Postprint: Image Steganalysis Methods Based on Convolutional Neural Networks

Authors: Peixian Gao, Wei Lixian, Liu Jia, Liu Mingming

Date: 2018-05-20T00:00:00+00:00

Abstract

To enhance the classification performance of convolutional neural networks (CNNs) in the field of image steganalysis, a novel convolutional neural network model (steganalysis-convolutional neural networks, S-CNN) is constructed for steganalysis. The model employs two convolutional layers and two fully connected layers, thereby reducing the number of convolutional layers; incorporates batch normalization layers prior to activation functions to mitigate overfitting during training; and eliminates pooling layers to minimize the loss of embedded information, consequently improving classification effectiveness. Experimental results demonstrate that, compared with conventional image steganalysis approaches, the proposed model streamlines the steganalysis procedure while achieving superior steganalysis accuracy.

Full Text

Preamble

Journal Article: http://www.arocmag.com/article/02-2019-01-038.html
ChinaXiv Partner Journal: Computer Application Research

Image Steganalysis Based on Convolutional Neural Networks

Gao Peixiana,b, Wei Lixiana,b, Liu Jiaa,b, Liu Mingminga,b
aKey Laboratory for Network & Information Security of Chinese Armed Police Force; bDepartment of Electronic Technology, Engineering University of Chinese Armed Police Force, Xi' an 710086, China

Abstract: To improve the classification performance of convolutional neural networks (CNN) in image steganalysis, this paper constructs a new CNN model (steganalysis-convolutional neural networks, S-CNN) for steganalysis. The model employs two convolutional layers and two fully connected layers, reducing the number of convolutional layers. By adding batch normalization

layers before activation functions, the model is optimized to avoid overfitting during training. The pooling layers are removed to reduce the loss of embedded information, thereby improving classification performance. Experimental results demonstrate that compared with traditional image steganalysis methods, the proposed model reduces steganalysis steps and achieves higher steganalysis accuracy.

Keywords: image steganalysis; convolutional neural network; batch normalization; activation function

Classification: P309.2
DOI: 10.3969/j.issn.1001-3695.2017.07.0692

0 Introduction

Image steganalysis has become a research hotspot in the field of information security [1,2]. Traditional image steganalysis methods consist of two steps: first, feature extraction, such as wavelet histogram features, Markov features, and co-occurrence matrix features of discrete cosine transform coefficients [3,4]; second, feature classification, such as Fisher algorithms and support vector machines [5,6]. Due to low reliability or time-consuming training processes, these traditional methods adversely affect steganalysis efficiency.

In recent years, with the excellent performance of deep learning in image classification, an increasing number of researchers have applied deep learning to their domains. In 2015, Qian et al. [7] used deep learning to replace traditional two-step machine learning for steganalysis, proposing a five-layer CNN model that employs a high-pass filter (HPF) for convolutional preprocessing, Gaussian nonlinear activation functions, and average pooling. Testing on the BOSSbase dataset achieved an accuracy only 3% to 4% lower than the Spatial Rich Model (SRM) [8] + Ensemble Classifier (EC) [9], representing a significant advancement for image steganalysis. In May 2016, Guanshuo Xu et al. [10] proposed a five-layer CNN model that added an absolute value layer (ABS) after the first convolutional layer to enhance the learning capability of subsequent convolutional layers. To avoid overfitting, they constrained data ranges in early layers and adopted 1×1 convolution kernels in deeper layers, achieving 80.24% detection accuracy against the S-UNIWARD steganographic algorithm in perspective scenarios. These results demonstrate the tremendous potential of CNN in image steganalysis.

To improve the accuracy and reliability of image steganalysis, this paper constructs a two-layer CNN model for image steganalysis. Experimental results show that compared with Guanshuo Xu et al.' s CNN model, the proposed model improves detection accuracy by 8.68%.

1 Five-Layer CNN Framework

The model constructed by Guanshuo Xu et al. consists of five convolutional modules, as shown in [Figure 1: see original paper]. The equations within the flowchart boxes represent: (number of convolution kernels) × (height of convolution kernel × width of convolution kernel × number of input feature maps), while the equations below the boxes represent: (number of feature maps) × (height of image × width of image). The convolutional layers, pooling layers, and HPF layer all use padding. Each convolutional module begins with a convolutional layer for feature extraction and ends with an average pooling layer (global pooling in the fifth module). The first and second modules use TanH activation functions, while the remaining layers use ReLU activation functions. To constrain data ranges, an absolute value layer (ABS) is employed in the first convolutional module. To prevent the CNN training from falling into local optima, batch normalization layers (BN layers) are added after each activation function layer. The output module includes one fully connected layer and one loss function layer. This model achieved 80.24% detection accuracy for S-UNIWARD embedding algorithm on the BOSSBase database.

---

2 S-CNN Framework

Based on Guanshuo Xu et al.'s research, this paper constructs a new CNN model for image steganalysis, as shown in [Figure 2: see original paper]. The entire S-CNN framework is divided into input, convolutional, and output modules. First, original images undergo high-pass filtering preprocessing. The HPF layer can accelerate CNN model convergence and is a special convolutional layer with 5×5 convolution kernels initialized with fixed weights F. By setting the learning rate parameter of the HPF layer to 0, the weights F remain fixed and do not update during training.

The convolutional module consists of convolutional layers and activation function layers. Convolutional layers extract different features from original images through convolution operations. Since original images exhibit local stability, the convolutional layer can synthesize features from various regions to obtain whole-image features. Convolutional layers comprise multiple convolution kernels, each being a weight matrix; more convolution kernels extract more features. As shown in [Figure 2: see original paper], images filtered by HPF undergo matrix operations with convolution kernels to extract different features, with each kernel producing one feature map that serves as input to the activation function layer. Activation functions break the linear characteristics of linear filtering during convolution. Image steganography embeds noise into cover images, and steganalysis identifies this noise; pooling layers weaken such noise and negatively impact classification performance. Therefore, this paper removes pooling layers.

After two convolutional modules, feature maps enter the output module, which

includes two fully connected layers and a loss function layer. Each neuron in the fully connected layers connects to all neurons in the previous layer, integrating locally discriminative information from convolutional or sampling layers [11]. To enhance CNN network performance, ReLU functions are adopted as activation functions for all neurons in the fully connected layers. The loss function layer employs the softmax function, and selecting an appropriate loss function is crucial for specific classification problems. Finally, the softmax layer completes classification.

Activation function selection is a critical aspect of CNN construction. The proposed CNN model uses ReLU functions, and experiments compare the classification effects of TanH and ReLU functions. In [Figure 3: see original paper], the solid curve represents ReLU while the dashed curve represents TanH. The figure shows that ReLU function outputs do not saturate with increasing input. Compared with saturating nonlinear functions, non-saturating nonlinear functions can solve gradient explosion/vanishing problems while accelerating convergence [12]. Additionally, activation functions must be differentiable to compute backpropagated errors.

For neural networks, training is a complex process. Beyond input layer data, the distribution of input data for each subsequent layer continuously changes as network parameters update, accumulating and amplifying throughout the network. This phenomenon, known as "Internal Covariate Shift" [13], forces the network to relearn new data distributions, affecting training speed. Moreover, the essence of neural network learning is to learn data distributions; once training and testing data distributions differ, network generalization capability significantly decreases. To improve training efficiency and recognition performance, this paper adds batch normalization layers (BN layers) before activation functions.

BN algorithm consists of two steps: First, normalization before each layer' s input transforms previous layer outputs to have zero mean and unit variance:

$$\mu_K = \mathbb{E}[x_K], \quad \sigma_K^2 = \mathrm{Var}[x_K]$$

$$\hat{x}_K = \frac{x_K - \mathbb{E}[x_K]}{\sqrt{\mathrm{Var}[x_K]}}$$

where K represents the network layer, x_K denotes input images to the BN layer in layer K, $\mathbb{E}[x_K]$ is the mean of each mini-batch during training (not the entire dataset), and $\mathrm{Var}[x_K]$ is the variance of each mini-batch. This normalization can destroy features learned by the previous layer. To restore these features, the second step of BN algorithm performs transformation and reconstruction:

$$y_K = \gamma \hat{x}_K + \beta$$

where y_K is the BN layer' s output image, and γ, β are learnable parameters. When $\gamma = \sqrt{\mathrm{Var}[x_K]}$ and $\beta = \mathbb{E}[x_K]$, the features learned by the previous layer can be recovered. Through transformation and reconstruction, the spatial relationships of feature maps are preserved.

Compared with Guanshuo Xu et al.' s model, the proposed S-CNN reduces CNN layers while increasing the number of convolution kernels in convolutional layers, enabling the network to extract more features. Although pooling layers reduce computational complexity and removing them increases memory consumption, pooling layers cause information loss that is detrimental to steganalysis. Considering these trade-offs, this paper removes pooling layers. All activation functions in the proposed model are ReLU functions, and experiments demonstrate that using exclusively ReLU functions yields better recognition performance than using TanH for the first layer and ReLU for the second. Additionally, the two-layer fully connected structure also contributes to improved recognition performance. BN layers prevent the model from falling into local optima during training and improve recognition accuracy; therefore, the S-CNN model employs BN layers after convolutional layers.

---

3 Experiments

3.1 Dataset and Experimental Platform

The experiments use the BOSSbase V1.01 dataset, which contains 10,000 grayscale images in pgm format with 512×512 resolution and is currently the most commonly used dataset for image steganalysis research. Due to GPU memory limitations, the grayscale images were batch-cropped to 128×128 using Photoshop, and 40,000 images were randomly selected as cover images. The S-UNIWARD steganographic algorithm was applied to embed information at 0.4 bpp (bits per pixel). The dataset was split into 30,000 cover images and 30,000 stego images for training, and 10,000 cover images and 10,000 stego images for testing. Experiments were implemented using the open-source deep learning framework Caffe on Windows.

3.2 Experimental Process

According to Caffe requirements, the dataset was first converted to leveldb format, and sample means were computed. The CNN architecture was constructed in the network configuration file, and training logs were saved for performance analysis. To improve training efficiency, original images were normalized during preprocessing by subtracting the dataset mean, which significantly reduces computational resource consumption and accelerates training. Dropout and L2 regularization were removed to improve model generalization capability [13]. BN layers enable the model to select larger learning rates, greatly accelerating convergence and reducing training time. To improve recognition accuracy, the

original image order in the dataset was shuffled to prevent certain images from being called multiple times.

3.3 Experimental Parameters

Caffe provides six optimization algorithms; this paper selects Stochastic Gradient Descent (SGD). The base learning rate (base_lr) is 0.01, momentum is 0.9, and weight decay is 0.004. The learning rate policy (lr_policy) is set to "inv" , where the learning rate decreases with iteration count, avoiding manual tuning. Due to memory limitations, the batch size is 64. The maximum iteration count is 5000, and to better evaluate CNN model performance, testing is performed every 300 iterations for a total of 16 tests. For the HPF layer, the learning rate multiplier (lr_mult) is set to 0 to keep its parameters fixed.

3.4 Experimental Results

Compared with traditional CNN steganalysis methods, the S-CNN model reduces network layers, simplifies network structure, accelerates training efficiency, and improves recognition accuracy. Under identical experimental conditions, the training time and recognition accuracy of various models are shown in .

** Training time and recognition accuracy of different models**

Model	Training Time (h)	Accuracy
Tan et al.' s 3-layer CNN model [14]	-	76.19%
Qian et al.' s 5-layer CNN model [7]	-	-
Xu et al.' s 5-layer CNN model [10]	-	80.24%
S-CNN model	-	88.92%

Compared with the traditional two-step machine learning steganalysis method (RM+EC), the S-CNN model shows significant improvement in detecting the S-UNIWARD steganographic algorithm, as shown in .

** Comparison of recognition accuracy between models**

Model	Accuracy
RM+EC	79.53%
S-CNN model	88.92%

Based on GPU performance and considering both training time and recognition accuracy, this paper tested five CNN architectures with different convolution kernel sizes, with results averaged over 16 tests. To verify the impact of BN layers on recognition accuracy, comparative experiments with and without BN

layers were conducted, proving that BN layers effectively improve CNN model performance, as shown in .

** Recognition accuracy of CNNs with different kernel sizes**

Convolution Layer 1	Convolution Layer 2	Accuracy (no BN)	Accuracy (with BN)
-	-	-	-

Experiments were also conducted on activation function selection, demonstrating that using exclusively ReLU functions outperforms using TanH for the first layer and ReLU for the second, as shown in [Figure 4: see original paper].

For image steganalysis, the HPF layer can significantly improve model convergence speed. shows the loss values and accuracy with and without the HPF layer when training the optimal model for 5,000 iterations.

** Loss and accuracy with and without HPF layer**

Configuration	Loss	Accuracy
With HPF layer	-	-
Without HPF layer	-	-

The loss value reflects the error between estimated and actual values, representing model fitting degree. [Figure 5: see original paper] shows the loss variation during model training. As iteration count increases, loss decreases and stabilizes.

Currently, Generative Adversarial Networks (GAN) have become a hot topic in image steganalysis research. GANs achieve balance through adversarial training between generator and discriminator networks. The classification accuracy of the DCGAN model proposed by Denis et al. [15] is shown in .

** Comparison of recognition accuracy between DCGAN and S-CNN**

Model	Accuracy
GAN (same seed)	-
GAN (random seed)	-
S-CNN	88.92%

The results show that when the generator network uses the same seed, recognition performance is better than S-CNN, but the generated images are not suitable for steganography. When using random seeds, recognition performance

decreases significantly below S-CNN, but produces better cover images for steganography.

Experimental results demonstrate that deep learning is a very promising tool for image steganalysis. Compared with traditional methods, it avoids manual feature extraction, greatly improves steganalysis efficiency, and effectively increases accuracy. Currently, CNN can only test simple steganographic algorithms; classification performance decreases when algorithms become complex or embedding rates are too low. Future work will focus on enhancing the generality of CNN-based steganalysis.

References

[1] Ren H E, Chang C W, Zhang J. Improved bilinear interpolation algorithm in information hiding [J]. Computer Application Research, 2010, 27(11): 4290-4292.
[2] Tao R, Zhang T, Ping X J. Blind detection resistant image steganography algorithm based on texture complexity and difference [J]. Computer Applications, 2011, 31(10): 2678-2681.
[3] Kodovsky J, Fridrich J, Holub V. Ensemble classifiers for steganalysis of digital media [J]. IEEE Trans on Information Forensics & Security, 2012, 7(2): 432-444.
[4] Luo W, Huang F, Huang J. Edge adaptive image steganography based on LSB matching revisited [J]. IEEE Trans on Information Forensics & Security, 2010, 5(2): 201-214.
[5] Pevný T, Bas P, Fridrich J. Steganalysis by subtractive pixel adjacency matrix [J]. IEEE Trans on Information Forensics & Security, 2010, 5(2): 201-214.
[6] Xiong G, Ping X J, Zhang T, et al. Image textural features for steganalysis of spatial domain steganography [J]. Journal of Electronic Imaging, 2012, 21(3): 033015.
[7] Qian Y, Dong J, Wang W. Deep learning for steganalysis via convolutional neural networks [C]// Proc of International Society for Optical Engineering. 2015: 9409: 94090J-94090J-10.
[8] Fridrich J, Kodovsky J. Rich models for steganalysis of digital images [J]. IEEE Trans on Information Forensics & Security, 2012, 7(3): 868-882.
[9] Kodovsky J, Fridrich J, Holub V. Ensemble classifiers for steganalysis of digital media [J]. IEEE Trans on Information Forensics & Security, 2012, 7(2): 432-444.
[10] Xu G, Wu H Z, Shi Y Q. Structural design of convolutional neural networks for steganalysis [J]. IEEE Signal Processing Letters, 2016, 23(5): 708-712.
[11] Sainath T N, Mohamed A R, Kingsbury B, et al. Deep convolutional neural networks for LVCSR [C]// Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. 2013: 8614-8618.
[12] Xu Bing, Wang Naiyan, Chen Tianqi, et al. Empirical evaluation of rectified activations in convolution network [EB//OL]. arXiv: 1505.00853v2, 2015.

[13] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [Z]. 2015.
[14] Tan S, Li B. Stacked convolutional auto-encoders for steganalysis of digital images [C]// Proc of IEEE Signal and Information Processing Association Summit and Conference. 2014: 1-4.
[15] Volkhonskiy D, Nazarov I, Borisenko B, et al. Steganographic generative adversarial networks [Z]. 2017.

Note: Figure translations are in progress. See original paper for figures.