CNN-based Steganalysis of MP3 Steganography in the Entropy Code Domain

Yuntao Wang
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China 100093
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China 100093
wangyuntao2@iie.ac.cn

Kun Yang
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China 100093
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China 100093
yangkun9076@iie.ac.cn

Xiaowei Yi[*]
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China 100093
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China 100093
yixiaowei@iie.ac.cn

Xianfeng Zhao
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China 100093
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China 100093
zhaoxianfeng@iie.ac.cn

Zhoujun Xu
Beijing Information Technology Institute,
Beijing, China 100094
pl_xzj@uestc.edu.cn

ABSTRACT

This paper presents an effective steganalytic scheme based on CNN for detecting MP3 steganography in the entropy code domain. These steganographic methods hide secret messages into the compressed audio stream through Huffman code substitution, which usually achieve high capacity, good security and low computational complexity. First, unlike most previous CNN based steganalytic methods, the quantified modified DCT (QMDCT) coefficients matrix is selected as the input data of the proposed network. Second, a high pass filter is used to extract the residual signal, and suppress the content itself, so that the network is more sensitive to the subtle alteration introduced by the data hiding methods. Third, the 1×1 convolutional kernel and the batch normalization layer are applied to decrease the danger of overfitting and accelerate the convergence of the back-propagation. In addition, the performance of the network is optimized via fine-tuning the architecture. The experiments demonstrate that the proposed CNN performs far better than the traditional handcrafted features. In particular, the network has a good performance for the detection of an adaptive MP3 steganography algorithm, equal length entropy codes substitution (EECS) algorithm which is hard to detect through conventional handcrafted features. The network can be applied to various bitrates and relative payloads seamlessly. Last but not the least, a sliding window method is proposed to steganalyze audios of arbitrary size.

[*]Corresponding author

CCS CONCEPTS

• **Security and privacy** → *Authentication*; • **Computing methodologies** → *Learning latent representations*;

KEYWORDS

CNN, entropy code, steganalysis, MP3, QMDCT coefficients, adaptive

ACM Reference Format:
Yuntao Wang, Kun Yang, Xiaowei Yi, Xianfeng Zhao, and Zhoujun Xu. 2018. CNN-based Steganalysis of MP3 Steganography in the Entropy Code Domain. In *IH&MMSec '18: 6th ACM Workshop on Information Hiding and Multimedia Security, June 20–22, 2018, Innsbruck, Austria,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3206004.3206011

1 INTRODUCTION

Steganography is the art of embedding secret messages into digital files, which is widely used as a secure method of communication between two parties. On the contrary, steganalysis aims to detect hidden messages in suspicious files. With the development of multimedia technology, recording equipment and editor software popularized in people's daily life, which brings convenience to steganography. Due to the high compression rate and the high quality of MP3, these compressed digital products become one of the most influential media recently. Furthermore, facilitated by the more complex coding principle and the better concealment, MP3 is regarded as one of the most suitable carriers for data hiding.

In the past decade, many MP3 steganographic schemes have emerged gradually. Majority of them are combined with the encoder. MP3Stego [14] is a well known steganographic method. The embedding operation is completed in the inner loop of the encoding. Messages are hidden based on the parity of part_2_3_length. Moreover, Liu *et al.* [13] proposed a data hiding method based on the energy of MDCT coefficients in adjacent frames. Gao *et al.* [4] and Yan *et al.* [21] presented steganographic algorithms based on Huffman codes substitution respectively, which establishes a mapping relationship between secret messages and Huffman codes. And Yang *et al.* [22] proposed an adaptive MP3 steganographic algorithm using equal length entropy codes substitution (EECS). A content-aware distortion function is designed to achieve optimal masking effect via the psychoacoustic model in this algorithm, which makes the algorithm more secure than previous methods.

Until now, many statistical features are proposed for MP3 steganalysis, but most of them are against MP3Stego. Westfeld [19] detected MP3Stego based on the change of the part_2_3 length variance. In [15], a method for MP3Stego steganalysis was proposed, which is based on the second order differential quantized modified discrete cosine transform (QMDCT) coefficients, accumulative Markov transition features, and accumulative neighboring joint density features. Wan *et al.* [18] proposed a steganalytic approach based on Huffman table distribution and recoding scheme. Yan [20] found that the length of bit reservoir would change after steganography, so they presented a method based on the statistic characters of bit reservoir. For general schemes, Hamzeh *et al.* [5] extracted a set of features based on the reversed mel-frequency cepstral coefficients (R-MFCC) of audio and its second order differential signal. Jin *et al.* [10] proposed to extract Markov features from row and column of the QMDCT coefficients matrix. Besides, Ren *et al.* [16] extracted the Markov and joint density features of multi-order Intra and Inter frame to detect AAC steganography. All these general schemes have good versatility.

Recently, various deep learning architectures are proposed successively, which have achieved state-of-the-art results in many areas, such as computer vision, speech recognition, natural language processing, and reinforcement learning. However, very few deep learning based methods have been applied for audio steganalysis. Therein, Chen *et al.* [1] first proposed a novel CNN to detect ±1 LSB steganography in the time domain.

This paper presents a deep learning based method to detect MP3 steganographic algorithms in the entropy code domain. In our proposed network, we first obtain the QMDCT coefficients matrix of MP3 audio and extract the residual signal via the second order row differences to improve the sensitivity to the subtle alteration introduced by the algorithms. Then, several block convolutional layers are followed to capture the potential properties of the input data. Each block is a combination of the convolutional layer, the activation function layer, and the max pooling layer. Some blocks contain a batch normalization layer at the top of 1×1 convolutional layer. Finally, the fully connected layer and the batch normalization layer are connected in cascade at the end of the network. The performance of the architecture is optimized via fine-tuning, such as substitution of activation function and subsampling methods. Experiments demonstrate that our proposed network outperforms the traditional handcrafted features, and the performance is also good at low relative payloads especially. Besides, we introduce sliding window mechanism to steganalyze audios of arbitrary size.

Explicitly, our two contributions to MP3 steganalysis are as follows.

1. A CNN-based method is proposed which is against MP3 steganographic algorithms in the entropy domain. Our network is effective to detect MP3 data hiding methods in the entropy domain at several bitrates and relative payloads. In particular, this network can be applied to steganalyze the EECS algorithm, a novel adaptive MP3 steganography, which is hard to detect by traditional handcrafted features.

2. The 1×1 convolutional kernel and the batch normalization layer are introduced to accelerate the convergence of the backpropagation and to decrease the danger of overfitting. Besides, we analyze the effect of batch normalization layer on the final accuracy and convergence speed further. Redundant batch normalization layers will decrease the final accuracy. Thus, we finetune the batch normalization layer to make the network perform better.

The rest of this paper is organized as follows. The preliminaries are introduced in Section 2. The architecture of our network is presented in Section 3. And Section 4 shows the experimental settings and results. Finally, the conclusions are drawn in Section 5.

2 PRELIMINARIES

2.1 Overview of MP3 Encoder

The architecture of MP3 encoder is shown in Figure 1, which can be divided into six parts: framing and sub-band filter, psychoacoustic model (PAM), MDCT, quantization, Huffman encoding and bitstream formatting. The original audio is encoded using pulse code modulation (PCM), which is WAV format. The WAV audio is first segmented into frames of 1152 sample dots. The sub-band filter process divides the audio signal into 32 uniform frequency sub-bands. The PAM process analyzes the audio signal via Fast Fourier Transform (FFT) and computes the perceptual entropy (PE). The MDCT process performs the time-frequency mapping with four type of windows. Then the non-uniform quantized MDCT coefficients are encoded through Huffman encoding process. Therein, the value of SMR determines the number of bits in quantized coding. Finally, the codestream and side information are formatted into the MP3 file.

As described in Figure 2, a channel consists of 576 QMDCT coefficients, which are orderly divided into three regions: big-value region, count1 region, and zero region. For the big-value region, every two QMDCT coefficients are encoded into one Huffman code. The big-value region can be further segmented into region0, region1, and region2. Three subregions are encoded independently through the utilization of Huffman tables from Table 0 to Table 31. If the value of QMDCT coefficients is less than 15, it is coded directly. Otherwise, the exceeding value is represented by linbits. Besides, if the coefficient is nonzero, the sign bit is used. The value 0 indicates the positive number and 1 indicates the negative number. The value of QMDCT coefficients in count1 region belongs to $\{-1, 0, 1\}$, and every four coefficients are encoded into one Huffman code via Table 32 and 33. All coefficients in zero region are zero and those coefficients would not be encoded. The structure of Huffman code

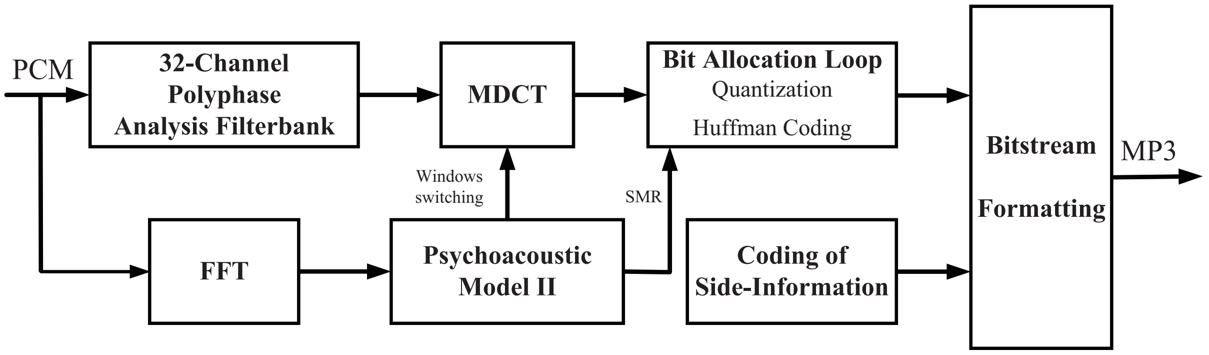


Figure 1: The procedure of MP3 encoding.

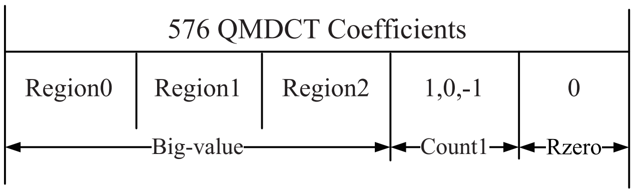


Figure 2: The structure of QMDCT coefficients in a channel.

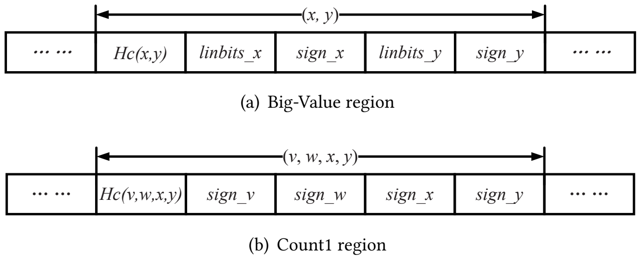


Figure 3: The structure of Huffman codestream.

is shown in Figure 3. H_c denotes a Huffman code of the QMDCT coefficient, *linbits_x* denotes the linbits of the first coefficient, *sign_x* denotes the sign bit of the first coefficient, *linbits_y* denotes the linbits of the second coefficient, *sign_y* denotes the sign bit of the second coefficient. Likewise, *sign_v*, *sign_w*, *sign_x*, and *sign_y* orderly denotes the sign bits of four QMDCT coefficients in count1 region.

2.2 Huffman Code Substitution Data Hiding Method

As more than 90% of the content of the MP3 compressed bitstream is Huffman code, the codewords are ideal steganography carrier. For Huffman code substitution algorithms (HCM), messages are directly embedded by the Huffman code substitution within the MP3 encoding, which is different from the previous MP3 data hiding method. Compared with other MP3 steganography algorithms, HCM can obtain higher capacity, better security, and lower computational complexity.

Random modification of the codewords causes confusion in the structure of the bitstream, which makes the decoder to fail. In order to keep the statistical distribution of Huffman codes and the codestream length, the structure of codestream in Figure 3 could not be changed arbitrarily. Otherwise, the MP3 encoder would collapse. Suppose that h_i^k is the i^{th} Hc in the k^{th} Huffman table, (x_i^k, y_i^k) is the corresponding QMDCT coefficients pair of h_i^k. Thus if $\forall i \neq j$, h_i^k and h_j^k is a pair of substitutable codes, it satisfies the following three conditions.

1. The length of h_i^k and h_j^k is equal,

$$L(h_i^k) = L(h_j^k) \tag{1}$$

2. The number of sign bits of (x_i^k, y_i^k) and (x_j^k, y_j^k) is equal,

$$S(x_i^k, y_i^k) = S(x_j^k, y_j^k) \tag{2}$$

3. The linbits of x_i^k and x_j^k is consistent, as well as y_i^k and y_j^k,

$$G(x_i^k) = G(x_j^k), G(y_i^k) = G(y_j^k) \tag{3}$$

Suppose the set Π^k contains all Huffman codes in the k^{th} Huffman table. For HCM algorithms, this set can be divided into two parts: Π_u^k and Π_v^k. Therein, Huffman codes in Π_v^k are available for embedding secret information. First, Π_v^k is set as $\varnothing$. For $\{\exists h_i | h_j \in \Pi^k \cap h_j \notin \Pi_v^k, (i \neq j)\}$, if (h_i, h_j) satisfies the three conditions, (h_i, h_j) moves to Π_v^k, otherwise it moves to Π_u^k. This process is repeated until Π^k is empty. Each Huffman code h_i in Π_v^k is numbered according to certain rules which keep the distribution characteristics of codewords to the utmost. The Huffman codes in Π_v^k are further divided into Π_0^k and Π_1^k. If the order of the codeword is odd, h_i is put into Π_1^k, or it is put into Π_0^k. Codes in Π_1^k indicate the bit 1 and codes in Π_0^k indicate the bit 0.

Until now, several HCM algorithms have been presented. In [4] (HCM-Gao), specific pairs of Huffman codes are extracted for substitution according to the similarities of codewords. In [21] (HCM-Yan), to minimize the modification of the QMDCT coefficients, Huffman codes are sorted in lexicographic order according to the value of the QMDCT coefficients. And, in [22] (EECS), zigzag scanning is presented in order to keep the distribution characteristics of Huffman

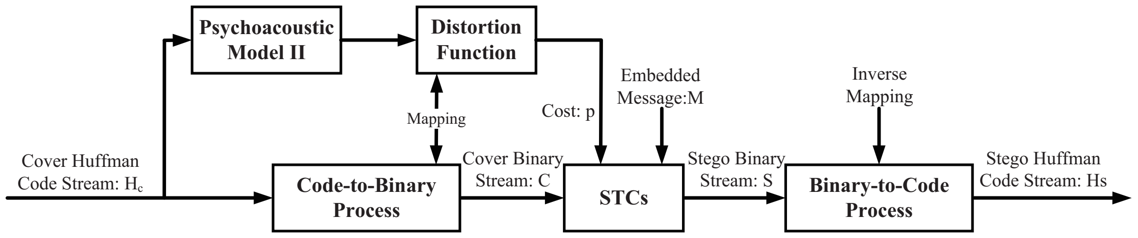


Figure 4: The procedure of EECS algorithm.

codes. Further, the EECS algorithm introduces distortion function based on PAM to hide secret messages adaptively, and the security of the algorithm is better than previous HCM algorithms. The flowchart of the EECS algorithm is shown in Figure 4. The algorithm is mainly divided into four steps: code-to-binary, distortion function, STC encode, and binary-to-code. Firstly, the cover Huffman code H_c is transformed into the cover binary stream C. Secondly, the distortion ρ is calculated according to the PAM and the distance of mapping Huffman codes. Thirdly, the secret messages M are concealed via STC [2]. Finally, the stego binary stream is mapped into stego Huffman code S.

3 THE PROPOSED CNN ARCHITECTURE

In this section, the architecture of the proposed CNN is elaborated, and every part of the network is analyzed in detail. The whole structure is shown in Figure 5.

First, the QMDCT coefficients matrix of MP3 with the size of 200×380 is extracted as the input data of the network. A high pass filter follows to get the residual signal in order to capture the minor modification introduced by the steganography algorithm better. Then, six block convolutional layers are followed in cascade. Each block is a combination of convolutional layers of 3×3 and 1×1 kernel, a Tanh activation function, and a max pooling layer. And we introduce the batch normalization layers in the last three blocks. The fully connected layers and the batch normalization layers are placed at the end. Finally, the cross-entropy loss is used to update the parameters of the network.

3.1 Input data – QMDCT Coefficients Matrix

There are three reasons why the QMDCT coefficients matrix is selected as the input data of the network. First, as described in Section 2, QMDCT coefficients are further encoded into Huffman codes, which means the substitution of Huffman codes is equivalent to the modification of QMDCT coefficients. Next, as shown in Figure 6 and Figure 7, the modification of QMDCT coefficients is larger than the modification of the sampling dots in the time domain by the data hiding methods. Finally, some paper [10, 15, 16] show that the statistical characteristics of the QMDCT coefficients matrix are effective to detect MP3 steganography algorithms. Overall, in order to steganalyze the algorithms in the entropy code domain, the QMDCT coefficients matrix is extracted as the input data. In consideration of the invariance of zero coefficients in steganography, the matrix is cropped to 200×380. To facilitate the description, the matrix is denoted as

$$M_Q = \begin{bmatrix} Q_{1,1} & & Q_{1,j} & & Q_{1,380} \\ & \ddots & & \ddots & \\ Q_{i,1} & & Q_{i,j} & & Q_{i,380} \\ & \ddots & & \ddots & \\ Q_{200,1} & & Q_{200,j} & & Q_{200,380} \end{bmatrix} \tag{4}$$

where the variable $i \in \{1, 2, ..., 200\}$ is the number of selected channels and $j \in \{1, 2, ..., 380\}$ is the index of QMDCT coefficients in a channel. The range of variable i depends on the selected frames N which satisfies $i \in [0, 4N]$.

3.2 High Pass Filter

High pass filter (HPF) is used to reduce the impact of content information and capture the minor modification introduced by the data hiding methods. In many image classification tasks, CNN can effectively extract the significant features that express the properties of the content. Nevertheless, the stego signal is far weaker than the content itself in steganography, which can be seen as high-frequency noise as shown in Figure 6 and Figure 7. Thus, for steganalysis, the introduction of the HPF layer is conducive to improve the detection accuracy. The HPF layer can be seen as a "fixed convolutional layer" in the network. Because the calculation of HPF is based on CPU, we just use some simple HPFs for more quick training.

The type of HPFs has a great influence on the final detection accuracy and the convergence speed of the network. In our experiments, a group of audio files is selected from dataset randomly. For the security of algorithms and the coding principle of MP3, zero values in QMDCT coefficients will not be changed. Therefore, we just calculate the proportion of modified points in nonzero coefficients at the situation of the origin, the first order row differences (Δ_r^1), the second order row differences (Δ_r^2), the first order column differences (Δ_c^1) and the second order column differences (Δ_c^2) separately. All results are shown in the Table 1. Herein, w is the width of the parity-check matrix which satisfies $\alpha = 1/\mathrm{w}$ and α represents the relative payload. As the results shown, the ratio can be expanded to triple of the original via the second order row differences which is selected as the HPF layer finally. Besides, a better HPF like Rich Model [3] is conducive to improve the detection accuracy significantly. It can be discussed for further study.

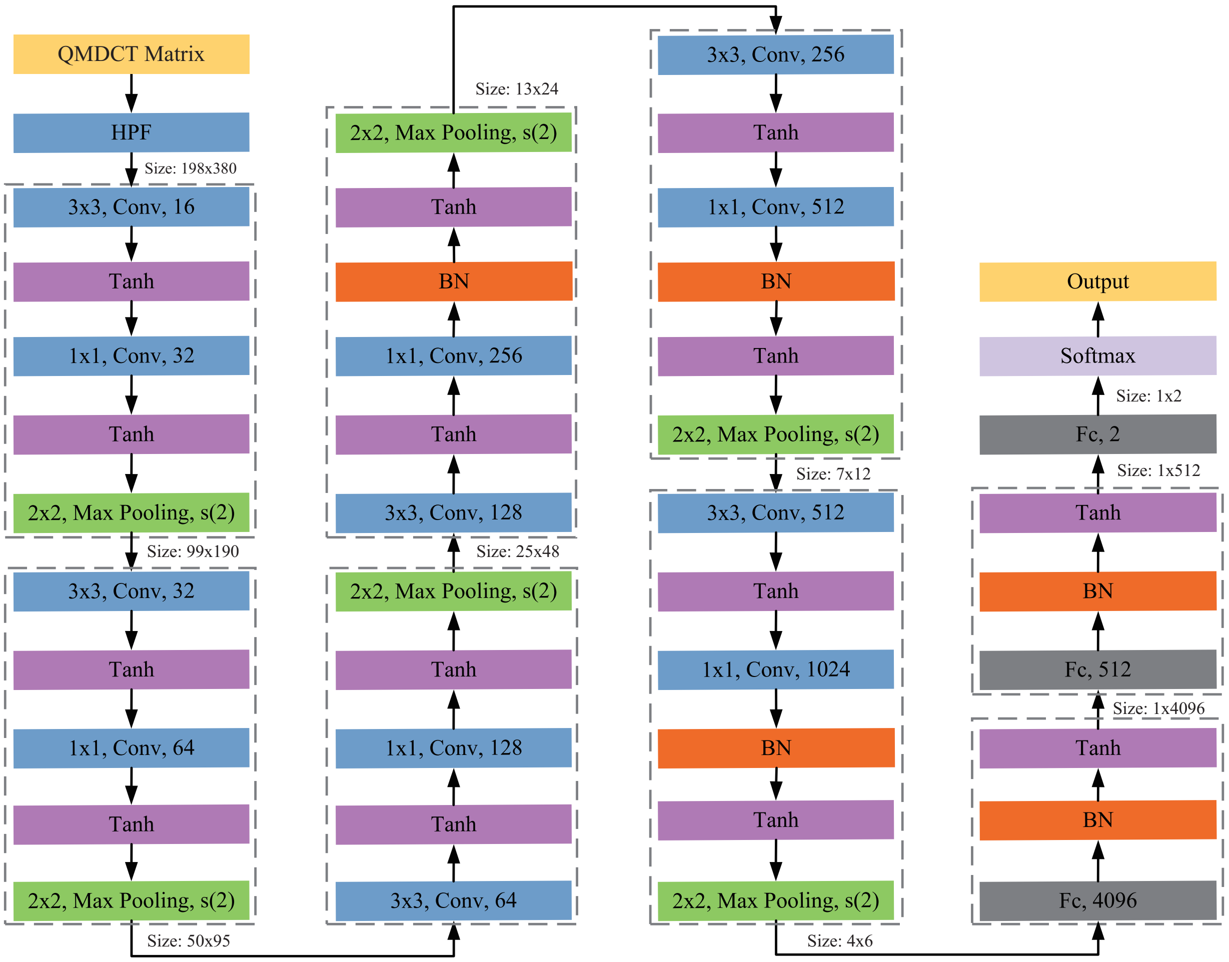


Figure 5: Structure of the proposed network. The parameters in each box are kernel size, layer type and the number of channels (the number of kernels). For example, "3x3, Conv, 32" means a convolutional layer with 3x3 kernel size and 32 output channels. s(2) means the stride of slide window is 2. "Fc, 4096" means the number of neurons in this Fc layer is 4096. "BN" and "Tanh" represents batch normalization layer and activation function respectively. The "Size 25x72" represents the output dimension of the block, which is the shape of feature maps.

The difference matrix is denoted as

$$M_{D_2Q} = \begin{bmatrix} D_2Q_{1,1} & & D_2Q_{1,j} & & D_2Q_{1,380} \\ & \ddots & & \ddots & \\ D_2Q_{m,1} & & D_2Q_{m,j} & & D_2Q_{m,380} \\ & \ddots & & \ddots & \\ D_2Q_{198,1} & & D_2Q_{198,j} & & D_2Q_{198,380} \end{bmatrix} \tag{5}$$

$$D_2Q_{m,j} = 2 \times Q_{i+1,j} - Q_{i,j} - Q_{i+2,j} \tag{6}$$

where the variable m and j is the row and col of the matrix separately. $m \in \{1, 2, ..., 198\}$ and $j \in \{1, 2, ..., 380\}$.

Table 1: The percentages (%) of modified points in QMDCT coefficients matrix algorithm for each pre-processing methods (EECS)

Bitrate	W	Origin	Δ_r^1	Δ_r^2	Δ_c^1	Δ_2^2
128	2	8.75	16.52	**23.68**	11.70	16.89
	3	5.14	9.93	**14.46**	6.94	10.03
	4	3.36	6.57	**9.65**	4.59	6.66
	5	2.42	4.77	**7.04**	3.32	4.81
	6	1.99	3.93	**5.81**	2.74	3.95
320	2	9.53	17.81	**25.42**	12.32	18.55
	3	6.08	11.61	**16.84**	7.93	12.03
	4	3.89	7.55	**11.06**	5.15	7.81
	5	3.24	6.29	**9.25**	4.26	6.48
	6	2.58	5.04	**7.43**	3.40	5.20

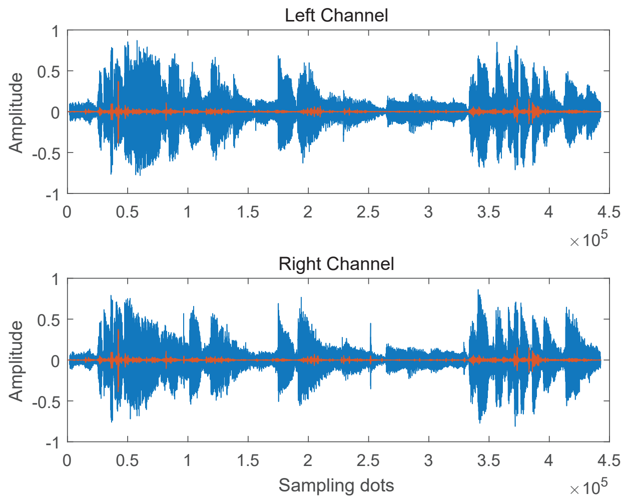


Figure 6: The waveform in the time domain (EECS, Bitrate=128kbps, W=2, and the blue line represents the cover signal, the red line represents the signal introduced by the data hiding method).

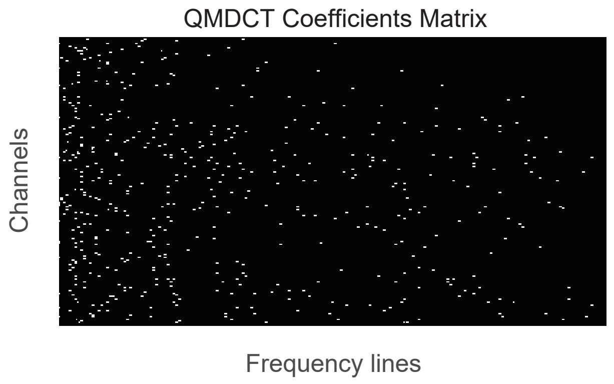
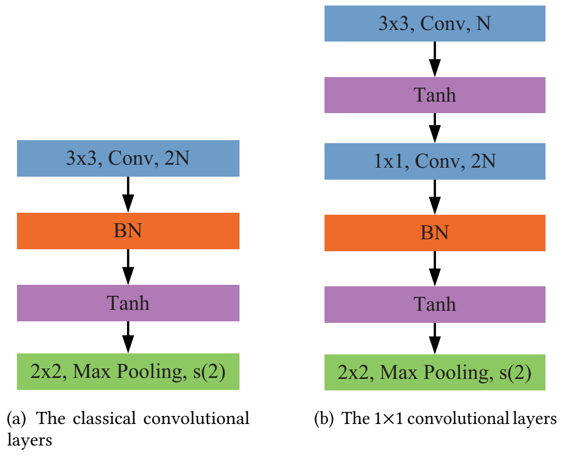


Figure 7: The diagram of the QMDCT coefficients matrix (EECS, Bitrate=128kbps, W=2, and the "white dot" represents the the signal introduced by the data hiding method).



(a) The classical convolutional layers

(b) The 1×1 convolutional layers

Figure 8: The structure of different block convolutional layers.

3.3 Convolutional Layer

The convolutional layer (Conv layer) is the main component in CNN. In our proposed network, convolutional kernels of two sizes, 3×3 kernel and 1×1 kernel, are used. The convolutional layer with the 3×3 kernel is mainly used to capture the features of input data, and the convolutional layer with the 1×1 kernel is applied for the interaction and information integration across channels, as well as the reduction of network parameters. We mainly discuss the function of the 1×1 convolutional kernel in this part. The structure of our block convolutional layers and the classical convolutional layers are shown in Figure 8.

Interaction and information integration across channels. The 1×1 convolutional kernel is valued in [12]. This kernel ignores the relationship between data points and other peripheral points. It is a linear combination of different feature maps so that the information can be integrated across channels.

Reduction of the network parameters. Another function of 1×1 convolutional kernel is to decrease the danger of overfitting. Suppose that the input channels of the 3×3 convolutional layer are N and the output channels are $2N$, the number of introduced parameters (including weights and biases) is

$$3 \times 3 \times N \times 2N + 2N = 18N^2 + 2N \tag{7}$$

If the 1×1 convolutional layer is applied, the number of introduced parameters is:

$$3 \times 3 \times N \times N + N + 1 \times 1 \times N \times 2N + 2N = 11N^2 + 3N \tag{8}$$

In this way, the number of parameters is reduced by $7N^2 - N$. When $N = 64$, the reduction of parameters is 28608.

3.4 Pooling Layer

The pooling layer is commonly used for the reduction of feature dimensions to retain main properties of input data and decrease parameters. The pooling layer can be regarded as a kind of fixed convolutional layers and it is mainly divided into two categories of max pooling and average pooling. The output of max pooling is the maximum value of the sliding window, which tends to retain the texture information. The output of average pooling is the average value of the sliding window that retains the background information. Besides, some experiments show that the convolutional layer with stride 2 or more also can be applied for subsampling. The feature extraction and subsampling are achieved at the same time. Subsampling via convolutional layer can be seen as an adaptive pooling method, but it needs more computational cost. In consideration of characteristics of steganography and complexity of models, only max pooling layer is used in our proposed network.

3.5 Activation Function

Due to the limited expression of the linear model, we introduce nonlinear factors through the activation function. For activation

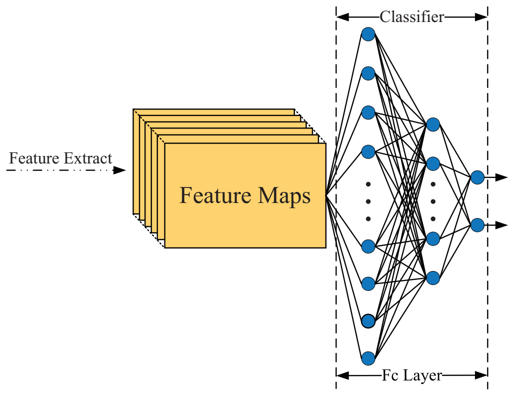


Figure 9: The structure of fully connected layer.

function, ReLu [17] is most commonly used, which can accelerate training. However, in our audio steganalysis task, Tanh is selected as the activation function given the finite range of Tanh.

3.6 Batch Normalization Layer

The Batch Normalization layer (BN layer) [9] is an effective trick widely used in CNNs, which can be placed at any position of the network, but before activation layer generally. The acceleration of convergence and increment of detection accuracy is achieved via the BN layer. The dependence on initialization method is weakened via the introduction of the BN layer. Besides, we decrease the danger of overfitting through BN layer instead of the dropout layer in our network. However, redundant BN layers will reduce the final accuracy.

3.7 Fully Connected Layer

The fully connected layer (Fc layer) acts as a role of "classifier" in the network. If the convolutional layer, the pooling layer and the activation function are used to map the input data into the hidden layer feature space, the Fc layer plays the role of mapping the features to the label space of samples as shown in Figure 9. The parameters of Fc layer accounts for 80% of the entire network. In our network, three Fc layers with the activation function Tanh are connected in cascade.

4 EXPERIMENTS

4.1 Experimental Setup

To evaluate the performance of our proposed network, a dataset which consists of 22671 stereo WAV audio clips with a sampling rate of 44.1kHz and duration of 10s is constructed. The WAV audios are encoded into MP3 files with the two common bitrates of 128kbps and 320kbps by LAME [8]. Three Huffman code substitution steganography algorithms, HCM-Gao [4], HCM-Yan [21] and EECS [22], are implemented to generate the stego MP3 audio files. For the two HCM algorithms, the secret information is embedded in the audio files during the encoding process at the relative embedding rate (RER) of 0.1, 0.3 and 0.5. The hidden messages in the EECS algorithm is encoded by Syndrome-Trellis Codes (STC), so the relative payload α is used to represent the embedding capacity. In consideration of $\alpha = 1/\text{W}$, we use the variable W, the constraint width of the parity-check matrix, to represent the relative payload instead of α. The secret information is embedded at W of 2, 3, 4 and 5, and the constraint height of parity-check matrix is fixed at 7, so that we get 20000 cover-stego pairs respectively. In every group, 16000 pairs are set for training, and the other 4000 pairs are for validation. The rest 2671 pairs are left for test in order to compare the network with traditional handcrafted features.

We train all networks on an NVIDIA Tesla P100 GPU with 16G graphics memory. For optimization, we use Adam [11] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The network is trained with an initial learning rate of 10^{-3} and we use exponential decay function with a decay rate of 0.9 and decay steps of 5000. The batch size of each iteration is 64 (32 cover/stego pairs) in training stage and 16 (8 cover/stego pairs) in validation stage. The weights of convolutional layers and fully connected layers are initialized via Xavier [6] method. And the initialization of biases is zero. To reduce the danger of overfitting further, we introduce the L2 regularization with the gain of 0.001. Last but not the least, all BN layers are removed at the test stage.

Two contrastive steganalysis features are used to compare with the proposed network, which are shown in experimental result tables as Ren [16] and Jin [10]. The features of these schemes are extracted from the QMDCT coefficients matrix and used to be against steganography algorithms in the compressed domain, such as MP3 and AAC. In [16], the multi-order differential coefficients of Intra and Inter-frame are calculated, then two correlation metrics including Markov transition probability and accumulative neighboring joint density are extracted for steganalysis. The threshold is fixed at 4. And in [10], the first order differential coefficients of Intra and Inter-frame is calculated and the Markov transition probability in row and column are extracted as the feature. The threshold is set to 6. All handcrafted features are trained via SVM with 10-ford validation.

4.2 The Fine-tune of Network Structure

To optimize the performance of the network, we fine-tune the structure appropriately. The type of activation function, the size of convolutional kernels, the number of batch normalization layers and the method of subsampling influence the final detection accuracy. The description, detection accuracy of each network variant and the number of iterations for convergence are shown in Table 2. Here, if the accuracy and loss of the network are almost constant in several consecutive epochs, the network can be regarded as convergence. The convergence speed of every network is not exactly the same, so we just record the final accuracy. In our experiments, it costs more time to train the network at the lower relative payload, which conforms to common sense. All experiments are implemented to detect the EECS algorithm with the bitrate of 128kbps and the relative payload W of 2. The performance of the network is measured according to the accuracy and convergence speed as shown in Table 2. And the detection curves of some networks are shown in Figure 10.

The function of BN layer. As the results shown in Table 2, the accuracy is greatly different whether there is BN layer or not.And the number of BN layers has a great impact on the final detection

Table 2: The description, detection accuracy and convergence iterations of each network variant (EECS, Bitrate=128kbps, W=2, the value of iterations is an approximate number, "-" means the network does not converge)

ID	The description of the modification	Accuracy (%)	Iterations
a	The proposed network	90.39	5000
b	Remove the batch normalization layer in the first group	84.51	4000
c	Remove the batch normalization layers in the first two groups	87.13	4500
d	Remove the batch normalization layers in the first four groups	79.46	12000
e	Remove all batch normalization layers	50.67	-
f	Remove the high pass filter layer	88.87	10000
g	Remove all 1×1 convolutional layers	86.73	7000
h	Average pooling layer is used for subsampling	56.21	-
i	Convolutional layer with stride 2 is used for subsampling	60.75	-
j	Replace the convolutional kernel with 5×5 kernel	90.36	9000
k	Introduce the ABS layer at the top of HPF layer	87.66	7000
l	Introduce the ABS layer at the bottom of HPF layer	88.35	8000
m	Replace all the activation function Tanh with ReLu and introduce the ABS layer at the top of HPF layer	58.27	-
n	Replace all the activation function Tanh with ReLu	51.09	-
o	Deepen the network to 7 blocks	88.54	7500

accuracy. The BN layer is conducive to improve the convergence speed and final accuracy of the network, but the redundant BN layers will decrease the accuracy. Thus, we remove the BN layers in the first three groups according to the results a, b, c in Table 2.

The function of HPF. From Table 2 and Figure 10(b), we can find that the high pass filter makes a great difference. Due to the similarity between cover and stego samples, it is little difficult for the network without HPF to capture some useful features efficiently. Thus, the introduction of the HPF can accelerate the convergence greatly and increase the final detection accuracy, especially at a lower relative payload. The design of HPF is independent of the network, but an effective HPF or data preprocessing method is conducive to the classification of cover and stego.

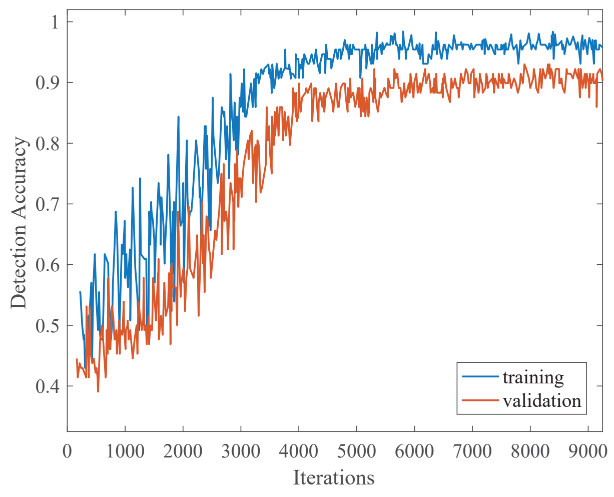
The function of 1×1 convolutional layers. As the result shown in the table 2. The detection accuracy of the network without 1×1 convolutional layers is lower than the proposed network and the iterations for convergence is more. It can be considered that the interaction and information integration across channels of 1×1 kernel as mentioned above contributes to the detection improvements. Besides, fewer parameters make the network more easily trained. Therefore, the introduction of the 1×1 convolutional layers is necessary.

The selection of the subsampling method. To analyze the influence of subsampling method on the final detection accuracy. The average pooling and the convolutional layer with stride 2 are both used for subsampling instead of the max pooling. But the detection accuracy of both networks is worse than the max pooling. It may be because the subtle secret message is hidden in the "texture region", the average pooling and convolutional pooling which tend to capture the content information are not suitable. Thus, the max pooling method is selected for subsampling.

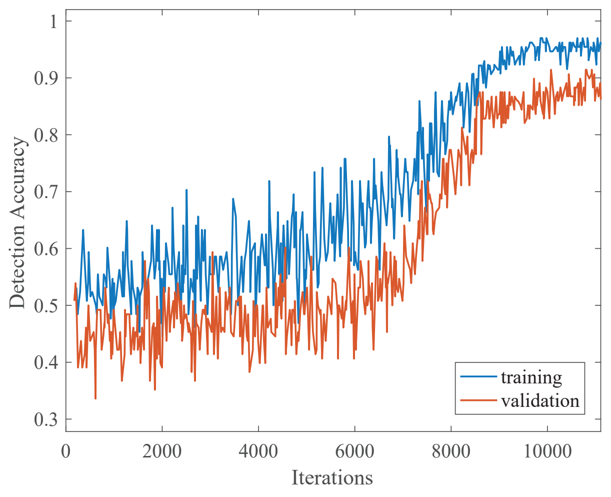
The selection of the convolutional kernel size. In consideration of the truth that every four QMDCT coefficients in count1 region are encoded into one Huffman codeword, the 5×5 convolutional kernel is applied instead of 3×3 kernel. Besides, the LRF of 5×5 kernel is larger, which may capture some properties on a large scale. However, the detection accuracy of two kernels is similar roughly as shown in Table 2 and Figure 10(c). And the parameters of the network with 3×3 kernels are fewer, which means the network with 3×3 kernel can be trained more easily. Hence, it's enough to select 3×3 convolutional kernel in our network.

The selection of activation function and the function of ABS layer. Different activation functions will affect the final detection accuracy. We replace Tanh with ReLu, but the experimental result shows that the activation function Tanh performs far better. The network with ReLu is difficult to converge and the detection accuracy drops sharply compared with Tanh. In order to find the influence of negative value of input data on the network, we introduce the ABS layer and ReLu is still selected as the activation function. The performance of the network is promoted not well. Besides, to obtain the effect of the ABS layer further, we placed the ABS layer at the top and bottom of the HPF layer respectively, but all experimental results are worse than the proposed network. One of the results is shown in Figure 10(d). Therefore, a conclusion can be drawn that the finite of Tanh activation function contributes to the detection and the ABS layer will reduce the difference between cover and stego which is detrimental to classification.

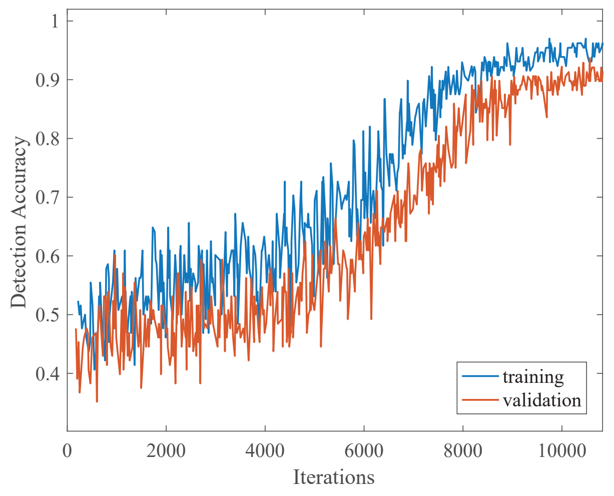
The depth of the network. In general, more features of the input data can be captured by a deeper network. Thus, we attempt to deepen the network to 7 blocks. However, two problems influence the train of deeper networks – overfitting and vanishing gradient that can be improved by ResNet [7] later, which means it is more difficult to train a deeper network. And as the result shown in Table 2. There is no significant improvement in the performance of the network with greater depth. Therefore, it is not advisable to stack or deepen the network simply.



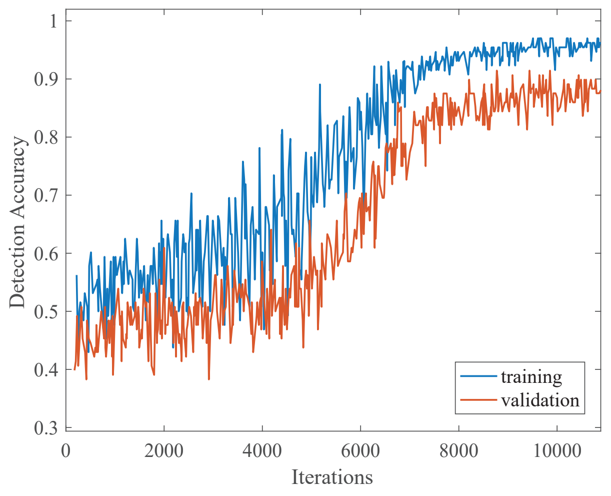
(a) The proposed network.



(b) Network without HPF layer.



(c) Network with 5 × 5 convolutional kernel.



(d) Network with ABS layer at the bottom of HPF.

Figure 10: The detection accuracy curves of some networks.

4.3 Comparison with Handcrafted Features

To assess our scheme comprehensively, two state-of-the-art handcrafted features (Ren [16] and Jin [10]) are compared with the proposed network. The results of each algorithm are shown in Table 3, 4 and 5. The experiments demonstrate that our network outperforms the traditional techniques in several bitrates and relative payloads. In the HCM-Gao algorithm, just specific codeword pairs are selected for substitution, so the embedding capacity is too little to detect. The embedding capacity of the HCM-Yan algorithm is larger and the traditional methods have a good performance for the detection of this algorithm at the RER of 0.5. But the performance is still less than the proposed network. Especially, the EECS algorithm is an adaptive steganography algorithm, which is hard to detect by the traditional method, but it can be well detected by our proposed network. For the three steganography algorithms, the detection accuracy of the proposed network is higher than the handcrafted features. Overall, our network can effectively detect the existing MP3 steganography algorithms in the entropy domain at various bitrates and relative payloads.

Table 3: Detection accuracy (%) of the HCM-Gao algorithm

Bitrate	RER	Ours	Ren [16]	Jin [10]
128	0.1	**70.72**	50.13	50.11
	0.3	**75.18**	51.41	52.34
	0.5	**78.53**	53.75	52.79
320	0.1	**73.83**	50.77	50.85
	0.3	**77.27**	55.18	53.63
	0.5	**80.71**	62.69	59.42

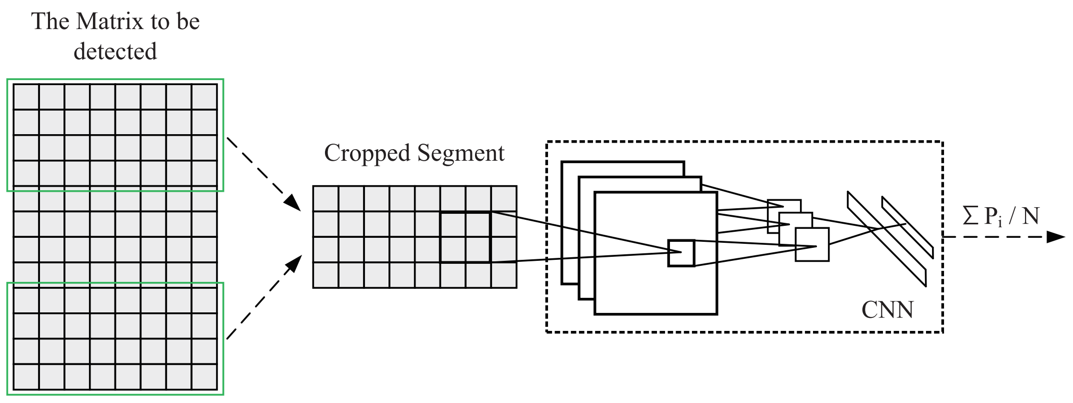


Figure 11: The procedure of steganalyzing audios of varying size.

Table 4: Detection accuracy (%) of the HCM-Yan algorithm

Bitrate	RER	Ours	Ren [16]	Jin [10]
128	0.1	**75.92**	50.94	51.25
	0.3	**81.39**	56.88	60.31
	0.5	**85.88**	76.56	72.50
320	0.1	**79.35**	64.38	54.69
	0.3	**83.09**	66.56	62.50
	0.5	**90.21**	77.81	77.19

Table 5: Detection accuracy (%) of the EECS algorithm

Bitrate	W	Ours	Ren [16]	Jin [10]
128	2	**90.39**	59.42	56.25
	3	**80.17**	53.37	53.81
	4	**67.82**	51.96	51.73
	5	**54.78**	50.56	50.35
320	2	**95.35**	71.38	69.69
	3	**83.09**	55.81	54.56
	4	**72.33**	52.24	52.13
	5	**56.46**	50.10	50.07

4.4 Steganalyzing audios of varying size

For a trained CNN, the input dimension of the Fc layer is invariant, which means the size of input data is fixed. If the size of test audios are not the same with trained audios, the network can't be used directly. In our experiments, the QMDCT coefficients matrix of 200×380 (almost duration of 1.3s) is selected as the minimum data unit. The matrix is variant on the dimension of channels as described in Section 2. Thus, a sliding window with 50% overlap is proposed to steganalyze audios of varying size as shown in Figure 11. Suppose the hidden messages are embedded into the whole audio file, otherwise this scheme is invalid. First, The matrix is cropped into several uniform fragments, and each part is 200×380. If the remaining segment does not satisfy the scale of 200×380, this segment is dropped directly. Then, the cropped segments are put into the network successively, thus the probability of classification can be obtained. Finally, calculate the average value of all results. If the value is more than 0.5, the audio is judged as stego. Otherwise, it is cover. In our experiments, the final detection accuracy is basically equivalent to the small fragments detection.

5 CONCLUSION

In this paper, we propose an effective steganalytic scheme based on CNN to detect MP3 steganography algorithms in the entropy code domain. The performance of the network is optimized via the substitution of activation function, kernel size and other parts which may have the influence on the final accuracy. Experiments demonstrate that our network performs far better than two state-of-the-art handcrafted features. The proposed network can be applied to various steganographic algorithms, bitrates, and relative payloads. In addition, our network can be used for steganalyzing the EECS algorithm, an adaptive MP3 steganographic method, which is hard to detect by conventional handcrafted features. Last but not the least, a sliding window strategy is presented to steganalyze audios of arbitrary size. All of our source code and datasets are now available via GitHub: https://github.com/Charleswyt/tf_audio_steganalysis.

Furthermore, better networks will be designed, which can be used for steganalysis in low embedding capacities effectively, such as a more efficient HPF layer or more excellent structure.

ACKNOWLEDGMENTS

This work was supported by National Key Technology R&D Program under 2016YFB0801003, NSFC under U1636102 and U1536105, and National Key Technology R&D Program under 2016QY15Z2500

REFERENCES

[1] Bolin Chen, Weiqi Luo, and Haodong Li. 2017. Audio Steganalysis with Convolutional Neural Network. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2017, Philadelphia, PA, USA, June 20-22, 2017*. 85–90.

[2] Tomáš Filler, Jan Judas, and Jessica J Fridrich. 2010. Minimizing embedding impact in steganography using trellis-coded quantization. In *Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010, Proceedings*. 754105.

[3] Jessica J Fridrich and Jan Kodovsky. 2012. Rich models for steganalysis of digital images. *IEEE Trans. Information Forensics and Security* 7, 3 (2012), 868–882.

[4] Haiying Gao. 2007. The MP3 steganography algorithm based on Huffman coding. *Acta Scientiarum Naturalium Universitatis Sunyatseni* 4 (2007), 009.

[5] Hamzeh Ghasemzadeh, Mehdi Tajik Khass, and Meisam Khalil Arjmandi. 2016. Audio steganalysis based on reversed psychoacoustic model of human hearing. *Digital signal processing* 51 (2016), 133–141.

[6] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*. 249–256.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778.

[8] Robert Hegemann, Alexander Leidinger, and RogÃĺrio Brito. 1998. LAME. https://sourceforge.net/projects/lame/files/lame/.

[9] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 448–456.

[10] Chao Jin, Rangding Wang, and Diqun Yan. 2017. Steganalysis of MP3Stego with low embedding-rate using Markov feature. *Multimedia Tools and Applications* 76, 5 (2017), 6143–6158.

[11] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980 (2014).

[12] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *CoRR* abs/1312.4400 (2013).

[13] Wei Liu, Shuo-zhong Wang, and Xin-peng Zhang. 2004. Audio watermarking based on partial mp3 encoding. *Acta Scientiarum Naturalium Univ. Sunyatseni* 43, S2 (2004), 26–33.

[14] Fabien Petitcolas. 2002. MP3Stego. http://www.petitcolas.net/steganography/mp3stego/.

[15] Mengyu Qiao, Andrew H Sung, and Qingzhong Liu. 2013. MP3 audio steganalysis. In *Information sciences*, Vol. 231. 123–134.

[16] Yanzhen Ren, Qiaochu Xiong, and Lina Wang. 2017. A Steganalysis Scheme for AAC Audio Based on MDCT Difference Between Intra and Inter Frame. In *Digital Forensics and Watermarking - 16th International Workshop, IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings*. 217–231.

[17] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556 (2014).

[18] Wei Wan, Xianfeng Zhao, Wei Huang, and Rennong Sheng. 2012. Steganalysis of MP3Stego based on Huffman table distribution and recording. *Journal of Graduate University of Chinese Academy of Science* 29, 1 (2012), 118–124.

[19] Andreas Westfeld and Andreas Pfitzmann. 1999. Attacks on steganographic systems. In *Information Hiding, Third International Workshop, IH'99, Dresden, Germany, September 29 - October 1, 1999, Proceedings*. 61–76.

[20] Diqun Yan and Rangding Wang. 2014. Detection of MP3Stego exploiting recompression calibration-based feature. *Multimedia tools and applications* 72, 1 (2014), 865–878.

[21] Diqun Yan, Rangding Wang, and Li-Guang ZHANG. 2011. A high capacity MP3 steganography based on Huffman coding. *Journal of Sichuan University (Natural Science Edition)* 6 (2011), 013.

[22] Kun Yang, Xiaowei Yi, Xianfeng Zhao, and Linna Zhou. 2017. Adaptive MP3 Steganography Using Equal Length Entropy Codes Substitution. In *Digital Forensics and Watermarking - 16th International Workshop, IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings*. 202–216.