Multimedia Systems (2021) 27:363–370
https://doi.org/10.1007/s00530-020-00697-y

SPECIAL ISSUE PAPER

Resampling parameter estimation via dual-filtering based convolutional neural network

Lin Peng[1] · **Xin Liao**[1,2] · **Mingliang Chen**[3]

Received: 1 July 2020 / Accepted: 20 September 2020 / Published online: 18 October 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
Resampling detection is an important problem in image forensics. Several exiting approaches have been proposed to solve it, but few of them focus on resampling parameter estimation. Especially, the estimation of downsampling scenarios is very challenging. In this paper, we propose a dual-filtering based convolutional neural network (CNN) to extract features directly from the images. First, we analyze the formulation of resampling parameter estimation and reformulate it as a multi-classification problem by regarding each resampling parameter as a distinct class. Then, we design a network structure based on the preprocessing operation to capture the specific resampling traces for classification. Two parallel filters with different highpass filters are deployed to the CNN architecture, which enlarges the resampling traces and makes it easier to achieve resampling parameter estimation. Next, concatenating the outputs of the two filters by a "concat" layer. Finally, the experimental results demonstrate our proposed method is effective and has better performance than state-of-the-art methods in resampling parameter estimation.

Keywords Image forensics · Resampling detection · Parameter estimation · Convolutional neural network · Dual-filtering

1 Introduction

Digital images play an important role in disseminating information, but it is very easy to manipulate images due to powerful image processing software. Currently, the authenticity of digital images has been doubtful. In reality, there are various methods of image tampering, such as rotation, scaling, clipping, synthesis and JPEG compression. Therefore, image forgery detection is significant and digital image forensics technology has become a meaningful research topic [1, 2].

In recent years, several remarkable works in image forensics have been reported for detecting different tampering operations, such as copy-move detection [3], splicing detection [4], median filtering detection [5], resampling detection [6] and operator chain detection [7]. Generally speaking, specific tampering traces are corresponding to different tampering operations. Thus, the majority of tampering detection approaches are based on inherent statistical characteristics. These contributions made in image forensics protect the authenticity and integrity of images.

Resampling is a common image processing manipulation in daily life, which is mainly used to satisfy the actual needs for the sizes of images. On the one hand, an image may be continuously adjusted to achieve the best visual quality by upsampling or downsampling. On the other hand, image tampers probably apply resampling to a specific area of the image for covering the tampering traces. Resampling detection can enable us to identify whether there is a tampered history trace. Therefore, resampling detection is a significant part of digital image forensics, and the existing resampling detection methods mainly depend on detecting the resampling period introduced by interpolation algorithms [8, 9].

The values of operation parameters will affect the extent of image tampering [10]. Specifically, the resampling parameter decides the size of an image after resampling. By resampling parameter estimation, we can realize roughly the size of the original image and the actual tampering extent. Thus, several works have been already

✉ Xin Liao
xinliao@hnu.edu.cn

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Science, Beijing 100093, China

[3] Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA

published for estimating resampling parameters. However, the performance of methods by manually extracting features is not very satisfactory. Recently, with the development of deep learning, convolutional neural networks (CNNs) which can automatically extract features are widely used for detecting various tampering operations in image forensics. For instance, Chen et al. presented a CNN model for median filtering detection [11], Lyu et al. proposed rotation-invariant CNN models for image enhancement forensics [12] and Liao et al. designed a two-stream convolutional neural network to detect image operator chain [13]. The superiority of deep learning approaches has been shown in these studies and the thought is applied to the proposed method as well.

In this paper, we consider resampling parameter estimation as a multi-classification problem by partitioning each resampling parameter into diverse subsets which represent different classes. The input images are processed simultaneously by dual-filtering consisting of a high-order highpass filter and a low-order highpass filter for suppressing the interference of image contents and highlighting resampling traces. Subsequently, we use the idea of feature fusion to concatenate the outputs of the two filters and conduct convolution training to extract the high-level feature of resampling traces. We experimentally demonstrate our proposed CNN-based parameter estimation method can identify each parameter and has better performance than state-of-the-art methods.

The main contributions of this paper include:

(1) We propose an end-to-end resampling forensics framework based on convolutional neural network. It can help us automatically learn the internal features from resampled images, achieving resampling parameter estimation. Moreover, the model also has high convergence speed and acceptable computation complexity.

(2) A dual-filtering structure is introduced to suppress the deviation caused by image contents and strengthen resampling traces, which achieves a complementary function in detecting resampling parameters.

(3) Experimental results demonstrate that our proposed model outperforms some state-of-the-art methods on resampling forensics. It obtains significant performance both in the upsampling and downsampling scenario.

The rest of this paper is organized as follows. Section 2 discusses several related works in resampling forensics. Section 3 formulates the resampling parameter estimation problem and Sect. 4 describes our network architecture in detail and provides the computation complexity analysis. In Sect. 5, we experimentally evaluate the performance of our proposed approach. In Sect. 6, we further highlight the main contributions of the proposed scheme and point out the drawbacks. Finally, we make a conclusion.

2 Related works

There have been several researches on resampling forensics until now. Early in [8], Popescu and Farid initiated the study about resampling forensics and proposed a method to detect resampling traces by utilizing the correlation between adjacent pixels without any digital watermark or digital signature, namely expectation-maximization (EM) algorithm. However, the EM algorithm has a strong dependency on the values of initial parameters which affect the convergence speed and final performance. Besides, this algorithm works poorly on downsampling detection. Then, Kirchner presented a spectrum analysis detector based on the EM algorithm that utilizes the variance of prediction residue [9]. His detector is a linear predictor, achieving lower computational complexity and better performance. Gallagher [16], Mahdian and Saic [17] detected resampling traces by using the specific statistical changes on the covariance structure of signals. Kirchner in [18] proved periodic interpolation artifacts of resampled images could be detected by a series of tailored row and column predictors.

Apart from the above resampling forensics works, Vázquez-Padín et al. [19] employed an optimum estimator to address the resampling factor estimation following the maximum likelihood criterion. Feng et al. proposed a new method to derive a 19-D feature vector used as input to an SVM-based classifier by examining the normalized energy density [6]. In [20], Zhu et al. presented a learning-to-rank framework to determine the image scaling factor based on the normalized energy density features and moment features. They estimated resampling parameters through a well-trained classifier based on handcrafted features. However, the majority of resampling algorithms are unable to carry out estimating parameter estimation efficiently. Moreover, since the characteristics of the downsampling scenario are much weaker, downsampling parameter estimation is a challenging problem.

Recently, there are many parameter estimation and identification studies based on the nonlinear systems. In [21], the authors identified successfully all of the unknown parameters in systems by implementing the hybrid modified function projective synchronization. Zhang et al. [22] further investigated the special module-phase synchronization. In addition, CNN is applied to image forensics to achieve classification by automatically capturing features instead of manually extracting features [23, 24]. Most of the existing forensics methods are based on the conventional approach and deep learning. Among these CNN-based forensics methods, Stamm et al. designed a kind of network which uses a new type of convolutional layer,

called constrained convolutional layer, for resampling parameter estimation [25]. The constrained CNN model achieved a breakthrough compared to previous methods. Nevertheless, such works mentioned above are not so satisfactory on downsampling forensics. It is challenging for our work to improve the accuracy of parameter estimation in the downsampling scenario.

3 Problem formulation

In this section, we focus our analysis on the formulation of resampling parameter estimation. Without loss of generality, we take the notation I as the original image and the notation I' as the resampled image. Mathematically, resampling involves an interpolation process and can be regarded as the convolution of the original image I and the interpolation function H that corresponds to the specific interpolation method.

$$I' = I * H, \tag{1}$$

where $*$ is the convolution operator and H denotes the bilinear interpolation function in our paper. The relationship between the original image and the resampled image can be expressed as

$$I' = r(I, s), s \in S, \tag{2}$$

where r indicates the resampling operation and S denotes the set of possible resampling parameters. s is one of the set S with $s > 1$ for upsampling and $s < 1$ for downsampling. In our study, we assume that S is discretely ordered and within a certain range.

We specify the resampling parameter s belongs in $\{s_1, s_2, s_3, \ldots, s_T, s_{T+1}\}$. Then, the resampling parameter set S can be divided into disjoint subsets $\{\omega_1, \omega_2, \omega_3, \ldots, \omega_T, \omega_{T+1}\}$.

$$\omega_t = \{s | x_t \le s < x_{t+1}\}, \tag{3}$$

where $t = 1, \ldots, T$ and

$$\omega_{T+1} = \{s | x_T < s \le x_{T+1}\}, \tag{4}$$

where x_t denotes the boundaries of each subset, x_1 is the smallest parameter s_1 and x_{T+1} is the largest parameter s_{T+1} in S. As mentioned above, the parameter of a resampled image from training data is definitely one of S. Thus, each resampling parameter s_i is corresponding to a subset ω_i.

$$s_1 \in \omega_1, s_2 \in \omega_2, \ldots, s_{T+1} \in \omega_{T+1}, \tag{5}$$

That is, all subsets ω_i constitute a minimal cover for S.

$$\bigcup_{i=1}^{T+1} \omega_i = S. \tag{6}$$

Subsequently, we specify each subset ω_i is marked as a class label l_i. $T + 1$ class labels $l_1, l_2, \ldots, l_{T+1}$ can be obtained.

$$\omega_1 \to l_1, \omega_2 \to l_2, \ldots, \omega_{T+1} \to l_{T+1}. \tag{7}$$

Each image in the training database owns a label. We can capture $T + 1$ different features corresponding to resampling parameters. Finally, an image to be measured can be identified by a trained model as the label it belongs to.

$$I' \to l \in \{l_1, l_2, l_3, \ldots, l_{T+1}\}. \tag{8}$$

That is, according to the class label, we can judge by which parameter subset the image was processed.

$$I' \to \omega_i \in \{\omega_1, \omega_2, \omega_3, \ldots, \omega_T, \omega_{T+1}\}. \tag{9}$$

Based on the above analysis, we match the parameter subset with the label to identify the class of the resampled image and then obtain the corresponding resampling parameter according to the class.

$$I' \to \omega_i = \{s_i\} \to l_i. \tag{10}$$

According to this general formulation, resampling parameter estimation is regarded as a multi-classification problem.

We have converted the study into making a classification problem. Thus, we need a classifier to directly take the image as input and automatically learn the internal features of the resampled image. And CNN can be viewed as a classifier to perform classification by capturing the specific resampling traces. As a result, our proposed method for resampling parameter estimation is based on CNN. We will elaborate on our method in the next section.

4 The proposed CNN for resampling parameter estimation

CNN is one of the deep learning models with a feed-forward structure, which is good at processing machine-learning problems related to images, especially large images. Compared with the traditional recognition algorithm, CNN can directly take the image as input. Through automatically learning the internal features from images, CNN is able to perform classification without manually extracting features [26]. A typical CNN usually consists of one data layer, one loss layer and multiple hidden layers. Generally, the hidden layers are composed of convolution layers, pooling layers, activation functions, and full connection layers, which are the core of the network structure for learning features. Finally, the CNN model will output the probability of one sample classified into each class through softmax function.

We design a CNN architecture as the classifier in this section shown in Fig. 1. To improve the accuracy of

classification, the setting of dual differential filtering is applied into our CNN structure.

4.1 The preprocessing operation

First of all, we describe the interpretability of dual-filtering operation. In related works, we have known that the resampling traces can be learned from specific features and many of the early methods for detecting image resampling are based on prediction residual [8, 9]. Hence, in our paper, we also consider the prediction residual R which can be expressed as

$$R = f(\mathcal{N}) - I', \tag{11}$$

where $\mathcal{N}$ is a local neighborhood of the resampled image I' and $f(.)$ is a predictor which can predict the value of the pixel based on its neighborhoods with different residuals. Generally speaking, the residual means the difference between a resampled image patch and its filtered patch according to the selected type $f(.)$. With the difference operation, our structure can obtain the prediction residual features and capture the traces that are beneficial for classification more easily.

Therefore, we apply a high-order highpass filter and a low-order highpass filter to enhance resampling traces and magnify the difference between different resampling parameters. Particularly, two filters are inserted into the architecture in parallel connection. A SQUARE5×5 [27] high-order highpass residual filter is set as follows:

$$K_1 = \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}. \tag{12}$$

Resampling will introduce the correlation and periodicity of pixels, thus, we select this highpass filter which makes use of more pixels for the prediction residual to reduce the interference of image contents, and the traces left by resampling with different parameters can be investigated. However, the detectability of downsampling is not very ideal. So we add another filter into our preprocessing operation to improve the detectability. The another filter is a low-order highpass filter with the kernel

$$K_2 = [-1, +1], \tag{13}$$

which is used for highlighting the resampling features [28]. Under the downsampling scenario, we get the downsampling images by throwing away some pixels from the original images. Hence, the correlation is destroyed and the resampled features can hardly be captured. Through this filter, each new pixel of every row in an image can be obtained.

$$y'_i = y_{i+1} - y_i, \tag{14}$$

where y_i is the original central pixel, y'_i is the new central pixel after filtering and y_{i+1} is the right neighbor of y_i. In other words, each new pixel of every row in an image is equal to the difference between the right neighbor of the pixel and this pixel. In fact, the correlation between y_{i+1} and y_i is really weak in downsampling scenario. In this way, each central pixel is associated with its right neighbor and some vertical features can be captured by using K_2 filter. So we select this filter to enlarge the downsampling traces. Finally, with the dual-filtering based structure, more valuable resampling traces for multi-classification can be captured.

The prediction residual R_i after highpass filtering can be expressed as follows:

$$\begin{aligned} R_i &= I' * K_i \\ &= r(I, s) * K_i, i = 1, 2. \end{aligned} \tag{15}$$

R_1 and R_2 are the image residual by filtering with K_1 and K_2, respectively. Then, we apply the "concat" layer to concatenate the output of the two filters R_1 and R_2. These parts constitute our preprocessing operation.

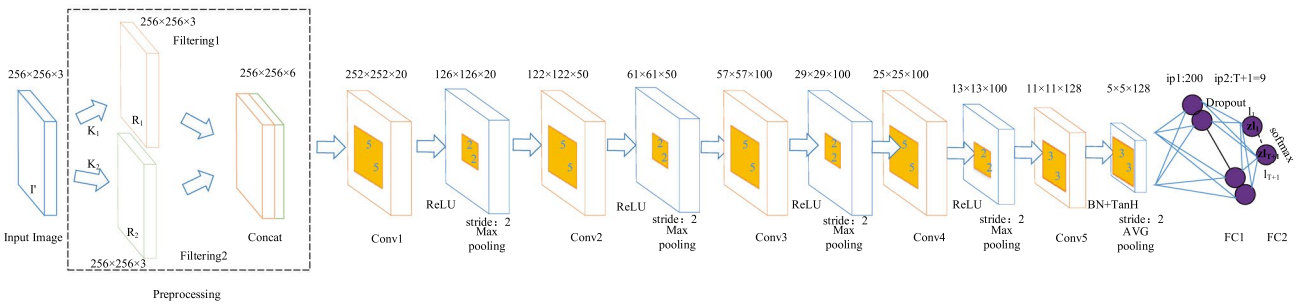

Fig. 1 The detailed structure of our proposed CNN for resampling parameter estimation

4.2 The architecture of proposed CNN model

After preprocessing, high-level features of resampled images are generated by the following layers. This model consists of five convolutional layers, five pooling layers, two full connection layers, and also consists of batch normalization layer, activation function, dropout layer, softmax layer and so on. The convolution layer is the core layer of network structure, including multiple convolution kernels. We use five convolution kernels with different sizes (5×5, 3×3). Two types of pooling are applied in this model to reduce the number and dimension of feature-map, including four max-pooling layers for retaining more texture information and one global average pooling layer for reducing parameters. To improve the expressive ability of models, convolution layers are followed by a non-linear function called activation function. ReLU has a faster convergent speed compared with Sigmoid and TanH, but it is fragile in training and some inputs may fall into the hard saturation zone, resulting in the corresponding weight cannot be updated. TanH has soft saturability and can expand the impact of features continuously because the mean value is 0. As a result, we apply two different activation functions into our CNN structure for the balance. ReLU is used in the four convolutional layers to accelerate, while TanH function is only applied in the last convolution. Furthermore, to minimize the internal covariate shift, we apply a zero-mean and unit-variance transformation of the data while training the CNN model. Hence, we use a batch normalization layer in our network. More details are shown in Fig. 1.

Finally, we use two full connection layers to perform classification. The last full connection layer is followed by a softmax activation function. It will output the probability of an image I' classified into each class such that

$$P(l = l_i|I') = \frac{e^{z_{l_i}}}{\sum_{i=1}^{T+1} e^{z_{l_i}}}, \tag{16}$$

where l_i is the label of the resampled image I' and z_{l_i} is the value of the i^{th} neuron in this layer. The number of neurons in the classification layer depends on the number of parameter classes $T + 1$.

4.3 The computation complexity analysis

We discuss the computation complexity analysis of the proposed CNN model. Firstly, suppose a convolutional layer have kernels of size $k_w \times k_h$, the numbers of input channel and output channel are C_{in} and C_{out}, the size of the output feature map is $f_w \times f_h$. The multiplication computation of this convolutional layer is $(k_w \times k_h \times C_{\text{in}}) \times f_w \times f_h \times C_{\text{out}}$, the addition computation is $(k_w \times k_h \times C_{\text{in}} - 1) \times f_w \times f_h \times C_{\text{out}}$, and the bias computation is $f_w \times f_h \times C_{\text{out}}$. Thus, the whole computation (mult-adds) of this convolutional layer is $2 \times (k_w \times k_h \times C_{\text{in}}) \times f_w \times f_h \times C_{\text{out}}$. According to the above calculation, the mult-adds of all convolutional layers in our proposed CNN model is 227.79 million.

Then, we analyze the computational complexity about dual-filtering operator. Based on analogous analysis, the computation (mult-adds) of filter operator is $(2 \times k_w \times k_h - 1) \times f_w \times f_h \times C_{\text{out}}$. The mult-adds of dual-filtering operator in our proposed CNN model is 1.02 million and the whole computations of the proposed CNN model for performing resampling parameter estimation is 228.81 million mult-adds. Therefore, we can observe that the preprocessing operation would not significantly increase the computational load and the overall complexity is acceptable.

5 Experimental results

In the section, we evaluate the performance of our proposed generic model to perform resampling parameter estimation through a set of experiments and compare its performance with state-of-the-art methods.

5.1 Dataset

We download images from the publicly available BOSS Database [29] and UCID Database [30] as the source of our training and testing datasets. 4,500 images from BOSS Database are resampled by using bilinear interpolation with nine different resampling parameters, i.e., 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5. Note that 1.0 means no resampling applied to an image. And then the central 256×256 blocks of the resampled images are retained to form the final datasets. Nine different image sets are created eventually. Next, we select 4,000 images from each image set as the training dataset and select the other 500 images from each image set as the validation dataset. In total, 36,000 RGB color images are used for training and 4,500 RGB color images are used for validation. Meanwhile, selecting 1,000 images from UCID Database to form a testing dataset through the same processing. 9,000 RGB color images are used for testing.

5.2 Training parameters

We set the batch size equal to 16 and the parameters of the Adaptive gradient as follows: momentum = 0.9, decay = 0.0005, and a learning rate = 0.001. The strategy of adjusting the learning rate is "fixed", which means the basic learning rate remains unchanged during the training. The maximum number of iterations of network training is set to 50,000. We run our experiments using NVIDIA TITAN XP

GPU with 12GB RAM in a modified version of the Caffe toolbox [31].

5.3 Experimental results

The performance of our designed method is evaluated on prepared training and testing datasets, and compared with that of Feng et al.'s method [6] and Stamm et al.'s method [25]. We download the relevant codes of their methods from the Internet and set the default parameter values according to their papers. The same datasets are used in three methods. For Feng et al.'s method, we first convert each image from the training dataset to a 19-D feature vector by extracting the normalized energy density and use SVM to train a model for classification. The performance of utilizing the trained SVM model for the testing dataset is the final result. The experimental process of Stamm et al.'s method is similar to ours.

Table 1 shows the accuracy of resampling parameter estimation using three methods. The average detecting accuracies are about 82.28%, 83.93%, and 93.19%, respectively. Our method improves average performance by over 10% compared to two other methods. From Table 1, it can be observed that our method has better performance than other methods, specifically in downsampling images. We can see that our method successfully achieve parameter estimation with an accuracy higher than about 80% with all resampling parameters. Noticeably, it can detect upsampling images with at least 92.4% accuracy especially for 110% upsampling images with 100% accuracy. Though capturing resampling traces in downsampling images is a challenging study, our method can still capture this to achieve parameter estimation. For example, it achieves 98.4% in the case of downsampling of 90%.

Table 2 shows the confusion matrix achieved by our method. We can see the estimation accuracy decreases with some downsampling parameters and the parameter of 80% seems to be difficult to distinguish from the parameter of 70%. From Table 2, it can be observed that the false-positive rates in the case of downsampling of 70% and 80% are much higher than other parameters. There is a performance gap between estimating downsampling and upsampling parameters. The reason behind this phenomenon is that the sampling process is different in cases of upsampling and downsampling. For upsampling, the size of the image increases, and there are more pixels in the resampled image than the original image. The additional pixels are quite relevant to their adjacent pixels. Thus, the correlation between adjacent pixels is enhanced and latent features can be learned better by the model. For downsampling, the size of the image decreases, and there are fewer pixels in the resampled image than the original image. In other words, some pixels of the original image are lost according to the corresponding interpolation algorithm, causing severe destruction to the correlation between adjacent pixels. Thus, latent features are too weak to be learned well by the model. Based on the above analysis, it is more challenging for our work to estimating downsampling parameters, leading to a performance gap between these cases. Even so, the experimental results show that our CNN is still able to accurately estimate resampling parameters and the overall effect of our method is acceptable.

In addition, we also consider resampling operation with nearest interpolation algorithm, and then estimate nine different resampling parameters by using the proposed CNN. Table 3 reports the confusion matrix and the average detection accuracy is 92.66%. Analogously, when images are

Table 1 Detection accuracy (%) for resampling parameter estimation by using different methods

Methods/parameters	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Feng et al.'s [6]	75.1	67.8	85.0	88.1	70.8	84.7	88.2	88.8	92.0
Stamm et al.'s [25]	76.5	43.5	70.9	92.1	92.2	94.6	93.1	94.5	98.0
Ours	**83.1**	**79.4**	**98.4**	**87.7**	**100**	**99.1**	**92.4**	**99.4**	**99.2**

Table 2 Confusion matrix for resampling parameter estimation (%) with bilinear interpolation. Tru and Det represent respectively the true label and detection label

Tru/det	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
0.7	**83.1**	10.2	0	6.7	0	0	0	0	0
0.8	13.9	**79.4**	0.2	6.3	0	0.1	0	0.1	0
0.9	0.6	0.2	**98.4**	0.8	0	0	0	0	0
1.0	4.6	7.3	0.4	**87.7**	0	0	0	0	0
1.1	0	0	0	0	**100**	0	0	0	0
1.2	0.1	0.5	0.1	0	0	**99.1**	0	0	0.2
1.3	0.3	0.1	0	0	0	0.5	**92.4**	4.6	2.1
1.4	0	0.1	0	0	0	0	0	**99.4**	0.5
1.5	0	0.1	0	6.7	0	0	0	0.7	**99.2**

Table 3 Confusion matrix for resampling parameter estimation (%) with nearest interpolation. Tru and Det represent respectively the true label and detection label

Tru/det	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
0.7	**97.5**	1.3	0.2	0	0.4	0.2	0.2	0	0.2
0.8	14.9	**82.6**	1.0	0.1	0.5	0.8	0.1	0	0
0.9	0.9	0.6	**97.0**	0.7	0.7	0	0.1	0	0
1.0	0.1	0	4.3	**94.7**	0.8	0.1	0	0	0
1.1	1.1	0.2	2.5	0.9	**93.9**	1.4	0	0	0
1.2	3.8	0	0	0	0.1	**95.4**	0.7	0	0
1.3	1.2	0.9	0	0	0	0.1	**93.5**	4.3	0
1.4	0.7	0.6	0	0.1	0	0.2	0.3	**84.1**	14.0
1.5	0	0.3	0	0.8	0	0.1	0	3.6	**99.2**

resampled by using cubic interpolation algorithm, the average detection accuracy obtained by our model is 82.12%. Compared with results about resampling with bilinear interpolation algorithm, it can be observed that the proposed method can obtain good performance for resampling parameter estimation with different interpolation algorithms. To further demonstrate the effectiveness of the model, we evaluate it with a much larger dataset. We select the other 4,000 images from BOSS Database which are not overlapping with our training dataset and the new testing dataset totals 36,000 images. The average detection accuracy is 92.28%. The result shows that the proposed method still obtains excellent performance in the case of a larger testing dataset.

6 Discussion

We have to admit that the explanations of the improvement in accuracy need to be discussed. From the experimental results, it can be observed that our method improves average performance compared to two other methods, which mainly owes to our proposed design.

The contribution of our work is deserving of encouraging. In this paper, an end-to-end resampling forensics framework based on convolutional neural network is presented. Deep learning and neural networks play a great role in digital image forensics, showing the superiority compared to some traditional methods. Specifically, the dual-filtering structure is designed to suppress image contents and highlight the resampling features, which achieves a complementary function in detecting resampling parameters. Experimental results show that our proposed model not only achieves upsampling parameter estimation in a high accuracy but also obtains acceptable performance in the downsampling scenario. Besides, there is a little running time on performing resampling forensics by the proposed method.

Though our method improves average performance by 10% compared to two other methods, the accuracy in a certain parameter is not ideal. For instance, it only obtains 79.4% accuracy in the case of downsampling of 80% with bilinear interpolation. The detection result is not satisfactory but acceptable. There is still much room for improvement in these cases. In the future, we intend to improve the performance of performing resampling forensics, especially in the downsampling scenario. For instance, we can insight further the essence of the CNN and resampling operation to develop an elaborate CNN network for higher accuracy.

7 Conclusion

In this paper, we present an end-to-end CNN model to perform resampling parameter estimation, which automatically learns the resampled features. The design of the dual-filtering structure plays a complementary role in estimating different parameters by suppressing image contents and strengthening resampling traces. Besides, the convergence speed and computation complexity of our proposed model are acceptable. The experimental results show that our proposed method has better performance than previous methods. However, our proposed method is effective only on the uncompressed images. JPEG is the most widely used format in our life. In the future, we will extend our work to investigate resampling parameter estimation on JPEG images.

Acknowledgements This work is supported by National Natural Science Foundation of China (Grant Nos. 61972142, 61772191, 61672222), Hunan Provincial Natural Science Foundation of China (No. 2020JJ4212), Open Project Program of National Laboratory of Pattern Recognition (Grant No. 201900017).

Conflict of interest We declare that we do not have any conflict of interest in connection with the work submitted.

References

1. Farid, H.: Image forgery detection. IEEE Signal Process Mag. **26**(2), 16–25 (2009)
2. Fridrich, J.: Digital image forensics. IEEE Signal Process Mag. **26**(2), 26–37 (2009)

3. Wang, C., Zhang, Z., Li, Q., Zhou, X.: An image copy-move forgery detection method based on SURF and PCET. IEEE Access. **7**, 170032–170047 (2019)
4. Zhang, Q., Wei, Z., Wang, R., Li, G.: Digital image splicing detection based on markov features in block dwt domain. Multimed Tools Appl. **77**(23), 31239–31260 (2018)
5. Kang, X., Stamm, M.C., Peng, A., Liu, K.J.R.: Robust median filtering forensics using an autoregressive model. IEEE Trans Inf Foren Secur. **8**(9), 1456–1468 (2013)
6. Feng, X., Cox, I.J., Doerr, G.: Normalized energy density-based forensic detection of resampled images. IEEE Trans Multim. **14**(3), 536–545 (2012)
7. Gao, S., Liao, X., Liu, X.: Real-time detecting one specific tampering operation in multiple operator chains. J Real Time Image Process. **16**(3), 741–750 (2019)
8. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Trans Signal Process. **53**(2), 758–767 (2005)
9. Kirchner, M.: Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In: Proceedings of the 10th ACM workshop on Multimedia and security (ACM), pp. 11–20 (2008)
10. Liao, X., Huang, Z.: A framework for parameters estimation of image operator chain. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2787–2791 (2020)
11. Chen, J., Kang, X., Liu, Y., Wang, Z.J.: Median filtering forensics based on convolutional neural networks. IEEE Signal Process Lett. **22**(11), 1849–1853 (2015)
12. Chen, Y., Lyu, Z., Kang, X., Wang. Z.J.: A rotation-invariant convolutional neural network for image enhancement forensics. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2111–2115 (2018)
13. Liao, X., Li, K., Zhu, X., Liu, K.J.R.: Robust detection of image operator chain with two-stream convolutional neural network. IEEE J Sel Top Signal Process. **14**(5), 955–968 (2020)
14. Stamm, M.C., Liu, K.: Forensic detection of image manipulation using statistical intrinsic fingerprints. IEEE Trans Inf Foren Secur. **5**(3), 492–506 (2010)
15. Ding, F., Zhu, G., Dong, W., Shi, Y.Q.: An efficient weak sharpening detection method for image forensics. J Vis Commun Image Represent. **50**, 93–99 (2018)
16. Gallagher, A.C.: Detection of linear and cubic interpolation in jpeg compressed images. In: Canadian Conference on Computer and Robot Vision (CRV), pp. 65–72 (2005)
17. Mahdian, B., Saic, S.: Blind authentication using periodic properties of interpolation. IEEE Trans Inf Foren Secur. **3**(3), 529–538 (2008)
18. Kirchner, M.: Linear row and column predictors for the analysis of resized images. In: Proceedings of the 12th ACM workshop on Multimedia and security (ACM), pp. 13–18 (2010)
19. Vázquez-Padín, D., Comesaña, P.: Ml estimation of the resampling factor. In: International Workshop on Information Forensics and Security (WIFS), pp. 205–210 (2012)
20. Zhu, N., Deng, C., Gao, X.: A learning-to-rank approach for image scaling factor estimation. Neurocomputing. **204**, 33–40 (2016)
21. Luo, C., Wang, X.: Hybrid modified function projective synchronization of two different dimensional complex nonlinear systems with parameters identification. J Franklin Inst. **350**(9), 2646–2663 (2013)
22. Zhang, H., Wang, X., Lin, X.: Topology identification and module-phase synchronization of neural network with time delay. IEEE Trans Syst Man Cybern Syst **47**(6), 1–8 (2017)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105 (2012)
24. Boroumand, M., Fridrich, J.: Deep learning for detecting processing history of images. J Electron Imaging. **2018**, 213-1 (2018)
25. Bayar, B., Stamm, M.C.: A generic approach towards image manipulation parameter estimation using convolutional neural networks. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (ACM), pp. 147–157 (2017)
26. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature. **521**(7553), 436–444 (2015)
27. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Trans Inf Foren Secur. **7**(3), 868–882 (2012)
28. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Trans Inf Foren Secur. **5**(2), 215–224 (2010)
29. Bas, P., Filler, T., Pevný, T.: Break our steganographic system: The ins and outs of organizing BOSS. In: International Workshop on Information Hiding (IH), pp. 59–70 (2011)
30. Schaefer, G., Stich, M.: UCID—an uncompressed color image database. In: Storage and Retrieval Methods and Applications for Multimedia, pp. 472–480 (2004)
31. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia (ACM), pp. 675–678 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.