JPEGCNN: A Transform Domain Steganalysis Model Based on Convolutional Neural Network

Lin Gan[1], Jingjie Chen[1], Yuling Chen[2(✉)], Zhujun Jin[1], and Wenxi Han[1]

[1] School of Cyberspace Security,
Beijing University of Posts and Telecommunications, Beijing, China
[2] State Key Laboratory of Public Big Data, GuiZhou University, Guizhou, China
61997525@qq.com

Abstract. Convolutional Neural Network (CNN) has gained an overwhelming advantage in many fields of pattern recognition. Both excellent data learning ability and automatic feature extraction ability of CNN are urgently needed in image steganalysis. However, the application of CNN in image steganalysis is still in its infancy, especially in the field of JPEG steganalysis. In this paper, a steganalysis model based on CNN in gray image transform domain is proposed, which is called JPEGCNN. At the same time, on the basis of JPEGCNN, JPEGCNN is extended to the transform domain of color image by researching and designing different methods of feature extraction. RGBMERGE-JPEGCNN and RGBADD-JPEGCNN are proposed respectively, which make up for the lack of research on steganalysis model based on convolution neural network in the transform domain of color image. Experiments show that JPEGCNN, RGBMERGE-JPEGCNN and RGBADD-JPEGCNN proposed in this paper have good detection ability for steganography algorithm in transform domain.

Keywords: Steganalysis · Convolutional Neural Network · Transform domain

1 Introduction

Steganalysis is a technique to determine whether additional information is hidden in the cover or not by analyzing the statistical characteristics of the carrier, and even to estimate the amount of information embedded in the carrier, and to obtain the content of the hidden information. At present, steganalysis is usually regarded as a two-class problem in the field of steganalysis. The goal is to distinguish between cover and stego. Under this circumstance, the existing methods mainly construct steganalysis detectors through the following two steps: feature extraction and classification. In the feature extraction step, a series of hand-crafted features are extracted from the image to capture the effects of the embedding operation. The effectiveness of steganalysis depends heavily on feature design. However, this work is complicated due to the lack of an accurate natural image model. At present, the most reliable characteristic design paradigm is to calculate the noise residuals and then model the residuals using the conditions of adjacent elements or joint probability distribution. With the increasing

X. Sun et al. (Eds.): ICAIS 2019, LNCS 11635, pp. 566–577, 2019.
https://doi.org/10.1007/978-3-030-24268-8_52

complexity of steganography, the more complex statistical characteristics of images need to be considered in the design process of the steganalysis domain, and the characteristics are gradually moving towards complexity and high dimensionality. For example, the representative steganalysis methods such as SRM (Spatial Rich Model) [1], PSRM (Projection Speciation Rich Model) [2] and other feature dimensions all exceed 10,000 dimensions. In the classification step, classifiers such as SVM or ensemble classifier learn the extracted features and use them for classification. Because the steps of feature extraction and classification are separated, they can't be optimized uniformly, which means that the classifier may not make full use of the useful information in feature extraction.

In order to solve the aforementioned problems, researchers have introduced deep learning theory into steganalysis in recent years. At the same time, according to the domain of steganography information extraction, image steganalysis can be divided into two categories: spatial domain steganalysis and transform domain steganalysis. There are many research results for spatial domain steganalysis, but few for transform domain steganalysis [3]. In 2016, Zeng et al. [4] applied the deep learning framework to transform do-main steganalysis for the first time and proposed a JPEG steganalysis model with three CNN subnetworks, which is called Hybrid CNN (HCNN). The final experimental results show that the accuracy of HCNN is higher than DCTR [5] and PHARM [6]. Zeng et al.'s findings demonstrate the feasibility of applying a deep learning framework for transform domain steganalysis. However, the HCNN is more complex than the previous deep learning based steganalysis model in spatial domain. The reasons are as follows: (1) The network has two additional steps quantitative and truncated. (2) The feature extraction module contains three paths, whereas the previous model only contains one path. In view of the vacancy of applying CNN theory to steganalysis in transform domain and the shortcomings of previous research results, an efficient steganalysis model based on CNN in transform domain is proposed, which is called JPEGCNN. Compared with HCNN, the proposed model has the following advantages: (1) The core of HPF layer in the preprocessing module is simpler. (2) Operation without quantization and truncation. (3) Parallel subnetwork structure is not required. These three points make the network proposed in this paper easier to implement and less computational overhead. In addition, on the basis of JPEGCNN, the general steganalysis model JPEGCNN for gray image transformation domain is extended to color image transformation domain by studying and designing different feature extraction methods. RGBMERGE-JPEGCNN and RGBADD-JPEGCNN are proposed respectively, which make up for the lack of research on steganalysis model based on convolution neural network in color image transformation domain.

The following contents are as follows: The second part mainly describes the structure of JPEGCNN proposed in this paper. The third part is the experiment of JPEGCNN. The fourth and fifth parts elaborate the color extended version of JPEGCNN and related experiments. The last part is conclusion and prospect.

2 JPEGCNN Model

2.1 Preprocessing Module

The preprocessing module usually has a HPF layer. The HPF layer is a special convolutional layer that is at the forefront of the entire network. In this layer, a pre-defined high-pass filter is usually used for filtering. the kernel used by the HPF layer in JPEGCNN is still from the SRM and size is 3×3. The absolute values of the weights in the kernel are distributed between 0 and 1, as follows.

$$\frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \tag{1}$$

The use of the filter described above is based on the following reasons: (1) Pixel residuals, rather than quantized DCT coefficients, are more conducive to steganalysis. The steganography algorithm usually has a small modification range for JPEG quantized coefficients, mostly plus or minus one. Therefore, the influence on the statistical characteristics of JPEG quantized coefficients is not significant. Using a filter similar to HCNN, there is no significant advantage in learning the difference between the two types of sample data from the transform domain. For spatial domain, although the quantization coefficient is only changed by 1 unit, the interference of the steganographic operation on the spatial pixels is further amplified by the amplification step of the quantization step, which is advantageous for the steganographic analysis. In addition, SRM has given a neighborhood pixel correlation model. There is a clear model in the spatial domain analysis as a guide. Pix-el residuals suppress the interference of image content. However, the correlation of neighborhood pixel correlation in the DCT domain is not modeled. The direct analysis of the DCT coefficients is inevitably affected by the image content, which further demonstrates the rationality of analyzing the residuals in the spatial domain. (2) A reasonably sized filter kernel function is more conducive to steganalysis. Research indicates that for complex images, neighborhood pixel correlation will decrease dramatically as the distance between the boundary and the center increases. The 5×5 filter used by GNCNN contains too many pixels with a distance from the center pixel greater than 2, and its ability to capture the neighborhood pixel correlation will decrease. Therefore, JPEGCNN uses a 3×3 size filter.

2.2 Feature Extraction Module

After residual is extracted by the preprocessing module, image is input into the feature extraction module composed of several convolutional layers.

Feature extraction module usually consists of multiple convolutional layers. Convolutional layer's input and output are a set of arrays called feature map, and each convolutional layer typically produces feature map in three steps, convolution, nonlinear activation, and pooling operation. The first step uses k convolutional kernels for filtering, resulting in k new feature maps. $F^n(X)$ denotes the feature map of the nth

layer output. $Weight^n$ denotes the nth layer convolutional kernel and $Bias^n$ denotes the offset. The convolutional layer can be expressed as follow:

$$F^n(X) = P\left(A^n\left(F^{n-1}(X) * Weight^n + Bias^n\right)\right) \tag{2}$$

In the formula, $F^0(X) = X$ represents input data and $A^n(\cdot)$ represents the nonlinear activation function. Nonlinear activation function is applied to each input element. Typical activation functions are sigmod, TanH, and ReLU. $P(\cdot)$ represents pooling operation, including average pooling operation and maximum pooling operation. In general, nonlinear activation function and pooling operation are optional in a specific layer, and a convolutional layer can also be set as untrainable.

In convolutional layer, each output feature map combines the features of multiple input feature maps by convolution. The structure of the convolutional layer involves the concepts of local perception and weight sharing. For local perception, each low-dimensional feature is only calculated from a subset of the inputs, such as the area of the pixel at a given location in the image. The local feature extractor shares the same parameters when applied to different adjacent input positions, which corresponds to the convolution of the image pixel values with the kernel containing the weight parameters. Weight sharing generates a shift-invariant operation, which also reduces the number of free variables, thereby increasing the generalization of the network.

The feature extraction module of JPEGCNN has 5 layers. Each of layer contains 16 convolution kernels of size 3×3 or 5×5. Compared to HCNN, the number of convolutional layers in JPEGCNN has been reduced by nearly half. The first convolutional layer of the feature extraction module uses the Gaussian activation function, and the deeper layers use ReLU activation function. Combined with the experimental results of ICNN, it is known that using ReLU as an activation function at a deeper level of CNN is a better choice. The average pooling operation is used at the end of each convolutional layer. The purpose of the pooling operation is to convert low-level feature representation into more useful feature identifier, thereby saving important information and discarding irrelevant detail. In general, deeper feature representation requires information from progressively larger input regions. The role of pooling is to merge information into a small set of local regions while reducing computational overhead, which is similar to the purpose of quantifying and truncating operations to aggregate useful features and discarding useless information.

2.3 Classification Module

The features extracted by the feature extraction module are input into the classification module for classification. The classification module finally outputs the category of image which is cover or stego. The classification module usually contains several fully-connected layers and a classifier [7].

Ye et al. [8] proposed that the fully-connected layer usually contains more training parameters. When the training set is not large enough, it easily leads to overfitting. One solution is to use only one fully-connected layer during training. There is another way to solve the problem of overfitting that is to use the dropout to regularize the fully-connected layer. When using dropout for training, the neuron output in the

corresponding layer is set to 0 with a certain probability. This technology can improve CNN's generalization ability to some extent.

The classification module has two 128-dimensional fully-connected layers and a softmax classifier. Dropout layers are added after each fully-connected layer to prevent overfitting. The dropout parameter is set to 0.5.

2.4 The Overall Structure

Combining the above three modules, the structure of JPEGCNN is shown in the Fig. 1. Taking the 256 × 256 size image as input. The input size of each layer is marked in the upper left corner.

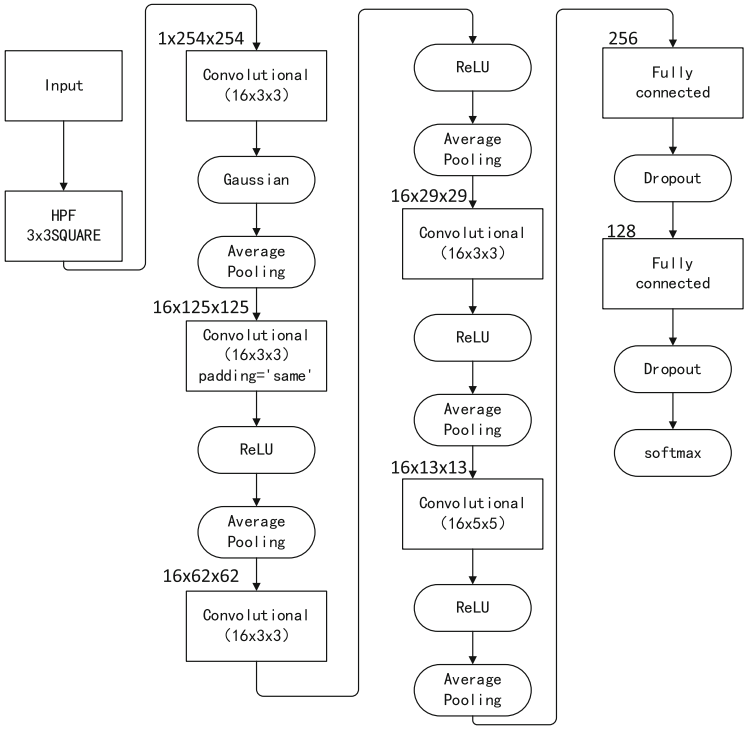

Fig. 1. The overall structure of JPEGCNN

3 Experiments and Analysis of JPEGCNN Model

3.1 The Data Set and Parameter

The data set used in this paper's experiment is the standardized data set BOSSBase 1.01 [9]. BOSSBase 1.01 contains 10,000 images. Each of image's size is 512 × 512. Due to limitation in computing resource, this paper resized the size of BOSSBase 1.01 images to size 256 × 256 in the experiment. 5 transform domain steganography algorithms

were used to evaluate the steganalysis ability of the JPEGCNN. These transform domain steganography algorithms were Jsteg, nsf5, MB1, MB2, and J-UNIWARD [10].

In experiment, steganography algorithms and cover were used to generate the corresponding stego. These 10,000 cover images and 10,000 stego images together formed the data of one experiment (10,000 pairs of cover-stego images). The training set, verification set, and test set used a ratio of 8:1:1, that was, the training set was 8000 pairs of cover-stego images, the verification set was 1000 pairs of cover-stego images, and the test set was 1000 pairs of cover-stego images.

The network learning rate in this paper was mostly set to 0.05 (occasionally 0.005 depending on experimental results). Adadelta [11] gradient descent algorithm was used during training. The size of mini-batch was 64. In the pre-processing module, the HPF layer initialization weight has been described in the second section and was set to non-trainable. The parameter of the Gaussian activation function was $\sigma^2 = 0.2$. In the feature extraction module, the convolutional kernel of each layer was initialized using a "Xavier" [12] initializer. The pooling size of each layer was 3×3 and the step size was 2. The total number of parameters of JPEGCNN is 63,212, and the number of parameters that could be trained is 63,202, while the number of parameters of HCNN is one million. JPEGCNN greatly reduces the number of parameters and improves computational efficiency.

3.2 The Experimental Results

The experimental result is shown in Fig. 2. The horizontal axis represents the steganography algorithms used, and each steganography algorithm uses three embedding rates. The vertical axis represents the accuracy of the JPEGCNN model on the test set. The embedding rate is calculated based on the bpn-zac (bits per nonzero AC DCT). It can be seen that JPEGCNN has good detection ability for five steganography algorithms.

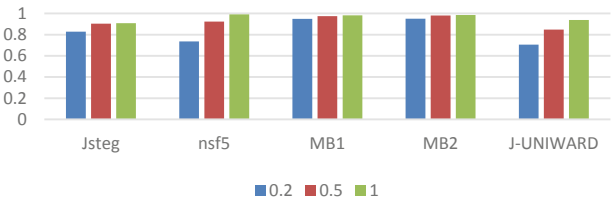

Fig. 2. The performance of JPEGCNN

4 JPEGCNN Color Extension Model

4.1 JPEGCNN Steganalysis Model Based on RGB Three-Channel Merge

According to the two strategies of steganalysis of color image, the most direct way of steganalysis of color image is to transform the three-channel color image directly into a single-channel gray image, and then use the gray image steganalysis model JPEGCNN

for steganalysis. This JPEGCNN model based on RGB three-channel merge is called RGBMERGE-JPEGCNN in this paper.

When transforming color image into gray image, it is necessary to consider the perception ability of human eyes to different colors and give appropriate weights to different channels in order to effectively retain useful information in each channel and reduce information loss. In this paper, the brightness equation is used to fuse the three channels of color image into a single channel gray image. The brightness equation is shown in Eq. 3 below.

$$y = 0.299R + 0.587G + 0.144B \tag{3}$$

Where y denotes the gray value of the pixel, R denotes the value of the red channel, G denotes the value of the green channel, and B denotes the value of the blue channel. The values before R, G and B are the conversion coefficients designed according to the sensitivity of human eyes to different colors of light.

Therefore, in the preprocessing module of RGBMERGE-JPEGCNN, besides HPF layer, an operation module of color channel fusion is needed. On the basis of JPEGCNN, the operation of color channel fusion is added to the preprocessing module, and the summary structure of RGBMERGE-JPEGCNN as shown in Fig. 3 below can be obtained.

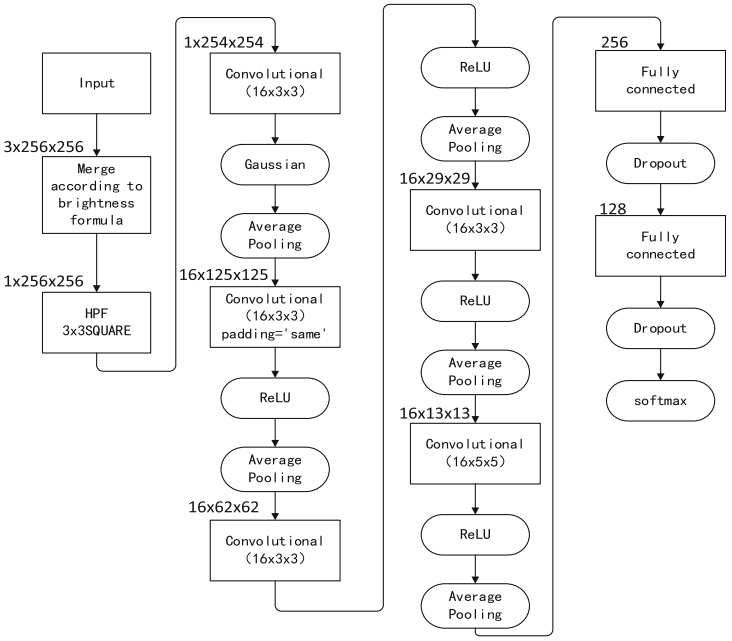

Fig. 3. The overall structure of RGBMERGE-JPEGCNN

4.2 JPEGCNN Steganalysis Model Based on RGB Three-Channel Superposition

R, G and B channels are merged by brightness formula, which may lose some useful information when fusing. In addition to the idea mentioned above, there is another way of thinking. For the three color channels R, G and B, each color channel can be regarded as a gray image. Therefore, the residual can be calculated on three color channels separately, then the residual can be superimposed, and then input into the network for feature extraction. The advantage of this method is to retain all residual information contained in the three color channels.

This JPEGCNN model based on RGB three-channel superposition is called RGBADD-JPEGCNN. In order to stack the residuals on the three color channels, the high-pass filter in the preprocessing module of RGBADD-JPEGCNN is extended to the size of 3x3x3, that is to say, the high-pass filter of JPEGCNN is expanded to a three-channel version. As shown in Formula 4 below.

$$\begin{aligned} \text{Layer 1}\ & \frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \\ \text{Layer 2}\ & \frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \\ \text{Layer 3}\ & \frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \end{aligned} \tag{4}$$

When the color image is inputted into RGBADD-JPEGCNN, the residuals of R, G and B channels can be extracted separately by three-dimensional high-pass filter and three-dimensional convolution operation of the inputted color image, and the results can be superimposed to the output of only one channel. The calculation process is as follows: (1) Two-dimensional convolution between the first layer and R channel, two-dimensional convolution between the second layer and G channel, and two-dimensional convolution between the third layer and B channel. (2) The final three-dimensional convolution results are obtained by adding the convolution result matrices of the first and third channels according to their corresponding positions.

After preprocessing the input image with the extended high-pass filter, the dimension of the data is reduced from three-dimensional to two-dimensional, and the operation can follow the idea of JPEGCNN to deal with the two-dimensional matrix. The summary structure of RGBADD-JPEGCNN is shown in Fig. 4 below.

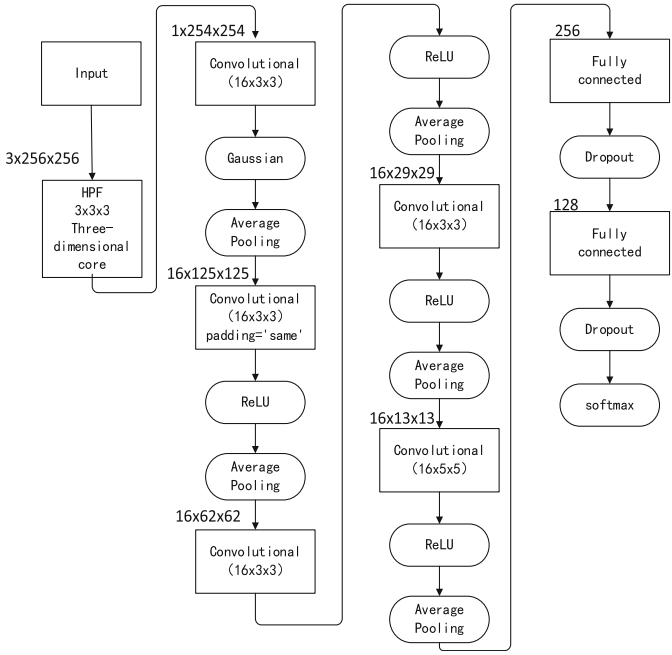

Fig. 4. The overall structure of RGBADD-JPEGCNN

5 Experiments and Analysis of JPEGCNN Color Extension Model

5.1 The Data Set and Parameter

The data set used in the experiment is the color version of the standardized data set BOSSBase v1.0. The content of the image includes people, scenery, architecture, dolls and other contents. The format of the image is cr2. In this paper, 2000 original images in cr2 format are used to make data sets. Firstly, the image in cr2 format is converted into uncompressed JPEG format. Then, the JPEG image is cut into four sub-images at the center. Finally, the four sub-images are scaled to 256 $\times$ 256, which is used as the carrier image of the steganography algorithm in the color transform domain. The carrier image is expanded to 8000 pieces by using enhanced data sets. Five steganalysis algorithms in transform domain are used to evaluate the steganalysis effect of the proposed network. They are: Jsteg, nsf5, MB1, MB2 and color-extended version of J-UNIWARD: Color-Jsteg, Color-nsf5, Color-MB1, Color-MB2 and Color-J-UNIWARD.

5.2 The Experimental Results

The results of RGBMERGE-JPEGCNN model analysis for five color image steganography algorithms in transform domain are shown in Fig. 8. The horizontal axis represents the steganography algorithm used, and each steganography algorithm uses

three embedding rates. The vertical axis represents the accuracy of RGBMERGE-JPEGCNN model on the test set after training.

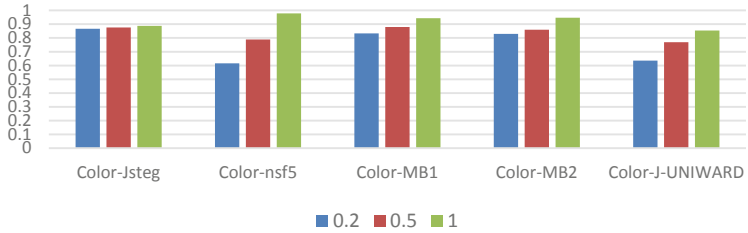

Fig. 5. The performance of RGBMERGE-JPEGCNN

From Fig. 5, we can see that RGBMERGE-JPEGCNN model has good analysis ability for five color image steganography algorithms in transform domain. When the embedding rate is 1.0, RGBMERGE-JPEGCNN model has a good analysis effect on the three algorithms of Color-nsf5, Color-MB1 and Color-MB2, with the accuracy of more than 90%. As the embedding rate decreases, the analysis accuracy of Color-Jstep algorithm decreases slightly, which indicates that RGBMERGE-JPEGCNN model is insensitive to the embedding rate transformation of the algorithm. At the low embedding rate of 0.2, RGBMERGE-JPEGCNN model still maintained more than 80% analysis accuracy for Color-Jsteg, Color-MB1 and Color-MB2. Generally speaking, the strategy of transforming color image into gray image by using brightness formula and steganalysis is feasible, and it has the potential of optimization from the results shown in the figure above.

Using RGBADD-JPEGCNN model, the analysis results of five steganography algorithms in the transform domain of color maps are shown in Fig. 6.

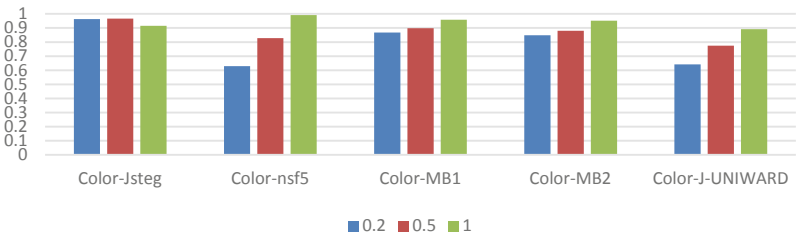

Fig. 6. The performance of RGBADD-JPEGCNN

From Fig. 6, we can see that RGBADD-JPEGCNN model has good analysis ability for five color image steganography algorithms in transform domain. When the embedding rate is 1.0, RGBADD-JPEGCNN model can analyze four steganography

algorithms: Color-Jsteg, Color-nsf5, Color-MB1 and Color-MB2 with an accuracy of more than 90%.

At the same time, we can see that with the decrease of embedding rate, the accuracy of RGBADD-JPEGCNN model for color-Jstep algorithm increases. The reason is that the RGBADD-JPEGCNN model with 0.5 embedding rate is initialized and fine-tuned by the model with 1.0 embedding rate, while the model with 0.2 embedding rate is initialized and fine-tuned by the model with 0.5 embedding rate. It can be roughly considered that the model with 0.5 embedding rate has undergone 400 rounds of training, and the model with 0.2 embedding rate has undergone 600 rounds of training. Although the embedding rate decreases, the difference of image modification degree between color-Jstep algorithm and high embedding rate is not obvious. The increase of training rounds brings more benefits than the loss of embedding rate. The experimental results show that although the embedding rate of steganography algorithm is decreasing, the accuracy of steganalysis is increasing.

From the analysis accuracy, although the RGBADD-JPEGCNN model is slightly superior to the RGBMERGE-JPEGCNN model in terms of channel characteristics. The comparison is shown in Fig. 7 below. The horizontal axis represents the steganography algorithm used, and each steganography algorithm uses three embedding rates. The vertical axis represents the difference between the analysis accuracy of RGBADD-JPEGCNN model and that of RGBMERGE-JPEGCNN model. It is speculated that the use of brightness equation in channel fusion results in the loss of some steganalysis features, while channel superposition preserves the steganalysis features in each channel.

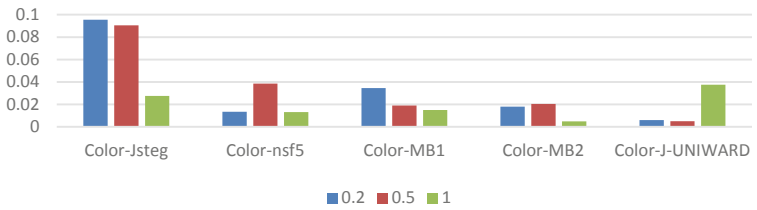

Fig. 7. Comparison of RGBADD-JPEGCNN and RGBMERGE-JPEGCNN

6 Conclusion

In view of the vacancy of applying convolutional neural network to steganalysis in transform domain, this paper proposes a gray image steganalysis model JPEGCNN based on convolutional neural network and two color image steganalysis models RGBMERGE-JPEGCNN and RGBADD-JPEGCNN based on convolutional neural network. In this paper, the related validation and testing are carried out on BOSSBase 1.01 data set. Experiments show that the proposed steganalysis model in transform domain has good analysis ability for steganography algorithm in transform domain.

Future research directions will focus on the following two points: First, the proposed network model still has room to improve the accuracy of Steganalysis for various steganography, especially in the case of low embedding rate. Second, the current generic steganalysis models are often limited to a specific domain, such as gray image spatial domain and color image transformation domain. Steganalysis between different domains may not be compatible. This is closely related to the design of steganalysis features. One of the trends of future general steganalysis research is whether we can design a more abstract steganalysis feature, which can span different domains and make the steganalysis model more general without domain restriction.

Acknowledgments. This work is supported by the National Natural Science Foundation of China grant (U1836205), Major Scientific and Technological Special Project of Guizhou Province (20183001), Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ014), Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ019) and Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (2018BDKFJJ022).

References

1. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **7**, 868–882 (2012)
2. Holub, V., Fridrich, J.: Random projections of residuals for digital image steganalysis. IEEE Trans. Inf. Forensics Secur. **8**, 1996–2006 (2013)
3. Chen, J., Wei, L., Yeung, Y., Xue, Y., Liu, X., Lin, C., Zhang, Y.: Binary image steganalysis based on distortion level co-occurrence matrix. CMC: Comput. Mater. Continua **055**(2), 201–211 (2018)
4. Zeng, J., Tan, S.: Large-scale JPEG steganalysis using hybrid deep-learning framework. IEEE Trans. Inf. Forensics Secur. **13**(5), 1200–1214 (2016)
5. Holub, V., Fridrich, J.: Low-complexity features for JPEG steganalysis using undecimated DCT. IEEE Trans. Inf. Forensics Secur. **10**(2), 219–228 (2015)
6. Holub, V., Fridrich, J.: Phase-aware projection model for steganalysis of JPEG images. In: Proceedings of SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XVII, vol. 9409 (2015)
7. Fang, W., Zhang, F., Sheng, V.S., Ding, Y.: A method for improving CNN-based image recognition using DCGAN. CMC: Comput. Mater. Continua **57**(1), 167–178 (2018)
8. Ye, J., Ni, J., Yi, Y.: Deep learning hierarchical representations for image steganalysis. IEEE Trans. Inf. Forensics Secur. **12**(11), 2545–2557 (2017)
9. Bas, P., Filler, T., Pevný, T.: "Break our steganographic system": the ins and outs of organizing BOSS. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 59–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24178-9_5
10. Qu, Z., Cheng, Z., Wang, X.: Matrix coding-based quantum image steganography algorithm. IEEE Access **7**, 35684–35698 (2019)
11. Zeiler, M.D.: ADADELTA: an adaptive learning rate method arXiv:1212.5701 (2012)
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of Aistats, vol. 9, pp. 249–256 (2016)