2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

A Novel Method for Detecting Image Forgery Based on Convolutional Neural Network

Na Huang
Faculty of Information Technology
Beijing University of Technology
Beijing 100124, China
nahuang@emails.bjut.edu.cn

Jingsha He
Faculty of Information Technology
Beijing University of Technology
Beijing 100124, China
jhe@bjut.edu.cn

Nafei Zhu
Faculty of Information Technology
Beijing University of Technology
Beijing 100124, China
znf@bjut.edu.cn

***Abstract*—Detection of image forgery is an important part of digital forensics and has attracted a lot of attention in the past few years. Previous research has examined residual pattern noise, wavelet transform and statistics, image pixel value histogram and other features of images to authenticate the primordial nature. With the development of neural network technologies, some effort has recently applied convolutional neural networks to detecting image forgery to achieve high-level image representation. This paper proposes to build a convolutional neural network different from the related work in which we try to understand extracted features from each convolutional layer and detect different types of image tampering through automatic feature learning. The proposed network involves five convolutional layers, two full-connected layers and a Softmax classifier. Our experiment has utilized CASIA v1.0, a public image set that contains authentic images and splicing images, and its further reformed versions containing retouching images and re-compressing images as the training data. Experimental results can clearly demonstrate the effectiveness and adaptability of the proposed network.**

Keywords-digital forensic, image forgery, tempering detection, deep feature, convolutional neural network

I. INTRODUCTION

Image editing has become more convenient along with the fast development of digital technology. It also has become more difficult for humans to identify image editing as digital images present themselves in the form of two-dimensional matrix.

In recent years, many incidents of image tampering have appeared in the fields of science, media, etc., incurring tremendous cost to judicature and social order. Given their easiness and effectiveness, copy-move [1], seam carve [2], re-compress [3] and retouch [4] are currently the most common techniques involved in image forgery, making them the main focus of forensic detection of image forgery. Forgery detection based on textual features of images is one of the most popular approaches. PRUN (photo-response non-uniformity) noise is another commonly used feature for detecting image forgery because the tampering manipulation can cause inconsistencies in terms of the noise of the target images [5].

Deep learning has been applied to forensic detection of image forgeries in recent years. Among the techniques, CNN (Convolutional Neural Network) has demonstrated remarkable performance in visual recognition [4,6]. Nevertheless, it has been pointed out that hierarchical features extracted by CNN could represent not only the visual concept for recognizing objects, but also the re-sampling or noise changes caused by tampering operations [7-11].

This paper proposes a novel method for detecting image tampering based on CNN. The work involves implementing a network based on CNN to automatically learn feature representations that can adapt itself to multiple tampering manipulations. Three commonly used optimizers, i.e., Adam (adaptive moment estimation), RMSProp and SGDM (stochastic gradient descent with momentum), are tested in the training stage. The work also includes experiment in which the performance is compared between Softmax and SVM to demonstrate the advantage.

The reminder of this paper is organized as follows. Section 2 describes the architecture of the implemented network. Section 3 presents the results of the experiment. Finally, Section 4 concludes this paper in which some future work is also discussed.

II. THE NETWORK

A. Architecture of the constructed network

Our proposed method involves the construction of an image-forgery-detecting network, that has a total number of nine layers including input layer, five convolutional layers for extracting representative features of images with feature fusion and max-pooling, two fully connected layers and a Softmax classifier. This architecture follows the existing networks that has been widely verified of good performance.

a) Input setup

Deep neural networks need representative data sets for learning. Therefore, we need a massive set of image samples that can provide the features of multiple clusters, e.g., spliced images, and rotated images. In the constructed network, the input layer is designed to accept RGB images each of which has the size 227×227×3. Original images could be resized to conform to this requirement.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00255

b) Convolutional layer

The major function of the convolutional layer is to extract features and to initialize fixed-size kernels to scan the input matrix. The size of the kernels and the stride of sliding are set when a network is to be constructed. Let a_{ij} denote the pixel at the i^{th} line and the j^{th} column of a feature map, computing of which follows Eq. (1).

$$a_{i,j} = f(\sum_{m=0}^{M}\sum_{n=0}^{N} w_{m,n}x_{i+m,j+n} + w_b) \tag{1}$$

where x_{ij} denotes the pixel at (i^{th}, j^{th}) of the input image and $w_{m,n}$ denotes the shared weight at the corresponding position of filters, w_b is the bias of filters and f denotes the activated function. The size of the kernel is $M \times N$, meaning that m ranges from 0 to M while n ranges from 0 to N.

The activated function used in the constructed network is ReLU to introduce nonlinear factors into linear convolutional operations so as to remove the redundancy in data, maximize the retention of characteristics of data and construct a sparse matrix. Function ReLU performs a threshold operation on each element expressed in Eq. (2), where x denotes the input element.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x \leq 0 \end{cases} \tag{2}$$

The standard function for an analog neuron output is generally tanh() or sigmoid() in which the output is basically unchanged even if the input data varies wildly in size. ReLU is a nonlinear, non-saturated function, which executes much faster than saturated functions during training. Moreover, this distorted linear function preserves not only the nonlinear expressive capability, but also the linear property, i.e., the positive value. Applying ReLU as the solution of the problem of Gradient dispersion enables the training of deep networks.

c) Feature fusion

The goals of pooling are to reduce the dimension of features and retain effective information to avoid over-fitting. Besides, pooling can resist rotation, translation, expansion, etc. The common operations adopted at the pool layer include maximum value sampling, average value sampling, summation area sampling and random area sampling. The pooling method utilized in this paper is maximum value sampling to reduce the offset of the estimated mean value caused by parameter errors in convolution layers and to retain more texture information of images.

d) Fully-connected layer

The fifth convolutional layer is followed by two fully-connected layers that are responsible for logical inference. The first fully-connected layer has a convolutional kernel to turn a three-dimensional matrix, an input of this layer, into a two-dimensional vector. Dropout is also employed in this layer to ignore random neurons and to resist over-fitting.

e) Output layer

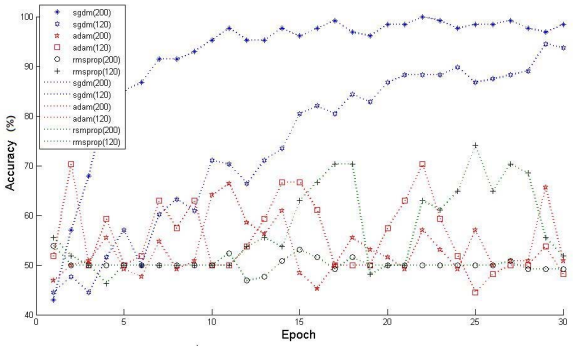
Softmax is used in this layer as a classifier after the fully-connected layers. The images are classified into the authentic cluster or the forged cluster. Let x be an input element, r be the r^{th} cluster, $r \in (1,k)$. For x, activated function of Softmax is expressed in Eq. (3).

$$\varphi_r(x) = \frac{\exp(c_r(x))}{\sum_{r=1}^{k}\exp(c_r(x))} \tag{3}$$

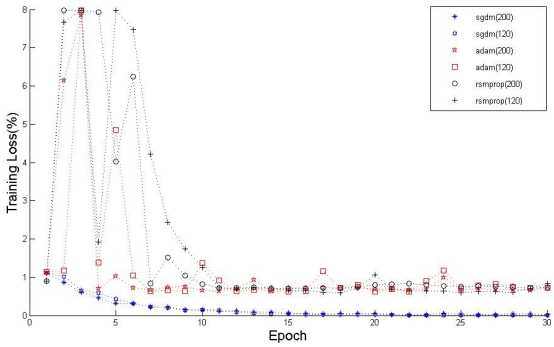
where $c_r(x)=\ln(P(x|c_r)P(c_r))$, $P(x|c_r)$ is the conditional probability that x belongs to the r^{th} cluster, $P(c_r)$ is the cluster prior probability, $\varphi_r(x)$ is the output of the operation Softmax, $0 \leq \varphi_r(x) < 1$. In the case of binary classification, the value of k is set to be 2 according to the number of clusters.

B. Network Training

Different variants of stochastic gradient descent can be used to train the network. Three commonly used optimizers are tested, i.e., Adam (adaptive moment estimation), RMSProp and SGDM (stochastic gradient descent with momentum). Comparison of the performance of the three optimizers is shown in figure 1 using two datasets with 200 images and 120 images, respectively.



(a) Training accuracy



(b) Training loss

Figure 1. Comparison of three optimizers in the training procedure

Finally, we adopt the SGDM optimizer to train the network for its superior performance. The update rule of the stochastic gradient descent with momentum is set in Eq. (4)

$$\theta_{\varsigma+1} = \theta_{\varsigma} - \alpha\nabla E(\theta_{\varsigma}) + \gamma(\theta_{\varsigma} - \theta_{\varsigma-1}) \tag{4}$$

where ς stands for the iteration number, $\alpha > 0$ is the learning rate, θ is the parameter vector and $E(\theta)$ is the loss function. The gradient of the loss function, $\nabla E(\theta)$, is evaluated using the entire training set and γ determines the contribution of the previous gradient step to the current iteration.

III. EXPERIMENT AND ANALYSIS

A. Image datasets

We utilized a public dataset, CASIA v1.0 [12], which is widely used for research in forgery detection. This dataset contains two types of images, authentic images and spliced images, both in JPG format. To test the performance of detecting some other tampering operations, i.e., retouch and re-compress, we further manipulate the authentic images with these two kinds of tampering. Table 1 shows the number of images in each dataset. CASIA v1.0 contains 800 authentic images and 921 spliced images and the other three datasets that we generated each contains 800 authentic and 800 tampering images.

TABLE I. THE IMAGE SETS USED IN THE EXPERIMENT

Image Set	Authentic images	Spliced images	Retouch	Re-compress
CASIA v1.0	800	921	×	×
CASIA-Cv	800	×	800	×
CASIA-Rc	800	×	×	800
CASIA-Un	800	800	800	800

B. Experiment using the implemented network

We used in the experiment the Neural Network Toolbox of Matlab 2018b in the Window 10 environment. We build five convolutional layers together with max-pooling, two fully connected layers and one Softmax layer. The input layer accept images of the size [227 227 3] (227×227 patch, 3 RGB channels). The first convolutional layer has 96 kernels with a sliding-window of size [11 11] and stride [4 4]. The basic setting of other layers was also done when created. The two fully connected layers each has 50% dropout function. These layers constitute our CNN network called DTnet for detecting universal tampering of images.

C. Performance

We used the datasets to train DTnet. Taking splice detection as an example, Figures 2 and 3 show the process of training of DTnet using CASIA v1.0 and CASIA-Un, respectively, from which we can see that the training accuracy reaches 100% and the training loss closes to 0 eventually. Table 2 shows the performance of detecting different kinds of image tampering in which three groups of experiment were performed through using different numbers of images as the training data and as the testing data. For universal detection, we modified the output-size in the classifying layer, which is the number of clusters, from 2 to 4 (Au, Sp, Rt and Rc).

To compare the difference between the Softmax classifier and the SVM classifier in the classification layer, we extracted the output features of CASIA v1.0 from the second fully connected layer and passed them into a SVM classifier. The results are presented in Table 3 from which we can see that the Softmax classifier achieves better performance in testing accuracy though the training accuracy of both classifiers eventually reaches 100%.

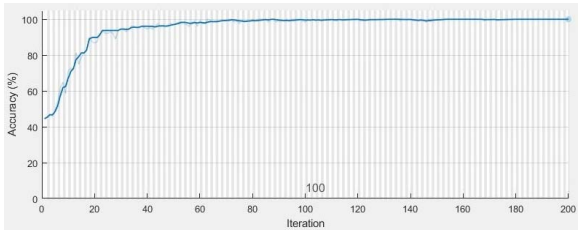


Figure 2. Performance of the training using CASIA v1.0.

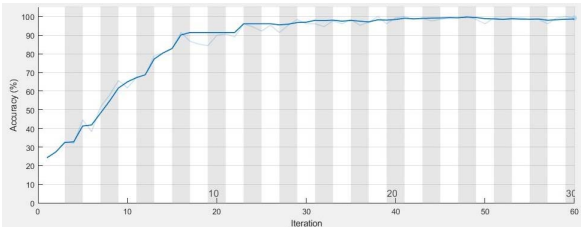


Figure 3. Performance of the training using CASIA-Un.

TABLE II. PERFORMANCE OF THE DETECTING NETWORK

Category	Number of images	Training : Testing	Accuracy (%)
Splice	60	8:2	98.50
	200	9:1	100
	1721	9:1	96.71
Retouch	60	8:2	100
	200	9:1	83.13
	1600	9:1	93.13
Re-compress	60	8:2	71.02
	200	9:1	68.15
	1600	9:1	64.44

TABLE III. COMPARISON BETWEEN SOFTMAX AND SVM

Dataset	Classifier	Number of images	Training : Testing	Accuracy (%)
CASIA v1.0	Softmax	60	8:2	75.00
		200	9:1	96.00
		1721	9:1	100
	SVM	60	8:2	55.00
		200	9:1	94.00
		1600	9:1	93.02

In the constructed network, authentic images are labeled with 'Au', splicing images with 'Sp', retouching images with 'Rt', and re-compressing images with 'Rc'. After training is completed through using the corresponding datasets, we can use the trained network to detect some images. Figure 4 shows the examples of tempering detection with the appropriate labels on the results.

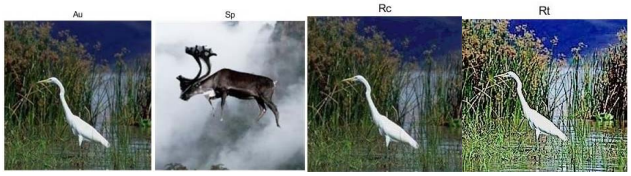


Figure 4. Forged images with the appropriate label as the title.

D. Comparison

Table 4 contains the comparison our proposed method for detecting image forgery to other image forgery detecting methods, Bayar [8], Zhang [10] and Rao [11] that also use deep features to detect image tampering but with different classifiers. For example, Zhang [10] used the threshold classifier with deep features to detect the forgery of image of types JPG and Tiff. Methods that use other features, such as DCT and PRUN noise, have also been compared. All presented methods can detect only one kind of tampering while DTnet is effective for three kinds and can be easily adapted to other kinds of forgery detection.

TABLE IV. COMPARISON OF THE AVERAGE ACCURACY WITH OTHER METHODS

Methods	Feature	Classifier	Splice	Re-touch	Recom-press
This method	Deep feature (CNN)	Softmax	0.95	0.93	0.71
Chierchia [5]	PRUN noise	Bayesian	-	×	×
Bayar [8]	Deep feature (CNN)	ET classifier	×	×	0.97
Zhang [10]	Deep feature (CNN)	Threshold classifier	0.91	×	×
Rao [11]	Deep feature (CNN)	SVM	0.97	×	×
Chen[13]	DCT	-	×	×	1
Guo[14]	Hue Saturatio-n Value	SVM	×	0.87	×
Bunk[15]	Re-samplig feature	Softmax	0.94	×	×

IV. CONCLUSION

In this paper, we constructed a convolutional neural network DTnet for detecting image forgery. The network has nine layers consisting of the image input layer, five convolutional layers, two fully connected layers and a Softmax classifier. We used an expanded version of the image dataset CASIA v1.0 that includes more tampering images as the training data. Compared to some previous methods, our network could easily be adapted to other types of tampering detection as it can learn features automatically. Testing results show the effectiveness and adaptability of our proposed method.

In the future, we will continue improving the network to achieve higher accuracy of detection. Source-device identification and image steg-analysis are two additional research directions that we seek to do more work.

ACKNOWLEDGMENT

The work in this paper has been supported by National Natural Science Foundation of China (61602456) and by National High-tech R&D Program (863 Program) of China (2015AA017204).

REFERENCES

[1] T. Mahmood, A. Irtaza, Z. Mehmood, and M. Tariq Mahmood, "Copy-move forgery detection through stationary wavelets and local binary pattern variance for forensic analysis in digital images," Forensic Science International, vol. 279, pp.8-21, 2017.

[2] Q. Liu and Z. Chen, "Improved approaches with calibrated neighboring joint density to steganalysis and seam-carved forgery detection in jpeg images," ACM Transactions on Intelligent Systems and Technology, vol. 5(4), pp.1-30, 2015.

[3] A. Taimori, F. Razzazi, A. Behrad, A. Ahmadi, and M. Babaie-Zadeh, "A novel forensic image analysis tool for discovering double JPEG compression clues," Multimedia Tools And Applications, vol. 76(6), pp.7749-7783, 2017.

[4] A. Bharati, R. Singh, M. Vatsa, and K. W. Bowyer, "Detecting facial retouching using supervised deep learning," IEEE Transactions on Information Forensics and Security, vol. 11(9), pp.1903-1913, 2016.

[5] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection," IEEE Transactions on Information Forensics & Security, vol. 9(4), pp.554-567, 2014.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for scene segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39(12), pp.2481-2495, 2017.

[7] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," Electronic Imaging, vol. 2017(7), pp.77-86, 2017.

[8] B. Bayar and M. C. Stamm, "On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection," in Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, pp.2152-2156, 2017.

[9] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp.5-10, 2016.

[10] Y. Zhang, L. L. Win, J. Goh, and V. L. Thing, "Image region forgery detection: A deep learning approach," in Proceedings of the Singapore Cyber-Security Conference 2016: Cyber-Security by Design, vol. 14, pp.1-11, 2016.

[11] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in Proceedings of the IEEE International Workshop on Information Forensics and Security, pp.1-6, 2017.

[12] J. Dong and W. Wang, CASIA tampered image detection evaluation (TIDE) database, v1.0 and v2.0, available at: http://forensics.idealtest.org/, 2014.

[13] Y. L. Chen, and C. T. Hsu, "Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection," IEEE Transactions on Information Forensics & Security, vol. 6(2), pp.396-406, 2011.

[14] Y. Guo, X. Cao, W. Zhang, and R. Wang, "Fake Colorized Image Detection," IEEE Transactions on Information Forensics & Security, vol. 13(8), pp.1932-1944, 2018.

[15] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, and B. S. Manjunath, "Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning," in the Proceeding of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.1881-1889, 2017.