SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Image Processing Operations Identification via Convolutional Neural Network

Bolin CHEN[1], Haodong LI[2], Weiqi LUO[1*] & Jiwu HUANG[2]

[1]*Guangdong Key Laboratory of Information Security Technology, School of Data and Computer Science, Sun Yat-sen University, Guangzhou* 510006*, P.R. China;*
[2]*College of Information Engineering, Shenzhen University, Shenzhen* 518052*, P.R. China*

Appendix A Supplemental Figures and Tables

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Figure A1 The four high-pass filters used in the proposed model.

Table A1 Parameters of the image processing operations used in the experiments

Operation type	Parameters
Gamma correction(GC)	γ: 0.5, 0.6, 0.7, 0.8, 0.9, 1.2, 1.4, 1.6, 1.8, 2.0
Histogram equalization(HE)	n/a
Unsharp masking sharpening(UM)	σ: $0.5-1.5$; λ: $0.5-1.5$
Mean filtering(MeanF)	window size: 3×3, 5×5, 7×7
Gaussian filtering(GF)	window size: 3×3, 5×5, 7×7; σ: 0.8 - 1.6
Median filtering(MedF)	window size: 3×3, 5×5, 7×7
Wiener filtering(WF)	window size: 3×3, 5×5, 7×7
Scale(Sca)	up-sampling: 1, 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 (%) down-sampling: 1, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45 (%)
Rotation(Rot)	degree: 1, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45 (°)
JPEG	quality factor: $75-99$
JPEG 2000(JP2)	compression ratio: $2.0-8.0$

* Corresponding author (email: luoweiqi@mail.sysu.edu.cn)

Table A2 Detection accuracy (%) for different image processing operations. The best accuracy for each operation is highlighted.

	GC	HE	UM	MeanF	GF	MedF	WF	Sca	Rot	JPEG	JP2	Avg
Proposed	**96.4**	**99.9**	**99.2**	99.9	**100**	**100**	99.9	**97.4**	**99.2**	**99.7**	**99.8**	**99.2**
Li [1]	94.3	**99.9**	97.1	**100**	**100**	**100**	**100**	96.2	99.0	98.8	**99.8**	98.6
Chen [2]	79.2	98.5	96.5	99.9	99.8	99.8	99.9	93.4	98.3	98.5	99.1	96.6
Bayar [3]	64.5	98.15	91.7	99.8	99.9	99.2	99.7	94.0	97.3	98.0	98.3	94.6
Xu [4]	76.0	97.6	94.9	99.9	99.9	99.9	99.8	95.1	98.6	99.0	99.7	96.4

Table A3 Confusion matrix for identifying image processing operations using the proposed method. The asterisk (*) here means that the corresponding value is less than 1%.

Act/Pred	Orig	GC	HE	UM	MeanF	GF	MedF	WF	Sca	Rot	JPEG	JP2
Orig	**97.5**	1.4	*	*	*	*	*	*	*	*	*	*
GC	4.0	**95.1**	*	*	*	*	*	*	*	*	*	*
HE	*	*	**99.3**	*	*	*	*	*	*	*	*	*
UM	*	*	*	**98.6**	*	*	*	*	*	*	*	*
MeanF	*	*	*	*	**99.1**	*	*	*	*	*	*	*
GF	*	*	*	*	*	**99.4**	*	*	*	*	*	*
MedF	*	*	*	*	*	*	**99.8**	*	*	*	*	*
WF	*	*	*	*	*	*	*	**99.3**	*	*	*	*
Sca	1.5	*	*	*	*	*	*	*	**98.3**	*	*	*
Rot	*	*	*	*	*	*	*	*	*	**99.9**	*	*
JPEG	*	*	*	*	*	*	*	*	*	*	**99.2**	*
JP2	*	*	*	*	*	*	*	*	*	*	*	**99.5**

Table A4 Average results (%) along the diagonal line of the corresponding confusion matrix. The best accuracy of the five methods is highlighted.

Method	Proposed	Li [1]	Chen [2]	Bayar [3]	Xu [4]
Accuracy	**98.8**	96.3	91.5	87.2	85.9

Table A5 Average classification accuracies(%) for different image sizes. The best accuracy of the five methods is highlighted.

Image size	256	128	64	32
Proposed	**98.8**	**96.5**	**91.3**	**81.5**
Li [1]	96.3	92.4	86.1	75.4
Chen [2]	91.5	85.5	75.2	58.8
Bayar [3]	87.2	80.3	69.3	48.8
Xu [4]	85.9	80.0	69.5	57.1

Table A6 Average classification accuracies(%) for different image sizes on the BOSSbase.

Image size	256	128	64	32
Proposed	98.1	95.8	90.3	80.6

Appendix B Validation of the Designs of the Proposed Model

Based on previous studies, most CNN based models are highly dependent on the investigated problem. In this paper, we aim to identify various image processing operations via convolutional neural network. In this section, we try to validate the rationality of the proposed model. Three parts of the proposed model have been considered, including the high-pass filter bank, the channel expansion layer and the last pooling layer. In addition, we also consider the activation functions used in the proposed model.

Appendix B.0.1 *High-Pass Filter Bank*

As described previously, four fixed high-pass filters were used to get the image residuals of the input images. Some related works, such as [5] and [6], have used high-pass filters to initialize the first convolutional layer in their models, and to make the weights of the filters trainable. In this experiment, therefore, we consider the four different cases for the high-pass filter bank: using the fixed four filters (denoted as "untrainable"), using the four filters for initialization ("trainable"), using the same size of filters with random initialization ("random") and removing the high-pass filter bank altogether ("no high pass", i.e. just copying the input image to the four channels).

The experimental results are shown in Figure B1 (left); we can observe that the proposed method (i.e. using fixed high-pass filters) achieves the best performance. The trainable high-pass filter bank can achieve a similar detection accuracy to our method when the number of iterations is large. However, the detection accuracy seems unstable during the early stages of the training. Replacing the high-pass filter bank with a random initialized convolutional layer shows poor performance (lower than 90%). In addition, its performance seems unstable during the entire training stage. Removal of the high-pass filter has the lowest accuracy of all these approaches, and this converges much more slowly than the others.

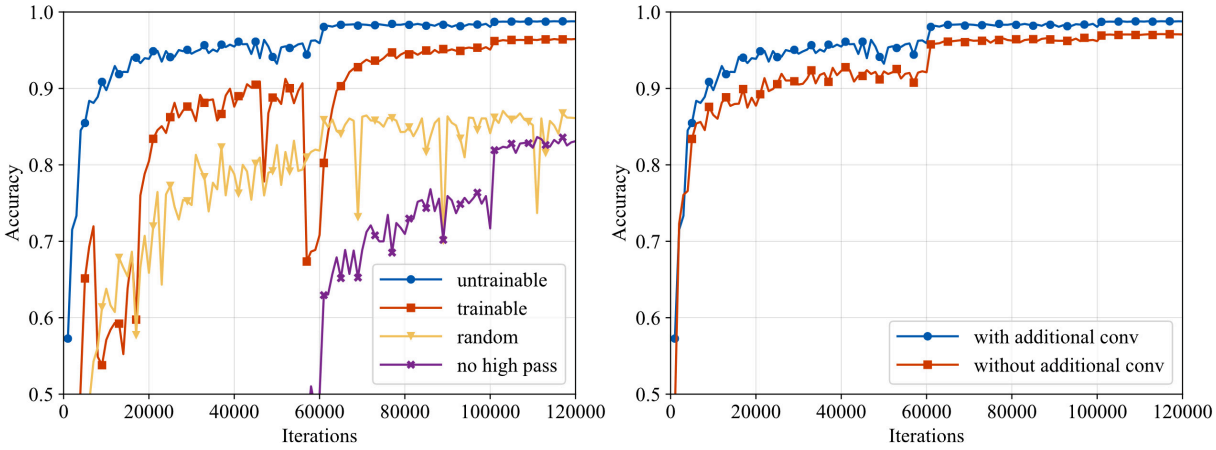

Figure B1 **Left**: Comparison of different settings in the high-pass filter bank. **Right**: Comparison of with/without the channel expansion layer.

Appendix B.0.2 *Channel Expansion Layer*

The channel expansion layer, the convolutional layer following the high-pass filter bank, aims to mix up the residuals produced by the high-pass filter bank and increase the channel number. This layer is used to combine the information of different residuals and provide more input features for the six subsequent layer groups. In this experiment, we present the detection performance when this layer is removed. The experimental results are shown in Figure B1 (right); we can observe that the channel expansion layer slightly improves the detection performance. In addition, we add a pooling layer after the channel expansion layer, meaning that the network has seven similar groups of layers. Our experiments show that this leads to worse performance(accuracy: 96.35%).

Appendix B.0.3 *Last Pooling Layer*

The pooling layer is an important component of a CNN. Usually, CNNs adopt pooling with a stride of two to downsample input feature maps. Most CNN-based methods in image forensics and steganalysis, such as [2, 3, 7] simply use this type of pooling layer throughout the whole network. Only a few works, such as [4], use global average pooling (GAP) in the last pooling layer. Unlike a conventional pooling layer, GAP aims to reduce the input feature maps to a size of 1×1. In our method, we also adopt GAP in the last pooling layer.

In this experiment, we evaluate different pooling methods for the last pooling layer, including max pooling with a stride of two, average pooling layer with a stride of two, and GAP. Please note that replacing that GAP with other pooling layers need to change the fully connected layer correspondingly. The comparative results are shown in Figure B2 (left); it can be seen that the GAP works better than the two other methods, while the common max pooling has the worst performance.

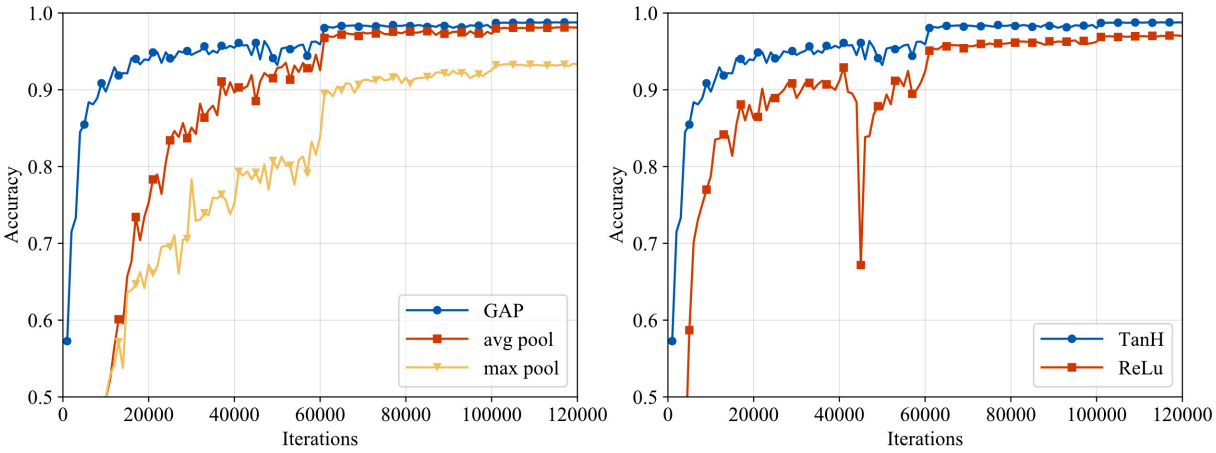

Figure B2 **Left**: Comparison of different last pooling layers. **Right**: Comparison of different activation functions.

Although average pooling outperforms max pooling in the last pooling layer, it does not mean that the use of average pooling instead of max pooling in the whole network will show better performance. In fact, according to our experiments, replacing all the max pooling in the network with average pooling means the training is stalled from the beginning.

Appendix B.0.4 *Activation Functions*

The activation function is another important issue in CNN, and the commonly used activation functions include Sigmoid, TanH and ReLu [8]. In this experiment, we evaluate the proposed model using these three activation functions. The experimental results are not shown in Figure B2 (right); the results for the Sigmoid function in this figure, since using the Sigmoid stalls the learning at the beginning, meaning that the Sigmoid function is useless in the proposed model. From Figure B2 (right), we can observe that TanH performs better and is more stable than ReLu.

It is well known that ReLu usually produces sparser features than TanH. In many tasks in computer vision, the key features [9] distinguishing different classes of objects tend to be prominent, and ReLu usually works better than TanH in these cases. However, the investigated forensic problem is quite different, since any image processing operation will introduce artifacts into the image as a whole rather than the local region within an image. Thus, it is expected that TanH would be better than ReLu, which fits our experimental results very well.

References

1 Li H D, Luo W Q, Qiu X Q, et al. Identi?cation of various image operations using residual-based features. IEEE Trans Circuits Syst Video Technol, 2018, 28: 31-45
2 Chen J S, Kang X G, Liu Y, et al. Median ?ltering forensics based on convolutional neural networks. IEEE Signal Process Lett, 2015, 22: 1849-1853
3 Bayar B, Stamm M C. A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of ACM Workshop on Information Hiding and Multimedia Security, Vigo, 2016. 5-10
4 Xu G S, Wu H Z, and Shi Y Q. Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett, 2016, 23: 708-712
5 Ni J Q, Ye J, and Yi Y. Deep learning hierarchical representations for image steganalysis. IEEE Trans Inf Forensics Security, 2017, 12: 2545-2557
6 Rao Y and Ni J Q. A deep learning approach to detection of splicing and copy-move forgeries in images. In: Proceedings of IEEE International Workshop on Information Forensics and Security, Abu Dhabi, 2016. 1-6
7 Qian Y L, Dong J, Wang W, et al. Deep learning for steganalysis via convolutional neural networks. SPIE Media Watermarking, Security, and Forensics, 2015, 9409: 94090J-94090J
8 Nair V and Hinton G E. Rectified linear units improve restricted boltzmann machines. In: Proceedings of IMLS International Conference on Machine Learning, Haifa, 2010. 807-814
9 Glorot X, Bordes A, and Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of PMLR International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, 2011. 315-323