Wei KK, Luo WQ, Tan SQ *et al.* CTNet: A convolutional Transformer network for color image steganalysis. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 40(2): 413-427, Mar. 2025. DOI: 10.1007/s11390-023-3006-3, CSTR: 32374.14.s11390-023-3006-3

CTNet: A Convolutional Transformer Network for Color Image Steganalysis

Kang-Kang Wei[1, 2] (魏康康), Wei-Qi Luo[1, 2, *] (骆伟祺), *Senior Member, CCF, IEEE*
Shun-Quan Tan[3] (谭舜泉), *Senior Member, IEEE*, and Ji-Wu Huang[4, 5, 6] (黄继武), *Fellow, IEEE*

[1] *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China*
[2] *Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China*
[3] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*
[4] *Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China*
[5] *Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China*
[6] *Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518129, China*

E-mail: weikk5@mail2.sysu.edu.cn; luoweiqi@mail.sysu.edu.cn; tansq@szu.edu.cn; jwhuang@szu.edu.cn

Received December 3, 2022; accepted October 16, 2023.

Abstract Compared with convolutional neural network (CNN), Transformer can obtain global receptive field features more effectively and has recently achieved great success in natural language processing and computer vision. Due to the particularity of steganography, however, almost all existing steganalytic networks just employ CNN with local receptive fields to detect embedding artifacts. In this paper, we propose a novel convolutional Transformer network for color image steganalysis. Specifically, we firstly obtain various image residuals for each color channel of an input image in the pre-processing module. To capture more comprehensive steganalytic features, the truncated residuals after channel concatenation will pass through a feature extraction module composed of a CNN group and a Transformer group. The CNN group aims to extract local receptive fields features, while the Transformer group with multi-head self-attention as the key tries to extract global steganalytic features. Finally, we employ a global covariance pooling (GCP) and two fully-connected (FC) layers with dropout for classification. Extensive comparative experiments demonstrate that the proposed method can significantly improve the detection performances in color image steganalysis and achieve state-of-the-art results. Although the proposed method is originally designed for color images, it can also obtain competitive results for grayscale images compared with the current best detector. In addition, we provide numerous ablation studies to verify the rationality of the proposed network architecture.

Keywords steganalysis, steganography, convolutional neural network, Transformer, color image

1 Introduction

Image steganography aims to embed secret information into digital cover images in an imperceptible manner. On the contrary, steganalysis tries to detect those stego images with hidden messages according to the embedding artifacts left by steganography. Like the cat-and-mouse game, steganography and steganalysis mutually promote each other. Since most modern steganography methods (e.g., [1–3]) are mainly based on image contents that are difficulty to model, image steganalysis faces great challenges.

Image steganalysis methods include two categories, namely, traditional methods based on handcrafted features[4–7] and modern methods based on deep learning. Taking traditional methods for instance, Fridrich and Kodovsky[4] proposed a rich model method called SRM to construct image noise com-

Regular Paper

The work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61972430 and U19B2022.

*Corresponding Author

ponents by considering many qualitatively different relationships between pixels. Denemark *et al.*[5] proposed a method called maxSRM based on SRM that utilizes the approximate knowledge of the selection channel. Tang *et al.*[6] proposed an adaptive steganalytic method based on embedding probabilities of pixels. In the past few years, many modern convolutional neural network (CNN) based steganalytic methods (e.g., [8–13]) have better performance than traditional methods in terms of detection accuracy. For instance, Ye *et al.*[8] presented a CNN-based framework called YeNet that well optimizes key steps in steganalysis, and this method achieved superior performance compared with SRM[4] and maxSRM[5]. Boroumand *et al.*[9] proposed a deep residual network called SRNet for steganalysis, which provides better detection results for both spatial and JPEG image steganography. Deng *et al.*[10] presented a fast and effective model by designing four CNN groups and introducing global covariance pooling. Zhang *et al.*[11] proposed an efficient framework called ZhuNet based on depth-wise separable convolutions and multi-level pooling.

The above mentioned methods are designed for grayscale images, and they cannot effectively extend to detect those color steganographic methods[14–17] emerged in recent years. Until now, there are several traditional steganalytic methods (e.g., [18–21]) that have been presented for color images. For instance, Goljan *et al.*[18] presented an extension of the SRM[4] called CRMQ1 for steganalysis of color images, and the proposed color rich model features are extremely powerful for detecting color images steganography. Abdulrahman *et al.*[19] proposed a steganalysis method based on the RGB channel feature correlation and an ensemble classifier, which achieved better performance compared with CRMQ1[18]. Recently, several CNN-based steganalyzers have been proposed for color images, and have achieved better performance than traditional methods. For instance, Zeng *et al.*[22] presented a wide-and-shallow steganalysis model called WISERNet, which achieved better detection accuracy compared with some traditional methods. Butora *et al.*[23] investigated the performance of a pretraining CNN on ImageNet when it is applied to image steganalysis. Although the pre-trained EfficientNet can achieve satisfactory results for JPEG images, it is not so effective as SRNet in detecting spatial domain images. Wei *et al.*[24] presented a steganalytic network called UCNet based on color channel representation, which achieves the current best results in the spatial domain.

Different from CNN, Transformer[25] employs the multi-head self-attention (MHSA) mechanism to capture global perception field information effectively. Many recent literatures show that Transformer outperforms CNN on natural language processing (e.g., [26]) and computer vision (e.g., ViT[27] and swin Transformer[28]). Unlike typical classification tasks in computer vision, the two categories (i.e., cover and stego) to be detected in steganalysis are visually imperceptible since recent steganography modifications are content adaptive and are relatively minor (i.e., ± 1). Most steganalytic methods (e.g., SRM-based methods) usually extract statistical artifacts of local regions within an image. To the best of our knowledge, only little steganalytic work[29] using Transformer has been proposed until now. In [29], the designed method directly combines two existing network architectures, that is, ResNet[30] and vision Transformer (ViT)[27], and achieves similar results to SRNet for grayscale image steganalysis.

Many existing steganalytic methods focus primarily on grayscale images, thereby limiting their applicability given the proliferation of color images, especially in the realm of social media. In addition, existing color image steganalytic methods often rely on CNN-based structures. While these methods are generally effective, they tend not to fully exploit the potential of steganalytic features that could enhance detection performance. This underutilization underscores the need for a steganalyzer capable of extracting more comprehensive steganalytic features for color images.

In this paper, we propose a novel steganalytic network called CTNet for color image steganalysis. We carefully design the three modules (i.e., pre-processing, feature extraction, and classification) in CTNet, and provide extensive experiments to demonstrate the superiority of the proposed method. The major contributions of this paper are as follows.

- We develop a unique two-tiered feature extraction module composed of a CNN group, designed to effectively capture local steganalytic features, and an efficient Transformer group, aiming at extracting global steganalytic features. This combination allows us to harness the strengths of both CNN and Transformer in feature extraction.
- In the classification module, we deviate from the conventional usage of global average pooling (GAP) and adopted global covariance pooling (GCP). The

proposed method has proven more efficient in enhancing the classification capability of the extracted steganalytic features. Furthermore, we incorporate two fully-connected (FC) layers with dropout to improve the robustness of the proposed model.

• We provide extensive comparative results to show that the proposed method can achieve the state-of-the-art results for color image steganalysis, and also achieve competitive results for grayscale images simultaneously. Furthermore, we give ablation and additional experiments to verify the rationality and robustness of the proposed method.

The rest of the paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 presents some comparative experimental results and ablation studies, etc. Finally, the conclusions and future work are given in Section 4.

2 Proposed Method

As illustrated in Fig.1, the network architecture of the proposed method includes three modules, that is, pre-processing, feature extraction, and classification, which will be described in the three following subsections. Besides, we will give a brief summary of the similarities and differences between the proposed method and some related ones.

2.1 Pre-Processing Module

The pre-processing module aims to reduce the influence of the image content features on the detection of steganographic modifications signals. To this end, this module includes three steps, i.e., channel separation, high-pass filters and truncated linear unit operation, and channel feature map concatenation.

2.1.1 *Channel Separation*

Let C be a color input image of size $M \times N$, and let C_1, C_2, and C_3 be the pixel maps[①] in the red (R), green (G), and blue (B) channels of C, respectively. Note that color image steganography would modify the three color channels simultaneously during message embedding. Thus, some inherent statistical characteristics among color channels within a cover image would be changed inevitably. To capture this change more effectively, we first separate the RGB channels of the input color image, and then perform the subsequent analysis on each color channel separately.

2.1.2 *High-Pass Filters and Truncated Linear Unit Operation*

In general, the steganographic modifications are relatively weaker than image contents. Therefore,

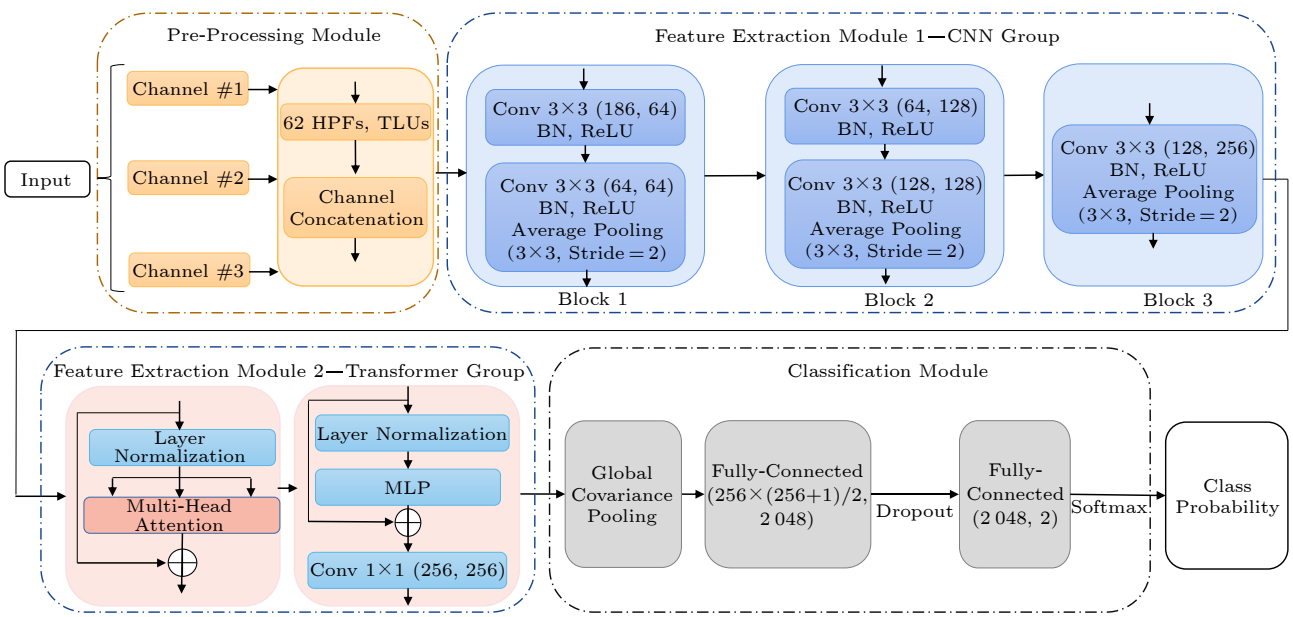

Fig.1. Framework of the proposed CTNet for steganalysis. For each convolutional layer, "Conv $k \times k$ (x, y)" denotes the layer with the filter size $k \times k$ for x input channels and y output channels.

[①]Note that if the input is a grayscale image C, we then duplicate it into three identical channels, that is, $C_1 = C_2 = C_3 = C$. These three channels (C_1, C_2, C_3), are then fed into the pre-processing module of our proposed method.

some high-pass filters (HPFs) are usually applied to transform the input image to the residual domains. Most existing CNN-based steganalyzers also employ these HPFs to suppress the image content in the early stages, and show that this operation can effectively facilitate the final detection performance. Thus, these HPFs are also used in the proposed method. Based on our extensive experiments, we use 62 fixed HPFs of size 5×5 consisting of 30 SRM filters from [4] and 32 Gabor filters from [31].

Note that the use of fixed HPFs can make the network focus on the steganographic artifact rather than the image content, which helps the network converge faster[8, 11]. For the residuals obtained after HPFs, the truncated linear unit (TLU) is then used to limit their dynamic ranges to obtain more valuable steganalytic features.

2.1.3 *Channel Feature Map Concatenation*

For various truncated image residuals, we then perform channel concatenation operation rather than channel summation which is widely used in existing CNN-based steganalyzers[24, 32, 33]. In this way, we can preserve the steganographic artifacts in different channels well. Thus, the feature maps of the three channels are concatenated to 186 (62×3) feature maps, which are input to the subsequent modules.

Note that several settings in the pre-processing module may affect the detection accuracy of the proposed method, such as the different HPFs and the truncation threshold T. These settings would be considered in our ablation studies in Subsection 3.2.

2.2 Feature Extraction Module

The proposed feature extraction module is mainly used to extract sufficient steganalytic features from the concatenation feature map (i.e., a feature map with size $M\times N\times186$) generated by the pre-processing module. Here, in the feature extraction module, the CNN group is set in the early stage for extracting local steganalytic features, and the Transformer group is placed in the later stage for further extracting global steganalytic features. As shown in Fig.1, the feature extraction module consists of a CNN group and a Transformer group, and their structures are as follows.

2.2.1 *CNN Group*

In general, the convolutional layer performs filter sliding on local regions to extract local features. Hence, the proposed CNN group is set in the early stage of the feature extraction module to extract the local steganalytic information of the input residual map. The CNN group consists of three different blocks, that is, Blocks 1, 2, and 3.

- Block 1 contains two identical cascade layers, each consisting of a 3×3 convolution, BN, and ReLU. The input/output channels of the first and second cascade layers are (186, 64) and (64, 64), respectively, and an average pooling layer with a 3×3 convolution and the stride of 2 is appended after the second cascade layer.
- Block 2 also contains two cascade layer structures, where the input/output channels of each cascade layer are (64, 128) and (128, 128), respectively, and an average pooling layer with a 3×3 convolution and the stride of 2 is appended after the second cascade layer.
- Block 3 contains a cascade layer, where the input/output channels are (128, 256). Then an average pooling layer with a 3×3 convolution and the stride of 2 is appended.

2.2.2 *Transformer Group*

After the CNN group, we use a Transformer group to extract global steganalytic features due to its huge receptive fields. In the proposed method, we do not use the previous split-patch and patch embedding operations in vision Transformer[27, 29], and only use the Transformer group with a multi-head self-attention (MHSA) as the key to extract global steganalytic features. In this way, the proposed method needs less training time and memory requirements.

As illustrated in Fig.1, the Transformer group consists of a layer normalization (LN), an MHSA, a multi-layer perception (MLP), and a 1×1 convolutional layer. At the beginning, we reshape the output Out_{CNN} (whose size can be denoted as $(b,\ n,\ h,\ w)$) of the CNN group into a vector of the form $(b,\ n,\ h\times w)$ as the input to the Transformer group, where b is the batch size, n is the number of channels, and h and w are the height and width of the feature maps, respectively. The input Out_{CNN} first passes through the LN layer to normalize the activation values of each layer to obtain the output $\boldsymbol{X}$, and then the MHSA layer immediately can combine the information learned from the different head sections, while the starting input has a residual connection after these two layers get the output T'_ℓ. This process can be expressed as follows:

$$T'_\ell = MHSA(LN(Out_{\text{CNN}})) + Out_{\text{CNN}}.$$

The output of the MHSA is obtained by concatenating the outputs of the two heads after a self-attention (SA) operation. As shown in Fig.2(a), the MHSA first multiplies the output $\boldsymbol{X}$ after the LN with a matrix $\boldsymbol{W}$ to obtain $\boldsymbol{a}^i$, and then $\boldsymbol{a}^i$ is multiplied by three different transformation matrices $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ to obtain three different matrices $\boldsymbol{q}$ (query), $\boldsymbol{k}$ (key), and $\boldsymbol{v}$ (information to be extracted), respectively. Then, the $\boldsymbol{q}^i$ generated by $\boldsymbol{a}^i$ is further multiplied by two transformation matrices to become $\boldsymbol{q}^{i,1}$ and $\boldsymbol{q}^{i,2}$. Similarly, the $\boldsymbol{k}^i$ and $\boldsymbol{v}^i$ are further calculated to obtain $\boldsymbol{k}^{i,1}$, $\boldsymbol{k}^{i,2}$ and $\boldsymbol{v}^{i,1}$, $\boldsymbol{v}^{i,2}$ respectively. Finally, the $\{\boldsymbol{q}^{i,1}, \boldsymbol{q}^{i,2}\}$, $\{\boldsymbol{k}^{i,1}, \boldsymbol{k}^{i,2}\}$, and $\{\boldsymbol{v}^{i,1}, \boldsymbol{v}^{i,2}\}$ are calculated by SA, which can be expressed as:

$$SA(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^{\text{T}}/\sqrt{d_k})\boldsymbol{V},$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are matrices consisting of $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$, respectively, and d is the length of the vector.

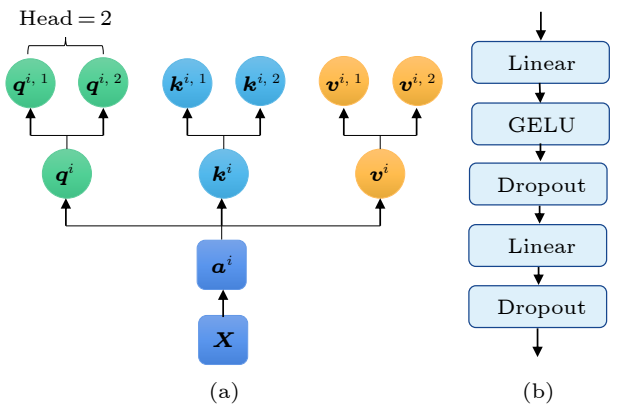

Fig.2. Illustration of MHSA and MLP. (a) Diagram of MHSA (e.g., head = 2). (b) Structure of MLP.

Then, a backbone consisting of an LN layer and an MLP (i.e., it consists of linear, GELU, and dropout layers, and the structure is shown in Fig.2(b)) layer is immediately followed by a residual connection. The output T_ℓ can be expressed as:

$$T_\ell = MLP(LN(T'_\ell)) + T'_\ell.$$

Finally, T_ℓ is reshaped to the form (b, n, h, w) and passes through a 1×1 convolutional layer with 256 input and output channels to obtain the output Out_{tf} of the Transformer group. It can be denoted as:

$$Out_{\text{tf}} = Conv(reshape(T_\ell)).$$

Note that the structure of the feature extraction module and the parameters of the Transformer group can affect the performance of the proposed method. Based on our experimental results (referring to Subsection 3.2.3), the structure of the feature extraction module is the CNN group first and then the Transformer group, where the parameters {*depth, head, mlp_head*} in the Transformer group are {1, 2, 4} and {1, 2, 2} for the color images and the grayscale images, respectively. In addition, the details of the configuration of each module in the proposed framework are shown in Table 1.

2.3 Classification Module

In the classification module, the output of the final layer is the classification probability predicted by "cover" and "stego". First, a global covariance pooling (GCP)[34] layer is used to convert the output feature map of the feature extraction module into a feature vector. Then, two fully-connected (FC) layers are employed, where the second FC layer uses a softmax function to distinguish between the cover and the stego images. Different from existing steganalytic methods, we add a dropout layer between the two FC layers in order to enhance the generalization ability of the proposed model. Based on our experiments, we find that different dropout rates can affect the performances, referring to Subsection 3.2.5.

2.4 Similarities and Differences with Existing Steganalyzers

The major similarities and differences between the proposed method and some related ones (i.e., YeNet[8], SRNet[9], CovNet[10], ZhuNet[11], the method of [29], WISERNet[22], and UCNet[24]) are illustrated in Table 2. From Table 2, we obtain the following observations.

• For the pre-processing module, all filters in SRNet are randomly initialized and learned during training, while YeNet, CovNet, ZhuNet, the method of [29], and WISERNet employ 30 learned or fixed SRM filters. The proposed method uses 62 fixed SRM+Gabor filters and TLU to obtain the truncated residuals as UCNet.

• For the feature extraction module, all steganalyzers employ the CNN structure. Specifically, SRNet uses the most (i.e., 22) convolutional layers, while WISERNet uses the least (i.e., 3) convolutional layers. The method of [29] is a combination of the ResNet[30] and ViT[27] structures. The proposed method consists of a CNN with five convolutional layers and a well designed Transformer for extracting steganalytic features.

Table 1. Detailed Configuration of the Proposed Convolutional Transformer Steganalytic Framework

Module	Name	Input Kernel Size	Output Size
Pre-processing	HPFs	$(5\times5)\times62$	$(256\times256)\times62$
	TLU	$(5\times5)\times62$	$(256\times256)\times62$
	Concatenation	$(5\times5)\times62$	$(256\times256)\times186$
Feature extraction	CNN group B1	$(3\times3)\times186$	$(256\times256)\times64$
		$(3\times3)\times64$	$(256\times256)\times64$
		AvgPool: (3×3), stride=2	$(128\times128)\times64$
	CNN group B2	$(3\times3)\times64$	$(128\times128)\times128$
		$(3\times3)\times128$	$(128\times128)\times128$
		AvgPool: (3×3), stride=2	$(64\times64)\times128$
	CNN group B3	$(3\times3)\times128$	$(64\times64)\times256$
		AvgPool: (3×3), stride=2	$(32\times32)\times256$
	Transformer	LN: $256\times(32\times32)$	$256\times1\,024$
		MHSA: $256\times1\,024$	$256\times1\,024$
		LN: $256\times1\,024$	$256\times1\,024\times4$
		MLP: $256\times1\,024\times4$	$256\times1\,024$
		Conv: $(1\times1)\times256$	$(32\times32)\times256$
Classification	GCP	$(32\times32)\times256$	$256\times(256+1)/2$
	Fully-connected	$256\times(256+1)/2$	2 048
	Dropout	2 048	2 048
	Fully-connected	2 048	2

Note: The input kernel size and output size are both denoted as $(width\times height)\times depth$, where *width* and *height* are the width and height of the feature map, respectively, and *depth* is the number of channels.

Table 2. Similarities and Differences Between Existing Modern Steganalyzers and the Proposed Method

Steganzlyzer	Pre-Processing	Feature Extraction	Classification	Scenario
YeNet[8]	30 learned SRM	CNN (10 layers)	1 FC	Grayscale
SRNet[9]	64 learned random filters	CNN (22 layers)	GAP & 1 FC	Grayscale
CovNet[10]	30 fixed SRM	CNN (10 layers)	GCP & 1 FC	Grayscale
ZhuNet[11]	30 learned SRM	CNN (6 layers)	SPP & 2 FC	Grayscale
Luo *et al.*[29]	30 learned SRM	CNN (10 layers) & ViT	MLP head	Grayscale
WISERNet[22]	30 learned SRM	CNN (3 layers)	GAP & 4 FC	Color
UCNet[24]	62 fixed SRM+Gabor	CNN (11 layers)	GAP & 1 FC	Color
Proposed	62 fixed SRM+Gabor	CNN (5 layers) & Transformer	GCP & 2 FC	Color & grayscale

• For the classification module, SRNet, CovNet, ZhuNet, WISERNet, and UCNet use pooling and FC operation, while YeNet and the method of [29] adopt an FC layer and an MLP head, respectively. The proposed method uses a GCP and two FC layers, and there is a dropout layer between the two FC layers.

• For application scenarios, SRNet is applied to grayscale images in both the spatial domain and the JPEG domain, while UCNet is for color images. WISERNet is designed for color images in the spatial domain, while other methods are applicable to grayscale images. To the best of our knowledge, the proposed model is the only steganalytic network that is valid for both color images and grayscale images in the spatial domain.

3 Experimental Results

In our experiments, 20 000 color images of size 256×256 are randomly selected from the ALASKA II dataset②. These images are divided into three non-overlapping parts: 15 000 for training, 1 000 from validation and the rest 4 000 for testing. Three typical color embedding schemes (i.e., CMD-C[14], ACMP[16], and GINA[17]) combined with two steganographic methods (i.e., S-UNIWARD[35], and HILL[3]) at 0.2, 0.3, and 0.4 bit per channel (bpc), respectively, are considered. For grayscale images, two popular

②https://alaska.utt.fr, Mar. 2025.

databases (i.e., BOSSBase[36] and BOWS2③) are used, and each database includes 10 000 images of size 512×512. As with many existing methods (e.g., [8–10]), all images are firstly resampled to 256×256 using the "imresize" in Matlab with default setting. The training set includes 14 000 images (4 000 are from BOSSBase, and 10 000 are from BOWS2), the validation set includes 1 000 images from BOSSBase, and the testing set includes 5 000 images from BOSSBase. Three steganographic methods (i.e., S-UNIWARD[35], WOW[37], and HILL[3]) with 0.2, 0.3, and 0.4 bit per pixel (bpp), respectively, are considered. To achieve more convincing results, we randomly divide the dataset three times and report the average results in the following experiments. Note that in the tables in the subsequent subsections, the values marked with an asterisk (*) and bolded indicate the best result in the corresponding case. Moreover, the values underlined denote the second best results in the corresponding cases.

During the training stage, the SGD optimizer with a momentum of 0.9 is used. In the convolutional layers, the filter weights are initialized with the He initializer and 5×10^{-4} $L2$ regularization. The batch size is 32 (i.e., 16 cover-stego image pairs). The initial learning rates are 0.01 and 0.02 for the color images and the grayscale images, respectively. The training is conducted for 200 epochs and the learning rate is divided 80, 130, and 170, respectively, during the training, and the best validation snapshot in the last 30 epochs is taken as the result of training. Based on our experiments Subsection 3.2.4, for the lower payloads (i.e., 0.3 and 0.2 bpc/bpp), we employ training from scratch and curriculum learning strategies for the color images and the grayscale images, respectively. The source codes for the proposed model are available on GitHub④ so that readers can repeat the experimental results easily.

3.1 Comparative Studies with Related Methods

In this subsection, we compare the proposed method with related methods for the color images and the grayscale images separately.

- *Evaluation on Color Images.* One traditional detector (i.e., CRMQ1[18]) and four modern CNN-based detectors (i.e., CovNet[10], SRNet[9]⑤, WISERNet[22], and UCNet[24]) are considered. Tables 3, 4, and 5 show the average detection results. From these tables, we observe that the proposed method can achieve the best performances in all cases. Compared with the current best detector (i.e., UCNet), we can achieve 2.85%, 1.94%, and 4.03% average improvements for

Table 3. Detection Accuracy (%) for CTNet and Five Steganalytic Methods for Color Images Under the CMD-C Strategy

Steganalyzer	CMD-C-SUNIWARD (bpc)			CMD-C-HILL (bpc)		
	0.4	0.3	0.2	0.4	0.3	0.2
CRMQ1[18]	73.68	68.83	63.80	71.95	67.29	62.42
CovNet[10]	66.45	62.95	58.30	70.56	66.62	62.85
SRNet[9]	75.96	68.48	66.60	76.20	73.45	68.40
WISERNet[22]	76.15	70.40	63.68	74.96	69.12	60.45
UCNet[24]	80.05	76.85	72.37	82.85	80.75	78.05
Proposed	**82.75***	**80.00***	**74.75***	**86.60***	**83.73***	**80.21***

Table 4. Detection Accuracy (%) for CTNet and Five Steganalytic Methods for Color Images Under the ACMP Strategy

Steganalyzer	ACMP-SUNIWARD (bpc)			ACMP-HILL (bpc)		
	0.4	0.3	0.2	0.4	0.3	0.2
CRMQ1[18]	75.42	71.96	66.31	73.61	69.37	63.78
CovNet[10]	70.55	66.75	62.55	78.49	71.33	64.80
SRNet[9]	83.20	78.85	68.45	86.83	81.45	74.40
WISERNet[22]	77.50	71.05	63.30	79.48	70.23	61.75
UCNet[24]	85.60	81.05	76.50	89.52	84.81	79.55
Proposed	**88.20***	**82.94***	**77.65***	**90.40***	**87.10***	**82.40***

③http://bows2.ec-lille.fr, Mar. 2025.
④https://github.com/revere7/CTNet_Steganalysis, Mar. 2025.
⑤Note that CovNet[10] and SRNet[9] are originally designed for grayscale images. To detect color images, we adjust the input channel of the first convolution layer from 1 to 3 as it did in [24, 32, 33].

Table 5. Detection Accuracy (%) for CTNet and Five Steganalytic Methods for Color Images under the GINA Strategy

Steganalyzer	GINA-SUNIWARD (bpc)			GINA-HILL (bpc)		
	0.4	0.3	0.2	0.4	0.3	0.2
CRMQ1[18]	70.74	67.13	61.80	69.63	65.11	60.63
CovNet[10]	63.90	58.80	52.88	67.62	65.48	61.85
SRNet[9]	70.05	65.20	59.20	76.80	72.90	66.30
WISERNet[22]	73.63	68.25	56.90	71.30	66.55	59.50
UCNet[24]	76.10	73.35	67.30	81.50	77.75	73.46
Proposed	**80.75***	**76.80***	**73.10***	**83.50***	**81.75***	**77.75***

detecting the CMD-C, ACMP, and GINA schemes, respectively, which is a significant improvement in image steganalysis.

• *Evaluation on Grayscale Images.* Four modern CNN-based steganalyzers (i.e., YeNet[8], ZhuNet[11], SRNet[9], and CovNet[10]) are included for a comparative experiment. Table 6 shows the average detection results. From Table 6, we observe that the proposed method can achieve the best performances in most cases, and have quite similar results in another two cases (i.e., S-UNIWARD at 0.2 bpp and HILL at 0.2 bpp) than the current best steganalyzer (i.e., CovNet). Compared with SRNet, we can obtain 0.19%, 0.93%, and 0.90% average improvements for detecting the S-UNIWARD, WOW, and HILL methods, respectively.

3.2 Ablation Study

In this subsection, we compare the proposed method with many variants via assigning different settings in the following five model components, including HPFs, truncation thresholds, settings in the feature extraction module, parameters in the Transformer, and dropout rates in the classification module. For simplicity, two color steganographic methods (i.e., CMD-C-HILL and GINA-SUNIWARD at 0.4 bpc) and two grayscale steganographic methods (i.e., S-UNIWARD and WOW at 0.4 bpp) are considered.

3.2.1 *High-Pass Filters*

Most of existing steganalysis methods usually use SRM or Gabor filters to obtain the residuals of the input image. In our experiment, we evaluate the detection accuracy of the proposed method with three different HPF sets, that is, 30 basic filters from SRM[4], 32 Gabor filters from [31], and a hybrid set (i.e., SRM+Gabor). The comparative results are shown in Table 7. From Table 7, we obtain the following two observations. 1) First of all, SRM outperforms Gabor for both the color images and the grayscale images. 2) The proposed method combing SRM and Gabor always obtains the best results. On average, it can achieve improvements of 2.17% and 1.99% compared with SRM and Gabor, respectively.

3.2.2 *Truncation Thresholds*

The truncation process can effectively limit the residuals after HPFs to a small range, which is beneficial to the subsequent steganalytic feature extraction.

Table 6. Detection Accuracy (%) for CTNet and Four Steganalytic Methods for Grayscale Images

Steganalyzer	S-UNIWARD (bpp)			WOW (bpp)			HILL (bpp)		
	0.4	0.3	0.2	0.4	0.3	0.2	0.4	0.3	0.2
YeNet[8]	87.69	83.63	77.78	89.98	87.19	82.22	83.63	80.03	74.63
ZhuNet[11]	84.37	78.70	71.11	89.76	86.55	81.59	81.62	77.61	72.04
SRNet[9]	89.70	85.57	79.56	90.87	88.24	84.05	85.46	81.07	75.73
CovNet[10]	89.88	85.98	**80.05***	91.93	88.93	84.33	85.87	82.07	**77.10***
Proposed	**90.00***	**86.06***	79.75	**92.26***	**89.12***	**84.58***	**86.21***	**82.21***	76.54

Table 7. Detection Accuracy (%) Comparison Using Different HPF Sets in CTNet

Filter	Color		Grayscale	
	CMDC-HILL	GINA-SUNIWARD	S-UNIWARD	WOW
SRM	85.30	79.00	89.58	91.40
Gabor	83.35	78.40	86.15	89.45
SRM+Gabor	**86.60***	**80.75***	**90.00***	**92.26***

In this experiment, we will evaluate the proposed model with five different thresholds, $T \in \{3, 5, 7, 10, 15\}$. The detection results are shown in Table 8. From Table 8, we observe that the detection accuracy tends to decrease as the thresholds are increased from 5 to 15. When $T = 5$, it achieves the best detection performance for both the color images and the grayscale images. Thus, the proposed method sets $T = 5$ in the pre-processing module.

3.2.3 *Settings in Feature Extraction Module*

In the proposed model, there are two groups (i.e., CNN and Transformer) in the feature extraction module. In this subsection, we will compare the proposed method with three other settings in this module.

- Setting 1: CNN. Just the CNN group is used in the feature extraction module.
- Setting 2: Transformer. Just the Transformer group is used in the feature extraction module. Note that since the sizes of input and output feature maps of the Transformer group are the same, it cannot be used directly in the feature extraction module alone. Therefore, we use ViT[27] with patch embedding and a Transformer encoder as the feature extraction module.
- Setting 3: Transformer-CNN. The Transformer group is placed in front of the CNN group. Note that since the input and output sizes of the Transformer group are consistent, setting it directly in front of the CNN group requires huge memory. Due to this limitation, we assign the Transformer group in front of block 3 in the CNN group, while preserving others unchanged.

The comparative results are shown in Table 9. From Table 9, we can observe that the different combination strategies for the CNN group and the Transformer group can affect the final detection results. Specifically, we obtain two following observations. 1) Just using the CNN group in the feature extraction module achieves good detection accuracy, while just using the Transformer group achieves very poor results (i.e., random guessing). The main reason may be that Transformer usually captures the global information while the steganographic modifications typically change some local statistics within cover, which is relatively easier to be modeled by CNN. 2) Proper combination of CNN and Transformer can further enhance the detection performance of CNN. To balance detection accuracy and efficiency, we firstly put the CNN group to extract local steganalytic features, and then assign the Transformer group to explore the global steganalytic features. By leveraging the strengths of both the CNN group and the Transformer group, our proposed steganalyzer can capture both the local steganalytic feature and the global steganalytic feature, leading to improved detection accuracy. In this way, the proposed method can achieve over 0.56% and 0.18% average improvements compared with the CNN for the color images and the grayscale images, respectively. Note that such an improvement is difficult for the proposed method and it is satisfactory in image steganalysis. However, if the Transformer group is placed in front of the CNN group, the corresponding accuracy would drop around 4.29% and 5.71%, respectively.

3.2.4 *Parameters in Transformer*

The parameter settings in the Transformer group

Table 8. Detection Accuracy (%) Comparison Using Different Truncation Thresholds in CTNet

Threshold	Color		Grayscale	
	CMDC-HILL	GINA-SUNIWARD	S-UNIWARD	WOW
T=3	85.85	80.65	89.81	91.48
T=5	**86.60***	**80.75***	**90.00***	**92.26***
T=7	85.75	80.55	89.79	91.22
T=10	85.50	80.60	89.00	91.12
T=15	85.00	78.70	89.50	90.67

Table 9. Detection Accuracy (%) Comparison Using Different Structural Setting of the Feature Extraction Module in CTNet

Structural Setting	Color		Grayscale	
	CMD-C-HILL	GINA-SUNIWARD	S-UNIWARD	WOW
CNN	85.91	80.30	89.85	92.04
Transformer	50.25	50.10	50.15	50.15
Transformer-CNN	79.28	79.50	84.72	86.12
CNN-Transformer (proposed)	**86.60***	**80.75***	**90.00***	**92.26***

also have an impact on the final detection results. In this experiment, we evaluate the model performances of different parameter combinations {*depth*, *head*, *mlp_head*} in the Transformer group, wherein, *depth* represents the number of the Transformer group, *head* represents the number of heads in the MHSA layer, and *mlp_head* represents the input dimension of a head. Table 10 shows the comparative detection results. From Table 10, we can observe that the best performance is achieved when using {1, 2, 4} and {1, 2, 2} for the color images and the grayscale images, respectively. On average, the proposed model obtains improvements of 0.58% and 0.29% compared with the other combinations for the color images and the grayscale images, respectively.

Table 10. Detection Accuracy (%) Comparison Using Different Parameters of the Transformer Group in CTNet

Parameter Setting	CMD-C-HILL	S-UNIWARD
{1, 2, 2}	85.83	**90.00***
{1, 2, 4}	**86.60***	89.59
{1, 2, 8}	85.85	89.83
{1, 3, 4}	86.37	89.95
{1, 3, 8}	86.14	89.85
{1, 3, 16}	86.38	89.95
{2, 3, 4}	85.80	89.69
{2, 3, 8}	85.50	89.69
{2, 3, 16}	86.15	89.97
{3, 3, 4}	86.00	89.26
{3, 3, 8}	85.95	89.41
{3, 3, 16}	86.23	89.66

Note: The parameter setting indicates {*depth*, *heads*, *dim_head*}.

3.2.5 *Dropout Rates in Classification Module*

During a training stage, a dropout operation can effectively prevent overfitting of the model, and different dropout rates also affect the performance of the network. In our experiment, we evaluate the detection results on three different dropout rates and without a dropout layer in the classification module, that is, 0.3, 0.5, 0.7 and without dropout (denoted w/o). Table 11 shows the comparative detection performance. From Table 11, we observe that the proposed method has the best results using dropout rates of 0.7 and 0.5 for the color images and the grayscale images, respectively. On average, we obtain improvements of 0.49% and 0.18% compared with the other three cases for the color images and the grayscale images, respectively.

3.3 Training from Scratch or Curriculum Learning

For detecting steganography with lower payloads (e.g., 0.3, 0.2 bpc/bpp), existing CNN-based steganalyzers[9, 10] usually use training from scratch or curriculum learning[38] from a higher payload (e.g., 0.4 bpc/bpp). In this experiment, we compare the model performances with the two learning strategies. Note that when curriculum learning is used for a low embedding rate, the corresponding epoch is 150 and its decay epoch is [50, 80, 100]. The learning rates are 0.01 and 0.02 for the color images and the grayscale images, respectively. Table 12 shows the comparative results of the two strategies. From Table 12, we observe that using training from scratch can obtain improvements of 0.32% compared with curriculum learning for the color images. While for grayscale images, using curriculum learning can obtain improvements of

Table 11. Detection Accuracy (%) Comparison Using Different Dropout Rates R of the Classification Module in CTNet

Ratio	Color		Grayscale	
	CMD-C-HILL	GINA-SUNIWARD	S-UNIWARD	WOW
R=0.3	86.29	80.35	89.81	91.74
R=0.5	86.34	80.15	**90.00***	**92.26***
R=0.7	**86.60***	**80.75***	89.82	91.61
w/o R	85.71	80.15	89.84	91.61

Table 12. Detection Accuracy (%) Comparison Using Training from Scratch and Curriculum Learning for Payloads 0.3 and 0.2 bpc/bpp

Strategy	Payload	Color		Grayscale	
		CMD-C-HILL	GINA-SUNIWARD	S-UNIWARD	WOW
Training from scratch	0.4	86.60	80.75	90.00	92.26
	0.3	83.45 √	76.80 √	85.97	88.94
	0.2	80.23 √	73.10 √	79.04	84.50
Curriculum learning	0.3	83.73	77.25	86.06 √	89.12 √
	0.2	80.21	71.10	79.75 √	84.58 √

0.28% compared with training from scratch. In the proposed method, therefore, we use training from scratch and curriculum learning for the color images and the grayscale images, respectively.

3.4 Model Convergence and Execution Time

In this experiment, we compare the convergence performance of the proposed method with related modern detectors. For a fair comparison, all experiments are performed on the same GPU NVIDIA GeForce GTX 2080Ti with 12 GB memory. Our software environment includes the Pytorch 1.10 framework and the Python 2.7 programming language. Fig.3 and Fig.4 show the variation curves of the validation detection accuracy with increasing epochs for related CNN-based methods during the training stage. From Fig.3, we observe that CovNet (200 epochs), WISERNet (343 epochs), and SRNet (571 epochs) are slow to converge in the early training period and SRNet has large fluctuations. The UCNet and the proposed method converge quickly and the proposed method requires less epoch, and has a higher accuracy. From Fig.4, we observe that YeNet (1 000 epochs), SRNet (571 epochs), and CovNet (200 epochs) all have good convergence performance, with YeNet requiring the most epochs. The proposed method has the fastest convergence rate in the early stage, while ZhuNet (400 epochs) has fluctuations and low accuracy.

Besides the convergence performance, the execution time is also considered. Table 13 shows the comparative results of related methods. From Table 13, we observe that the proposed method requires only 24 hours of training to achieve the satisfactory detection accuracy for both color images and grayscale images, which is acceptable compared with other related methods.

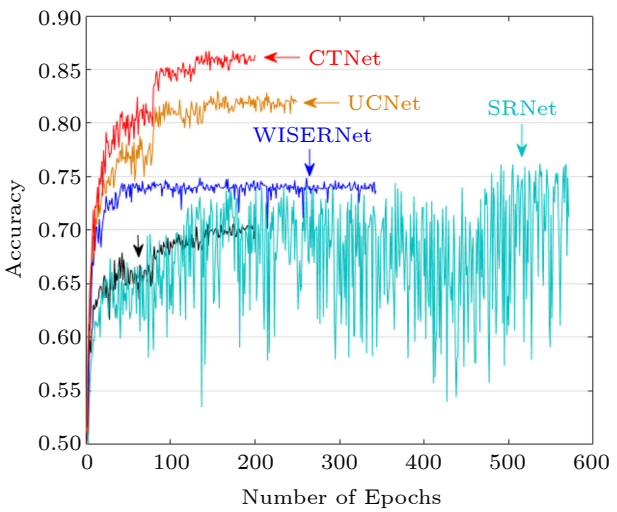

Fig.3. Validation accuracy of different steganalytic methods with increasing epochs for CMD-C-HILL (0.4 bpc) staganography.

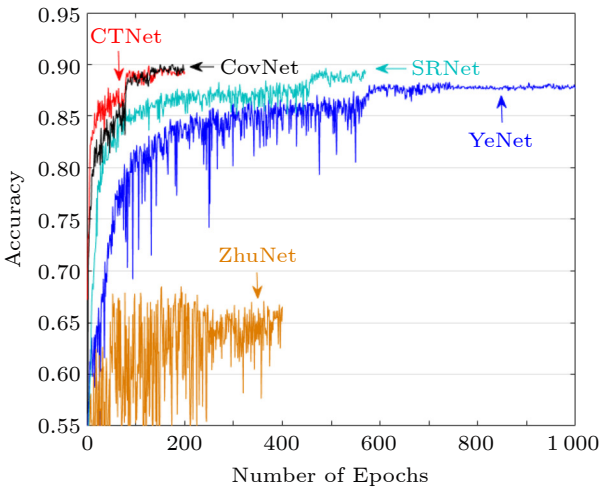

Fig.4. Validation accuracy of different steganalytic methods with increasing epochs for S-UNIWARD (0.4 bpp) staganography.

Table 13. Training Time Comparison for CTNet and Related Methods for CMD-C-HILL (0.4 bpc) and S-UNIWARD (0.4 bpp)

Scenario	Method	Training Time (h)
Color	CovNet[10]	13
	SRNet[9]	51
	WISERNet[22]	12
	UCNet[24]	22
	Proposed	24
Grayscale	YeNet[8]	32
	SRNet[9]	48
	ZhuNet[11]	21
	CovNet[10]	12
	Proposed	24

3.5 Performance on Larger ALASKA II

To further validate the effectiveness of the proposed steganalytic detector, we conduct experiments on a larger color dataset and grayscale dataset (i.e., ALASKA II[④] containing 80 005 images). In our experiment, the training set, the validation set, and the testing set contain 35 000, 5 000, and 40 005 images, respectively. The initial learning rate is 0.01 for color

[④]https://github.com/revere7/CTNet_Steganalysis, Mar. 2025.

images, and the number of epochs and decay epoch of the proposed model are 200 and [80, 130, 170] respectively. In addition, to avoid the model overfitting, we adjust the learning rate of the proposed network for detecting grayscale images to 0.001, and the decay epoch setting is the same as that for color images. The comparative results are shown in Table 14 and Table 15. From the two tables, we observe that the proposed method still outperforms the compared modern steganalyzers on larger color ALASKA II and grayscale ALASKA II. On average, the proposed method has a 2.82% and 0.66% improvement compared to the current best detectors, that is, UCNet for the color images and YeNet for the grayscale images.

Table 14. Detection Accuracy (%) Comparison for CTNet and Related Steganalyzers for CMD-C-HILL (0.4 bpc) on Larger ALASKA II

Method	CMD-C-HILL
CovNet[10]	70.79
SRNet[9]	82.69
WISERNet[22]	76.72
UCNet[24]	84.53
Proposed	**87.35***

Table 15. Detection Accuracy (%) Comparison for CTNet and Related Steganalyzers for S-UNIWARD (0.4 bpp) on Larger ALASKA II

Method	S-UNIWARD
YeNet[8]	68.68
SRNet[9]	66.35
ZhuNet[11]	50.00
CovNet[10]	67.25
Proposed	**69.34***

3.6 Performance on Cover-Source Mismatch

To further investigate the robustness of the proposed method under cover-source mismatch conditions, we conduct corresponding experiments under two color datasets (i.e., ALASKA II and BOSSBose). Similar to previous methods[22, 24], we initially generate 10 000 color images of size 256×256 from raw data in BOSSBase, termed as BOSS-PPG-BIL (abbreviated as BOSSBase in subsequent description). For ALASKA II and BOSSBose, the training set and the test set are divided into 14 000/5 000 and 7 000/2 500, respectively. It means that when the training set is ALASKA II and the test set is ALASKA II or BOSSBase, the corresponding numbers of the training set and the test set are 14 000 and 5 000, respectively. The detection accuracy of the proposed method and other relevant methods under cover-source mismatch are presented in Table 16. From Table 16, we observe that our method not only excels in situations of cover-source mismatch but also consistently surpasses the other methods evaluated. This evidences the high degree of universality in our proposed method.

Table 16. Detection Accuracy (%) of the Proposed Method and Related Methods for Detecting CMD-C-HILL at 0.4 bpc Under Cover-Source Mismatch Conditions

Method	Training Set	Test Set	
		ALASKA II	BOSSBase
CovNet	ALASKA II	70.56	76.17
	BOSSBase	58.60	85.46
SRNet	ALASKA II	76.20	77.48
	BOSSBase	61.88	96.50
WISERNet	ALASKA II	74.96	80.34
	BOSSBase	59.34	91.80
UCNet	ALASKA II	82.85	87.56
	BOSSBase	59.28	96.80
Proposed	ALASKA II	**86.60***	**91.20***
	BOSSBase	**62.04***	**97.50***

3.7 Performance on Color JPEG Images

To verify the performance of the proposed method for the color JPEG images, we conduct experiments using a dataset of 20 000 color JPEG images from ALASKA II. The steganographic methods used are J-UNIWARD[35] and J-MiPOD[39] at quality factor (QF) 75 and 0.4 bpnzac, respectively, and the division of the dataset follows the approach described in Subsection 3. We compare our results with those from J-XuNet[40], SRNet[9], LC-Net[41], and UCNet[24]. As presented in Table 17, the experimental results indicate that our proposed method surpasses the performance of J-XuNet, SRNet, and LC-Net, but does not match up to UCNet. Despite this, the results showcase the efficacy of our proposed steganalyzer in detecting steganography in color JPEG images, thereby demonstrating its competitive advantage.

Table 17. Detection Accuracy (%) of the Proposed Method and Related Methods for Detecting J-UNIWARD and J-MiPOD at QF 75 and 0.4 bpnzac

Method	J-UNIWARD	J-MiPOD
J-XuNet[40]	68.93	74.22
SRNet[9]	89.53	88.95
LC-Net[41]	89.45	88.78
UCNet[24]	**90.02***	**90.20***
Proposed	89.85	89.05

3.8 Discussion

The proposed method achieves better performance for detecting both the color stego images and the grayscale stego images compared with the existing methods, and has a superior performance on ALASKA II as well as in the case of cover-source mismatch. However, the proposed method also has some limitations, primarily in two aspects. Firstly, even though our method exhibits remarkable performance in detecting color stego images in the spatial domain, it falls short of delivering optimal results when applied to color JPEG images. Secondly, while our proposed network proves to be effective, there is room for improvement as the training time has not been fully optimized.

4 Conclusions

In this paper, we proposed a novel steganalytic network which employs the CNN and Transformer structures for color image steganalysis. Extensive comparative results show that the proposed method can significantly outperform modern steganalyzers. Compared with the current best UCNet, the proposed method can achieve 2.85%, 1.94%, and 4.03% average improvements for detecting the CMD-C, ACMP, and GINA schemes, respectively. In addition, it can be applied for grayscale image steganalysis directly, and also achieve competitive results compared with the existing detectors.

In future, several issues about the proposed method are worthy of further study. For instance, it may focus on refining the detection of steganography in color JPEG images and improving the computational efficiency of the network training process. Furthermore, many existing steganalyzers are designed for small size images. To have better practical implications, future steganalyzers should be specifically tailored for larger-sized images.

Conflicts of Interest The authors declare that they have no conflict of interest.

References

[1] Filler T, Fridrich J. Gibbs construction in steganography. *IEEE Trans. Information Forensics and Security*, 2010, 5(4): 705–720. DOI: 10.1109/TIFS.2010.2077629.

[2] Luo W, Huang F, Huang J. Edge adaptive image steganography based on LSB matching revisited. *IEEE Trans. Information Forensics and Security*, 2010, 5(2): 201–214. DOI: 10.1109/TIFS.2010.2041812.

[3] Li B, Wang M, Huang J, Li X. A new cost function for spatial image steganography. In *Proc. the 2014 IEEE International Conference on Image Processing*, Oct. 2014, pp.4206–4210. DOI: 10.1109/ICIP.2014.7025854.

[4] Fridrich J, Kodovsky J. Rich models for steganalysis of digital images. *IEEE Trans. Information Forensics and Security*, 2012, 7(3): 868–882. DOI: 10.1109/TIFS.2012.2190402.

[5] Denemark T, Sedighi V, Holub V, Cogranne R, Fridrich J. Selection-channel-aware rich model for steganalysis of digital images. In *Proc. the 2014 IEEE International Workshop on Information Forensics and Security*, Dec. 2014, pp.48–53. DOI: 10.1109/WIFS.2014.7084302.

[6] Tang W, Li H, Luo W, Huang J. Adaptive steganalysis based on embedding probabilities of pixels. *IEEE Trans. Information Forensics and Security*, 2016, 11(4): 734–745. DOI: 10.1109/TIFS.2015.2507159.

[7] Xia C, Guan Q, Zhao X, Xu Z, Ma Y. Improving GFR steganalysis features by using Gabor symmetry and weighted histograms. In *Proc. the 5th ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2017, pp.55–66. DOI: 10.1145/3082031.3083243.

[8] Ye J, Ni J, Yi Y. Deep learning hierarchical representations for image steganalysis. *IEEE Trans. Information Forensics and Security*, 2017, 12(11): 2545–2557. DOI: 10.1109/TIFS.2017.2710946.

[9] Boroumand M, Chen M, Fridrich J. Deep residual network for steganalysis of digital images. *IEEE Trans. Information Forensics and Security*, 2019, 14(5): 1181–1193. DOI: 10.1109/TIFS.2018.2871749.

[10] Deng X, Chen B, Luo W, Luo D. Fast and effective global covariance pooling network for image steganalysis. In *Proc. the 2019 ACM Workshop on Information Hiding and Multimedia Security*, Jul. 2019, pp.230–234. DOI: 10.1145/3335203.3335739.

[11] Zhang R, Zhu F, Liu J, Liu G. Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. *IEEE Trans. Information Forensics and Security*, 2020, 15: 1138–1150. DOI: 10.1109/TIFS.2019.2936913.

[12] Deng X Q, Chen B L, Luo W Q, Luo D. Universal image steganalysis based on convolutional neural network with global covariance pooling. *Journal of Computer Science and Technology*, 2022, 37(5): 1134–1145. DOI: 10.1007/s11390-021-0572-0.

[13] Wei K, Luo W, Liu M, Ye M. Residual guided coordinate attention for selection channel aware image steganalysis. *Multimedia Systems*, 2023, 29(4): 2125–2135. DOI: 10.1007/s00530-023-01094-x.

[14] Tang W, Li B, Luo W, Huang J. Clustering steganographic modification directions for color components. *IEEE Signal Processing Letters*, 2016, 23(2): 197–201. DOI: 10.1109/LSP.2015.2504583.

[15] Qin X, Li B, Tan S, Zeng J. A novel steganography for

spatial color images based on pixel vector cost. *IEEE Access*, 2019, 7: 8834–8846. DOI: 10.1109/ACCESS.2019.2891316.

[16] Liao X, Yu Y, Li B, Li Z, Qin Z. A new payload partition strategy in color image steganography. *IEEE Trans. Circuits and Systems for Video Technology*, 2020, 30(3): 685–696. DOI: 10.1109/TCSVT.2019.2896270.

[17] Wang Y, Zhang W, Li W, Yu X, Yu N. Non-additive cost functions for color image steganography based on inter-channel correlations and differences. *IEEE Trans. Information Forensics and Security*, 2020, 15: 2081–2095. DOI: 10.1109/TIFS.2019.2956590.

[18] Goljan M, Fridrich J, Cogranne R. Rich model for steganalysis of color images. In *Proc. the 2014 IEEE International Workshop on Information Forensics and Security*, Dec. 2014, pp.185–190. DOI: 10.1109/WIFS.2014.7084325.

[19] Abdulrahman H, Chaumont M, Montesinos P, Magnier B. Color image steganalysis using correlations between RGB channels. In *Proc. the 10th International Conference on Availability, Reliability and Security*, Aug. 2015, pp.448–454. DOI: 10.1109/ARES.2015.44.

[20] Liao X, Chen G, Yin J. Content-adaptive steganalysis for color images. *Security and Communication Networks*, 2016, 9(18): 5756–5763. DOI: 10.1002/sec.1734.

[21] Yang C, Kang Y, Liu F, Song X, Wang J, Luo X. Color image steganalysis based on embedding change probabilities in differential channels. *International Journal of Distributed Sensor Networks*, 2020, 16(5): 1550147720917826. DOI: 10.1177/1550147720917826.

[22] Zeng J, Tan S, Liu G, Li B, Huang J. WISERNet: Wider separate-then-reunion network for steganalysis of color images. *IEEE Trans. Information Forensics and Security*, 2019, 14(10): 2735–2748. DOI: 10.1109/TIFS.2019.2904413.

[23] Butora J, Yousfi Y, Fridrich J. How to pretrain for steganalysis. In *Proc. the 2021 ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2021, pp.143–148. DOI: 10.1145/3437880.3460395.

[24] Wei K, Luo W, Tan S, Huang J. Universal deep network for steganalysis of color image based on channel representation. *IEEE Trans. Information Forensics and Security*, 2022, 17: 3022–3036. DOI: 10.1109/TIFS.2022.3196265.

[25] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, {\L}ukasz and Polosukhin, Illia. Attention is all you need. *Advances in neural information processing systems*, 2017, 30.

[26] Qiu X, Sun T, Xu Y, Shao Y, Dai N, Huang X. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 2020, 63(10): 1872–1897. DOI: 10.1007/s11431-020-1647-3.

[27] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. the 9th International Conference on Learning Representations*, May 2021.

[28] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp.10012–10022. DOI: 10.1109/iccv48922.2021.00986.

[29] Luo G, Wei P, Zhu S, Zhang X, Qian Z, Li S. Image steganalysis with convolutional vision transformer. In *Proc. the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2022, pp.3089–3093. DOI: 10.1109/ICASSP43922.2022.9747091.

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.770–778. DOI: 10.1109/cvpr.2016.90.

[31] Song X, Liu F, Yang C, Luo X, Zhang Y. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In *Proc. the 3rd ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2015, pp.15–23. DOI: 10.1145/2756601.2756608.

[32] Yousfi Y, Butora J, Fridrich J, Giboulot Q. Breaking ALASKA: Color separation for steganalysis in JPEG domain. In *Proc. the 2019 ACM Workshop on Information Hiding and Multimedia Security*, Jul. 2019, pp.138–149. DOI: 10.1145/3335203.3335727.

[33] Yousfi Y, Butora J, Khvedchenya E, Fridrich J. ImageNet pre-trained CNNs for JPEG steganalysis. In *Proc. the 2020 IEEE International Workshop on Information Forensics and Security*, Dec. 2020, pp.1–6. DOI: 10.1109/WIFS49906.2020.9360897.

[34] Li P, Xie J, Wang Q, Gao Z. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.947–955. DOI: 10.1109/cvpr.2018.00105.

[35] Holub V, Fridrich J, Denemark T. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014, 2014: 1–13. DOI: 10.1186/1687-417X-2014-1.

[36] Bas P, Filler T, Pevnỳ T. "Break our steganographic system": The ins and outs of organizing BOSS. In *Proc. the 13th International Conference on Information Hiding*, May 2011, pp.59–70. DOI: 10.1007/978-3-642-24178-9_5.

[37] Holub V, Fridrich J. Designing steganographic distortion using directional filters. In *Proc. the 2012 IEEE International Workshop on Information Forensics and Security*, Dec. 2012, pp.234–239. DOI: 10.1109/WIFS.2012.6412655.

[38] Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In *Proc. the 26th Annual International Conference on Machine Learning*, Jun. 2009, pp.41–48. DOI: 10.1145/1553374.1553380.

[39] Cogranne R, Giboulot E, Bas P. Efficient steganography in JPEG images by minimizing performance of optimal detector. *IEEE Trans. Information Forensics and Security*, 2022, 17: 1328–1343. DOI: 10.1109/TIFS.2021.3111713.

[40] Xu G. Deep convolutional neural network to detect J-UNIWARD. In *Proc. the 5th ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2017, pp.67–73. DOI: 10.1145/3082031.3083236.

[41] Huang J, Ni J, Wan L, Yan J. A customized convolutional neural network with low model complexity for JPEG steganalysis. In *Proc. the 2019 ACM Workshop on Information Hiding and Multimedia Security*, Jul. 2019, pp.198–203. DOI: 10.1145/3335203.3335734.

Kang-Kang Wei received his M.E. degree in computer technology from Jiangxi University of Finance and Economics, Nanchang, in 2020. He is currently a Ph.D. candidate of School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou. His research interests include image steganalysis, steganography, and deep learning.

Wei-Qi Luo received his Ph.D. degree in computer science and technology from Sun Yat-sen University, Guangzhou, in 2008. He is currently a professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, and also a researcher with the Guangdong Key Laboratory of Information Security Technology, Guangzhou. His current research interests include digital multimedia forensics, steganography, and steganalysis.

Shun-Quan Tan received his B.S. degree in computational mathematics and applied software and his Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, in 2002 and 2007, respectively. In 2007, he joined the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, where he is currently an associate professor. He is also the vice director of the Shenzhen Key Laboratory of Media Security. His current research interests include multimedia security, multimedia forensics, and machine learning.

Ji-Wu Huang received his B.S. degree from Xidian University, Xi'an, in 1982, his M.S. degree from Tsinghua University, Beijing, in 1987, and his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He is currently a professor with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen. His current research interests include multimedia forensics and security.