Journal of Visual Communication and Image Representation 51 (2018) 162–168

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

Median filtering detection of small-size image based on CNN

Hongshen Tang, Rongrong Ni*, Yao Zhao, Xiaolong Li

Beijing Jiaotong University, Institute of Information Science, China
Beijing Key Laboratory of Advanced Information Science and Network Technology, China

ARTICLE INFO

Keywords:
Median filtering forensics
CNN
Nearest neighbor interpolation

ABSTRACT

Existing median filtering detection methods are no longer effective for small size or highly compressed images. To deal with this problem, a new median filtering detection method based on CNN is proposed in this paper. Specifically, a new network structure called MFNet is constructed. First, for preprocessing, the nearest neighbor interpolation method is utilized to up-sample the small-size images. The property of median filtering can be well preserved by the up-sampling operation and enlarged difference between the original image and its median filtered version can be obtained. Then, the well-known mlpconv structure is employed in the first and second layers of MFNet. With mlpconv layers, the nonlinear classification ability of the proposed method can be enhanced. After that, three conventional convolutional layers are utilized to finally derive the feature maps. The experimental results show that the proposed method achieves significant improved detection performance. Moreover, the proposed method performs well for highly compressed image of size as small as 16×16.

1. Introduction

With the development of image processing technology, it becomes more difficult to identify the authenticity of digital images. Image manipulation as a way of content tampering is widely employed. For example, a portion of one image can be pasted into another. Then, filtering operation is used to cover the pasting traces. Usually, people can hardly distinguish the tampered images from the original images visually. Therefore, the related digital media forensic technologies arise at the historic moment. Currently existing forensic techniques involve the detection of re-sampling [1–3], JPEG compression [4,5], blurring [6,7], contrast enhancement [8], median filtering [9,13–17], image copy [18,19], and so on.

As image processing technology, the operation of median filtering (MF) as a nonlinear operation is widely applicated to remove noise, preserve edges and smooth regions within an image [14]. So the forger can use median filtering to make their fakes appear more realistic, such as destroying statistical traces of blocking artifacts left by the JPEG compression. A lot of effective methods have been proposed to detect MF [13–17]. In [13], Kirchner and Fridrich proposed to use the statistics of pixels value differences as features to detect median-filtered images. In [14], Cao et al. proposed a method by calculating the probability of zero values on the first order pixel difference in textured image regions. These two methods are effective for uncompressed images, however, they cannot provide satisfactory detection results for compressed images. In [15], a feature set containing 44 features has been proposed for MF detection. This method achieves relatively good performance for high quality JPEG images. In [16], considering the first- and the second-order image differences and the non-linear local image correlation, Chen et al. constructed a new 56 dimensional feature set named GLF. Their method is effective when detecting low resolution and JPEG compressed images. In [17], Kang et al. proposed to use the fitted autoregressive model's coefficients as features for median filtering detection. This method is experimentally verified slightly better than [16].

Recently, convolutional neural networks (CNN) have shown remarkable performance in computer vision and natural language processing [10,11,20]. It can learn feature representations and fulfill classification automatically. There are also some applications about forensics using CNN methods [21,27,28]. In [21], Chen et al. firstly applied CNN in median filtering forensics. In their CNN model, the first layer is a filter that accepts an image as the input and outputs its median filtering residual (MFR), then, the CNN framework is used to learn hierarchical representations for further classification. The results show significant improvements compared to [15–17].

Although many methods have been proposed for MF detection, however, they are mostly unable to detect the MF operation on small size and JPEG compressed images effectively; especially when the image size is less than 64×64. In small-size images, the traces of MF are too weak to be detected. There are just little differences between the small original image and the MF tampered image. So detecting the median filtering operation is a challenging task when the image block is

* Corresponding author at: Beijing Jiaotong University, Beijing 100044, China.
E-mail address: rrni@bjtu.edu.cn (R. Ni).

https://doi.org/10.1016/j.jvcir.2018.01.011
Received 31 March 2017; Received in revised form 24 December 2017; Accepted 15 January 2018
Available online 31 January 2018
1047-3203/

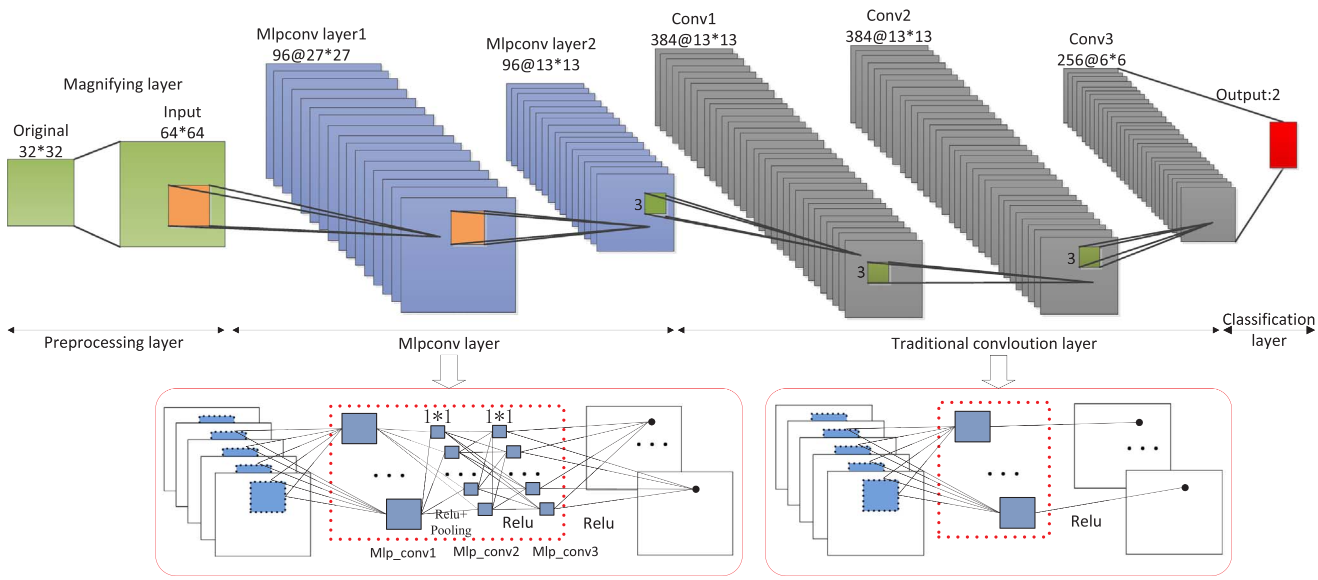


Fig. 1. The proposed framework.

relatively small.

To combat the problem that the traces of MF are weak in small block, we propose to magnify the small size images to enlarge the differences between the median filtered image and its original version. Then we propose a new CNN model (named MFNet) which contains two mlpconv layers [20] with enhanced nonlinear expression ability to learn feature representations and fulfill classification. With the proposed MFNet, median filtering detection of small-size and JPEG compressed image has achieved good results.

The rest of this paper is organized as follows. We will focus on the proposed scheme in Section 2. Then, Section 3 will present the experimental results and performance comparison. Finally, the conclusion is drawn in Section 4.

2. The proposed scheme for MF detection

Median filtering is a nonlinear operation that operates by replacing a pixel's value with the median value of the pixels in a small window surrounding it. It usually produces continuous constant or constant regions [12]. The detection methods in [13–17] are based on manually designed features, then classifiers, such as SVM, are learned to estimate the results. In this paper, we magnify the small images to enlarge the difference between the tampered and the original image, and then use the proposed CNN model to learn the perfect features and make decision automatically. In order to distinguish the designed architecture from the traditional CNN model, we name it MFNet in this paper.

2.1. The framework of MFNet

Convolutional neural networks have shown remarkable performance in computer vision. It generally consists of convolution layer, pooling layer, fully connection layer and classification layer. The convolution layer generates feature maps by linear convolutional filters followed by nonlinear activation functions (sigmoid, tanh, ReLUs, etc.) [20]. Pooling layer can fuse the feature information extracted from the convolution layer, thus more global information can be obtained. The classification layer usually consists of several full connection layers and a softmx function. The feature maps of the last convolution layer are fed into the full connection layers. Then a softmax is used for classification. All the weights of the whole architecture will be updated via back propagation [10].

Identifying whether the images are median filtered or not is a binary classification problem. Hence, it is possible to learn a complex CNN model that both extracts a set of discriminative features and estimates whether the test image is tampered or not. However, directly using the detecting images as the input of CNN model to detect MF is verified not well, so a residual filter layer was added to suppress the interference caused by image edges and textures [21]. We have also tested the existing CNN models developed for image classification as median filtering forensic models. The performance is not satisfying without considering the characteristics of forensic problems. In fact, there is a big difference between image classification and MF forensics. For image classification, the difference between classes is significant. For example, the feature difference between a cat image and a dog image is obvious. For the median filtering forensics, the difference is hardly perceptible because the traces of the MF operation is slight. Thus, the images should be preprocessed to make the difference obviously, or the CNN model should be improved to be suitable for MF detection. We proposed a novel CNN model named MFNet to detect median filtering images, taking into account the ability of capturing the weak traces of MF in small size images. In the MFNet, a preprocessing layer is added to magnify the input image along horizontal and vertical directions, which can enlarge the differences between the tampered image and its original version. Considering the MF's non-linearity, to extract good data representations at high levels of abstraction, we combine two mlpconv layers [20] into the MFNet to enhance the nonlinear ability of the network. Then, the new constructed CNN model is used to extract features from the magnified images and perform classification automatically.

Fig. 1 illustrates the overall architecture of MFNet, which includes one preprocessing layer, two mlpconv layers, three convolution layers and a classification layer. The testing images will be magnified by nearest neighbor interpolation firstly. The MFNet accepts magnified images as input and learn hierarchical features to do classification. For the rest of this section, we will elaborate on several key parts exploited in the design successively.

2.2. The prepocessing layer: magnify the small size image

The image processing layer is employed to magnify testing images, thus, enlarged difference between the tampered image and original version can be obtained. In this layer, we make a magnifying operation with nearest neighbor interpolation, which selects the value of the nearest point and does not consider the values of other neighboring points at all, yielding a piecewise-constant interpolation [22]. Generally, the traces of median filter in small size images are imperceptible; and the modified values of pixels are small as well. Therefore detecting the median filtering operation is a challenging task when the image block is relatively small. Hence, with the nearest neighbor interpolation, we aim to strengthen the weak MF fingerprints and expand the pixel data of the small patches. This can provide a more distinguishable inter-class difference than the original size images to drive the whole network, thus achieve better performance as compared to directly feeding the non-magnified images to CNN model.

Actually, there are many methods for amplifying an image. However, other magnifying methods will introduce new gray values, which can destroy the traces of median filtering. While the value interpolated by nearest neighbor interpolation is completely copied around the pixel. It won't introduce new gray values. Besides, the nearest neighbor interpolation is simple enough and does not introduce extra computation and time consumption.

We evaluate several typical statistical features to show the enlarged difference in the frequency domain. To this end, we randomly select 1000 images (cropped to 32 × 32) from BOSSbase 1.01 [23] and then perform 3 × 3 median filtering and nearest neighbor upsampling to obtain the median filtered image set and their magnified version. Fourier transform is performed on them respectively to transform into frequency domain. Then, we compute the coefficient of variation, skewness and kurtosis separately on the original size image's frequency coefficients as well as the corresponding coefficients with interpolating amplification. The distributions of these statistical features are shown in Fig. 2. The horizontal axis represents the difference between the original size image and its median filtered version. The vertical axis is the difference between the resized image and its resized median filtered version. The statistical difference is defined as

$$Dif = Feature(abs(Fourier(med(x)))) - Feature(abs(Fourier(x))) \tag{1}$$

where $med(\cdot)$ is the median filtering operation, $Fourier(\cdot)$ represents Fourier transform, $abs(\cdot)$ calculates the amplitude of the Fourier coefficients and $Feature(\cdot)$ computes the coefficient of variation, skewness and kurtosis, respectively. From Fig. 2, it is clearly observed that the difference becomes bigger after magnifying than original size, meaning that amplifying the small image can enlarge the difference between the median filtered image and its original version, as described previously. So it is possible to differentiate those images after median filtering from the original ones with a high probability.

2.3. Mlpconv layer: enhance nonlinear ability of the network

In our framework, instead of the entire traditional convolutional layers used in the model, there are two mlpconv layers fused in MFNet. Due to MF's non-linearity, theoretical analysis of the general relation between the input and output distribution of the median filter is highly non-trivial. However, traditional convolutional layer is just a linear convolution. Then a non-linear activation function is applied to the output of every convolutional layer. It has been proved that traditional convolutional layer cannot learn effective enough feature representation of MF. So it is crucial to enhance the non-linear analyzing ability of the network for good data representations at high levels of abstraction. One encouraging news is that Lin et al. [20] proposed an enhanced nonlinear structure called mlpconv layer which can achieve a better abstract representation of the data. Compared with traditional convolution layer, the mlpconv layer is deeper and has more nonlinear operations. This is consistent with the nonlinear characteristics of the median filtering operation. Based on the above consideration, to effectively capture sufficient feature for MF detection, we propose a scheme that adding two mlpconv layers into traditional neural network, as shown in Fig. 1. The nonlinear operations of the mlpconv layers can improve the nonlinear modeling ability of the whole architecture.

Fig. 3 illustrates the difference between traditional convolution layer and mlpconv layer. The mlpconv layer maps the input local patch to the output feature vector with a multilayer perceptron consisting of multiple fully connected layers with nonlinear activation functions [20]. The first layer of the mlpconv layer is a linear convolution operation that its kernel size usually is m × m (3 × 3, 5 × 5, etc.). The second and third layers are 1 × 1 convolutional kernels. At the end of each layer there is a rectified linear unit (ReLU) used as non-linear activation function. Multiple nonlinear operations is the key of mlpconv layer. It can represent complex features at high levels of abstraction. The formula of mlpconv layer [20] is as follows:

$$\begin{aligned} f_{i,j_1}^1 &= \max\left(\sum x_i * w_{j_1}^1 + b_{j_1}, 0\right)\\ &\vdots\\ f_{i,j_n}^n &= \max\left(\sum f_i^{n-1} * w_{j_n}^n + b_{j_n}, 0\right) \end{aligned} \tag{2}$$

Here n is the number of layers in the mlpconv layer, i and j are used to index the channels of the feature map, f is the output map, * denotes convolution, b is the bias to the convolution, w is the filter kernel (weights).

2.4. The rest part of MFNet

Following the two mlpconv layers, there are three traditional convolution layers before classification layer. It is used to further extract and fuse features. There are also three max pooling layers in the network which can be found from Fig. 1 and Table 1. The first and second pooling layers are placed in the mlpconv layers, and the third one is between the conv2 and conv3 layers. Pooling layer can fuse the feature information extracted from the convolution layer, thus the more global information can be obtained. Simultaneously, it will reduce the spatial resolution of the feature map. The feature maps of the last convolution layer are fed into the classification layer. Firstly, they pass through one fully connected layer to converge to two values. That is different from other conventional CNN models that employ two or more fully

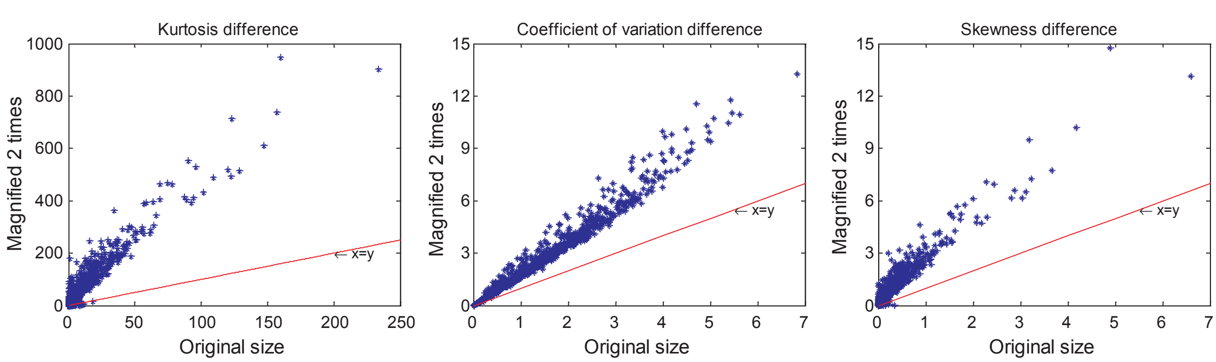
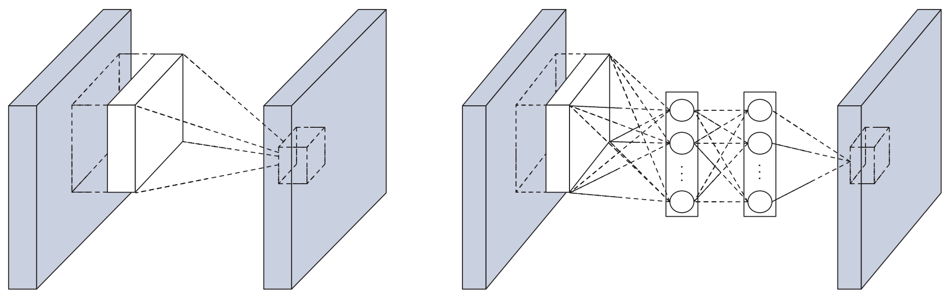


Fig. 2. The differences about kurtosis, coefficient of variation and skewness between the frequency coefficients of median filtered image and its non-operated version. The horizontal axis is about original size, and the image of vertical axis is magnified 2 times.



(a) Traditional convolution layer (b) Mlpconv layer

Fig. 3. Comparison of traditional convolution layer and mlpconv layer. The middle part represent convolution kernel. The traditional convolution layer includes a linear filter while the mlpconv layer includes a micro network [20].

connected layers. This is because the fully connected layers usually occupy too many parameters to be learned, which are prone to overfitting, especially when the training data is not big enough. Then, a softmax activation function "squashes" the two values in the range $[0, 1]$ and guarantees that them sum up to one. Two classes represent the positive (median filtered) and the negative (original), respectively. The well-known back propagation algorithm is used to train the CNN model. So the classification result can be fed back to guide the feature extraction automatically and the learning mechanism can be established [21].

2.5. Detail settings of the MFNet

In brief, we only explain the testing image size of 32×32 in detail. Magnifying 2 times as an example, all parameter settings about the network can be seen in Fig. 1. Before the magnified images entered into the network, they will be cropped to 63×63 firstly. The detail settings of Mlpconv layer1 and layer2 are shown in Table 1. The ReLU is applied to the output of every convolution layer. Meanwhile, the Pool1 and Pool2 layers are followed by a local normalization scheme (LRN) [11] which aids generalization. It is worth noting that there is a pooling layer behind conv3, so the size of feature map is 6×6 in Fig. 1.

We also consider four other sizes of input image, i.e. 16×16, 32×32, 96×96 and 128×128. When the size of an input image is not 64×64, there is just a little difference about the architecture in mlpconv layer1. If the input size is 16×16 or 32×32, the kernel of Mlp_Conv1 will be replaced by 5×5 and there will be no padding, other settings are the same as 64×64 except that 32×32 will be firstly cropped to 31×31. When the input image is 96×96 pixels, the only difference of the architecture settings is that the image will be cropped into 91×91 and there is also no padding. As for the size of 128×128, 119×119 pixels will be obtained by cropping and no padding will be added to the block, the rest settings remain exactly the same as the case of 64×64 input.

The proposed model is implemented by using Caffe. The training parameters of the stochastic gradient descent are set as follows: the batch size is 128, the learning policy is "inv", the momentum value is 0.9, the weight decay is 5×10^{-4} and a fixed learning rate 0.01 over all iterations.

3. Experimental results

We evaluate the proposed scheme on a composite image database containing 14,800 images. These images are mainly taken from four widely used image databases: the BOSSbase 1.01 [23], the UCID database [24], the NRCS Photo Gallery database [25] and the BOSS RAW database [26]. The BOSSbase database contributes 10,000 images, and the rest three databases contribute 1338, 791 and 1543 images, respectively. In order to get enough data, there are also 1128 natural images stemming from [5]. All images are converted to gray-scale images before any further processing. Each image from the original composite database is used to constitute the negative class and its median filtered version serves as the positive class. We randomly selected 3/5 images as the training set, 1/5 as the validation and the rest as the testing set. All the experiments are done with a single GPU card of type GeForce GTX Titan X manufactured by Nvidia. We measure the efficiency of the MFNet by looking at the maximum accuracy rate after convergence.

3.1. The effectiveness of magnifying

In order to validate the magnification actually enlarges the difference between median filtered image and its original version, the last convolution layer's feature maps of the proposed network are displayed in Fig. 4. The left 2 columns' input are 32×32 and non-magnified, the right 2 columns' input are magnified by 2 times. As seen in Fig. 4, the features of non-magnified original images are similar to the median filtered images. There is a sharp contrast in the feature maps of magnified original images compared to magnified median filtered images. It's obvious that more distinct perceptible difference of units are shown in the feature maps between the magnified pairs. This proves that magnifying the small detecting images can enhance the correlation properties of median filtering and expand the difference between original and median filtered.

3.2. Verifying the validity of the MFNet without magnifying layer

Before conducting the experiments, it's important to evaluate the effectiveness of mlpconv layer. By adding two mlpconv layers in the model, we achieve the accuracy of 84.10% which is the best case in the

Table 1
The detail settings of Mlpconv layer1 and layer2.

Input size	Mlpconv layer1				Mlpconv layer2			
	Mlp1_Conv1	Pool1	Mlp1_Conv2	Mlp1_Conv3	Mlp2_Conv1	Pool2	Mlp2_Conv2	Mlp2_Conv3
64 × 64	Kernel:11 Stride:1 Pad:1 Num_out: 96	Kernel:3 Stride:2	Kernel:1 Stride:1 Num_out: 96	Kernel:1 Stride:1 Num_out: 96	Kernel:5 Stride:1 Pad:2 Num_out:256	Kernel:3 Stride:2	Kernel:1 Stride:1 Num_out:96	Kernel:1 Stride:1 Num_out: 96

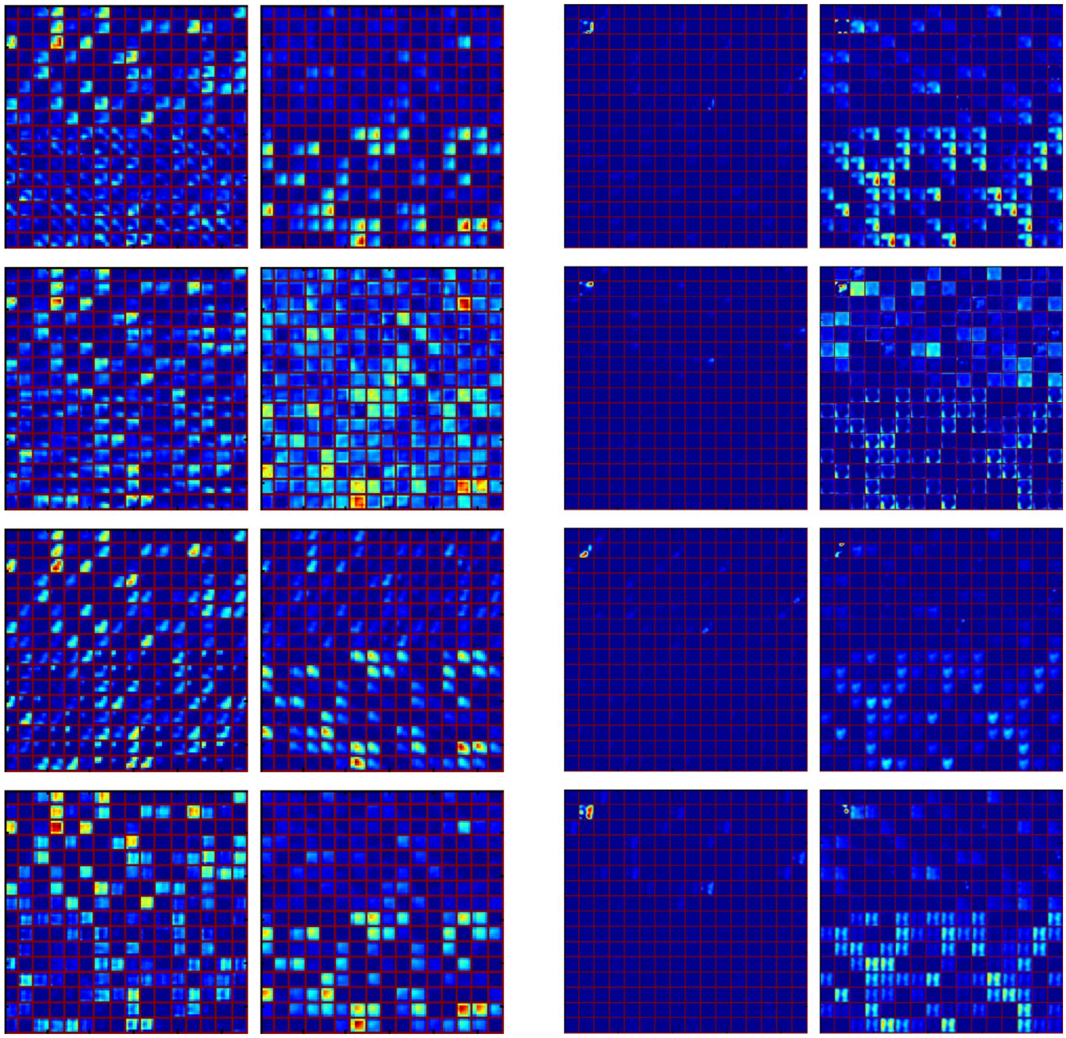


Fig. 4. The last convolution layer's feature maps of four testing images, the left 2 subgraphs's input are non-magnified, the right 2's input are magnified by 2 times. And the left one is the original, the right one is median filtered.

Table 2
The experimental results about Mlpconv layers (JPEG70 vs. MF3 + JPEG70).

Method	No mlpconv	One mlpconv	Two mlpconv	Three mlpconv	Four mlpconv
Accuracy	82.53	83.77	84.10	83.85	83.54

results of Table 2. If more than three mlpconv layers are added into the network, the performance become worse than the case of two mlpconv layers. For the case of three mlpconv layers, the model is deeper and the gradient information for back propagating may have some loss. What's more, in our experiments, we don't have enough labeled training data for such a deep net. Therefore, only two mlpconv layers are fused in our scheme.

We also compared the MFNet with Chen's method [21]. The detailed setting of data is similar to [21]. Except median filtering residual (MFR) which is generated by computing the difference between a testing image and its median filtered version [21], we also considered the testing image without MFR as input of our proposed model. The results can be seen from Table 3. "JPEG90" denotes that the image without median filtering but JPEG compressed with quality factor of 90, "MF5 + JPEG90" denotes that the image with composite operation of 5 × 5 median filtering and JPEG compression with quality factor 90. "MFR" and "image" denote that median filtering residual and the testing image pixels without MFR are used as input of CNN model

Table 3
Detection accuracy compared with [21].

Image size	Input	Method	JPEG70 vs. MF3 + JPEG70	JPEG70 vs. MF5 + JPEG70	JPEG90 vs. MF3 + JPEG90	JPEG90 vs. MF5 + JPEG90
64 × 64	MFR	Chen [21]	86.62	92.37	93.28	94.56
		Proposed	89.49	93.81	**95.30**	**95.62**
	Image	Proposed	**89.96**	**94.20**	95.18	95.51
32 × 32	MFR	Chen [21]	79.01	84.50	88.39	90.24
		Proposed	82.30	87.33	90.91	92.97
	Image	Proposed	**84.10**	**89.58**	**91.50**	**93.57**

Bold values are designed to highlight the better results.

Table 4
Detection accuracy for different magnification times.

Method	Magnification Times	JPEG70 vs. MF3 + JPEG70	JPEG70 vs. MF5 + JPEG70	JPEG90 vs. MF3 + JPEG90	JPEG90 vs. MF5 + JPEG90
Proposed	1 (32 × 32)	84.10	89.58	91.50	93.57
	2 (64 × 64)	85.23	91.67	93.06	95.71
	3 (96 × 96)	85.67	91.87	93.34	96.23
	4 (128 × 128)	85.68	91.92	93.95	96.32
Chen [21]	MFR (32 × 32)	79.01	84.50	88.39	90.24

Table 5
Verification results on 16 × 16 and 64 × 64 patches.

Magnification times	JPEG70 vs. MF3 + JPEG70	JPEG70 vs. MF5 + JPEG70	JPEG90 vs. MF3 + JPEG90	JPEG90 vs. MF5 + JPEG90
1 (64 × 64)	89.96	94.20	95.18	95.51
2 (128 × 128)	91.07	94.86	95.97	96.62
1 (16 × 16)	73.46	81.87	81.28	86.33
2 (32 × 32)	77.04	84.86	86.42	90.67

separately. It's noted that our model performs better than Chen's [21]. Almost in all cases of our model, using the image pixels without MFR as input performs better than MFR as input except for JPEG90 with the size of 64 × 64. This proves that our model has strong nonlinear analyzing ability. In the following experiment, we will use the image pixels as the input.

3.3. Detection on nearest neighbor interpolated images

We mainly consider the image blocks with the size of 32 × 32. Firstly, the 32 × 32 image patches are cropped from the center of full-resolution images. Then 3 × 3 median filtering (MF3) and 5 × 5 median filtering is operated on the small image patches separately. And JPEG compression is considered in our experiments. For these small image patches, 2, 3 and 4 times are magnified separately by nearest neighbor interpolation.

The detection accuracy results are reported in Table 4, which indicate that the image magnification method by nearest interpolating is helpful for small size MF detection. Compared to original patches, the detection accuracy is increased by 1.13–2.14% when two times magnified. We believe that our proposed model and the nearest neighbor interpolation can extract MF feature effectively.

3.4. The effectiveness on other size patches

To confirm the effectiveness of our scheme, we also conduct the experiments on 16 × 16 and 64 × 64 patches. Since there is a little promotion in experiment 2 when 3 or 4 times magnified, we just design the experiments on two times magnified. The results are presented in Table 5.

Specially, for the 16 × 16 patches, a more remarkable improvements can be obtained. That provides us a reliable method to detect MF when the manipulation occurs in small area of one image. The performances on 64 × 64 are also very good. Compared to non-magnified patches, magnifying improves the accuracy by 0.66–1.11%.

4. Conclusion

In this paper, we propose to use the nearest neighbor interpolation to magnify the small-size testing images, so that we can enlarge the difference between the original images and its median filtered versions to better distinguish the small size tampered images from unaltered images. We also propose an effective CNN model called MFNet which is different from traditional architecture. By using two mlpconv layers in front of the model, the more nonlinear operations of the mlpconv layers can enhance the nonlinear analyzing ability of the whole architecture which is consistent with the nonlinear characteristics of the median filtering operation. We have evaluated the performance of our MFNet using different size images. Results show the effectiveness of magnifying and MFNet on median filtering detection. We believe that the magnifying idea is also useful for other digital image forensics.

Acknowledgements

This work was supported in part by the National Key Research and Development of China (2016YFB0800404), National NSF of China (61332012, 61672090), Fundamental Research Funds for the Central Universities (2015JBZ002).

References

[1] A.C. Popescu, H. Farid, Exposing digital forgeries by detecting traces of resampling, IEEE Trans. Signal Process. 53 (2) (Feb. 2005) 758–767.
[2] M. Kirchner, On the detectability of local resampling in digital images, Proceedings of SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Contents, Dover, NY, 200868190F–68190F-11.
[3] Y.T. Kao, H.J. Lin, Wang, C.W. Wang, Y.C. Pai, Effective detection for linear up-sampling by factor of fraction, IEEE Trans. Image Process. 21 (8) (2012) 3443–3453.
[4] R. Neelamani, R. De Queiroz, Z. Fan, R. Baraniuk, Jpeg compression history estimation for color images, IEEE Trans. Image Process. 15 (8) (2006) 1365–1378.
[5] W.Q. Luo, J.W. Huang, G.P. Qiu, Jpeg error analysis and its applications to digital image forensics, IEEE Trans. Inf. Forensics Secur. 5 (3) (Sep. 2010) 480–491.
[6] P. Kakar, S. Natarajan, W. Ser, Detecting digital image forgeries through inconsistent motion blur, in: Proceedings of IEEE ICME, 2010, pp. 486–491.
[7] S. Bayram, I. Avcubas, B. Sankur, N. Memon, Image manipulation detection, J. Electron. Imag. 15 (4) (2006) 04110201–04110217.
[8] M.C. Stamm, K.J.R. Liu, Forensic detection of image manipulation using statistical intrinsic fingerprints, IEEE Trans. Inf. Forensics Secur. 5 (3) (2010) 492–506.
[9] X. Kang, M.C. Stamm, A. Peng, K.J.R. Liu, Robust median filtering forensics using an autoregressive model, IEEE Trans. Inf. Forensics Secur. 8 (9) (2013) 1456–1468.
[10] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[11] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2) (2012) 1097–1105.
[12] A.C. Bovik, Streaking in median filtered images, IEEE Trans. Acoust. Speech Signal Process. 35 (4) (Apr. 1987) 493–503.
[13] M. Kirchner, J. Fridrich, On detection of median filtering in digital images, in: k. liu (Ed.), Proceedings of SPIE, Electron. Imaging, Media Forensics, Security II, vol. 7541, 2010, pp. 1–12.
[14] G. Cao, Y. Zhao, R. Ni, L. Yu, Forensic detection of median filtering in digital images, Proceedings of IEEE ICME, 2010, pp. 89–94.
[15] H.D. Yuan, Blind forensics of median filtering in digital images, IEEE Trans. Inf. Forensics Secur. 6 (4) (2011) 1335–1345.
[16] C. Chen, J. Ni, J. Huang, Blind detection of median filtering in digital images: a difference domain based approach, IEEE Trans. Image Process. 22 (12) (2013) 4699–4710.
[17] X. Gui, M.C. Stamm, A. Peng, K.J. Ray Liu, Robust median filtering forensics using an autoregressive model, IEEE Trans. Inf. Foren. Sec. 8 (9) (2013) 1456–1468.
[18] L. Zhou, Y.L. Wang, Q.M.J. Wu, C.N. Yang, X.M. Sun, Effective and efficient global context verification for image copy detection, IEEE Trans. Inf. Forensics Secur. 12

(1) (2017) 48–63.
[19] Jian Li, Xiaolong Li, Bin Yang, Xingming Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 507–518.
[20] M. Lin, Q. Chen, S. Yan, Network in network, in: Proceedings of ICLR, 2014.
[21] J. Chen, X. Kang, Y. Liu, Z. Wang, Median filtering forensics based on convolutional neural networks, IEEE Sig. Process. Lett. 22 (11) (2015) 1849–1853.
[22] [Online] < https://en.wikipedia.org/wiki/Nearest_neighbor > .
[23] P. Bas, T. Filler, T. Pevný, Break our steganographic system: the Ins and Outs of organizing BOSS, in: International Conference on Information Hiding, vol. 96, 2011, pp. 59–70.
[24] G. Schaefer, M. Stich, UCID: an uncompressed color image database, in: Proceedings of SPIE, Storage Retrieval Methods and Applications for Multimedia, 2004, pp. 472–480.
[25] [Online] < http://photogallery.nrcs.usda.gov > .
[26] [Online] < http://exile.felk.cvut.cz/boss/BOSSFinal/index.php?mode=VIEW&tmpl=materials > .
[27] Bayar B, Stamm MC. A deep learning approach to universal image manipulation detection using a new convolutional layer, in: ACM Workshop on Information Hiding and Multimedia Security, Vigo, Spain, June 2016.
[28] Amel Tuama, Marc Chaumont, Frederic Comby, Camera model identification with the use of deep convolutional neural networks, in: IEEE International Workshop on Information Forensics and Security, Abu Dhabi, UAE, December 4–7, 2016.