A fast and effective image steganalysis model based on convolutional neural network

Fengsong Hu, Rong Xu, Zhekun Cheng
College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
hfs@hnu.edu.cn

Abstract. In recent years, the performance of deep learning in image steganalysis applications has become more and more outstanding, but at the same time the training time has also greatly increased. Some models need to be trained for several days, and the research efficiency is very low. In this article, we propose an image steganalysis model in spatial domain based on a three-layer convolutional neural network. The model does not use a pooling layer, and uses the global average pooling layer instead of the fully connected layer. Experimental results show that the training time of the model is greatly shortened, and the accuracy of detecting the three steganography algorithms with an embedding rate of 0.4bpp exceeds 85%.

1. Introduction

The history of the research and development of steganalysis technology began in the mid-1990s, and it has been more than 20 years since then, from the specific steganography algorithm (special steganalysis method). Up to now, continuous progress has been made in various adaptive steganography algorithms (general steganalysis methods). Contemporary steganalysis techniques can be divided into two categories: one is the steganalysis technique based on artificial engineering features and traditional classifiers, and the other is the steganalysis technique based on deep learning.

The classic steganalysis methods of artificially designed features include high-order statistical features based on wavelet decomposition proposed by Lyu and Farid et al. [1], Holotyak et al. [2] proposed using statistical moments in the wavelet transform domain, and Fridrich et al. [3] proposed hidden image calibration based on Writing analysis method, Sullivan et al. [4] proposed a steganalysis feature based on the correlation between pixels in the image domain. When steganography performs a steganography operation on an image, it changes certain statistical characteristics of the image, which cannot be distinguished by the naked eye. Therefore, the early steganalysis methods focused on how to design appropriate image statistics to make steganography. The change in operation becomes obvious.

Different from traditional steganalysis methods, convolutional neural networks can automatically extract image features and have achieved great success in a wide range of applications. Tan et al. [5] first used convolutional neural networks in steganalysis in 2014. In 2015, Qian et al. [6] first proposed a steganalysis method using Gaussian nonlinear activation function and mean pooling layer, but the performance was not good. In 2017, Ye et al. [7] proposed a steganalysis method based on deep residual network, which has significantly improved performance. In 2019, Boroumand et al. [8] proposed a 12-layer deep residual network SR-Net. The detection accuracy of the encrypted image modified by the WOW steganography algorithm with a detection embedding rate of 0.4bpp has exceeded 90%.

The research on steganalysis methods based on convolutional neural networks at home and abroad has surpassed the traditional artificial design feature method in recognition accuracy, but the training
time of very excellent models such as SR-Net takes more than three days, and the deployment of experimental equipment is very expensive. It is not conducive to the practical application of research results and the further development of the subject. The main goal of this paper is to shorten the training and testing time of the model as much as possible, and to improve the efficiency of model deployment without significant loss of recognition accuracy.

2. QX-Net for Image Steganalysis

We named the proposed steganalysis model QX-Net. Fig. 1 illustrates the overall architecture of our CNN. QX-Net consists of an image pre-processing block, three convolution blocks and a classification block. The network produces a probability distribution over the two class labels.

In order to suppress the image content and enhance the steganographic embedding which is a hidden weak signal, QX-Net uses the high-pass filter (HPF layer) commonly used by researchers in the traditional steganalysis field for image pre-processing. The HPF layer is actually a special convolutional layer, its convolution kernel size is 5×5, also called KV kernel, it is a linear filtering kernel. It usually has the following in formula 1. \( N_{ij} \) is the area pixel of \( X_{ij} \), excluding the equal sign. \( \hat{X}_{ij}(\cdot) \) is the predicted value of \( X_{ij} \). \( R_{ij} \) is the residual value obtained by calculation, and the high-pass filtered image is usually called the residual image.

\[
R_{ij} = \hat{X}_{ij}(N_{ij}) - X_{ij}
\] (1)

In addition to the image pre-processing layer, QX-Net also has three convolution modules and one classification module. The convolution module performs convolution operation on the feature map obtained by image pre-processing. The classification module produces the classification decision (cover or stego). See Fig. 1. Each block is made of the following steps:
1. a Convolution Layer. Similar to S-CNN [9], The first convolution module uses 64 convolution kernels with a size of 3×3, and the second and third convolution modules both use 100 1×1 convolution kernels. Using multiple 1×1 convolution kernels can improve the ability to perform more complex calculations on the feature map, and integrate the information in the feature map across channels. In order to reduce the size of the feature map of the convolutional layer and at the same time reduce the amount of calculation of the convolution operation, we set the stride of the convolutional layer to 2.

2. an Absolute Value activation (ABS) layer. This ABS layer is only used in the first convolution module similarly to Yedroudj-Net [10]. It allows statistical modeling to consider the sign symmetry of the noise residual and discards the negative information of 0 symmetry in the feature map.

3. a Batch Normalization (BN). The batch normalization operation performed by the batch normalization layer (BN layer) can speed up the convergence speed of the network and suppress overfitting to a certain extent. We only use the BN layer in the first convolution module to reduce the loss of feature information in the convolution module 1. The second and third convolution modules do not set the BN layer to shorten the training time.

4. Activations. The activation function can make the convolutional neural network have a nonlinear expression ability, and can greatly improve the fitting ability of the convolutional neural network. The QX-Net model all uses the ReLU activation function, see formula 2. The ReLU function is a piecewise linear function with one-sided inhibition. That is, the negative numbers are all 0 after activation, and the positive numbers remain unchanged. This makes the neurons in the network have sparse activation and the calculation speed is very fast.

\[ \text{ReLU} = \max(0, x) \]  

5. Global average pooling. Similar to S-NIN [11], QX-Net's classification module uses a global average pooling layer instead of a fully connected layer. It can perform full average pooling on each feature map, so that each feature map corresponds to an output. This setting can further reduce the number of network parameters and improve the training efficiency and recognition effect of the model.

3. Experiments

3.1. Dataset and software platform

The data sets we use are BOSSbase V1.01 and BOWS-2. Each of these two data sets contains 10,000 512×512 grayscale images. Because the experimental equipment configuration is GeForce GTX 1080 Ti, 12GB video memory; memory is 64GB. The computing power is very limited, so we cut the 512×512 image into a 128×128 size using Photoshop's batch processing command. Then use a simple script written in python to randomly select 20,000 images (a total of 40,000) from BOSSbase V1.01 and BOWS-2 for the experiment, and then randomly select 30,000 from the 40,000 images as the original image training set.

We use S-UNIWARD [12], WOW [13], and HILL [14] steganography algorithms with an embedding rate of 0.4bpp to embed secret information on 30,000 original images to obtain the corresponding training set of encrypted images. The remaining 10,000 images were made into a test set. All the experiments in this section are carried out on the Caffe1.0 platform under the Windows10 environment.

3.2. Hyper-parameters

Before using Caffe1.0 for model training, you need to use the convert.imageset.exe program to convert the data set .pgm format into the platform-specific Leveldb format. Then build the QX-Net model in the network model.prototxt file, and set the experimental model parameters in the hyperparameter file solver.prototxt. The specific settings of the parameters are as follows:

1. Solver_type is the stochastic gradient descent algorithm (SGD).
2. The basic learning rate (base_lr) is 0.01.
3. The learning rate strategy (lr_policy) is "inv". As the number of iterations increases, the learning rate will decrease.
4. The weight of the last update (momentum) is 0.9.
5. The weight decay (weight_decay) is 0.004.
6. Due to limited video memory, each batch size during training is 32.
7. The maximum number of training iterations (max_iter) is 15000.
8. The test interval (test_interval) is 300, and a test is performed every 300 iterations. The initialization method of all convolutional layers is "Xavier".

3.3. Reference model overview

Since the steganalysis model constructed in this paper refers to the model architectures of Yedroudj-Net and S-CNN, this section will give an overview of these two model architectures.

The S-CNN network architecture is shown in Figure 2. The input image size of S-CNN is 128×128, and only one HPF high-pass filter is used for image pre-processing. The HPF layer can accelerate the convergence speed of the CNN model. The convolution module contains only two layers, the same BN layer and ReLU activation function. In the S-CNN model, the setting of the pooling layer is cancelled, in order to avoid the pooling layer from weakening the image noise that needs to be recognized by the steganalysis. The output module contains two fully connected layers with 1000 neurons per layer, and a loss function layer that uses the softmax function to complete the classification.

As shown in Figure 3, the Yedroudj-Net input image size is 256×256, and 30 SRM high-pass filter cores with a size of 5×5 are used for image pre-processing. Then in the 5-layer convolution module, only the absolute value (ABS) layer and the batch normalization (BN) layer are used in the first layer; the truncation function Trunc is used in the first and second layers to limit the range of data values, and prevent deeper layers from modeling large values. The truncation function Trunc is given in formula 3 and is parameterized by $T \in \mathbb{N}$ (threshold). For layers 3 to 5, the classic ReLU activation function is
used, and the average pooling layer is dedicated to the second to fifth layers. For the last layer, global average pooling is performed to generate an element for each corresponding feature map, thereby preventing Statistical modeling obtains the position information of embedded pixels from the training data. The classification module is composed of three fully connected layers. At the end of the module, the softmax activation function is used to generate distributions on two class labels.

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

Figure. 3. Yedroudj-Net CNN architecture.

3.4. Results
First, we conducted experiments on the detection accuracy of the three steganography algorithms of QX-Net, Yedroudj-Net and S-CNN at an embedding rate of 0.4bpp. The experimental results are shown in Table 1. The QX-Net constructed in this paper has a higher recognition accuracy rate than Yedroudj-Net and lower than S-CNN. In order to further determine the cause of the gap, we conducted a comparative experiment with or without the pooling layer. The experiment tested the S-UNIWARD steganography algorithm under 0.4bpp. The results are shown in Table 2. The preliminary analysis of
the dimensionality reduction effect of the pooling layer has a certain negative impact on the recognition accuracy. Yedroudj-Net originally required higher experimental environment configuration requirements than the experimental equipment we used, so it is normal for its recognition accuracy to decrease.

Table 1. Comparison results of detection accuracy of the three models.

| Models       | Accuracy  |
|--------------|-----------|
|              | WOW 0.4bpp| S-UNIWARD 0.4bpp | HILL 0.4bpp |
| Yedroudj-Net | 75.81%    | 76.45%         | 73.66%      |
| S-CNN        | 89.34%    | 89.76%         | 89.42%      |
| QX-Net       | 85.77%    | 86.06%         | 83.61%      |

Table 2. Comparison of experimental results with or without pooling layer.

| QX-Net         | Loss  | Accuracy | Training time (15,000 iterations/h) |
|----------------|-------|----------|-------------------------------------|
| With pooling layer | 0.0971 | 83.92%   | 1.13h                               |
| Without pooling layer | 0.1004 | 86.03%   | 0.91h                               |

Second, we conducted experiments on the influence of the presence or absence of the BN layer on the recognition accuracy and training time of the S-UNIWARD steganography algorithm under 0.4bpp. The experimental results are shown in Table 3. The setting of the BN layer has an effect on the recognition accuracy, but will increase model training time.

Table 3. Comparison of experimental results with or without BN layer.

| QX-Net      | Accuracy | Training time (15,000 iterations/h) |
|-------------|----------|-------------------------------------|
| With BN layer | 87.41%   | 1.44h                               |
| Without BN layer | 86.19%   | 1.02h                               |

Third, we conducted experiments on the effect of replacing the fully connected layer with the global average pooling layer in the QX-Net output module. The experimental results are shown in Table 4. Among them, the recognition accuracy is based on the S-UNIWARD steganography algorithm under 0.4bpp as the detection object. It can be seen from Table 4 that the QX-Net output module adopts the global average pooling layer to achieve a higher recognition accuracy than the fully connected layer by 0.3%, and the training and testing time is shorter.

Table 4. The effect of global average pooling layer and fully connected layer on model performance experimental results.

| classification module | Accuracy | Training time (15,000 iterations/h) | Test time (100 sheets/s) |
|-----------------------|----------|-------------------------------------|--------------------------|
| fully connected layer | 85.17%   | 1.21h                               | 52s                      |
| global average pooling layer | 85.46%   | 0.94h                               | 26s                      |
In order to further study the performance of QX-Net in training and testing time, we next performed comparative experiments on the above models under the condition of 15,000 iterations. The results are shown in Table 5. Choosing the number of 100 test images can make the time easier to observe, and in order to reduce the error, we tested 10 times and averaged as the experimental result.

Table 5. Training and testing time comparison results.

| Models     | Training time (15,000 iterations/h) | Test time (100 sheets/s) |
|------------|------------------------------------|--------------------------|
| Yedroudj-Net | 7.31h                              | 210s                     |
| S-CNN      | 2.70h                              | 66s                      |
| QX-Net     | 0.92h                              | 26s                      |

The above experiments show that the overall performance of the QX-Net model is better than Yedroudj-Net, the recognition accuracy is about 4.36% lower than that of S-CNN, the training time is shortened by about 65.93%, the test time is shortened by about 60.61%, and the model efficiency is obtained a big improvement.

4. Conclusion

Based on the results and discussions presented above, the conclusions are obtained as below:

(1) It is shown that the cancellation of the pooling layer has a positive effect on the recognition accuracy. The setting of the BN layer has an effect on the recognition accuracy, but it will increase the model training time.

(2) In this model, a global average pooling layer is used instead of a fully connected layer, and each feature map is fully averaged pooled so that each feature map corresponds to an output. Using the global average pooling layer can reduce the number of network parameters and improve the training efficiency of the model.

Acknowledgments

This work was financially supported by the Research on Key Technologies of Cloud Computing Security Service Platform of Changsha Huixian Education Technology Co., Ltd(H201991420122).

References

[1] Lyu, S. and farid, H. (2002) Detecting Hidden Messages Using Higher-Order Statistics and Superpoint Vector Vector Machines. Springer, Berlin. A Mashko. Review of approaches to the problem of driver fatigue and drowsiness[C]. In: Smart Cities Symposium Prague. Prague: IEEE, 2015, 1-5.
[2] Holotyak. T, Fredrich. J, Voloshynovsky. S. Blind statistical steganalysis of additive steganography using wavelet higher order statistics[C]/Springer, IFIP Internation Communication on Communications and Multimedia Security. [S. l.]: SPringer, 2005: 273-274.
[3] Friedrich J. Feature-based steganalysis for jpeg images steganographic schemes[C] //Springer, International Workshop on Information Hiding. [S. l.]: Springer, 2004: 67-81.
[4] Sullivan K, Madhow U, Chandrasekaran S, et al. Steganalysis for markov cover data withapplications to images[J].IEEE Transformation June 2006, 1(2): 275-287.
[5] Tan S, Li B. Stacked convolutional auto-encoders for steganalysis of digital images[C]//Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific. IEEE, 2014: 1-4.
[6] Y. Qian, J. Dong, W. Wang. Deep learning for steganalysis via convolutional neural networks[C]// Electronic Imaging, Media Watermarking, Security, and Forensics. San Francisco, CA, USA: SPIE, 2015:94090J-94090J-10.
[7] J. Ye, J. Ni, Y. Yi. Deep Learning Hierarchical Representations for Image Steganalysis[J]. IEEE
Transactions on Information Forensics & Security, 2017, 12(11):2545-2557.

[8] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. 2019. Deep Residual Network for Steganalysis of Digital Images. IEEE Transactions on Information Forensics and Security 14, 5 (2019), 1181–1193.

[9] Lixian Wei, Peixian Gao, Jia Liu, Mingming Liu. Image steganalysis method based on convolutional neural network[J]. Application Research of Computers, 2019, 36(01): 235-238.

[10] Yedroudj M, Comby F, Chaumont M. Yedrouj-Net: An efficient CNN for spatial steganalysis. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 2092-2096.

[11] Peixian Gao, Lixian Wei, Jia Liu, Mingming Liu. Structure improvement of convolutional neural network for image steganalysis[J]. Computer Engineering, 2018, 44(10): 309-313.

[12] Vojtech Holub, Jessica Fridrich, and Tomas Denemark. Universal cuisine function for stegan graph in an arbitrary domain. EURASIP Journal on Information Security, 2014 (1): 1-13.

[13] Holub V, Fridrich J. Designing Steganographic Distortion Using Direction Filters [A]. 2012.

[14] Li B, Wang M, Hang J, et al. A new cost function for spatial image steganography. [J]. 2015: 4206-4210.