Computers, Materials & Continua CMC, vol.62, no.1, pp.1-16, 2020

Digital Forensics for Recoloring via Convolutional Neural Network

Zhangyi Shen[1], Feng Ding[2, *] and Yunqing Shi[1]

Abstract: As a common medium in our daily life, images are important for most people to gather information. There are also people who edit or even tamper images to deliberately deliver false information under different purposes. Thus, in digital forensics, it is necessary to understand the manipulating history of images. That requires to verify all possible manipulations applied to images. Among all the image editing manipulations, recoloring is widely used to adjust or repaint the colors in images. The color information is an important visual information that image can deliver. Thus, it is necessary to guarantee the correctness of color in digital forensics. On the other hand, many image retouching or editing applications or software are equipped with recoloring function. This enables ordinary people without expertise of image processing to apply recoloring for images. Hence, in order to secure the color information of images, in this paper, a recoloring detection method is proposed. The method is based on convolutional neural network which is quite popular in recent years. Unlike the traditional linear classifier, the proposed method can be employed for binary classification as well as multiple labels classification. The classification performance of different structure for the proposed architecture is also investigated in this paper.

Keywords: Image forensics, machine learning, convolutional neural network, recoloring.

1 Introduction

Digital image is an important information channel [Xia, Zhu, Sun et al. (2018)]. It can serve as evidence utilized at court or on investigation of business or communal department. With the development of multimedia technology and digital devices at present, people can enhance the image quality, hide messages in images, and even tamper the contents of images easily. In order to protect the authenticity and integrity of images, many scientists devote themselves to the research of image forensics [Fridrich (2009); Farid (2009); Piva (2013)]. It is a booming research area in recent years.

Considering the diversity of image manipulations, there are a lot of problems need to be solved in image forensics. The tampered contents are required to be exposed [Zhou, Yang, Chen et al. (2016)], the hidden message should be revealed [Chen, Lu, Yeung et al.

[1] Electrical and Computer Engineering Department, New Jersey Institute of Technology, 323 Martin Luther King BLVD, Newark NJ 07102, USA.

[2] Computer Science Department, State University of New York Albany, 1400 Washington Ave, Albany NY 12222, USA.

[*] Corresponding Author: Feng Ding. Email: fding@albany.edu.

(2018); Lyu and Farid (2002)]. Besides, all possible manipulations should be verified [Cui, McIntosh and Sun (2018); Stamm, Wu and Liu (2013); Wen, Qi and Lyu (2018)] to completely estimate the editing history of images. The detection of image recoloring belongs to one of the forensics problems to identify manipulation fingerprints. Literally, the manipulation of image recoloring [Reinhard, Adhikhmin, Gooch et al. (2001)] substitute the original color of image area with colors. The image area could be objects or background in images. For example, switching the color from red to green is a simple linear recoloring. Other than this, there are other non-linear recoloring manipulations involving the alternation of hue, saturation and lightness. For instance, in many commercial image editing software, there are functions to change the tone of entire image to bring different vision effect to human eyes. Some algorithms make images look warmer, some enhance the color contrast of images to make them sharper, and some may generate an overall blue image for special purpose. Summarized from above, there are plenty of image manipulation algorithms which could be defined as image recoloring.

In order to solve image forensics problems, in the past, most methods of image forensics used to design handcraft feature and feed the feature into supervised linear classifier for classification [Ding, Zhu, Dong et al. (2018); Wang, Dong and Tan (2009); Ding, Zhu, Yang et al. (2014); Ding, Shi, Zhu et al. (2019); De Rosa, Marco, Matteo et al. (2015)]. This situation changed after the resurrection of neural network in recent years. The current version of neural networks is employed by researchers to successfully solve many challenging problems in variety areas. It initiates a new era in artificial intelligence. In the category of modern neural network, convolutional neural network (CNN) is the most typical method of deep learning. Instead of relying on the handcraft features, the CNN is capable of learning features automatically from the given data. What's more, the ability of backward passing enables the optimization of learning procedure that can further boost the performance of CNN. These properties make the CNN an outstanding machine learning method for classification.

The research for deep learning can be traced back to the end of 1980s. In 1989, a neural network to recognize handwritten digits is proposed. This network was named ‘Net-5’. By reducing the number of parameters as well as forcing hidden units to learn from local information, ‘Net-5’ gained better generalization and performance in detection. Meanwhile, the back-propagation algorithm was executed throughout the whole training process. Later, an enhanced ‘Net-5’ was adopted by U.S. Mail. In ‘Net-5’, the number of hidden layers was expanded from 2 to 3. Besides, the network has two convolutional layers and one fully connected layer. After that, another network, which was evolved from ‘Net-5’, named ‘LeNet-5’ was proposed in 1998. ‘LeNet-5’ has become the rudiment of all the deep CNN structures later although it is still shallow.

Limited by the performance of hardware at that time, the research of deep learning stayed in silence for quite some time till 2012. In 2012, the first deep CNN structure named ‘AlexNet’, which aroused an upsurge of deep learning, is published. The authors won the ILSVRC-2012 competition by using ‘AlexNet’, which can be considered as a big progress in deep learning. This network achieved the best testing error rate of 15.3% on the ImageNet database in that competition while the second-best result, which is generated by traditional method, was 26.2%. After the remarkable success of ‘AlexNet’,

numerous brand-new CNN structures, such as 'ZF Net' [Zeiler and Fergus (2014)], 'VGGNet' [Simonyan and Zisserman (2014)], 'GoogLeNet' [Szegedy, Liu, Jia et al. (2015)], 'ResNet', have been occurred in succession. The CNN structures have gone deeper and deeper relying on powerful graphics cards at present.

CNN has started being adopted by image forensics in recent years. By now, it has made tremendous achievements on many forensics cases, such as steganalysis [Fridrich and Kodovsky (2012)], median filter detection [Chen, Kang, Liu et al. (2015)]. In this paper, a convolutional neural network structure is proposed as a recoloring forensics tool. The proposed CNN model can not only detect the traces of recoloring, but also identify which recoloring strategy is applied.

The rest of this paper is organized as follows. In Section 2, several image recoloring algorithms are introduced, and the effects of these algorithms are shown. Section 3 shows our proposed CNN structure and introduces the functions of layers. The experimental results are shown in Section 4. At last, the Section 5 made a conclusion for this paper.

2 Image recoloring

Nowadays, people are willing to use graphics editor, such as Photoshop, to retouch their photos. Actually, a large proportion of pictures processed by common graphics editors were implemented by image recoloring algorithms. Although the final impressions of these pictures varied in many different categories of visual effects, the manipulation of image recoloring can be generally divided into the operation based on three elements: Hue, Saturation and Luminance.

Among image recoloring algorithms, color transfer is one of the most typical one, which belongs to hue transfer. In color transferring process, the color space of image will be transferred from *RGB* to others, such as *Lαβ*. The specific steps of this algorithm are shown below.

Step 1. Give a target image and a source image, obtain *R*, *G*, *B* of them.

Step 2. Convert *RGB* to *Lαβ* space. The transfer rule is shown below.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

$$L^* = 116 f\left(\frac{Y}{1.0}\right) - 16 \tag{2}$$

$$\alpha^* = 500\left[f\left(\frac{X}{0.9502456}\right) - f\left(\frac{Y}{1.0}\right)\right] \tag{3}$$

$$\beta^* = 200\left[f\left(\frac{Y}{1.0}\right) - \frac{Z}{1.088754}\right] \tag{4}$$

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & if\ t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29}, & otherwise \end{cases} \tag{5}$$

Step 3. Calculate transferred $L\alpha\beta$. The formula is shown as follow.

$$l_k = \frac{\sigma_t^k}{\sigma_s^k}\left(S^k - mean\left(S^k\right)\right) + mean\left(T^k\right) \quad k = (l, \alpha, \beta) \tag{6}$$

where l_k, σ_t^k, σ_s^k, S^k, T^k stand for the transferred $L\alpha\beta$, the variance of k on target image, the variance of k on source image, the value of k on source image, the value of k on target image, respectively.

Step 4. Convert transferred Lαβ back to RGB space. The transfer rule is shown below.

$$Y = 1.0f^{-1}\left(\frac{1}{116}(L^* + 16)\right) \tag{7}$$

$$X = 0.950456f^{-1}\left(\frac{1}{116}(L^* + 16) + \frac{1}{500}\alpha^*\right). \tag{8}$$

$$Z = 1.088754f^{-1}\left(\frac{1}{116}(L^* + 16) - \frac{1}{200}\beta^*\right) \tag{9}$$

$$f^{-1}(t) = \begin{cases} t^3, & \text{if } t > \frac{6}{29} \\ 3(\frac{6}{29})^2(t - \frac{4}{29}), & otherwise \end{cases} \tag{10}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.5371500 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{11}$$

Fig. 1 shows a sample of color transfer. Giving a source image and a target image, the color transfer algorithm can change the hue style of target image closing to source image. As shown in Fig. 1, The target image is bright-coloured because of the vivid color of leaves initially. However, the color of leaves was transferred from orange to green though color transfer process resulting in a new recolored image which has cold hue style visually. Although the visual effect was changed, the recolored image still seems natural. In other word, it is hard to be exactly recognized as a recolored image, which has been manipulated by color transfer algorithm, though the subjective judgment of human. As a result, the research of image forensics on recolored image is meaningful. In our experiment, we used a popular type of color transfer named Aibao, which was created by a Chinese photographer in 2008. The effect of Aibao is shown on the 2nd row of Fig. 2. It is a tone closing to cyan-blue.

Nonlinear mapping is another common method of image recoloring. It can be also regarded as a color transferring without sample target. In this case, considering the RGB channel, all pixels in original images are non-linearly converted. Typically, it is used to change the tone of given image such as creating a warm tone. The red components are largely amplified especially for the pixels with less red components to bring a warm view to human eyes. The 3rd row of Fig. 2 is an example of warming images.

Besides Aibao and warming, there are also three different styles of image recoloring, which were used in our experiment. Their descriptions were listed below.

High-dynamic range (HDR). High dynamic range is an image format that can expand the range of brightness level. In image editing tools, a filter that can produce a similar effect is named as HDR filter. HDR filter expands the luminance difference of image, making the bright area much brighter and the dark area much darker. Three samples of HDR images are shown at the 2nd row of Fig. 3. Compared with the original images at 1st row, the differences are visually obvious on first two images but not so obvious on the last image since the major areas on the last image are moderate-brightness.

Retro. Retro filter turns pictures to old photo look. Three samples of Retro images are shown at the 3rd row of Fig. 3. Compared with original images, the overall hue of Retro images is partial to yellow, which presents a sense of age.

Post youth. Post youth filter also turns pictures to old photo look. However, Post youth images are much brighter than Retro images. Three samples of Post youth image are shown at the 4th row of Fig. 3. These images look sprightly, presenting a sense that filled with youth memories.

3 Convolutional neural network (CNN)

Convolutional neural network (CNN), which is built up with plenty of neurons that has learnable weights and biases, has made remarkable achievement on image forensics. The network has a hierarchical structure containing two kinds of modules: the convolutional module and the classification module. In training procedure, CNN can adjust the weights of neurons through back-propagation (BP) to fit the training dataset. In this section, an effective CNN structure for recoloring detection is proposed. Before this, the function of each modules should be introduced.

3.1 Convolutional module

Convolutional module, which has the function of turning images to feature maps or feature vectors, contains several layer groups. Each layer group has three kinds of layers: the convolutional layer, the activation layer and the pooling layer. Each of them has its unique effect for features generation.

3.1.1 Convolutional layer

Convolutional layer, which is mostly adopted for feature extraction in neural networks, is composed with multiple filters. Each filter parallel processes on the input to generate feature maps. Consequently, the number of output feature maps or feature vectors is equal to the number of filters in that convolutional layer. Fig. 4 shows an example of convolution that a 3×3 filter applies on an input image in size of 5×5. The filter scans the whole image in sequence with stride 1. The elements in the scanning window should be multiplied by the elements on the corresponding position in the filter and the sums of the products in scanning windows should be filled into the output feature maps in order. A new feature map in size of 3×3 is obtained after scanning.

However, unlike grayscale image, color image has 3 channels (R, G, B). When color image is used as input image, the filter in convolutional module should be in 3 channels too. As a result, the generated feature map also has 3 channels.

(a) (b) (c)

Figure 1: Sample of color transfer. (a) Source image, (b) Target image, (c) Recolored image

Figure 2: The comparison of original images, Aibao images, Warming images. 1st row: original images. 2nd row: Aibao images. 3rd row: Warming images

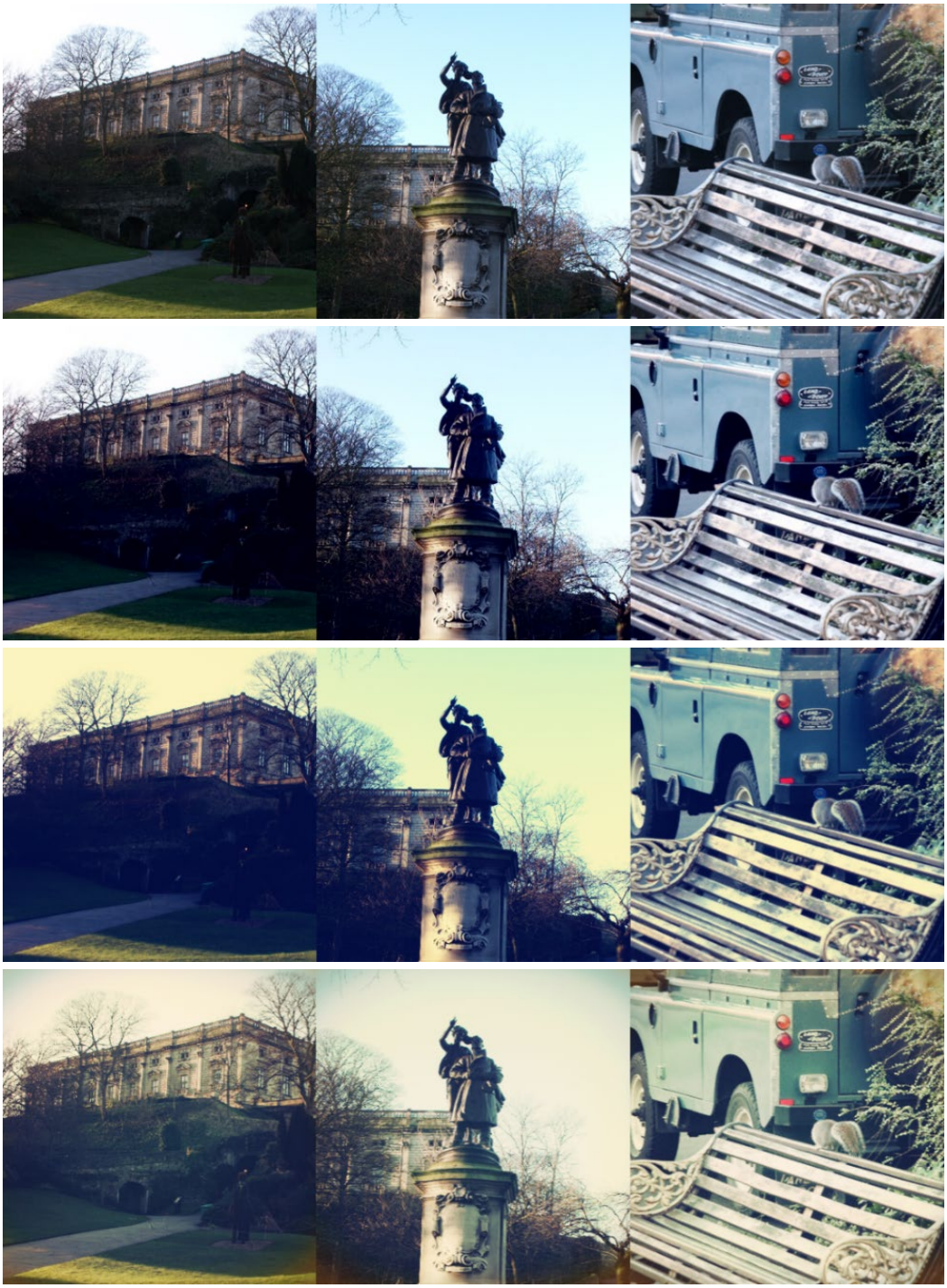


Figure 3: The comparison of original images, HDR images, Retro images, Post youth images. 1st row: original images. 2nd row: HDR images. 3rd row: Retro images. 4th row: Post youth images

3.1.2 Activation layer

Activation layer takes on a non-substitutable effect for optimizing the data model. This layer keeps in step with the convolutional layer bringing nonlinear factors to the data model, which contains many feature maps or feature vectors, generated from the preceding convolution layer. By using activation layer, CNN can solve complicated classification problems better.

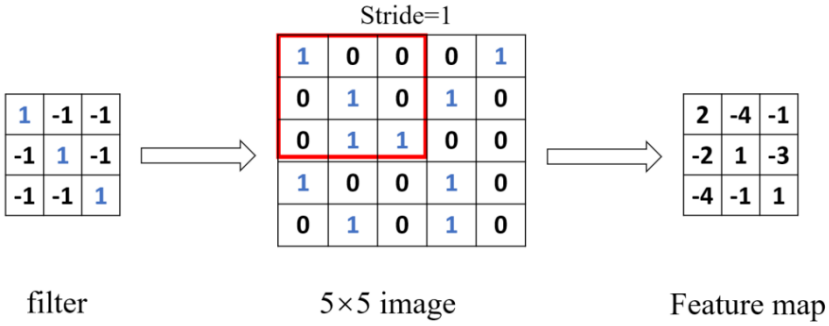
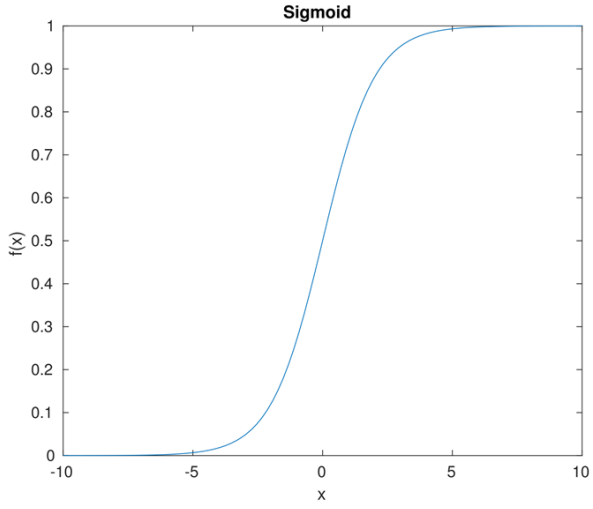
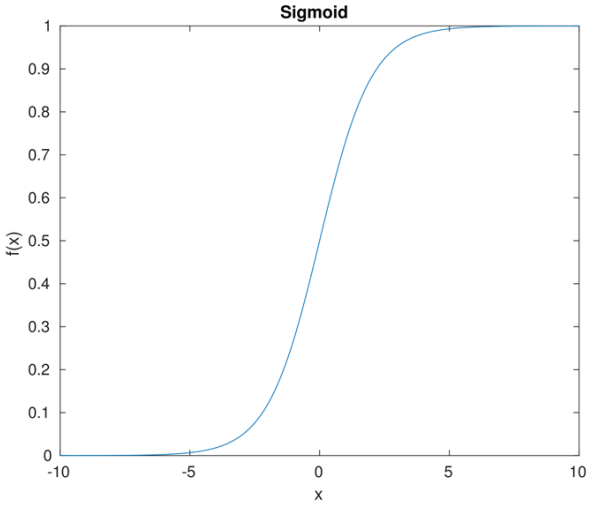


Figure 4: Feature map learning procedure in convolution layer

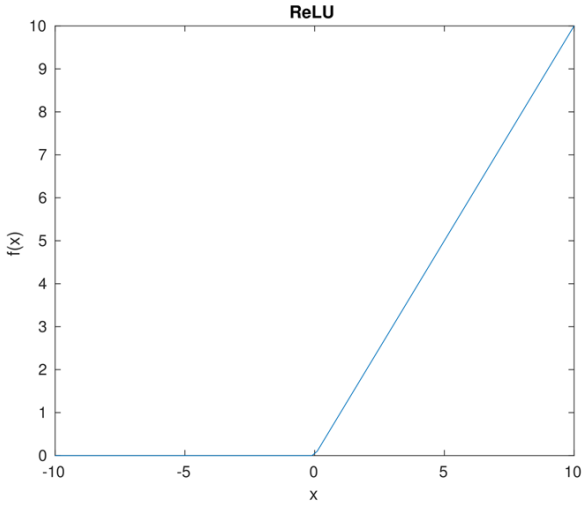
Sigmoid, TanH and ReLU are three types of activation function. They are plotted in Fig. 5. AlexNet has proved that ReLU can shorten the learning cycle much faster than Sigmoid. Also, ReLU can efficiently speed up the convergence of training loss and process learning optimizing easily because of its piecewise linear nature. In consideration of the speed for convergence and the accuracy of whole network, ReLU is the prior choice for activation layer in CNN.



(a)



(b)



(c)

Figure 5: (a) Sigmoid (b) TanH (c) ReLU

3.1.3 Pooling layer

Since multiple filters are adopted in convolutional module, the size of the whole data model becomes larger and larger. In order to reduce the data size, pooling layer is hired in the end of each convolutional module.

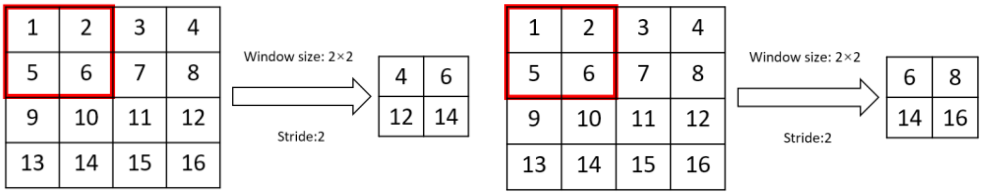
Average pooling and max pooling are two types of pooling layers used in CNN. With fixed values of the pooling window size and the stride for window scanning, the pooling

window scans feature map just like what the filters in convolutional layer do. The example of pooling is shown in Fig. 6. Average pooling calculates the average value for each pooling window and reserves all the average values in order on the output feature map. Different from average pooling, max pooling collects the maximum elements for each window.

In addition, since the number of parameters in network is obviously reduced by pooling layer, the whole network structure has less chance to suffer from overfitting in training.

3.2 Classification module

After going through several convolutional modules, the generated feature maps or vectors will get into the classification module and then be transformed to probabilities for each class. In our proposed classification module, there are two kinds of layers-the fully-connected layer and the SoftMax layer.



(a) (b)

Figure 6: (a) Average pooling (b) Max pooling

3.2.1 Fully-connected layer

Fully-connected layer (FC) has the function of mapping the feature maps or vectors, which are generated from the previous convolutional module, to the setting classes of the network and giving scores of this feature set for each class. FC has some hidden layers and each hidden layer has neurons with learnable weights and setting biases. The input vectors or feature maps are fully connected with the first hidden layers, same with the neurons between adjacent hidden layers. After going through all the hidden layers, the input feature will be transformed to scores for each class. The higher the score on one class means that the input image is more likely belonging to that class according to the judgment of the network. On the contrary, the lower the score on one class suggests that the input image is considered as that it is not belonging to that class by the network.

3.2.2 SoftMax layer

SoftMax layer, which is on the deepest position of the whole network, transforms scores got from the previous FC layer to probabilities for each class and ensure the sum of them as 1 at the same time. The formula of SoftMax algorithm is shown as follow:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{12}$$

where y_i stands for the output probability for i class,Z_i and Z_j are input scores, k is a constant, which stands for the number of classes.

3.3 Proposed method

The proposed method is based on CNN. The CNN structure, motivated by "LeNet-5", is drawn in Fig. 7. It is composed with four convolutional modules and one classification modules.

UCID image dataset [Schaefer and Stich (2003)] used for our experiments. All the images are cropped into the size of 384×384. As a result, the parameters in our proposed CNN are all optimized to fit this size. The input layer are the original images and the images that have been processed by recoloring. Four convolutional modules are set behind the input layers, displayed as "Layer Group 1", "Layer Group 2", "Layer Group 3" as well as "Layer Group 4" in Fig. 5. Each layer group contains three layers begin with convolutional layer to filter the input feature maps. In first layer group, the input feature map is in size of 1× (384×384) and it goes through 8 filters in size of 1× (3×3). In following modules, there are 16 filters of size 8× (3×3) in layer group 2, 32 filters of size 16× (3×3) in layer group 3 and 64 filters of size 32× (3×3) in layer group 4, respectively. Then, the feature maps generated from convolutional module will get into activation layers, which has the ability to optimize the statistical model. ReLU is chosen as the activation function for all layer groups. At the end of each layer group, pooling layer processes down sampling to feature maps. In first three layer groups, max pooling layer in size of 5×5 with stride 2 is performed to get the local maximum. The size of feature maps will be reduced by 3/4 through each max pooling. As for the last layer group, average pooling in size of 48×48 is hired to make sure the output are 64 feature vectors. Finally, 64 feature vectors go through classification module and the final probabilities comes out. These probabilities represent the accuracy of filters in common image editing software detection achieved by this CNN structure. By the way, fully-connected layer contains two hidden layers in our proposed designing.

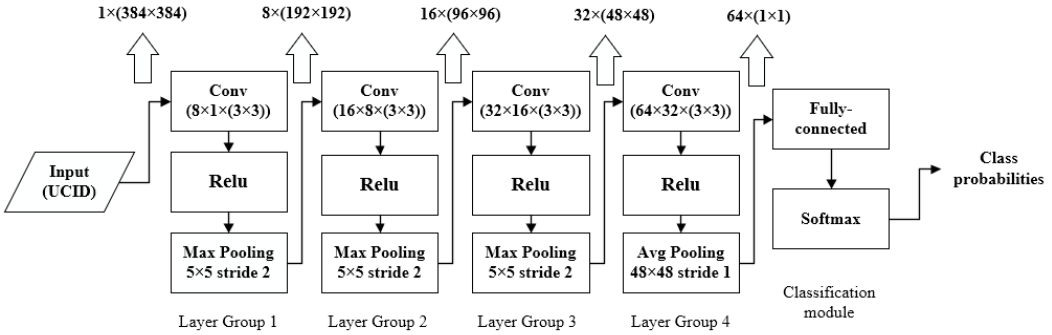


Figure 7: The structure of proposed CNN. The configuration of each layer is displayed inside the boxes. The sizes of feature maps are listed on the top, shown as number of feature maps × (height×width). The sizes of convolutional filter groups are shown in the boxes follows number of filters × number of input feature maps × (height×width)

4 Experiment results

4.1 Dataset

UCID [Schaefer and Stich (2003)] was employed as the image dataset in our experimental. It contains 1,338 uncompressed color images with size of 384×512 or 512×384 in "tiff" format. For convenience purposes, all the images were cropped into the size of 384×384. In addition, consider of that "tiff" is not a mainstream image format at present, all the images were converted to "png" format. Overall, the experimental image database contains totally 1,338 uncompressed colorful images with size of 384×384 in "png" format.

4.2 Platform and settings

TensorFlow, the most popular deep learning framework nowadays, is selected to compose our CNN structure. The open source feature and high expansibility of TensorFlow hastened the network building. The version number of the TensorFlow for this experiment is 1.11.0. All the experiment codes were implemented on Spyder 3.3.1, which is a common python development environment. Adam optimizer, as the most common optimizer in TensorFlow using, was applied to train the whole network. Two graphics cards were employed for training process. The model of the graphics cards is NVIDIA GeForce GTX 1080Ti with 10 GB memory. The training batch size was set to 64, meaning that for each iteration, 64 images will get into the network. The training iteration was fixed as 5000 for two-category classification and 20000 for six-category classification. One epoch means all the images in training dataset get into the network once. So, for two-category classification, the total number of epochs is 152. As for four-category classification, the total number of epochs is 200.

4.3 Results

First, 5 training processes of two-category classification, which were HDR images with original images, retro images with original images, post youth images with original images, aibao images with original images as well as warming images with original images, were performed individually. Then, a model to distinguish all the six styles of images was trained. The classification accuracy and the consumed epochs for convergence are recorded in Tab. 1.

Table 1: The performance of the proposed method towards the recoloring algorithms

Cases	HDR	Retro	Post youth	Aibao	Warming	All
Accuracy	100%	100%	100%	100%	93.75%	96.88%
Epoch	34	6	19	46	68	175

As shown in Tab. 1, the proposed CNN emerged its overwhelming ability to recognize the images not only between recolored images and original images, but also between different styles of recolored images. In addition, the curves of the training losses and the test errors on Post youth case and all classes case were plotted on Fig. 8. As shown in the figure, the convergence of two-category classification occurred in 25 epochs and the convergence of six-category classification occurred in 60 epochs. However, there are

some small fluctuations after convergence occurred.

Besides, we also made some comparison on different depths of our proposed CNN as well as different combinations of activation functions in convolutional module.

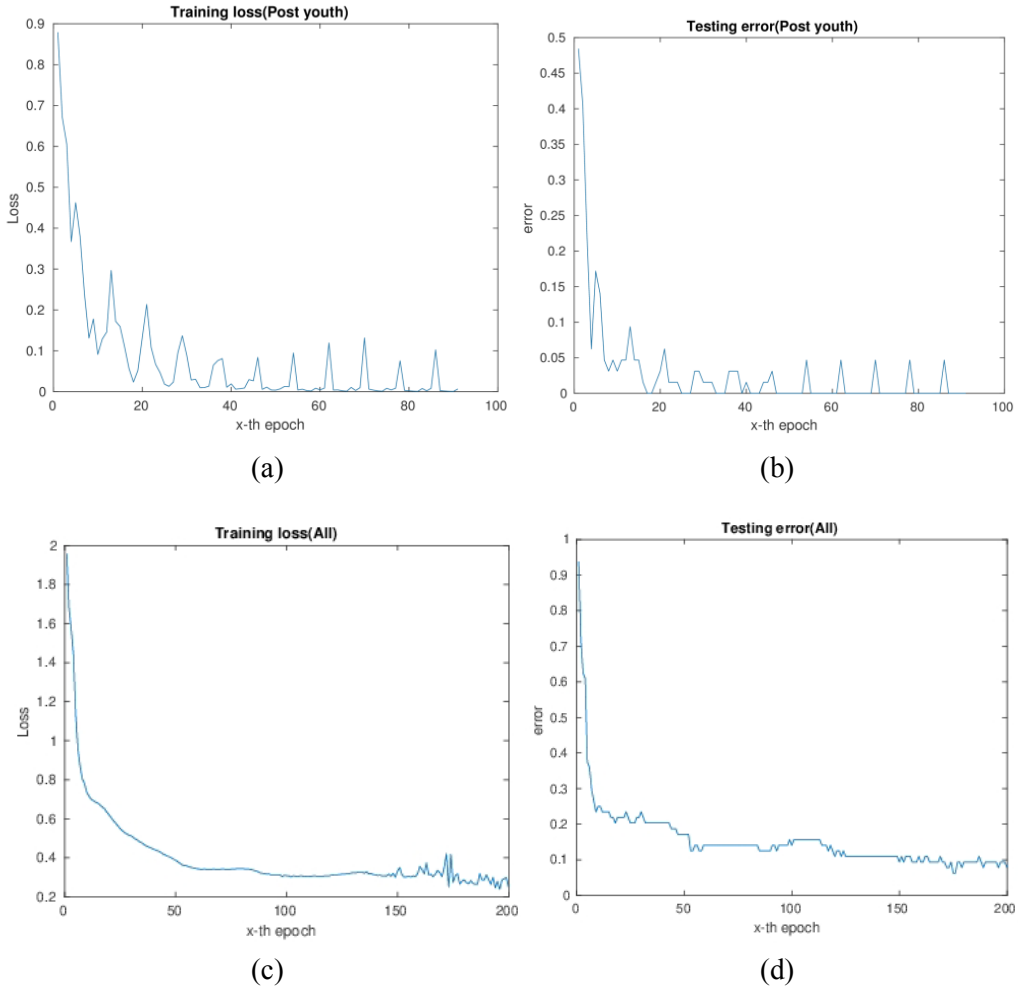


Figure 8: (a) Training loss of two-category classification (Post youth and original); (b) Testing error of two-category classification (Post youth and original);(c) Training loss of six-category classification (All 6 styles); (d) Testing error of six-category classification (All 6 styles)

In our proposed CNN, there are totally 12 layers (4 layer groups) in convolutional modules, which is a comparatively shallow CNN. In order to find out the most suitable depth of the network, we compared our 4 layer groups structure with 3 layer groups, 5 layer groups as well as 6 layer groups on two-category classification and six-category classification, respectively. The results of two-category classification, which is HDR images against original images, is shown in Tab. 2. As shown in the table, all the depths

of the network can achieve 100% in this two-category classification case. 5 layer groups and 6 layer groups is even faster than 4 layer groups to achieve the best accuracy. However, it does not mean that 5 or 6 layer groups will be a better choice on detecting recolored images. Tab. 3 recorded the results of six-category classification case. In Tabs. 3, 4 layer groups outperformed other depths on detecting accuracy. As a result, although deeper network structures can speed up convergence of training process, they are easier to suffer from overfitting with the increasing of data size and then have a performance degradation. This comparison proved that 4 layer groups is still the best choice on image recoloring detection.

Table 2: The comparison of the proposed method with different depths for binary classification of HDR and original image

Depths	3 layer groups	4 layer groups	5 layer groups	6 layer groups
Accuracy	100%	100%	100%	100%
Epoch	74	34	17	18

Table 3: The comparison of the proposed method with different depths towards the classification of all recoloring algorithms

Depths	3 layer groups	4 layer groups	5 layer groups	6 layer groups
Accuracy	93.75%	96.88%	92.19%	93.75%
Epoch	72	175	38	61

In some forensics works based on CNN, authors used TanH to replace ReLU as activation layer in the last few convolutional layers and this strategy lead to an immense improvement for their methods. However, all of the activation functions in our proposed CNN structure are ReLU. We tried two other combinations of activation layers, which are replacing all ReLU with TanH and replacing all ReLU with TanH except the last layer group, also on two-category classification and six-category classification, respectively. The Warming as recoloring algorithm, which is the most challenging problem other than the other methods, is chosen as the subject of two-category classification for our experiment. The classification results are shown in Tab. 4. Although all three kinds of combinations have the similar performance on two-category classification, the combination of all ReLU, which is adopted in our proposed CNN structure, can achieve better accuracy on six-category classification.

Table 4: The comparison of different combinations of activation layers

Cases	Two-category classification			Six-category classification		
Combinations	All ReLU	All TanH	3 TanH+1 ReLU	All Relu	All TanH	3 TanH+1 ReLU
Accuracy	93.75%	93.75%	93.75%	96.88%	93.75%	93.75%
Epoch	68	143	11	175	95	160

5 Conclusion

In this paper, a method for image recoloring forensics is proposed. It is based on CNN structure that consists of 12 layers. The performance of the proposed CNN is excellent regardless if it is binary or multiple label classification. It is capable of detecting recoloring as well as identify which recoloring algorithm is applied. The proposed method can reach accuracies over 90% under all circumstances. In addition, we also discussed about the impact of the network with different depths and activation strategies for the problem. Besides, the oscillation can be observed for the convergence of training procedure during our experiment. In future, we will try to address this issue by optimizing the neural network and organizing more proper data. Furthermore, deeper research on recoloring forensics will be conducted.

References

Chen, J. J.; Lu, W.; Yeung, Y.; Xue, Y. J.; Liu, X. J. et al. (2018): Binary image steganalysis based on distortion level co-occurrence matrix. *Computers, Materials & Continua*, vol. 5, no. 2, pp. 201-211.

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.

Ding, F.; Zhu, G. P.; Dong, W. Q.; Shi, Y. Q. (2018): An efficient weak sharpening detection method for image forensics. *Journal of Visual Communication and Image Representation*, vol. 50, pp. 93-99.

Ding, F.; Zhu, G. P.; Yang, J. Q.; Xie, J.; Shi, Y. Q. (2014): Edge perpendicular binary coding for USM sharpening detection. *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 327-331.

Ding, F.; Shi, Y. X.; Zhu, G.; Shi, Y. Q. (2019): Smoothing identification for digital image forensics. *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8225-8245.

De Rosa, A.; Marco F.; Matteo M.; Piva, A; Barni, M. (2015): Second-order statistics analysis to cope with contrast enhancement counter-forensics. *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1132-1136.

Farid, H. (2009): Image forgery detection. *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16-25.

Fridrich, J. (2009): Digital image forensics. *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26-37.

Fridrich, J.; Kodovsky, J. (2012): Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882.

Lyu, S.; Farid, H. (2002): Detecting hidden messages using higher-order statistics and support vector machines. *International Workshop on Information Hiding*, pp. 340-354.

Piva, A. (2013): An overview on image forensics. *ISRN Signal Processing*.

Ravi, H.; Subramanyam, A. V.; Emmanuel, S. (2015): ACE-an effective anti-forensic contrast enhancement technique. *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 212-216.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. (2001): Color transfer between

images. *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34-41.

Schaefer, G.; Stich, M. (2003): UCID: An uncompressed color image database. *Storage and Retrieval Methods and Applications for Multimedia 2004, International Society for Optics and Photonics*, vol. 5307, pp. 472-481.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Stamm, M. C., Wu, M.; Liu, K. R. (2013): Information forensics: an overview of the first decade. *IEEE Access*, vol. 1, pp. 167-200.

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

Wang, W.; Dong, J.; Tan, T. N. (2009): Effective image splicing detection based on image chroma. *16th IEEE International Conference on Image Processing*, pp. 1257-1260.

Wen, L. Y.; Qi, H. G.; Lyu, S. W. (2018): Contrast enhancement estimation for digital image forensics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2, pp. 49.

Xia, Z. H.; Zhu, Y.; Sun, X. M.; Qin, Z.; Ren, K. (2018): Towards privacy-preserving content-based image retrieval in cloud computing. *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 258-264.

Zeiler, M. D.; Fergus, R. (2014): Visualizing and understanding convolutional networks. *European Conference on Computer Vision*, pp. 818-833.

Zhou, Z. L.; Yang, C. N.; Chen, B. J.; Sun, X. M.; Liu, Q. et al. (2016): Effective and efficient image copy detection with resistance to arbitrary rotation. *IEICE Transactions on Information and Systems*, vol. 99, no. 6, pp. 1531-1540.