 CMC, vol.55, no.2, pp.229-241, 2018

Identifying Materials of Photographic Images and Photorealistic Computer Generated Graphics Based on Deep CNNs

Qi Cui[1, 2, *], Suzanne McIntosh[3] and Huiyu Sun[3]

Abstract: Currently, some photorealistic computer graphics are very similar to photographic images. Photorealistic computer generated graphics can be forged as photographic images, causing serious security problems. The aim of this work is to use a deep neural network to detect photographic images (PI) versus computer generated graphics (CG). In existing approaches, image feature classification is computationally intensive and fails to achieve real-time analysis. This paper presents an effective approach to automatically identify PI and CG based on deep convolutional neural networks (DCNNs). Compared with some existing methods, the proposed method achieves real-time forensic tasks by deepening the network structure. Experimental results show that this approach can effectively identify PI and CG with average detection accuracy of 98%.

Keywords: Image identification, CNN, DNN, DCNNs, computer generated graphics.

1 Introduction

With the development of related algorithms and software, computer-generated images have become more and more realistic. In the background of cloud computing, it is difficult for the naked eye to distinguish between photographic images (PI) and photorealistic computer generated images (CG) [Fu, Shu, Wang et al. (2015)]. Fig. 1 shows example images of photographic images and computer generated graphics. The acquisition of a photographic image (PI) requires photoelectric conversion, interpolation, and post-processing of the sensor to add noise to the image. CG images, on the other hand, are produced by software and have a smoother texture than PI. Due to recent advances in the field of artificial intelligence, algorithms based on deep neural networks (DNN) are more suitable for classification tasks. At the same time, when compared with existing methods, CNN-based algorithms can achieve real-time detection more effectively.

From a legal point of view, the distinction between PI and CG is also very important. For example, someone can convert a CG image into PI for profit, but current PI and CG forensics technology can discern the truth. DNNs can extract image features efficiently and automatically train the classification features. In the field of image forensics, methods

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Ning Liu Road, No. 219, Nanjing, 210044, China.

[2] Jiangsu Engineering Centre of Network Monitoring, Ning Liu Road, No. 219, Nanjing, 210044, China.

[3] Computer Science Department, New York University, New York, NY 10012, USA.

* Corresponding author: Qi Cui. Email: cuiqiloveslife@gmail.com.

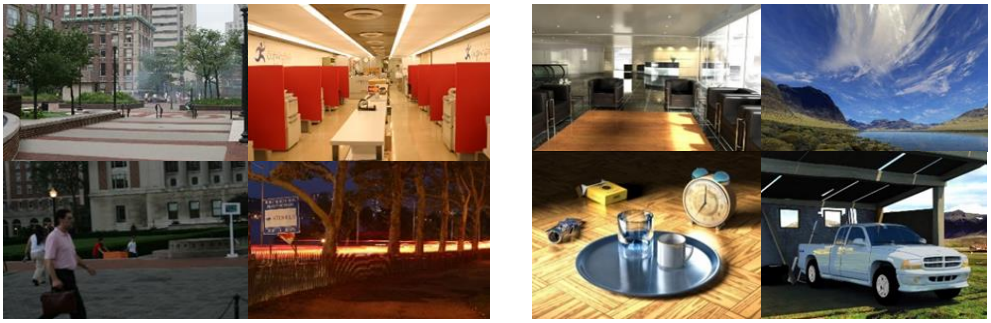
based on convolutional neural networks (CNNs) [Li, Luo and Huang (2015); Chen, Kang, Liu et al. (2015); Ye, Ni and Yi (2017)] are the most popular implementations of DNNs and the most effective for solving these types of image forensics problems.

This paper is structured as follows: Section 2 describes related work in this field. In Section 3, we introduce the proposed 50-layer CNN structure for real-time PI and CG forensics processing. Section 4 describes our experiments and provides an analysis of the results.

2 Related works

In the field of biometric image forensics, computer-generated face and natural facial forensics have been explored. Conotter et al. [Conotter, Bodnari, Boato et al. (2015); Dang-Nguyen, Boato and De Natale (2012)] represent the development of forensics in this area. Mader et al. [Mader, Banks and Farid (2017)] pointed out that human beings can improve recognition accuracy by adding incentives in the feedback process. For example, in the field of near-duplicate image detection and retrieval, Zhou et al. [Zhou, Yang, Chen et al. (2016); Zhou, Wu, Huang et al. (2017); Zhou, Wang, Wu et al. (2017); Cao, Zhou, Sun et al. (2018); Zhou, Mu and Wu. (2018); Zhou, Wu, Yang et al. (2017)] proposed the use of the traditional features of the image to apply to coverless information hiding. Yuan et al. [Yuan, Li, Wu et al. (2017)] proposed a method based on CNNs for fingerprint liveness detection which raised average detection accuracy to 95.43%, proving the effectiveness of CNNs in the classification of complex and highly-similar textured images. Gurusamy et al. [Gurusamy and Subramaniam (2017)] used a machine learning approach for MRI Brain Tumor Classification. In the research of PI and CG forensics, Peng et al. [Peng and Zhou (2014)] used PRNU (Photo Response Non-Uniformity) to achieve average recognition accuracy of 94.29% when the feature dimension was set to 48 on 1,200 natural images and 1,200 computer generated images selected from the Columbia University datasets [Ng, Chang, Hsu et al. (2005)]. Long et al. [Long, Peng and Zhu (2017)] used PI's unique PRNU attribute to improve the average recognition accuracy to 99.83%. Wang et al. [Wang, Li, Shi et al. (2016)] proposed a novel set of features based on Quaternion Wavelet Transform (QWT) for digital image forensics. The corresponding results show that the proposed scheme achieves 18% improvements on the detection accuracy than Farid's scheme and 12% than Ozparlak's scheme.

This work aims to design and implement a method to efficiently identify PI and CG in real-time. Considering that CNNs can describe image features with high performance, this paper will use CNNs to extract image features and train the recognition classification process. CNNs are mainly used to identify image displacement, scaling, and other issues. Convolutional layers in the CNN structure utilize high-dimension vectors extracted from convolution kernels as image features. These high-dimension features are finally sent to the classification layer to complete the model training.



(a) (b)

Figure 1: Example images of PI and CG in columbia photographic images and photorealistic computer graphics dataset (Version 1). (a) PI. (b) CG

3 Proposed approach

As a kind of deep neural network method, CNNs are widely used in the field of image processing. The high-dimensional vectors extracted by CNNs are used as image features, and the network classification layer uses the extracted features to train a model for classification. The local perception strategy and parameter sharing strategy of CNNs can effectively reduce the number of parameters and improve training efficiency. Using the local network connection method separates out all but the closely-connected pixels in the image. By combining these local high-level features, the global features of the image can be obtained. A typical CNN includes a feature extraction layer and a feature mapping layer. The input of each neuron is connected to the local acceptance of the previous layer of the network to extract the local feature. Multiple feature maps in the network represent different planes. All the neurons in the plane share the weights. In this way, the number of network parameters is reduced and the training efficiency is improved. The input image is decomposed by the convolutional layer into feature maps reflecting local dependency [Liu and Yao (1996)]. With CNNs, pooling operations can reduce the association of feature maps to prevent over-fitting. Due to the fact that the high-level feature dimensions in the network can reach upwards of one thousand dimensions, the training data may have a large amount of redundant information which can easily lead to over-fitting. For example, as a common pooling operation, maximum pooling outputs the largest value of the selected area in the feature map, which allows for faster convergence [Scherer, Müller and Behnke (2010)]. The average pooling is added after the convolution layer in order to remove redundant information and prevent the network from getting over-fit due to the small size of the training data. Each convolution layer is activated by the *relu* function, which is expressed as:

$$relu = \max(0, x) \tag{1}$$

In the process of training, we choose the way of learning rate policy as $step$. Suppose the basic learning rate is $base\_lr$. The learning rate changes once after $stepsize$ iterations. At the $iter$th iteration, the learning rate is:

$$lr = base\_lr \times \gamma^{floor(iter/stepsize)} \tag{2}$$

At the end of the network, the *softmax* connection will output the input samples to the probability output of each class. The network outputs the highest probability value as the prediction category. Assuming there are m samples, which are (X_i, Y_i), in a total of L networks. The output of the last layer is $f(X_i)$. Then the loss function of the network is:

$$loss = -\frac{1}{m} \times \sum_{i=1}^{m} Y_i' \log f(X_i) + \lambda \sum_{k=1}^{L} sum(\|W_k\|^2) \tag{3}$$

Where λ is represents the penalty of weight W_k, Y_i is the category of input X_i.

Physical characteristics of the image, including CFA interpolation and pattern noise (mainly PRNU), have been shown to be effective for digital image forensics. PRNU is produced by the image sensor of photographic equipment. It is a kind of noise with uneven optical response and has been widely used in image source recognition [Peng and Zhou (2014)]. In our work, we leverage the known non-homogeneity of PI image noise when compared to CG. In this paper, we describe the use of deep neural networks to extract the features of PI and CG and train them to finish the classification of the two types of images. Similar to the existing PRNU forensic methods [Peng and Zhou (2014); Long, Peng and Zhu (2017); Peng, Zhou, Long et al. (2017)], the input images are first subjected to high-pass filtering during the training process. Then the images are fed into the deep CNNs as the network input. Finally, the network output classification model is developed. The test step uses the model file generated in the previous step. The input image is also first filtered with a high-pass filter and then fed into the network to obtain the probability values belonging to both classes respectively, and the discrimination result is given by the probability value.

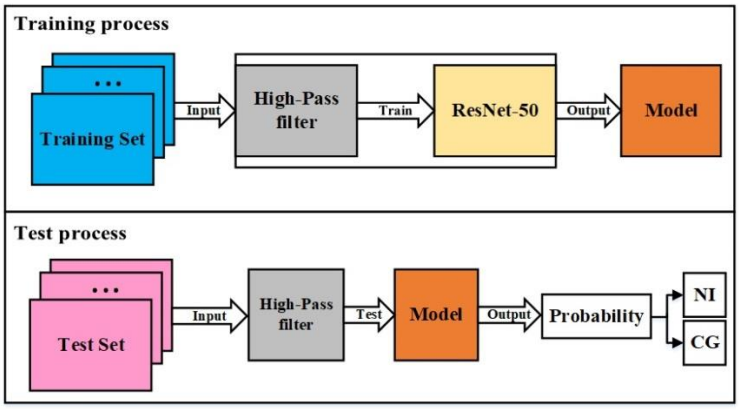


Figure 2: The proposed CNN structure for discriminating between PI and CG

Fig. 2 shows the whole flowchart of PI and CG classification based on the CNN structure proposed in this paper. In this model, each input image is grayed out first and then subjected to high-pass filtering to add high-frequency components. In our classification approach, there are differences in the local correlation between PI and CG. Namely, the noise is amplified by high-pass filtering, and the subsequent convolution layers of the network structure can fully learn the features and use them to classify. CNNs can learn the texture features of both types of images and can use an optimization algorithm, such as gradient

descent, to lower the model loss value when categorizing. With the network and model efficiently identify the input information, we can classify the two images.

(a) (b) (c) (d)

Figure 3: Examples of high-pass filtering operations with PI and CG. (a), (c), respectively, the original PI, CG images. (b), (d), respectively, PI, CG after the high-pass filter operation

Following this pre-processing we execute CNN processing which consists of 50 layers of CNNs and a global average pooling layer. The CNNs are responsible for learning the optimization function and converting each pre-processed input image into a 2048-dimensional vector for classification. All of the convolution layers are followed by the Bach-Normalization (BN) layer and the Rectified Linear Unit (ReLU) layer as the activation functions for each layer. Inspired by Szegedy et al. [Szegedy, Liu, Jia et al. (2015); He, Zhang, Ren et al. (2016)], a short-cut connection was added in the 50-layer convolution structure to prevent gradient vanishing during the training process. Only the residuals were learned by the convolutional structure of the middle layers [He, Zhang, Ren et al. (2016)]. The short-cut connection in this work is shown in detail in Fig. 5. These operations make the depth of the network inconsistent. A network can represent a variety of depth, which makes the network more efficient. The final structure of the network is a fully connected layer and a softmax layer, which converts the feature vector into a posteriori probability for each class. Finally, we make the final prediction by selecting the maximum posteriori probability value for each category label. This article has modified the CNNs network structure with a depth of 9 and 50 for experiments to prove that the deeper CNNs more suitable for this problem. In order to illustrate the effectiveness of the short-cut for this problem, in 4.1, the accuracy of the network model and the loss of training are compared in detail in the case of short-cut.

This paper proves the validity of short-cut connection by modifying CaffeNet and comparing the original structure. CaffeNet was slightly modified in AlexNet [Krizhevsky, Sutskever and Hinton. (2012)], who won the 2012 ImageNet [Deng, Dong, Socher et al. (2009)] Large-Scale Visual Identity Challenge (ILSVRC). The CaffeNet architecture used in this paper is shown on the left in Fig. 4. The network consists of 5 convolutional layers and 3 fully connected layers. The output from the last layer in the experiment was changed from 1000 for the original task to 2, indicating that the goal of this task is to classify PI and CG into two categories. The right side of Fig. 4 shows the network structure of the original CaffeNet after adding the short-cut connection. The solid curve represents the direct short-cut connection and the dashed curve represents the deformed short-cut connection.

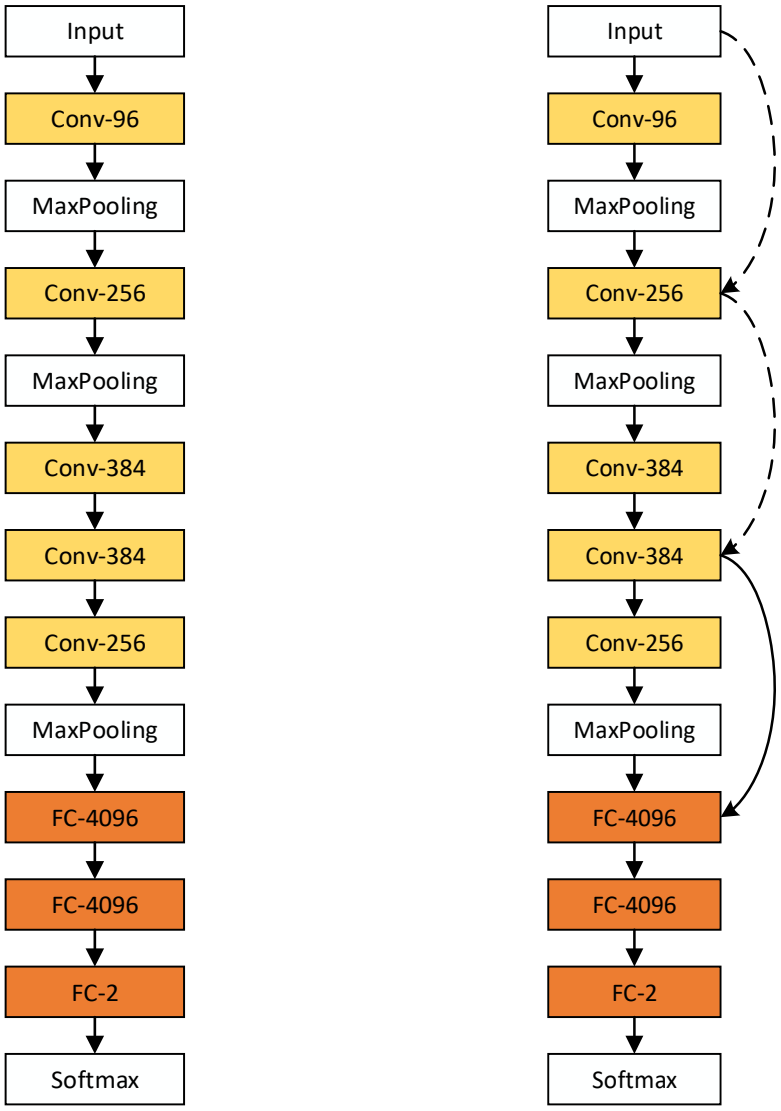
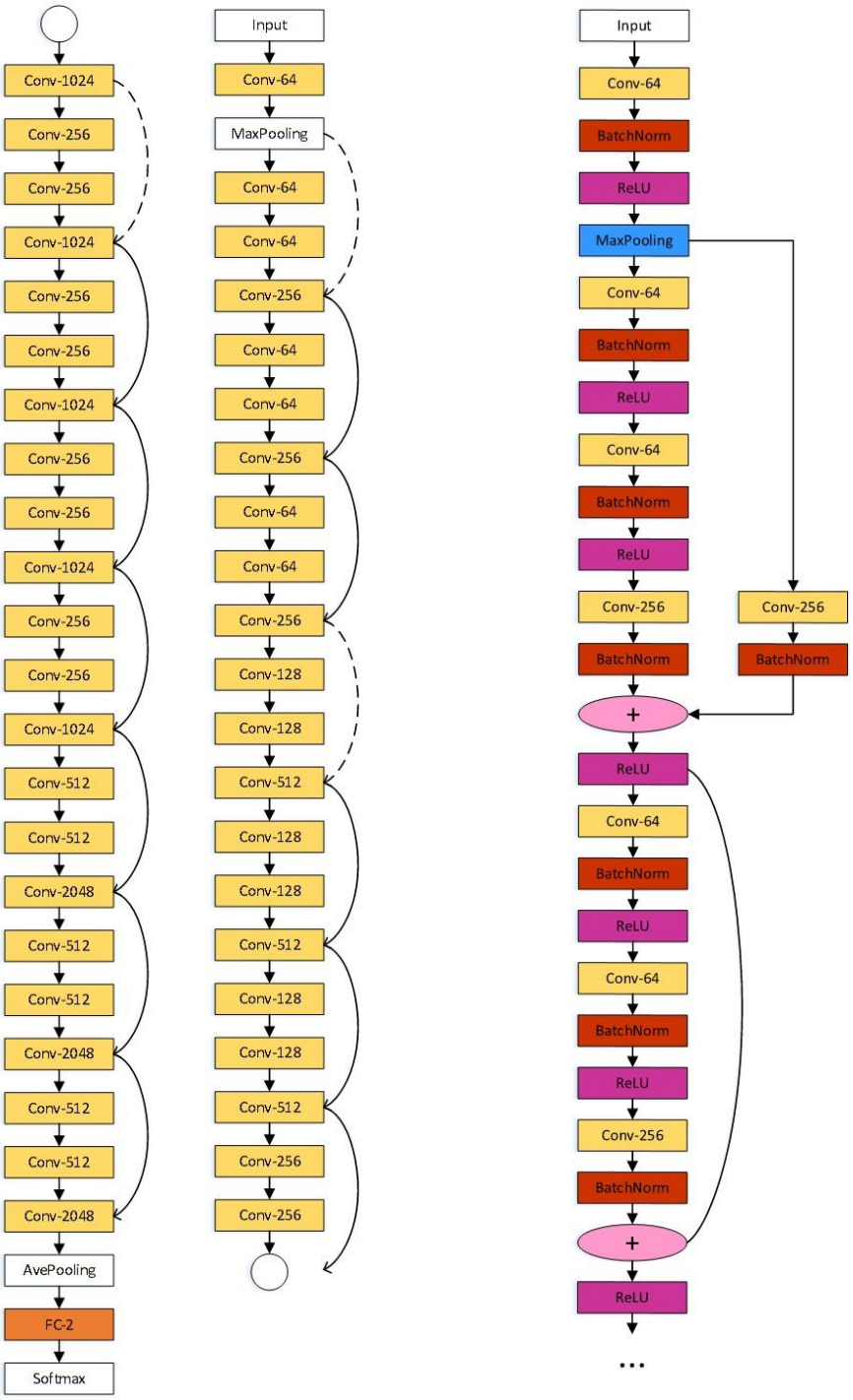


Figure 4: The left part is CaffeNet structure. The right part is CaffeNet with short-cut connection



(a) (b)

Figure 5: Part (a) is a 50-layer convolutional network with a short-cut connection. The solid curve represents the direct short-cut and the dashed curve represents the deformed

short-cut. The white circles represent the connection symbol of the whole network. Part (b) is an expanded view of the convolutions of the first seven layers on the left, where the solid convolutions between the convolutions of layers 2 and 4 represent the transitions with the transition short-cut (the connection represented by the dashed curve on the left). The solid curve between the 7-layer convolutional structures indicates that there is a direct short-cut connection (the connection represented by the solid curve on the left). The plus sign indicates the element addition operation

4 Experiments

The PI and CG selected in this paper come from Columbia Photographic Images and Photorealistic Computer Graphics Dataset (Version 1) [Ng, Chang, Hsu et al. (2005)]. This work uses the deep learning framework Caffe [Jia, Shelhamer, Donahue et al. (2014)] to train the network and generate the model. The data set is divided into a training set consisting of 1400 images and a test set consisting of 200 images. In order to achieve the purpose of random input data, all experiments read into images randomly.

4.1 Training process

The training process uses Caffe's GPU mode. The GPU device used in the experiment is NVIDIA GTX1080Ti. The programming language and framework used in the experiment include Python, Matlab, Shell, Cuda and so on.

4.1.1 The training process on CaffeNet

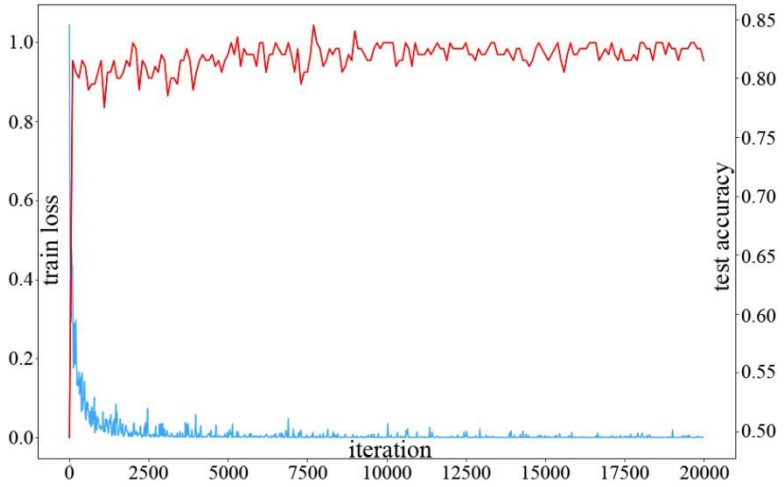


Figure 6: Test accuracy and loss of training process on CaffeNet. The red line represents the correct rate of the testing process network model. The blue line represents the loss value of the training network model

When training CaffeNet, we choose a fine-tune training strategy to achieve rapid network convergence. Fine-tune training requires pre-trained model files. The CaffeNet model selected by this experiment is provided by Caffe, which is trained on 1,000,000 images on

ImageNet. The weight of each layer in the model is used as the initial parameter to continue training the network.

In this experiment, the basic learning rate was set as 0.0001, and the learning rate decreased to 0.00001 after 10,000 training sessions. The purpose of this operation is to keep the trained model in a slow speed of adjustment on the new dataset. *momentum* is set to 0.9, and *weight _ decay* is set to 0.0005. The test accuracy of the training process is shown in Fig. 6. The curve rises rapidly and tends to stabilize. After 20,000 epochs of training, the best accuracy is 85%, and the final average accuracy is 83%.

4.1.2 The training process on the modified CaffeNet

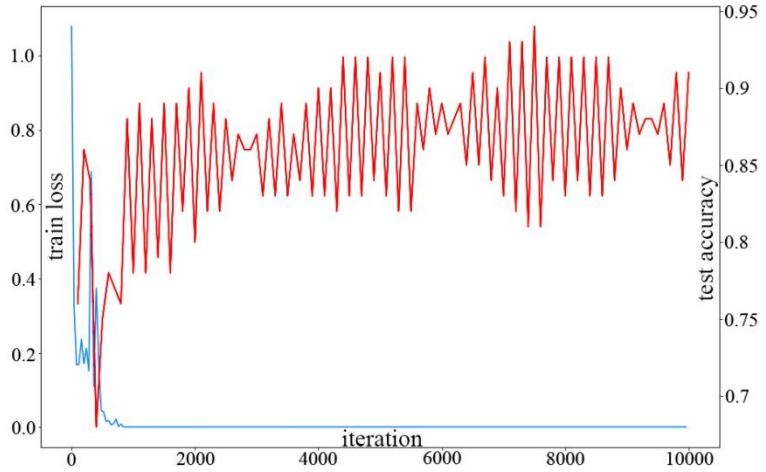


Figure 7: Test accuracy and loss of training process on the increase short-cut connections over CaffeNet. The red line represents the correctness of the network model during testing and the blue line represents the loss value of the training network model

In the same way as mentioned above, after modifying the CaffeNet network and adding the short-cut connection, achieving the 2 classification task, the output of the third fully connected layers is modified to 2. The basic learning rate was set as 0.01. We select the learning rate adjustment method of *poly* and set *power* to 0.5. The gradient update weight *momentum* is set to 0.9, and the error function penalty *weight _ decay* is set to 0.0002. Fig. 7 shows the test accuracy curve during training on CaffeNet with the addition of short-cut connections. The accuracy of the front part shows an oscillating trend and the rear part tends to be steady. After 10,000 trainings, the final average accuracy was 87%.

4.1.3 The training process on the 50-layer CNN with short-cut connection

When training the 50-layer CNN network, we implemented the training part of Fig. 8. After data input, MATLAB's *rgb2gray* function is used to convert the input image to a grayscale image, then the high-pass filter operation is applied to it and fed into the network to train. The basic learning rate *base _ lr* is set to 0.05, learning rate adjustment is selected to

multistep . We set *gamma* to 0.1 and *momentum* to 0.9. As shown in Fig. 9, after 200,000 trainings, the final average accuracy was 98%. Training time is about 44,813.81 seconds.

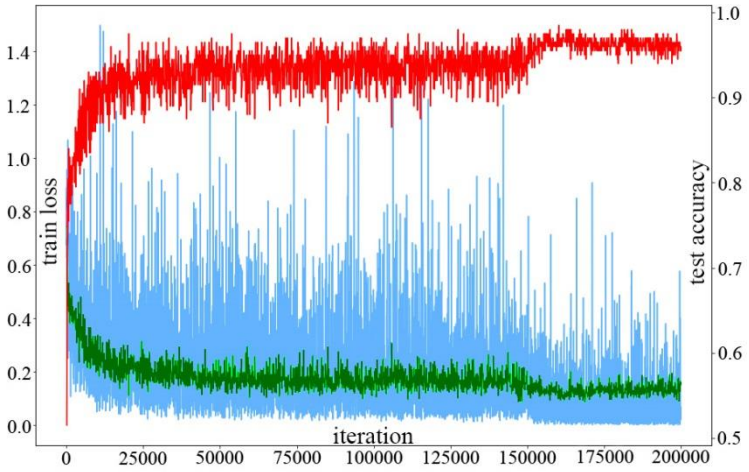


Figure 8: Test accuracy and loss of training process on the 50-layer convolutional network with a short-cut connection. The red line represents the correct rate of the testing process network model. The blue line represents the loss value of the training network model. Green represents the loss value of the test process network model

4.2 Comparison and analysis

This section compares the network model with or without a short-cut connection, proving that the short-cut connection is helpful to improve the accuracy of the detection. In the experiment, we prove that the network depth can be improved by comparing the depth of the network to improve the detection accuracy.

4.2.1 Comparison of network with or without short-cut connection

Comparing Fig. 6 and Fig. 7, we can find that after increasing the short-cut connection, the detection accuracy of the network is improved from 83% to 87%, and the loss function during the training process drops more quickly, indicating that the short-cut connection can solve the gradient vanishing problem. At the same time, the network model will pay more attention to the classification features of input images.

4.2.2 Comparison of the depth of the network

Comparing Fig. 7 with Fig. 8 we can find that after increasing the depth of the network, the detection accuracy can be increased to 98%, which indicates that deepening the network layers can effectively improve the detection accuracy.

4.3 Robustness and real-time analysis

As input to the image at the input layer of the network, parameters are set to turn the image to increase the sample size. It can be proven that the proposed depth CNN method can effectively resist image rotation attack.

This section proves the real-time capabilities of the proposed method by using a trained network model. According to the experimental results, the average test time of single images in the training set of 1,400 images is 1.02 seconds. Tab. 1 shows the comparison with other existing methods. Compared with other existing methods, the model of the proposed method is adaptable. Since the method of feature extraction and classification is an integrated process, it does not need to be segmented. The sum of feature extraction time and classification time is 1.02 seconds, which lags behind that of Fan's method [Fan, Wang, Zhang et al. (2012)]. However, the accuracy of the proposed method is 4.49% higher than their method.

Table 1: Real-time comparison between the proposed method and the existing methods

Methods	Feature Extraction Time (s)
Long's method	20.11
Peng's method	9.48
Peng's method	60.82
Peng's method	9.34
Fan's method	0.88
Proposed method	1.02

4 Conclusion and future work

This paper solves the classification of PI and CG using a method based on a 50-layer CNN structure with a short-cut connection. The method proposed in this paper is a training method based on deep CNNs, and the image features are automatically learned by the network. In other words, feature extraction and classification training process together. The average test accuracy of 50-layer CNNs on the Columbia Photographic Images and Photorealistic Computer Graphics Dataset (Version 1) dataset was 98%. Through the research of this paper, we find that convolution neural network architecture can effectively solve the classification problem of PI and CG, and the network structure will have broader application to other areas of image forensics. For future work, we will continue to use deeper network structure for research on this issue.

Acknowledgement: This work is supported, in part, by the National Natural Science Foundation of China under grant numbers U1536206, U1405254, 61772283, 61602253, 61672294, 61502242; In part, by the Jiangsu Basic Research Programs-Natural Science Foundation under grant numbers BK20150925 and BK20151530; In part, by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund; In part, by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

References

Cao, Y.; Zhou, Z.; Sun, X.; Gao, C. (2018): Coverless information hiding based on the molecular structure images of material. *Computers, Materials & Continua*, vol. 54, no. 2, pp. 197-207.

Chen, J.; Kang, X.; Liu, Y.; Wang, Z. J. (2015): Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849-1853.

Conotter, V.; Bodnari, E.; Boato, G.; Farid, H. (2015): Physiologically-based detection of computer generated faces in video. *IEEE International Conference on Image Processing*, pp. 248-252.

Dang-Nguyen, D. T.; Boato, G.; De Natale, F. G. (2012): Discrimination between computer generated and natural human faces based on asymmetry information. *Signal Processing Conference (EUSIPCO), IEEE, 2012 Proceedings of the 20th European*, pp. 1234-1238.

Dang-Nguyen, D. T.; Boato, G.; De Natale, F. G. (2013): Identify computer generated characters by analysing facial expressions variation. *Information Forensics and Security (WIFS), 2012 IEEE International Workshop*, pp. 252-257.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K. et al. (2009): Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

Fan, S.; Wang, R.; Zhang, Y.; Guo, K. (2012): Classifying computer generated graphics and natural images based on image contour information. *Journal of Information & Computational Science*, vol. 9, no. 10, pp. 2877-2895.

Fu, Z.; Shu, J.; Wang, J.; Liu, Y.; Lee, S. (2015): Privacy-preserving smart similarity search based on simhash over encrypted data in cloud computing. *Journal of Internet Technology*, vol. 16, no. 3, pp. 453-460.

Gurusamy, R.; Subramaniam, V. (2017): A machine learning approach for MRI brain tumor classification. *Computers, Materials & Continua*, vol. 53, no. 2, pp. 91-108.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J. et al. (2014): Caffe: Convolutional architecture for fast feature embedding. *22nd ACM International Conference on Multimedia*, pp. 675-678.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097-1105.

Li, H.; Luo, W.; Huang, J. (2015): Anti-forensics of double JPEG compression with the same quantization matrix. *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 6729-6744.

Liu, Y.; Yao, X. (1996): Evolutionary design of artificial neural networks with different nodes. *IEEE International Conference on Evolutionary Computation*, pp. 670-675.

Long, M.; Peng, F.; Zhu, Y. (2017): Identifying natural images and computer generated graphics based on binary similarity measures of PRNU. *Multimedia Tools and Applications*, pp. 1-18.

Mader, B.; Banks, M. S.; Farid, H. (2017): Identifying computer-generated portraits: The importance of training and incentives. *Perception*, vol. 46, no. 9, pp. 1062-1076.

Ng, T. T.; Chang, S.; Hsu, J.; Pepeljugoski, M. (2005): Columbia photographic images and photorealistic computer graphics dataset. *Columbia University, ADVENT Technical Report*, pp.1-23.

Peng, F.; Zhou, D.; Long, M.; Sun, X. (2017): Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis. *AEU-International Journal of Electronics and Communications*, vol. 71, pp. 72-81.

Peng, F.; Li, J.; Long, M. (2015): Identification of natural images and computer-generated graphics based on statistical and textural features. *Journal of Forensic Sciences*, vol. 60, no. 2, pp. 435-443.

Peng, F.; Zhou, D. (2014): Discriminating natural images and computer generated graphics based on the impact of CFA interpolation on the correlation of PRNU. *Digital Investigation*, vol. 11, no. 2, pp. 111-119.

Scherer, D.; Müller, A.; Behnke, S. (2010): Evaluation of pooling operations in convolutional architectures for object recognition. *International Conference on Artificial Neural Networks*, pp. 92-101.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Wang, J.; Li, T.; Shi, Y.; Lian, S.; Ye, J. (2016): Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics. *Multimedia Tools & Applications*, vol. 76, no. 22, pp. 1-17.

Ye, J.; Ni, J.; Yi, Y. (2017): Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 11, pp. 2545-2557.

Yuan, C.; Li, X.; Wu, Q.; Li J.; Sun, X. (2017): Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. *Computers, Materials & Continua*, vol. 53, no. 4, pp. 357-371.

Zhou, Z.; Mu, Y.; Wu, Q. (2018): Coverless image steganography using partial-duplicate image retrieval. *Soft Computing*, pp. 1-12.

Zhou, Z.; Wang, Y.; Wu, Q.; Yang, C.; Sun, X. (2017): Effective and efficient global context verification for image copy detection. *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48-63.

Zhou, Z.; Wu Q.; Huang F.; Sun, X. (2017): Fast and accurate near-duplicate image elimination for visual sensor networks. *International Journal of Distributed Sensor Networks*, vol. 13, no. 2.

Zhou, Z.; Wu, Q.; Yang, C.; Sun, X.; Pan Z. (2017): Coverless image steganography based on histograms of oriented gradients-based hashing algorithm. *Journal of Internet Technology*, vol. 18, no. 5, pp. 1177-1184.

Zhou, Z.; Yang, C.; Chen, B.; Sun, X.; Liu, Q. et al. (2016): Effective and efficient image copy detection with resistance to arbitrary rotation. *IEICE Transactions on Information and Systems*, no. 6, pp. 1531-1540.