Median Filtering Forensics Based on Convolutional Neural Networks

Jiansheng Chen, Xiangui Kang, Ye Liu, and Z. Jane Wang

***Abstract*—Median filtering detection has recently drawn much attention in image editing and image anti-forensic techniques. Current image median filtering forensics algorithms mainly extract features manually. To deal with the challenge of detecting median filtering from small-size and compressed image blocks, by taking into account of the properties of median filtering, we propose a median filtering detection method based on convolutional neural networks (CNNs), which can automatically learn and obtain features directly from the image. To our best knowledge, this is the first work of applying CNNs in median filtering image forensics. Unlike conventional CNN models, the first layer of our CNN framework is a filter layer that accepts an image as the input and outputs its median filtering residual (MFR). Then, via alternating convolutional layers and pooling layers to learn hierarchical representations, we obtain multiple features for further classification. We test the proposed method on several experiments. The results show that the proposed method achieves significant performance improvements, especially in the cut-and-paste forgery detection.**

***Index Terms*—Convolutional neural networks, deep learning, hierarchical representations, median filtering forensics.**

I. Introduction

MULTIMEDIA forensics has been an active research area during the last decade. Blind forensics techniques generally utilize statistical fingerprints to verify the authenticity of multimedia data without access to the original source. However such imperceptible fingerprints may be destroyed by various manipulations. Recently more efforts have been made to expose the processing history of digital images, such as filtering [1]–[5], re-sampling [6], compression [7], and contrast enhancement [8], since their blind detections are forensically important [1]–[5].

Widely employed as a popular noise removal and image enhancement tool, median filtering has the properties of non-linearity and preserving edge information of an image. These characters have been utilized by anti-forensics methods, e.g., removing statistical traces of blocking artifacts left by the JPEG compression [9], or destroying linear correlations between adjacent pixels for the purpose of hiding the trace of re-sampling [10].

A number of works have been proposed for median filtering forensics [1]–[5]. Perfect performance was reported when detecting median filtering on uncompressed and large-size images. However, an image is usually saved in a compressed format such as the JPEG format and we could face the scenario that a portion of a filtered image is pasted into the original image. Therefore the detection of median filtering from small blocks of a compressed image is forensically important, and it remains a challenge for median filtering forensics.

Existing median filtering forensics techniques mainly depend on manually selected features, and the feature extraction and the classification are generally separated and not optimized simultaneously in an iterative way. Furthermore, the performance could degrade due to no exact model of a natural image. To address this concern, in this paper, instead of constructing better artificial features, we plan to automatically learn feature representations jointly with the classification for median filtering forensics.

Deep learning frameworks are able to learn feature representations and fulfill classification automatically, and also can utilize the classification result to guide the feature extraction via the back propagation algorithm. Motivated by how human brains process information, researchers have explored to train deep multi-layer neural networks, such as Deep Boltzmann Machines [11], Deep Auto encoders [12], and Convolutional Neural Networks [13]. These methods have shown impressive performances in artificial intelligence (AI) tasks such as object recognition [13] and natural language processing [14].

However, the trace in an image left by median filtering may be too weak to be detected by the conventional CNN approach. Therefore, we modify the conventional CNN model by adding a filter layer, and design a specific CNN-based framework for median filtering forensics. Extensive experiments have shown that the proposed method can achieve better detection performance than the state-of-the-art schemes.

II. The Proposed CNN Model for Forensics

A. Architecture of our Modified CNNs

CNNs automatically learn features and perform the classification. It has deep architectures that consist of multiple levels of non-linear operations. A typical CNN has several types of

Manuscript received March 30, 2015; revised May 22, 2015; accepted May 22, 2015. Date of publication June 01, 2015; date of current version June 04, 2015. This work was supported by the National Science Foundation of China under Grants 61379155, U1135001, and 61332012, the 973 Program under Grant 2011CB302204, and the NSF of Guangdong province under Grant s2013020012788. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yao Zhao.

J. Chen, X. Kang, and Y. Liu are with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: isskxg@mail.sysu.edu.cn).

Z. J. Wang is with the Electrical and Computer Engineering Department, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: zjanew@ece.ubc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2438008

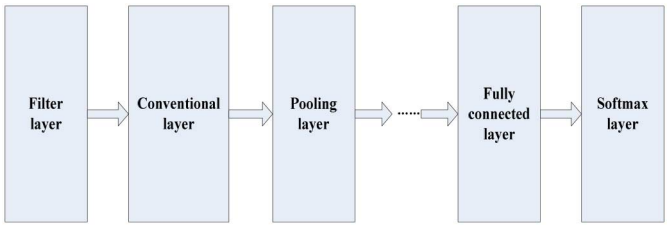

Fig. 1. The proposed CNN for median filtering forensic.

layers, such as convolutional layers, pooling layers and classification layers. At each convolutional layer, the output feature-map usually combines convolutions with multiple inputs. They can capture local dependencies among neighbor elements. The convolutional outputs from all inputs are then transferred into element-wise non-linearity [15]. Pooling can reduce the spatial resolution of each feature-map and translates information into more global one. [16] reported that max pooling can lead to faster convergence and improved generalization while [17] analyzed the theoretical aspect of feature pooling. Via alternating convolutional layer and pooling layer, the output feature vectors are fed into the classification layer. Finally, the classification layer will output the probability of one sample classified into each class through softmax connection.

In our preliminary study, we directly employed conventional CNN models [13] as median filtering forensic models, and they didn't perform well, suggesting that existing CNN models can hardly capture the important statistical forensic properties. In median filtering forensics, since the fingerprint caused by median filtering is heavily affected by image edges and textures, using conventional CNN models directly (i.e., using the raw image pixels as inputs to CNNs) leads to poor performance. Therefore we propose modifying the conventional CNN model by adding a filter layer due to the following intuitive reason: The added filter layer can suppress interference caused by the presence of image edges and textures, and therefore the trace left by median filtering can be successfully exposed. The proposed framework based on CNNs for median filtering forensics is shown in Fig. 1.

Filter Layer: Since in median filtering forensics, using conventional CNN models with the raw image pixels as inputs didn't yield good performances, one additional layer, the filter layer in Fig. 1, is added to the conventional model. Through this filter layer, the median filtering residual (MFR) of an image is obtained. Then the output MFR is fed into conventional networks. The filter layer is important in the proposed method since it can suppress the interference caused by image edges and textures, as shown in [1]. With eliminating/suppressing the interference of irrelevant information (e.g., image edges and textures), the trace left by median filtering can be investigated.

The MFR is defined as follows. Applying the $w \times w$ median filtering window on a test image $x(i,j)$ and obtain the output image $y(i,j)$. The MFR is:

$$\begin{aligned} d(i,j) &= med_w(x(i,j)) - x(i,j) \\ &= y(i,j) - x(i,j) \end{aligned} \tag{1}$$

where w is chosen to be 3 in our work, $x(i,j)$ is original pixel value at point (i,j), $y(i,j)$ is median filter value of $x(i,j)$ and $d(i,j)$ means the MFR, which is the difference between $y(i,j)$ and $x(i,j)$.

Convolutional Layer: A conventional convolutional layer consists of two operations: convolution and non-linearity. The response of a convolutional layer is called feature map. Actually, each feature map is a particular feature representation of the input in a certain area. The convolution operation can be denoted as

$$x_j^l = \sum_{i=1}^{n} x_i^{l-1} * w_{ij}^{l-1} + \mathrm{b}_j^l \tag{2}$$

where $*$ denotes convolution, x_j^l is the j-th output map in layer l, w_{ij}^{l-1}(also called weight) is the trainable convolutional kernel connecting the i-th output map in layer $l-1$ and the j-th output map in layer l, b_j^l is the trainable bias parameter for the j-th output map in layer l. The convolution operation main includes the theory of receptive field and shared weights. Receptive field means each low level feature will be computed from only a subset of the input, it controls the numbers of pixels in connection. Additionally, the share of parameters reduces the number of free variables, hence increases the generalization performance of the network [18]. Following the convolution, the non-linearity operation is obtained by applying an element-wise non-linear activation function (sigmoid, tanh, etc.) to each element of feature maps. The Rectified Linear Units (ReLUs) is used in our work because it can lead to fast convergence in the performance of large models trained on large datasets [19]. Based on equation (2), the operation is expressed as

$$f_{m,n} = \max(x_{m,n}^l, 0) \tag{3}$$

where (m,n) means the pixel index in the feature map, and $x_{m,n}$ stands for the input patch centred at location (m,n).

Pooling Layer: After obtaining feature maps using convolution, we can use all extracted features for classification. However this can be computationally challenging and prone to overfitting. Thus only the mean (or max) value of a particular feature over a region of the image is calculated. The aggregation operation is called pooling. Average pooling and max pooling are two typical pooling methods, which propagate the average and the maximum value within the local region to the next layer respectively. The loss of spatial information is translated to an increasing number of higher level feature representations [16].

Classification Layer: In general, the classification layer consists of a few fully connected layers. When the learned features pass through the first or two fully connected layers, they will be fed to the top layer of the CNNs, where a softmax activation function is used for classification. The back propagation algorithm is used to train the CNN. As described in [13], the weights and the bias can be renewed adaptively in the convolutional and fully connected layers following the error propagation procedure. In this way, the classification result can be fed back to guide the feature extraction automatically and the learning mechanism can be established.

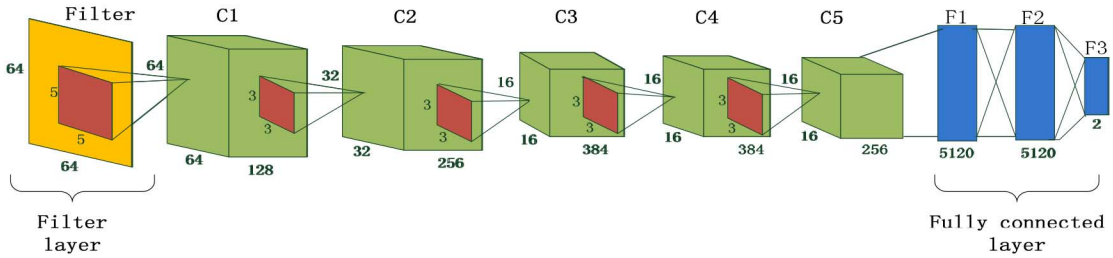

Fig. 2. The framework of the proposed method.

B. Parameters and Settings of Our Modified CNNs

The framework of the proposed model is shown in Fig. 2, where we describe the detailed settings of the architecture, and it is a supplementary explanation to the model in Fig. 1. The architecture contains nine layers. The first layer is a filter layer, the second to the sixth are convolutional layers, the last three are fully-connected layers. In Fig. 2, the feature maps and kernels are marked in green and red respectively. Pooling layers will be explained in the next paragraph.

In this work, we address the challenge of detecting median filtering from a small-sized and compressed image block. We consider two sizes of an input image, i.e. 64×64 and 32×32 pixels. For the sake of brevity, we only explain one size choice in detail. Let us take a gray scale image of size 64×64 as the input to the architecture shown in Fig. 2. Firstly, the filter layer gets the MFR of an image. Then the first convolutional layer convolves them with 128 kernels of size 5×5. The size of the output (C1) is $64 \times 64 \times 128$, which means the number of feature maps is 128 and the resolution of feature maps is "64×64". Then the second convolutional layer takes the output of the first layer (C1) as the input and filters it with 256 kernels of size 3×3. The third, fourth, and fifth convolution layers apply convolutions with 384 kernels of size 3×3, 384 kernels of size 3×3, 256 kernels of size 3×3 respectively. The Rectified Linear Units (ReLUs) is applied to the output of every convolutional layer. Meanwhile, the first, second, and fifth convolutional layers are followed by an overlapping max pooling operation with window size 3×3 and step size 2, which operate on each feature map in the corresponding convolutional layer, and lead to the same number of feature maps with the decreasing spatial resolution. Each of the fully-connected layers (F1 and F2 in Fig. 2) has 5120 neurons. In both fully-connected layers (F1 and F2), a recently-introduced technique, i.e., "dropout" [19], is used. The last fully connected layer (F3) has two neurons. Its output is fed to a two-way softmax.

When the size of an input image is 32×32 pixels, the only difference of the architecture settings is that there is no max pooling layer followed the C1 convolutinal layer. Other settings remain the same as in the 64×64 case.

III. Experimental Results

To evaluate the performance of the proposed model and compare its performance with other methods, we test on a composite image database containing 15352 images. These images are from five widely used image databases: the BOSSbase 1.01 [20], the UCID database [21], the BOSS RAW database [22], the Dresden Image Database [23] and the NRCS Photo Gallery database [24]. Each of the 4 databases contributes 1338 images, and BOSSbase database contributes 10000 images. All images are converted to gray-scale images before any further processing. Each image from the original composite database belongs to the negative class and its median filtered version belongs to the positive class. The training set contains half number of images in each class, while the other half of images compose the testing set. Detection accuracy (Ac) is used to evaluate the performance:

$$A_c = \frac{c}{n} \tag{4}$$

where c is the number of correctly predicted samples and n is the number of total testing images. We compare the proposed model with existing works: the AR method [1], median filter feature (MFF) method [2] and GLF method [4]. We perform SVM training and testing for the three conventional methods. For the proposed model, all experiments are conducted on GPU using the C++ programming language.

A. Detecting Median Filtering from Small and JPEG Compressed Image Blocks

We first crop image blocks with size of 64×64 and 32×32 from the center of a full-resolution image, and then build a corresponding training set and testing set. 3×3 median filtering (MF3 in short form) and 5×5 median filtering (MF5 in short form) are considered in our experiments. The detection accuracy results are reported in Table I. "JPEG_70" denotes that the image without median filtered but JPEG compressed with quality factor (QF) of 70, "MF3 + JPEG_70" denotes that the image with composite operation of median filtering and JPEG compression with QF 70. It is noted that the proposed model performs the best in all cases. Considering that the detection accuracy of the proposed model is about 1%–8% better than that of the state-of-the-art methods in different cases, we believe that the deep learning feature representations are effective.

It is clear that the filter layer for obtaining MFR is important in the proposed CNN model. The experimental results in Table II show that, without the filter layer, i.e., the image pixels are used as the input to the layer C1 directly, the model cannot

TABLE I
DETECTION ACCURACY (%) FOR MEDIAN FILTERING DETECTION AGAINST JPEG COMPRESSION. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Image size	Method	JPEG_70 vs. MF3+JPEG_70	JPEG_90 vs. MF3+JPEG_90	JPEG_70 vs. MF5+JPEG_70	JPEG_90 vs. MF5+JPEG_90
64×64	Proposed CNN	**85.14**	**94.04**	**94.12**	**96.84**
	MFF [2]	78.75	85.61	86.88	89.79
	GLF [4]	83.07	91.06	91.23	94.77
	AR [1]	83.12	90.22	88.64	93.17
32×32	Proposed CNN	**79.42**	**87.71**	**88.65**	**93.21**
	MFF [2]	73.99	80.32	82.49	85.91
	GLF [4]	78.15	85.43	87.28	91.57
	AR [1]	75.63	83.52	80.80	86.26

TABLE II
DETECTION ACCURACY (%) WITH OR WITHOUT MFR

Image size	Method	JPEG_70 vs. MF3+JPEG_70
64×64	With MFR	85.14
	Without MFR	77.92

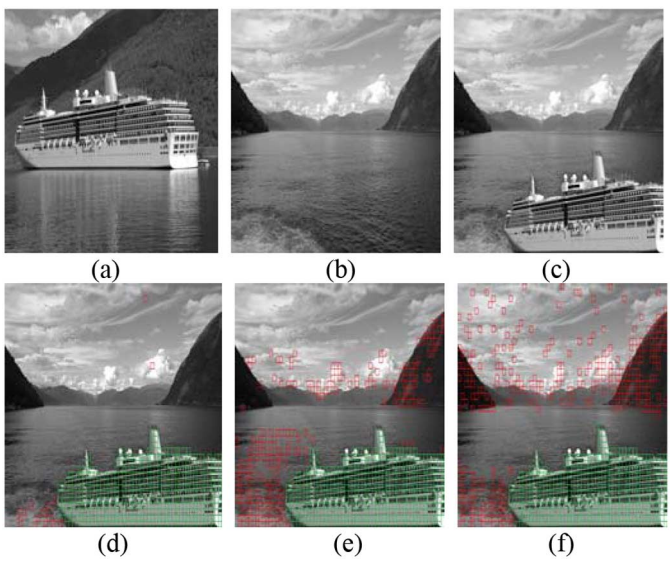

Fig. 3. A cut and paste forgery detection example, showing (a) the median filtered image from which an object is cut; (b) the unaltered image into which the cut object is pasted; (c) the composite image which is JPEG compressed using a quality factor of 90. The detected blocks of boat (the true positives) are marked in green, and other detected blocks outside of the boat (the false alarms) are marked in red;.(d) using the proposed model; (e) using the GLF method [4]; and (f) using the AR method [1].

catch the forensic fingerprint well. Because the results for different median filtering and different JPEG QFs are similar, only the comparisons of 3×3 median filtering (MF3) detection under JPEG 70 compression are shown in Table II.

B. Cut and Paste Forgery Detection

The ability to detect median filtering in low-resolution images and image windows is essential for detecting forgeries when a portion of a median filtered image is inserted into a non-median filtered image. An example of cut-and-paste image forgery and the corresponding forensic detection results are shown in Fig. 3. Fig. 3(a) shows the 3×3 median filtered image from which an object (the boat) was cut. Fig. 3(b) shows the unaltered image into which the cut object was pasted. Fig. 3(c) shows the composite image, which was JPEG compressed using QF 90. In order to detect the forgery, the composite image was first segmented into 64×64 pixel blocks, and then each block was tested for evidence of locally applied median filtering. In this example, each detection method was trained on corresponding training images, i.e. the JPEG 90 compressed images with size of 64×64. Blocks corresponding to median filtering detections were boxed and outlined. Fig. 3(d) shows the result of blockwise detections on the composite image using our proposed CNNs method. Fig. 3(e) shows the result using the GLF method in [4]. Fig. 3(f) shows the result with the AR method in [1]. In Fig. 3, the detect blocks of the boat (the true positives) are outlined in green, and other detected blocks outside of the boat (the false alarms) are marked in red. It is clear that the proposed method achieves high detection rate with the lowest false alarm rate. This example illustrates that the proposed CNN-based method outperforms the three state-of-the-art approaches in the cut-and-paste forgery detection.

IV. CONCLUSION

In this paper, we propose a median filtering forensic method based on deep learning. The contributions are outlined as follows: Different from exiting conventional median forensics techniques, the feature extraction and classification steps are unified in a modified CNN-based model with adding a filter layer, and hierarchical feature representations are learned; Using feature representations learned automatically from a deep learning model, we can achieve better detection accuracy results when compared with the state-of-art methods using handcrafted features. We have demonstrated that the proposed CNN-based method can detect median filtering in small and JPEG compressed image blocks and is able to identify cut-and-paste forgeries well.

REFERENCES

[1] X. Kang, M. C. Stamm, A. Peng, and K. J. Ray Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1456–1468, Sep. 2013.

[2] H. Yuan, "Blind forensics of median filtering in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1335–1345, Dec. 2011.

[3] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *Proc. SPIE, Electron. Imaging, Media Forensics, Security II*, 2010, vol. 7541, pp. 1–12.

[4] C. Chen, J. Ni, R. Huang, and J. Huang, "Blind median filtering detection using statistics in difference domain," in *Proc. Inf. Hiding*, Berkeley, CA, USA, May 2012.

[5] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2010, pp. 89–94.

[6] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[7] W. Q. Luo, J. W. Huang, and G. P. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.

[8] M. C. Stamm and K. J. R. Liu, "Forensic estimation and reconstruction of a contrast enhancement mapping," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, USA, 2010, pp. 1698–1701.

[9] M. C. Stamm and K. J. Ray Liu, "Anti-forensics of digital image compression," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1050–1065, Sep. 2011.

[10] M. Kirchner and R. Bohme, "Hiding traces of resampling in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 582–592, Dec. 2008.

[11] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 448–455.

[12] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1–40, 2009.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition, in," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[15] Y. Liu and X. Yao, "Evolutionary design of artificial neural networks with different nodes," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, pp. 670–675.

[16] D. Scherer, A. Muller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. Int. Conf. Artif. Neural Netw.*, 2010.

[17] Y. Boureau, F. Bach, Y. LeCun, and j. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2010, pp. 2559 –2566.

[18] M. Browne and S. S. Ghidary, "Convolutional neural networks for image processing: An application in robot vision," in *Proc. AI 2003: Advances in Artif. Intell.*, 2003, pp. 641–652.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.

[20] P. Bas, T. Filler, and T. Pevny, "Break our steganographic system:The ins and outs of organizing boss," *Inf. Hiding*, pp. 59–70, 2011.

[21] G. Schaefer and M. Stich, "UCID-An uncompressed color image database," in *Proc. SPIE, Storage Retrieval Meth. Applicat. Multimedia*, 2004, pp. 472–480.

[22] [Online]. Available: http://exile.felk.cvut.cz/boss/BOSSFinal/index.php?mode=VIEW&tmpl=materials

[23] T. Gole and R. Bohme, "Dresden image database for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, Mar. 2010, pp. 22–26.

[24] United States Department of Agriculture. Natural resources conservation service photo gallery. [Online]. Available: http://photogallery.nrcs.usda.gov