Couturier R, Guyeux C, Salomon M (2016) Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key. arXiv:1605.07946v3
6. Denemark T, Sedighi V, Holub V, Cogranne R, Fridrich J (2014) Selection-channel-aware rich model for steganalysis of digital images. IEEE Workshop on Information Forensic and Security (WIFS)
7. Fridrich J, Goljan M (2002) Practical steganalysis of digital images - state of the art. Proc SPIE Photonics Imaging, Secur Watermarking Multimed Contents 4675:1–13
8. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensic Secur 7(3):868–882
9. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)
10. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385v1
11. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. arXiv:1502.01852v1
12. Holub V, Fridrich J (2013) Random projections of residuals for digital image steganalysis. IEEE Trans Inf Forensic Secur 8(12):1996–2006
13. Holub V, Fridrich J, Denemark T (2014) Universal distortion function for steganography in an arbitrary domain. EURASIP J Inf Secur 1(1):1–13
14. Holub V, Fridrich J (2012) Designing steganographic distortion using directional filters. IEEE Workshop on Information Forensic and Security (WIFS)
15. Kodovsky J, Fridrich J, Holub V (2012) Ensemble classifiers for steganalysis of digital media. IEEE Trans Inf Forensic Secur 7(2):432–444
16. Li B, Huang J, Shi YQ (2009) Steganalysis of YASS. IEEE Trans Inf Forensic Secur 4(3):369–382
17. Li B, Wang M, Li X, Tan S, Huang J (2015) A strategy of clustering modification directions in spatial image steganography. IEEE Trans Inf Forensic Secur 10(9):1905–1917
18. Li B, He J, Huang J, Shi YQ (2011) A survey on image steganography and steganalysis. J Inf Hiding Multimed Signal Process 2(2):142–172
19. Li B, Wang M, Huang J, Li X (2014) A new cost function for spatial image steganography. IEEE International Conference on Image Processing (ICIP) pp 4206–4210
20. Lu H et al (2016) Wound intensity correction and segmentation with convolutional neural networks. Concurrency and Computation: Practice and Experience
21. Lyu S, Farid H (2004) Steganalysis using color wavelet statistics and one-class support vector machines. SPIE Symposium on Electronic Imaging pp 35–45
22. Li Y et al (2016) Underwater image de-scattering and classification by deep neural network. Comput Electr Eng 54:68–77
23. Pibre L, Pasquet J, Ienco D, Chaumont M (2016) Deep learning for steganalysis is better than a rich model with an ensemble classifier and is natively robust to the cover source-mismatch. SPIE Media Watermarking, Security, and Forensics
24. Pevny T, Bas P, Fridrich J (2010) Steganalysis by subtractive pixel adjacency matrix. IEEE Trans Inf Forensic Secur 5(2):215–224
25. Provos N, Honeyman P (2002) Detecting steganographic content on the internet. Proceedings of Network and Distributed System Security Symposium (NDSS)
26. Qian Y, Dong J, Wang W, Tan T (2015) Deep learning for steganalysis via convolutional neural networks. SPIE Media Watermarking, Security, and Forensics, vol 9409
27. Ren T, Liu Y, Ju R, Wu G (2016) How important is location information in saliency detection of natural images. Multimed Tools Appl 75(5):2543–2564
28. Ren T, Qiu Z, Liu Y, Yu T, Bei J (2015) Soft-assigned bag of features for object tracking. Multimed Syst J 21(2):189–205
29. Sedighi V, Cogranne R, Fridrich J (2016) Content-Adaptive steganography by minimizing statistical detectability. IEEE Trans Inf Forensic Secur 1(2):221–234

30. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large scale image recognition. arXiv:1409.1556v6
31. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition
32. Tan S, Li B (2014) Stacked convolutional auto-encoders for steganalysis of digital images. Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA) pp 1–4
33. Wang H, Wang S (2004) Cyber warfare: steganography vs. steganalysis. Commun ACM 47(10):76–82
34. Wu H-T, Huang J, Shi YQ (2015) A reversible data hiding method with contrast enhancement for medical images. J Vis Commun Image Represent 31:146–153
35. Wu J, Zhong SH, Jiang J, Yang Y (2016) A novel clustering method for static video summarization. Multimed Tools Appl 2016:1–17
36. Xu G, Wu H, Shi YQ (2016a) Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett 23(5):708–712
37. Xu G, Wu H, Shi YQ (2016b) Ensemble of CNNs for steganalysis: an empirical study. Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security pp 103–107
38. Xu X, He L, Lu H, Shimada A, Taniguchi R (2016) Non-linear matrix completion for social image tagging. IEEE Access 2016:1–7
39. Zhong SH, Liu Y, Hua K (2016) Field effect deep networks for image recognition with incomplete data. ACM Trans Multimed Comput Commun Appl 12(Article):52
40. Zhong SH, Liu Y, Li B, Long J (2015) Query-oriented unsupervised multi-document summarization via deep learning. Expert Syst Appl 42(21):8146–8155

Songtao Wu received his B.Sc. in Electronic Information Engineering from Lanzhou University in 2009 and M.S. in Circuits and Systems from Peking University 2012. He is now pursuing the Ph.D. degree from Department of Computing, The Hong Kong Polytechnic University. His research interests include steganography and steganalysis, artificial intelligence.



Shenghua Zhong received her B.Sc. in Optical Information Science and Technology from Nanjing University of Posts and Telecommunication in 2005 and M.S. in Signal and Information Processing from Shenzhen University in 2007. She got her Ph.D. from Department Computing, The Hong Kong Polytechnic University in 2013. She worked as a Postdoctoral Research Associate in Department of Psychological & Brain Sciences at The Johns Hopkins University from 2013 to 2014. Currently, she is an Assistant Professor in College of Computer Science & Software Engineering at Shenzhen University in Shenzhen. Her research interests include multimedia content analysis, brain science, and machine learning.



Yan Liu received the BEng degree from the Department of Electronic Engineering at Southeast University and the MSc degree from the School of Business at Nanjing University in China. She received the PhD degree from the Department of Computer Science at Columbia University. Currently, she is an associate professor in the Department of Computing at The Hong Kong Polytechnic University. As a director of Cognitive Computing lab, she focuses her research in brain modeling, ranging from image/video content analysis, music therapy, manifold learning, deep learning, and EEG data analysis.