Robust Detection of Image Operator Chain With Two-Stream Convolutional Neural Network

Xin Liao, *Member, IEEE*, Kaide Li, Xinshan Zhu, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Many forensic techniques have recently been developed to determine whether an image has undergone a specific manipulation operation. When multiple consecutive operations are applied to images, forensic analysts not only need to identify the existence of each manipulation operation, but also to distinguish the order of the involved operations. However, image operator chain detection is still a challenging problem. In this paper, an order forensics framework for detecting image operator chain based on convolutional neural network (CNN) is presented. Two-stream CNN architecture is designed to capture both tampering artifact evidence and local noise residual evidence. Specifically, the new CNN-based method is proposed for forensically detecting a chain made of two image operators, which could automatically learn manipulation detection features directly from image data. Further, we empirically investigate the robustness of our proposed method in two practical scenarios: forensic investigators have no access to the operating parameters, and manipulations are applied to a JPEG compressed image. Experimental results show that the proposed framework not only obtains significant detection performance but also can distinguish the order in some cases that previous works were unable to identify.**

***Index Terms*—Image forensics, image operator chain, order detection, convolutional neural network.**

I. Introduction

DUE to the powerful and user-friendly image editing software, digital images can be easily altered without leaving perceptible artifacts. Nowadays, image forensics has been used for determining the authenticity, processing history, and the origin of digital images content [1]. It aims at reconstructing what has happened to digital content in order to answer who has done what, when, where, and how. A larger number of forensic methods have been proposed for detecting a specific image manipulation, such as resizing [2]–[4], median filtering [5], [6], contrast enhancement [7], [8], copy-move forgery [9], [10], etc.

In a realistic scenario, multiple processing operations are inevitably utilized to forge an image, which would weaken or even erase the traces left by the previous operations. Thus, researchers have focused on analyzing the manipulation chains of multiple operators. Multiple JPEG compressions are discriminated based on the statistical analysis of Benford-Fourier coefficients [11], [12]. Some forensic detectors have been reported to detect a heterogeneous processing chain composed of double JPEG compression interleaved by a specific operation (e.g., resizing [13], contrast enhancement [14], linear filtered [15]). Instead of targeted detecting methods, there has been significant interest in the development of universal image manipulation identification, designed to classify different types of image processing operations by a universal feature set [16]–[19]. However, these forensic methods are designed to identify the existence of a single operation in the presence of manipulation chains.

In fact, apart from identifying the operation applied to the images, investigators expect to detect the order of the involved operations, so as to obtain the complete processing history and determine who manipulated the images. Few positive efforts [20] have been made to study the fundamental question of when we can or cannot detect the order of image operations. The works in [21], [22] formulated the order detection into general multiple hypotheses testing problems, and then proposed an information-theoretical order forensics framework based on mutual information. Conditional fingerprints are defined in the framework to understand why the order of operations is not always detectable. However, since the proposed characteristic footprints would be weakened by a further post-processing operation, the order of image operations cannot be identified in some cases. In addition, the existing approaches still fail to detect the order of the operations in the previously JPEG compressed images, and their performance usually degrades significantly. It is necessary to further consider the practical scenario where manipulations are applied to a JPEG compressed image.

Recently, as a common deep learning network, convolutional neural network (CNN) [23] has attracted increasing attention due to the excellent performance, especially in image classification, document analysis, and natural language processing.

Manuscript received October 30, 2019; revised April 7, 2020 and June 8, 2020; accepted June 9, 2020. Date of publication June 15, 2020; date of current version August 24, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61972142, 61972282, and 61772191, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2020JJ4212, and in part by the Open Project Program of National Laboratory of Pattern Recognition under Grant 201900017. The guest editor coordinating the review of this manuscript and approving it for publication was Anderson de Rezende Roch. *(Corresponding author: Xin Liao.)*

Xin Liao is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: xinliao@hnu.edu.cn).

Kaide Li is with the Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: kaideli@hnu.edu.cn).

Xinshan Zhu Li is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: xszhu@tju.edu.cn).

K. J. Ray Liu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@umd.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors. The material includes some confusion matrices of experimental results and the detailed computation complexity analysis.

Digital Object Identifier 10.1109/JSTSP.2020.3002391

Researchers have begun to investigate the potential of CNN in image forensics, such as image median filtering forensics [24], resampling detection [25], camera model identification [26], [27], recapture forensics [28], copy-move forgery forensics [29], universal image manipulation identification [30], [31], and multiple JPEG compression detection [32]. Few attempts to investigate CNN in forensically determining the order of image processing operations. Recently, a novel CNN with a constrained convolutional layer is employed to detect the image operator chain [33], [34]. It could directly extract low-level pixel relationships that capture the unique forensic fingerprints induced by an ordered chain consisting of two image operations. Note that these previous existing CNN-based image forensic works basically assume that the operating parameters of training images and testing images are the same, i.e., forensic analysts have access to the operating parameters of suspected images. However, it is not reasonable in practical applications to assume that the operating parameter settings are available.

In this paper, we propose a CNN-based forensic framework for detecting the image operations. While traditional image forensic algorithms mainly extract features manually, the new data-driven framework could automatically learn and obtain manipulation features. A two-stream operator chain forensics CNN network is presented, where one stream explicitly detects tampering artifacts and another stream specially extracts local residual features. To evaluate our approach, some typical image operations are used to collectively constitute ordered chains. Experimental results demonstrate that the proposed framework not only can detect the image operator chain with high accuracy but also is able to identify the order of the operations that the existing works cannot. Considering the practical scenarios in which a forensic investigator has no access to the prior information about the operating parameters, we empirically investigate the robustness of the proposed method when the operating parameters are unknown but within certain ranges. Furthermore, our proposed method is also effective in realistic scenarios where the processed image is JPEG compressed.

The rest of the paper is organized as follows. In Section II, we formulate the problem of image operator chain forensics, and then study the robustness issues in two practical scenarios: detection without prior information and identification for JPEG compressed images. Section III first proposes the two-stream CNN-based image operator chain detection framework. The corresponding well-designed preprocessing operations are presented, and the transfer learning strategy for image forensics performance enhancement is also given. Extensive experimental comparisons and analysis are shown in the next section, demonstrating the effectiveness of our proposed framework. Section V provides the corresponding discussions. Finally, the conclusion is made in Section VI.

II. Detecting an Ordered Chain of Image Processing Operations

In this section, we first formulate the target problem of detecting the image operator chain. Then, we consider the practical scenarios where a forensic investigator has no access to the prior information about the operating parameters, and investigate how to identify the image operator chain without prior information. Finally, we discuss the forensic detector used to identify the processing history of JPEG images.

A. Problem Formulation

Assuming an image operator chain might contain two image operations A and B, the detection of image operator chain could be formulated as the following multiple classification problem. The possible processing history of a given image would fall into one of the five classes.

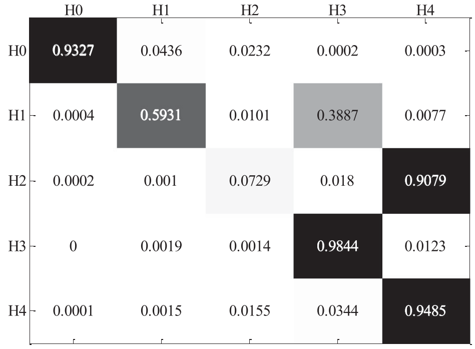
$$\begin{aligned} H_0 &: \text{ The image is unaltered.} \\ H_1 &: \text{ The image is altered by A only.} \\ H_2 &: \text{ The image is altered by B only.} \\ H_3 &: \text{ The image is altered by B then altered by A.} \\ H_4 &: \text{ The image is altered by A then altered by B.} \end{aligned} \tag{1}$$

The image operator chain is not always detectable due to the interplay between image processing operations. Later applied operations would affect and disguise the fingerprints left by earlier applied operations. Recently, researchers designed the hand-crafted features to detect the chain consist of image resizing and Gaussian blurring [22]. By observing the different fingerprints in the discrete Fourier transform (DFT) of an image's p-map [2], the above five classes could be distinguished. However, the detection results are unfavorable. In some cases, they can hardly tell the difference between H_3 and H_4, and thus the order of image resizing and Gaussian blurring cannot be determined. In fact, the forensic detectors are heuristically designed, and only two hand-crafted features about the suspected images are adopted. The feature selection depends heavily on the domain knowledge, and the classification performance is mainly determined by the threshold values. Further, the classification is independent of the feature extraction, and thus the prior feature extraction could not be optimized together with the classification.

In this paper, rather than traditional image forensic paradigm, our work focuses on extracting features and learning the hierarchical representations through multiple layers of nonlinear processing. In this way, a CNN-based forensic detector could combine feature extraction and classification steps in the unique framework. The manipulation detection features could be learned directly from the image data set. Forensic analysts do not need to think about the complicated feature selection and feature design.

B. Identification Without Prior Information

The knowledge of operation parameters for a forensic investigator plays an important role in determining the order of image processing operations. Note that most of the existing CNN-based image forensic approaches rely on the assumption that training and testing data are generated by image manipulations with the same parameter settings. However in a practical application, the operating parameters of suspected images are unknown, and the mismatch of the parameters should be considered [35]. Thus, the forensic investigator has to identify the ordered chain of image processing operations without prior information. If



	H0	H1	H2	H3	H4
H0	0.9327	0.0436	0.0232	0.0002	0.0003
H1	0.0004	0.5931	0.0101	0.3887	0.0077
H2	0.0002	0.001	0.0729	0.018	0.9079
H3	0	0.0019	0.0014	0.9844	0.0123
H4	0.0001	0.0015	0.0155	0.0344	0.9485

Fig. 1. Example of identification without prior information. Confusion matrix is obtained by directly applying the constrained CNN model in [33], [34] (training images are created by $s_1 = 1.2, \nu_1 = 0.7$) to test the suspected images (created by $s_2 = 1.5, \nu_2 = 1.0$). (A: Upsampling B: Gaussian blurring).

the mismatching trained CNN model is selected, the detection accuracy would decrease drastically. For example, assuming the constrained CNN model in [33], [34] is trained by the image data set created by image upsampling (scaling factor $s_1 = 1.2$) and Gaussian blurring (Gaussian blurring variance $\nu_1 = 0.7$), we directly apply it to test the suspected images generated by image upsampling ($s_2 = 1.5$) and Gaussian blurring ($\nu_2 = 1.0$). Fig. 1 reports the confusion matrix for image operator chain detection. It can be observed that the five classes are not distinguished. Thus, the image operator chain cannot be detected in this case.

In fact, identification without prior information is a more realistic and significant scenario, which would require the forensic detector to be more robust and general. In this paper, for a forensic investigator, we reasonably assume that the operating parameters are unknown but within certain ranges. For instance, an experienced analyst can have an estimation of the Gaussian blurring variance, but he/she is unlikely to know the exact value. Therefore, we will investigate the robustness of the proposed CNN in practical scenarios in which forensic analysts have no access to the parameter settings, but they only know a possible range of the parameter. In order to obtain a robust CNN model, we collect the image data in a much more extensive and systematic manner.

Specifically, in the deep learning procedure, we would train the CNN model by using a variety of images altered with a mixture of operating parameters, i.e., train the network using a limited set of possible parameter values as anchor points. Assume the operating parameter p_A of the image operation A is set as $p_A \in \{p_A^1, p_A^2, \ldots, p_A^{m_a}\}$ and the operating parameter p_B of the image operation B is set as $p_B \in \{p_B^1, p_B^2, \ldots, p_B^{m_b}\}$. For H_0 class, we collect N unaltered images. The altered images for H_1 class are respectively manipulated by using the operation A with the parameter p_A, and then a total of $N \times m_a$ images are created. Similarly, we obtain $N \times m_b$ altered images for H_2 class by applying operation B with the parameter p_B. For H_3 and H_4 classes, there are $m_a \times m_b$ combinations of operation parameters, and thus each class has $N \times m_a \times m_b$ images. Finally, a total of $N \times (2m_am_b + m_a + m_b + 1)$ images will be fed into the CNN model.

For some image operations with continuously varied parameters, our proposed solution does not require a prohibitively high number of points. The traces left by a specific manipulation with different values of operation parameters are similar. As long as the possible range of operation parameters are given, our solution could select limited parameter values as anchor points, train CNN model by using these anchor points, and finally achieve good performance for detecting image operator chain. The intervals of the selected anchor points should be as small as possible. In our paper, a simple equal step is used as the interval. In the future, we plan to investigate the relationship between the quantization of tampering traces and the interval selection. Furthermore, our proposed solution is simple yet efficient, and is also applicable to other CNN-based image forensics methods, which is validated by the experimental results in Section IV-F.

C. Robustness of JPEG Compressed Images

Nowadays, the JPEG standard is the most widely used compression technique of digital images. A vast amount of digital images taken by digital cameras are saved in the JPEG compressed format. Most forensics tools work well only for uncompressed images and their accuracies might drop significantly with JPEG compression. In order to create a forged JPEG image, the image is usually loaded into a photo editing software, manipulated by multiple heterogeneous processing operations and then re-saved in JPEG format again. Therefore, given a suspicious JPEG image, it is important to further identify the processing history.

In this paper, we will study the most common scenario in practice when the processed image is JPEG compressed. Specifically, we investigate double JPEG compressed images when two image operations A and B are applied between the two compressions. The following five classes should be distinguished when forensic analysts acquire a JPEG compressed image.

H_0 : The image is single compressed with quality factor QF1.
H_1 : The image is double compressed with quality factors QF1 then QF2 interleaved by A.
H_2 : The image is double compressed with quality factors QF1 then QF2 interleaved by B. (2)
H_3 : The image is double compressed with quality factors QF1 then QF2 interleaved by B then A.
H_4 : The image is double compressed with quality factors QF1 then QF2 interleaved by A then B.

Note that in [22], when two image manipulations resizing and Gaussian blurring are applied to a JPEG compressed image, the fingerprints would be weakened by the last applied JPEG compression. Thus, five classes are easily confused with each other, and thus the image operator chain could not be detected. In fact, in the presence of post-processing operation, the characteristic artifacts exploited to detect a specific operator would be severely suppressed. The post-processing operation would not only weaken or even erase the specific footprints left by previous processing operations, but also perturb the characteristic patterns presented in the DCT distribution of JPEG images. Therefore, we could exploit the peculiar traces left in the DCT

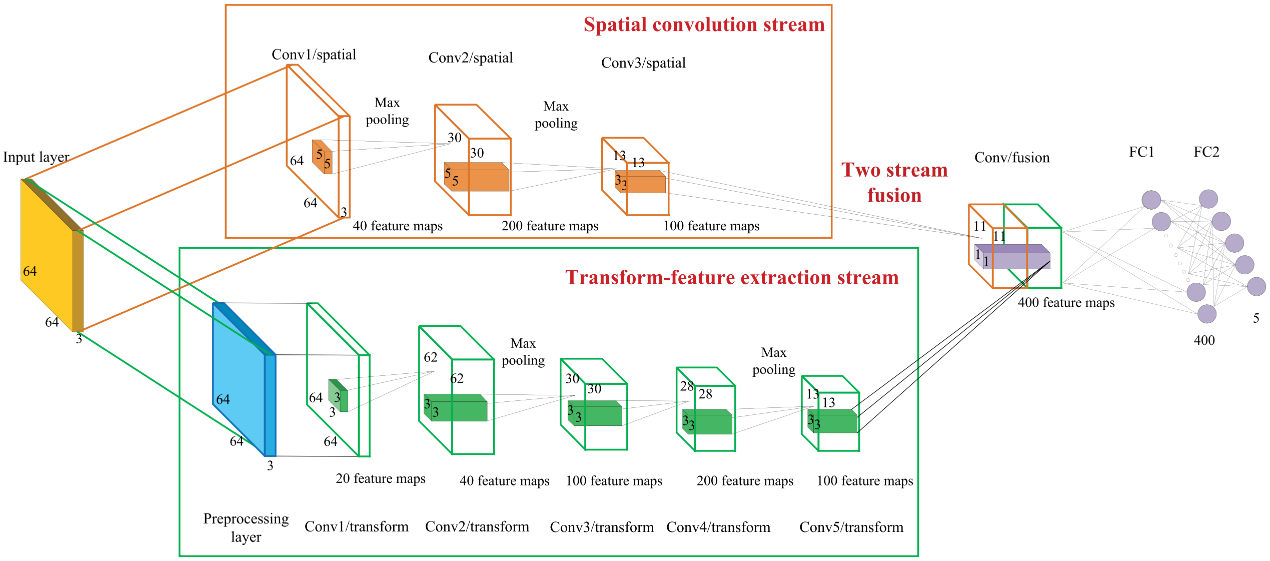


Fig. 2. Illustration of our proposed two-stream CNN network for image operator chain detection.

domain. In this paper, we try to capture the evidence by analyzing the anomalous statistical properties characteristic of DCT coefficients.

III. The CNN-Based Image Operator Chain Detection Framework

In this section, we first illustrate the detailed architecture of the proposed two-stream CNN network. Then, considering four typical image processing manipulations, we investigate various preprocessing operations for different image operator chains. Furthermore, a preprocessing operation for JPEG image operator chains is presented. Finally, the transfer learning strategy for image forensics performance enhancement is given.

A. The Architecture of Two-Stream CNN Network

CNN tries to study the relationship between the input and the output and store the learned experience in their filter weights. These networks are composed of multiple layers, each of which computes convolutional transforms, followed by nonlinearities and pooling operators. Assuming that F is nonlinearity activation function (e.g., Sigmoid, Tanh, ReLU), and "*pooling*" broadly refers to some forms of combining "nearby" signal values (e.g., averaging) or picking one representative value (e.g, maximization), we have

$$x_j^l = pooling\left(F\left(\sum_{i=1}^{n} x_i^{l-1} \times w_{ij}^l + b_j^l\right)\right) \tag{3}$$

where x_j^l is j-th feature map output in the hidden layer l, w_{ij}^l is the weight of trainable convolutional kernel connecting the i-th input feature map in the hidden layer $l-1$ and the j-th output feature map in the hidden layer l, and b_j^l is the training bias term for j-th output feature map in the hidden layer l. The weights and biases would be learned and renewed during the backpropagation.

The architecture of our proposed CNN network is illustrated in Fig. 2, where the size of filters in each layer and the dimensions of their corresponding feature maps are depicted. In our work, considering the computing capacity, the input layer is the color image patches with the size of $64 \times 64 \times 3$. We propose a novel two-stream CNN network to capture both tampering artifact evidence and local noise residual evidence. One of our proposed streams is named spatial convolution stream and the other one is called transform-feature extraction stream. The fusion of two streams could reveal both evidence of high-level visual tampering artifacts and low-level noise residual features.

The proposed spatial convolution stream consists of three convolutional layers and two pooling layers, which could learn visual tampering artifacts and generate more informative features. The first convolutional layer filters the input data with 40 kernels of size 5×5, and then the rectified linear units (ReLU) and max-pooling layer are applied. ReLU uses the nonlinearity activation function $F(x) = \max(x, 0)$, thus clipping negative values to zero. In the pooling layer, the filters of size 2×2 and stride of 2 could retain the maximum value and discard 75% of the activations, so as to decrease the spatial resolution and improve the convergence performance. Then 40 corresponding feature maps with the size of 30×30 are generated. The second convolutional layer takes the output of the first convolutional layer as the input and convolves them with 200 kernels of size 5×5. The nonlinearity activation function is ReLU, and the pooling layer is the max pooling, which would yield 200 feature maps with the size of 13×13. The third convolutional layer is 100 kernels of size 3×3. The ReLU function is also used to activate the outputs. Finally, we have 100 feature maps with the size of 11×11.

In order to reveal co-occurrence based local features and capture the hidden residual information, the proposed transform-feature extraction stream consists of one well-designed preprocessing layer, five convolutional layers, and two pooling layers. We specially design the preprocessing layer based on the characteristics of two involved manipulations in the image operator chain, which would be presented in detail in the next subsections. The first convolutional layer filters the input data with 20 kernels of size 3×3, and the ReLU function is used. Then 20 feature maps with the size of 62×62 are obtained. The second convolutional layer takes the output of the first convolutional layer as the input and convolves them with 40 kernels of size 3×3. We utilize the ReLU function to activate the outputs. The max pooling would yield 40 feature maps with the size of 30×30. The third, fourth, and fifth convolutional layers are 100 kernels of size 3×3, 200 kernels of size 3×3, and 100 kernels of size 3×3, respectively. The ReLU function is also applied to activate the outputs. After the fourth convolutional layer, the max pooling layer is utilized. Finally, we obtain 100 feature maps with the size of 11×11.

The features of two streams are fused to detect the image operator chain and obtain better performance. We fuse two proposed streams by using concatenation and 400 convolutional filter kernels of size 1×1. Note that 1×1 convolutional filters are used to effectively learn new relationships among different feature maps by doing dot product in three dimensions, which was first investigated and named "Network in network" [36]. The 1×1 convolutional filters are added to the proposed CNN network, and they are used mainly as dimension reduction modules to reduce the size of the network. Compared with a standard CNN architecture, the 1×1 convolutional filters could increase the number of feature maps with fewer parameters, and realize complex and learnable interactions of cross-channel information. The ReLU function is used to activate the outputs. Then, we have 400 neurons in the first fully-connected layers (FC1), which converts previous outputs into a vector. The second fully-connected layers (FC2) has 5 neurons corresponding to five classes in Eqs. (1), (2), and its output is fed into a softmax classifier. Finally, we determine whether the image operator chain could be detected.

In our proposed CNN network, no padding layer is included in the convolutional layers. A padding layer could prevent further dimensionality reduction and image border distortion. It is always useful for the tasks of object detection and localization. Generally, when the object is undetermined, image border distortion will result in poor performance. However, for our task of image operator chain detection, the traces left by the tampering manipulations are usually fragile. Zero elements introduced by padding layers would weaken the traces, which might have a negative effect on tampering detection. In addition, our paper focuses on global tampering (i.e., multiple consecutive operations are applied to the whole image), so we do not need to consider the object detection and localization. Experimental results in Section IV have also shown that our proposed CNN network without padding layer could achieve significant detection performance. Note that detection and localization of splicing, copy-move, removal are popular problems in multimedia forensics today. In future work, we plan to extend our approach to a data-driven order forensic framework for detecting image operator chains consisting of splicing, copy-move, removal operations (e.g., NIST image datasets [37]), and then the padding layer will become an indispensable component in the CNN network.

TABLE I
PREPROCESSING OPERATIONS FOR DIFFERENT IMAGE OPERATOR CHAINS

Image operator chains	Preprocessing operations
Upsampling + Gaussian blurring	$I' = \lvert DFT(I)\rvert$
Upsampling + Median filtering	A filter kernel K_S in image steganalysis
Gaussian blurring + Median filtering	$I' = Gaussian(I) - I$
Upsampling + USM sharpening	$I' = \lvert DFT\Big(Gaussian(I) - I\Big)\rvert$
Gaussian blurring + USM sharpening	A Laplacian filter kernel K_L
Median filtering + USM sharpening	LBP (local binary pattern)

B. Well-Designed Preprocessing Operations for Non-JPEG Images

In order to accurately capture local noise residual evidence and detect the image operator chain, the specific preprocessing operations are designed in the transform-feature extraction stream. In this subsection, we propose various preprocessing operations for different image operator chains, which are illustrated in Table I. Four typical image processing manipulations are considered, image upsampling with bilinear interpolation, Gaussian blurring, median filtering and unsharp masking (USM) sharpening. Thus, we have six different chains made of two image operations.

For the image operator chain of image upsampling and Gaussian blurring, the linear interpolation process in image upsampling will introduce periodic artifacts into an image, and thus lead to distinct peaks of DFT of the image's p-map. Note that there is no distinct periodic artifact in an unaltered image, and thus it can be considered as a periodic noise resulted from upsampling operation. Furthermore, in the Gaussian blurring operation, the convolution process would increase the co-correlations between neighboring pixels, so Gaussian blurring operation would leave high-frequency noise in the DFT of the image. Therefore, we could adopt DFT as the preprocessing operation described as follows. We believe that it would be easier to extract the noise residual when transforming an image from spatial domain to frequency domain.

$$I' = |DFT(I)| \tag{4}$$

where I is the original image, $DFT(I)$ means applying DFT transform on the image I, $|\cdot|$ is the modulus operation and I' is the output of preprocessing.

For the image operator chain of image upsampling and median filtering, the image upsampling operation will result in the image's periodic artifact. The median filtering is a nonlinear digital filtering technique, which could preserve edges while removing noise. When the median filtering operation is applied to an image, we will replace each pixel with the median of neighboring pixels. Note that the following filter kernel K_S, first introduced in the weighted stego image steganalysis [38], can be used for exposing both image's periodicity and image pixels changes. Therefore, we could apply the filter operation based on the kernel K_S as the preprocessing operation.

$$K_S = \begin{bmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{bmatrix} \tag{5}$$

For the image operator chain of Gaussian blurring and median filtering, both of the two manipulations will slightly modify the image texture and reduce image noise. Thus, the difference preprocessing operation based on the Gaussian filter could be utilized. Through this preprocess, the Gaussian filter residual of an image is obtained, which can suppress the interference caused by the presence of image content. By eliminating the interference of irrelevant information, the traces left by the tampering manipulations would be revealed. The difference preprocess is operated by using the following equation.

$$I' = Gaussian(I) - I \tag{6}$$

where I is the original image, $Gaussian(I)$ is the result of Gaussian blurring, the Gaussian blurring variance $\nu = 5$ and the window size of Gaussian blurring is chosen to be 5×5. I' represents the Gaussian filter residual, which is the difference between $Gaussian(I)$ and I.

For the image operator chain of image upsampling and USM sharpening, the upsampling manipulation would bring about the image's periodic artifact, and the USM sharpening operation is a typical filter that amplifies the high-frequency components of an image. The differential operation can extract the image's high-frequency texture features, and the DFT operation is useful in exposing the image's periodic artifact. Thus, we jointly utilize both to capture evidence from the manipulated images. The preprocessing procedure is presented as below.

$$I' = |DFT\left(Gaussian(I) - I\right)| \tag{7}$$

where the Gaussian blurring variance $\nu = 0.5$ and the window size of Gaussian blurring is chosen to be 3×3. We specially adopt the Gaussian blurring with the small variance and small window size, in order to efficiently reveal the subtle residuals and yield better performance for tampering detecting.

For the image operator chain of Gaussian blurring and USM sharpening, the convolution process in the Gaussian blurring operation would increase the co-correlations between neighboring pixels. It will lead to the result that the manipulation features have a relatively high similarity with its reference one. Moreover, for a given image, the USM sharpening operation would amplify the high-frequency parts. By using the Laplacian operator, we can obtain the second derivative of an image, which could expose the image's high-frequency texture features and reveal the local structural relationships among the pixels of manipulated images. Thus, we apply the filter operation based on the following Laplacian filter kernel K_L as the preprocessing operation in the transform-feature extraction stream.

$$K_L = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{8}$$

For the image operator chain of median filtering and USM sharpening, the USM sharpening operation would expand the high-frequency portions. The process of calculating the median value in the median filtering operation would erase the image details. Note that LBP (local binary pattern) [39] is a powerful descriptor for the image texture, which could characterize the magnitude relationship between the central element and its neighbors. Thus, we utilize LBP as the preprocessing operation to expose the changes of the image texture. Given a center pixel I_c and its neighbor pixels I_p $(p \in \{0, \ldots, P-1\})$ on a circle of radius R, the LBP operator is defined as below.

$$LBP = \sum_{p=0}^{P-1} s(I_p - I_c) \times 2^p \tag{9}$$

where $s(x)$ is an indicator function defined as

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{10}$$

In our work, the number of selected neighboring pixels $P = 4$, and the radius $R = 1$.

For non-JPEG images, our paper designs the well-directed preprocessing operations for detecting different image operator chains. In order to capture local noise residual evidence, these specific preprocessing operations are used to suppress the interference from image content, and reveal the local structural relationships among the pixels of manipulated images. We believe that the choice of preprocessing operations could be helpful in extracting the image's pixels changes, periodicity artifact, and high-frequency texture features. Furthermore, not all the image manipulations would leave distinct tampering artifacts in a given feature space. Even if a universal preprocessing operation could be used for detecting different image operator chains, the targeting of the forensic detector should be considered, and the detection performance might be improved by designing the well-directed preprocessing operation. Thus, different preprocessing operations are tailored to the specific image operator chains.

C. A Preprocessing Operation for JPEG Images

For detecting different JPEG image operator chains, we design DCTR with 5×5 DCT basis patterns, as a preprocessing operation in the transform-feature extraction stream. Note that DCTR and its variation [40] are the efficient features set for JPEG images steganalysis, which could be viewed in the JPEG domain as a projection-type model with orthogonal projection vectors. They have exhibited excellent performance in detecting steganography in JPEG images. We believe that the DCTR features contain much more information that would be quite useful for JPEG image forensics, and a substantial component of the fine-grain structure could be exploited. Therefore, in order to determine the processing history of a JPEG image, we select twenty-five 5×5 DCT basis patterns $\mathbf{B}^{(k,l)} = (B_{mn}^{(k,l)})$ as the specific kernels in the convolutional phase of the proposed CNN network, i.e., a well-designed preprocessing operation used in the transform-feature extraction stream for all possible processing operator chains. It would be considered as a general detector of different processing operations applied to an image prior to JPEG compression.

DCTR with 5×5 DCT basis patterns is a hand-crafted convolutional phase, which takes decompressed JPEG images as input and twenty-five residual maps as output. Twenty-five 5×5 DCT basis patterns $\mathbf{B}^{(k,l)} = (B_{mn}^{(k,l)})$ are defined as

$$B_{mn}^{(k,l)} = \frac{w_k w_l}{5} \cos\frac{\pi k(2m+1)}{10} \cos\frac{\pi l(2n+1)}{10} \tag{11}$$

where $0 \leq k, l \leq 4, 0 \leq m, n \leq 4$, w_i is defined as follows.

$$w_i = \begin{cases} 1, & i = 0 \\ \sqrt{2}, & 1 \leq i \leq 4 \end{cases} \tag{12}$$

Given a $M \times N$ JPEG input image, it is firstly decompressed to the corresponding spatial-domain version $\mathbf{I}$. $\mathbf{I}$ is convolved with $\mathbf{B}^{(k,l)}$ to twenty-five residual maps $\mathbf{R}^{(k,l)}$ with the size of $M \times N$ as below.

$$\mathbf{R}^{(k,l)} = \mathbf{I} * \mathbf{B}^{(k,l)} \tag{13}$$

D. Transfer Learning Strategy for Image Forensics Performance Enhancement

Though the exiting CNN-based image forensic approaches are effective for image tampering detection, the detection performance is not always satisfying when the images are altered by lower values of operation parameters. This is mainly due to the fact that manipulations with lower values of operation parameters might leave less forensic traces, which makes the tampered image much harder to detect. Note that there are some common forensic patterns shared between image operations with different values of operation parameters. Transfer learning [41] is usually used to explore the shared domain-specific knowledge contained in the related tasks and improve the performance of the target task, so it is a good way to address the above issue. We believe that feature representations learned with a pre-trained CNN for detecting manipulations with higher values of operation parameters can be efficiently transferred to improve the learning of features for detecting the same manipulations with lower values of operation parameters.

In CNN-based image operation chain detection framework, those manipulations fingerprints that are left by image operator chain with higher values of operation parameters (i.e., high-intensity manipulations) could be considered as the auxiliary information, which can be efficiently utilized to help the task of detection performance for image operator chain with lower values of operation parameters (i.e., low-intensity manipulations). The transfer learning strategy in CNN-based detecting image operator chain framework is illustrated in Fig. 3. We will introduce how the feature representations could be learned from the source task (order detection for high-intensity manipulations) and transferred to the target task (order detection for low-intensity manipulations). We first pre-train a CNN model on the images altered by high-intensity operations A with the parameter p_A^H and B with the parameter p_B^H (source task) by using the backpropagation. Then we transfer the weights of all the layers in the pre-trained model to the target task. That is to say, the CNN model in the target task would be initialized with the feature representations learned from the pre-trained model, instead of random initialization. Finally, we fine-tune it on the images altered by low-intensity operations A with the parameter p_A^L and B with the parameter p_B^L (target task) by continuing the backpropagation.

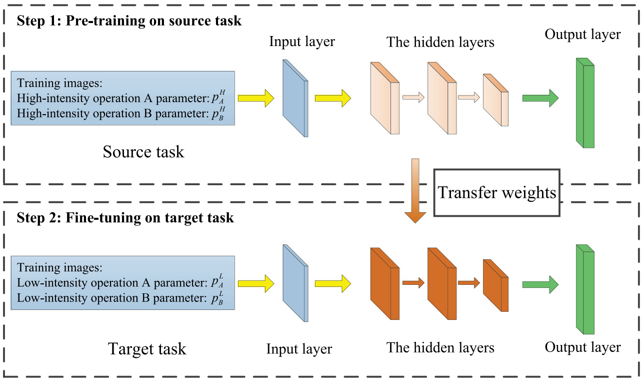

Step 1: Pre-training on source task
Input layer
The hidden layers
Output layer
Training images:
High-intensity operation A parameter: p_A^H
High-intensity operation B parameter: p_B^H
Source task
Transfer weights
Step 2: Fine-tuning on target task
Training images:
Low-intensity operation A parameter: p_A^L
Low-intensity operation B parameter: p_B^L
Target task
Input layer
The hidden layers
Output layer

Fig. 3. Illustration of transform learning strategy in CNN-based detecting image operator chain framework.

IV. Experimental Results

In this section, we conduct several experiments to demonstrate the effectiveness and robustness of the proposed CNN-based framework. Following the experimental setup, we first detect the image operator chain by using the proposed CNN with known parameter settings. To verify the robustness of our proposed framework, the order detection results of the proposed CNN without prior information are given. Then, the operator chain detection is examined when the processed image is stored in the JPEG format. Next, the impact of two-stream fusion will be illustrated in an ablation study. Finally, the detection performance comparisons of the proposed CNN method with state-of-the-art methods are provided.

A. Experimental Setup

We select 1,000 color images from UCID database [42] to generate training and validation databases. We crop them into image blocks with the size of 256 × 256, subdivide each image block into 16 non-overlapping 64 × 64 image blocks, and have 16,000 unaltered image blocks. A set of altered images is created by applying the corresponding operations to these selected images according to Eq. (1). Finally, a total of 80,000 image blocks are fed into the proposed CNN. 80% of these image blocks are used for training, while the remaining are utilized for validation. BOSSbase image set [43] consisting of 10,000 color images is used to acquire the testing image database. The corresponding operations are applied to generate the manipulated images, and then a total of 50,000 color images are obtained. We only crop each image in the center into 64 × 64 block and then obtain a total of 50,000 testing image blocks.

All the experiments are conducted by using a modified version of the Caffe Toolbox [44]. We run our experiments using NVIDIA TITAN XP GPU with 12 GB RAM. To facilitate this, we convert our datasets to the LMDB format. Mini-batch stochastic gradient descent is utilized to solve all the CNN in the experiments. The cross-entropy loss function is adopted to minimize the distance between the true label and the predicted label. In the training and validation phase, the batch size is set to 64 images, the momentum value is fixed to 0.9, the weight

TABLE II
SUMMARY OF AVERAGE DETECTION ACCURACIES OF DIFFERENT IMAGE OPERATOR CHAINS AND OPERATION PARAMETERS SETTINGS

	Parameter Settings in Image Operator Chain	**Average Detection Accuracies**
Image Operator Chain Detection with Known Parameter Settings	Upsampling factor $s = 1.5, 1.2$, Gaussian blurring variance $\nu = 1.0, 0.7$	96.23%, 94.05%, 91.21%, 93.77%
	Median filtering window size $w = 5 \times 5, 3 \times 3$, USM sharpening radius $r = 3, 2$	84.51%, 81.98%, 86.19%, 85.76%
	Gaussian blurring variance $\nu = 1.0, 0.7$, USM sharpening radius $r = 3, 2$	88.51%, 86.75%, 87.30%, 86.69%
	Upsampling factor $s = 1.5, 1.2$, Median filtering window size $w = 5 \times 5, 3 \times 3$	91.66%, 92.51%, 86.63%, 89.17%
	Gaussian blurring variance $\nu = 1.0, 0.7$, Median filtering window size $w = 5 \times 5, 3 \times 3$	94.02%, 92.61%, 91.99%, 91.74%
	Upsampling factor $s = 1.5, 1.2$, USM sharpening radius $r = 3, 2$	89.16%, 88.33%, 86.46%, 85.06%
	Upsampling factor $s = 1.5, 1.2$, Gaussian blurring variance $\nu = 1.0, 0.7$, cubic interpolation kernel (different interpolation kernel)	94.36%
	Source task: Upsampling factor $s = 1.3$, USM sharpening radius $r = 3$ Target task: Upsampling factor $s = 1.2$, USM sharpening radius $r = 1$ (Transfer Learning)	80.49% → 82.07%
	Source task: Gaussian blurring variance $\nu = 0.7$, USM sharpening radius $r = 3$ Target task: Gaussian blurring variance $\nu = 0.6$,, USM sharpening radius $r = 2$ (Transfer Learning)	81.90% → 83.40%
Image Operator Chain Detection without Prior Information	Upsampling factor $s \in (1.5, 1.8)$, Gaussian blurring variance $\nu \in (0.7, 1.0)$ (The parameters are inside the range)	95.13%
	Gaussian blurring variance $\nu \in (0.7, 1.0)$, Median filtering window size $w = 5 \times 5$ or 3×3 (The parameters are inside the range)	92.42%
	Upsampling factor $s = 1.4, 1.9$, Gaussian blurring variance $\nu = 1.1$ (The parameters are outside the range)	85.96%, 86.12%
Image Operator Chain Detection for JPEG Compressed Images	Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$, QF1=75, QF2=85	90.20%
	Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$, QF1=85, QF2=85	85.32%
	Upsampling factor $s = 1.2$, Gaussian blurring variance $\nu = 0.9$, QF1=70, QF2=90	85.88%
	Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$, QF1=75, QF2=85	88.18%
	Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$, QF1=85, QF2=75	84.07%
	Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$ training JPEG images: QF1=85, QF2=75, testing JPEG images: QF1=80, QF2=75 (non-matching QF1)	83.24%
	Upsampling factor $s = 1.5$, Median filtering window size $w = 5 \times 5$, QF1=75, QF2=85	86.75%
	Upsampling factor $s = 1.5$, Median filtering window size $w = 5 \times 5$, QF1=85, QF2=75	78.25%
	Gaussian blurring variance $\nu = 0.8$, Median filtering window size $w = 3 \times 3$, QF1=70, QF2=90	87.30%
	Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$, QF1=90, QF2=70	83.65%
	Upsampling factor $s = 1.5$, USM sharpening radius $r = 3$, QF1=80, QF2=90	85.35%
	Upsampling factor $s = 1.5$, USM sharpening radius $r = 3$, QF1=75, QF2=85	86.45%
	Upsampling factor $s \in (1.5, 1.8)$, Gaussian blurring variance $\nu \in (0.7, 1.0)$, QF1=80, QF2=90 (The parameters are inside the range)	85.35%

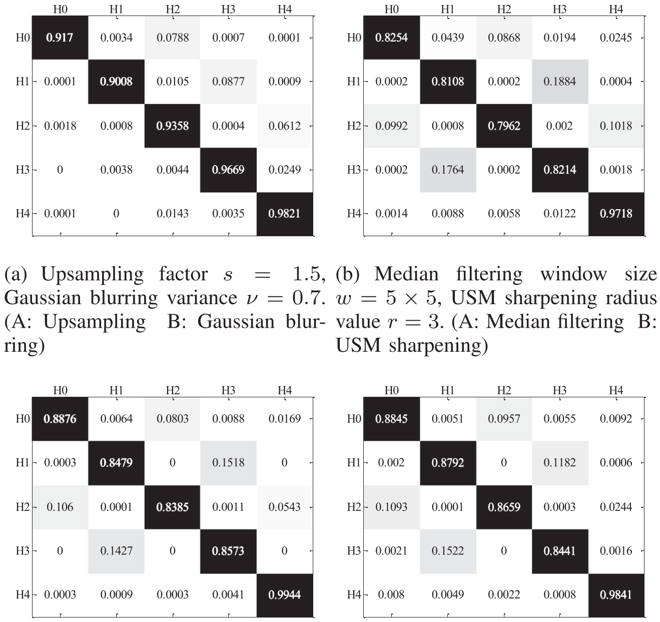
decay is 0.0005, the maximal iteration epoch is 180, and the learning rate is initialized to 0.001. The step size is 60 and the gamma is 0.2, which indicates that the learning rate is scheduled to decrease to 20% every 60 epochs. For the transfer learning, during the fine-tuning stage, we first initialize a CNN with the pre-trained model, and divide the initial learning rate by 10 and then train the CNN. The maximal iteration epoch and the stepsize would be reduced by half.

B. Image Operator Chain Detection Results of the Proposed CNN With Known Parameter Settings

In this subsection, we evaluate the effectiveness and feasibility of our proposed CNN to detect the image operator chain with two different manipulations and various operation factors. Following the existing CNN-based image forensic approaches, we will utilize the corresponding trained CNN model to test the images altered by using the same parameter settings. Table II provides the summary of average detection accuracies of different image operator chains and operation parameters settings. Some typical confusion matrices are shown in Fig. 4 and others are given in Figs. A1–A4 in the Appendix due to space limitations. It can be observed that the proposed CNN could distinguish the image operator chains with high accuracy.

Fig. 4(a) reports the confusion matrix obtained by applying our proposed CNN to detect the image operator chain of upsampling and Gaussian blurring, when the upsampling factor $s = 1.5$ and Gaussian blurring variance $\nu = 0.7$. The maximum element in each row locates at the diagonal line of the confusion matrices. In fact, the diagonal elements denote the classification accuracy of each class and the remaining are the error rates. The average classification accuracies are 94.05%, and the proposed CNN is able to distinguish each class in the image processing history with high accuracy. It is worth pointing out that [22] failed to determine the order of the operations in this case. We believe that the proposed CNN could learn subtle fingerprints left by the image operator chain of upsampling and Gaussian blurring.

Fig. 4(b) reports the confusion matrix obtained by using our proposed CNN, when median filtering window size $w = 5 \times 5$ and USM sharpening radius value $r = 3$. H_3 represents that the image is altered by USM sharpening then altered by median filtering, and H_4 represents that the image is altered by median filtering then altered by USM sharpening. In this sense, H_3 and H_4 can be considered as "inverted" operator chains. H_3 is really harder to predict than H_4. As far as we are concerned, there are two reasons: 1) As a powerful descriptor for image texture, our proposed preprocessing operation LBP is more efficient for detecting USM sharpening, and could capture the evidence left by the latter operation in H_4 (i.e., USM sharpening). 2) The process of calculating median values in the median filtering operation would modify image texture and erase image details,



(a) Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 0.7$. (A: Upsampling B: Gaussian blurring)

(b) Median filtering window size $w = 5 \times 5$, USM sharpening radius value $r = 3$. (A: Median filtering B: USM sharpening)

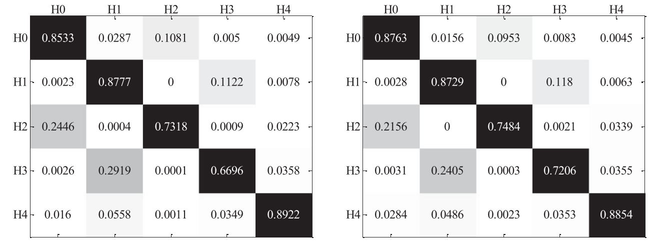
(c) Gaussian blurring variances $\nu = 1.0$, USM sharpening radius value $r = 3$. (A: Gaussian blurring B: USM sharpening)

(d) Upsampling factor $s = 1.5$, USM sharpening radius value $r = 3$. (A: Upsampling B: USM sharpening)

Fig. 4. Confusion matrices for order forensics of image operator chain by using the proposed CNN with known parameter settings.

so the trace left by USM sharpening will be extremely weakened by median filtering. In fact, the detection performance not only depends on our purposed CNN-based method, but also lies in the kinds of manipulations and the order between them.

We detect the image operator chains of Gaussian blurring and USM sharpening, upsampling and USM sharpening. Gaussian blurring variance $\nu = 1.0$, upsampling factor $s = 1.5$, USM sharpening radius value $r = 3$. The confusion matrices are shown in Fig. 4(c, d). It is shown that the proposed CNN could respectively achieve the average classification accuracies of 88.51% and 89.16%, and the maximum value in each row locates at the diagonal line of the confusion matrices. Since the last processing operation always leaves some traces, so H_1 and H_3 are usually difficult to tell apart, as well as H_2 and H_4. It is worth noting that H_2 (altered by USM sharpening only) is sometimes confused with H_0 (unaltered). The substantial reason is USM sharpening with a smaller radius has little influence on the original image. USM sharpening is a well-known technique used in photography to enhance the visual quality of an image by sharpening edges of the elements without increasing noise or blemish. Radius affects the size of the edges to be enhanced or how wide the edge rims become. Since higher radius values would cause halos at the edges (a detectable faint light rim around objects), our experiments utilize a smaller radius. The modification of the original image is small, and the traces left by USM sharpening will be relatively weak. Thus, H_0 and H_2 are also difficult to distinguish.



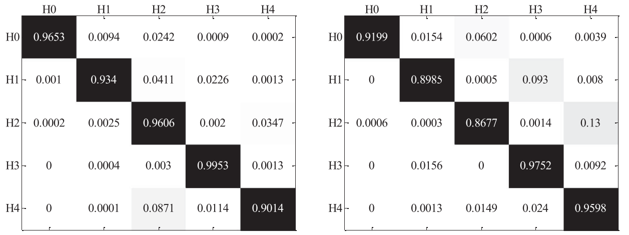
(a) Upsampling factor $s = 1.2$, USM sharpening radius value $r = 1$, without transfer learning strategy

(b) Upsampling factors $s = 1.2$, USM sharpening radius value $r = 1$, with transfer learning strategy

Fig. 5. Comparisons of confusion matrices for order forensics of image operator chain without and with transfer learning strategy. Source task: Upsampling factor $s = 1.3$, USM sharpening radius value $r = 3$. Target task: Upsampling factors $s = 1.2$, USM sharpening radius value $r = 1$. (A: Upsampling B: USM sharpening).

We also consider the image upsampling operation by using cubic interpolation, and detect the image operator chain of image upsampling (factor $s = 1.5$) and Gaussian blurring (variance $\nu = 1.0$). The average classification accuracy is 94.36%. The corresponding confusion matrices are given in Fig. A5 in the Appendix. It can be observed that the choice of interpolation kernel of image upsampling has little effect on the detection performance.

Fig. 5 shows the comparisons of confusion matrices for detecting image operator chain without and with the transfer learning strategy. In order to obtain the forensic detector for image operator chain of the image upsampling factor $s = 1.2$ and USM sharpening radius value $r = 1$, according to the descriptions in Section III-D, we first pre-train a CNN model on images created by image upsampling ($p_A^H = 1.3$) and USM sharpening ($p_B^H = 3$), then fine-tune the CNN model on images created by image upsampling ($p_A^L = 1.2$) and USM sharpening ($p_B^L = 1$). It is shown that the classification accuracies are improved, especially for H_3 class, the classification accuracy is increased by 5.1%. We also observe some slight performance loss on other classes. In CNN-based image operation chain detection framework, the loss function quantifies the amount by which the prediction deviates from the actual values for five classes. CNN attempts to find a global optimum by continuing the backpropagation, which could improve the overall detection performance. The CNN model in the target task is initialized with the feature representations learned from the pre-trained model, instead of random initialization. Since we transfer the weights of all the layers in the pre-trained model to the target task, the CNN model probably falls into another solution space, and decreases the detection accuracy of one or two classes. However, the average detection accuracy of five classes would be increased by using the transfer learning strategy. Moreover, we consider the image operator chain consisting of Gaussian blurring ($p_A^L = 0.6$) and USM sharpening ($p_B^L = 2$). The feature representations learned from Gaussian blurring ($p_A^H = 0.7$) and USM sharpening ($p_B^H = 3$) could be transferred to improve the average classification accuracy. The corresponding confusion matrices are given in Fig. A6 in the Appendix. Therefore, by using the transfer learning strategy, the proposed CNN-based framework could obtain improvements in detecting image operator chain.



(a) Upsampling factor $s \in (1.5, 1.8)$, Gaussian blurring variance $\nu \in (0.7, 1.0)$. (A: Upsampling B: Gaussian blurring)

(b) Gaussian blurring variance $\nu \in (0.7, 1.0)$, Median filtering window size $w = 5 \times 5$ or 3×3. (A: Gaussian blurring B: Median filtering)

Fig. 6. Confusion matrix for order forensics of image operator chain, by using the proposed CNN without prior information. The exact parameters are unknown, but inside a certain range.

C. Image Operator Chain Detection Results of the Proposed CNN Without Prior Information

In this subsection, further robustness tests will be made for a complete assessment of the proposed CNN in case of lack of prior information.

Following the analyses in Section II-B, it is reasonable to assume that the forensic investigators have no access to the parameter settings, but they only know a possible range of the parameter. The CNN model is trained by using a variety of images created by a mixture of operating parameters. Take the image operation chain of image upsampling and Gaussian blurring for example. Suppose that the image upsampling factors are set as $s \in (1.5, 1.8)$ and Gaussian blurring variances are set as $\nu \in (0.7, 1.0)$. We have 16,000 image blocks for H_0 class. For H_1 class, the altered images are respectively manipulated by applying upsampling operation with different factors $s = 1.5, 1.6, 1.7, 1.8$ and then a total of 64,000 image blocks are created. Similarly, we obtain 64,000 altered image blocks for H_2 class by applying Gaussian blurring operation with different variances $\nu = 0.7, 0.8, 0.9, 1.0$. For H_3 and H_4 classes, there are sixteen combinations of parameters, and thus each class has 256,000 manipulated image blocks. Finally, a total of 656,000 image blocks are fed into the proposed CNN model. The testing images are manipulated with the randomly generated parameters in the previous ranges, i.e., the exact parameters are unknown, but inside a certain (known) range. Specifically, for the testing images, the upsampling factor s varies from 1.5 to 1.8, the step size is 0.001, and then we randomly choose $s \in \{1.500, 1.501, 1.502, \ldots, 1.799, 1.800\}$. Moreover, the Gaussian blurring variance ν varies from 0.7 to 1.0, the step size is also 0.001, and then we randomly choose $\nu \in \{0.700, 0.701, 0.702, \ldots, 0.999, 1.000\}$. Our experimental results are summarized in Fig. 6 (a). It is shown that five classes in the image processing history are completely distinguished, and the proposed CNN achieves an overall classification accuracy of 95.13%. Therefore, it can be observed that, without modifying the architecture, the proposed CNN could be trained to detect the image operator chain of upsampling and Gaussian blurring when the operating parameters are unknown but within certain ranges. We also execute the experiments for the image operator chain consisting of Gaussian blurring and median filtering. Assume Gaussian blurring variances are set as $\nu \in (0.7, 1.0)$, and the window sizes of the median filtering are set as $w = 5 \times 5, 3 \times 3$. For the testing images, we randomly choose the Gaussian blurring variance $\nu \in \{0.700, 0.701, 0.702, \ldots, 0.999, 1.000\}$, and the window sizes of the median filtering w is 5×5 or 3×3. The confusion matrix of the image operator chain is given in Fig. 6(b). It can be observed that the maximum element in each row locates at the diagonal line of the confusion matrices, and the average classification accuracy is 92.42%.

We also consider the mismatching condition where the exact parameters are outside the range used for training. For example, the image upsampling factor s is 1.4 or 1.9, and Gaussian blurring variance ν is 1.1. The corresponding confusion matrices are shown in Fig. A7 in the Appendix. The average classification accuracies are 85.96%, 86.12%, and the image operator chains could be determined.

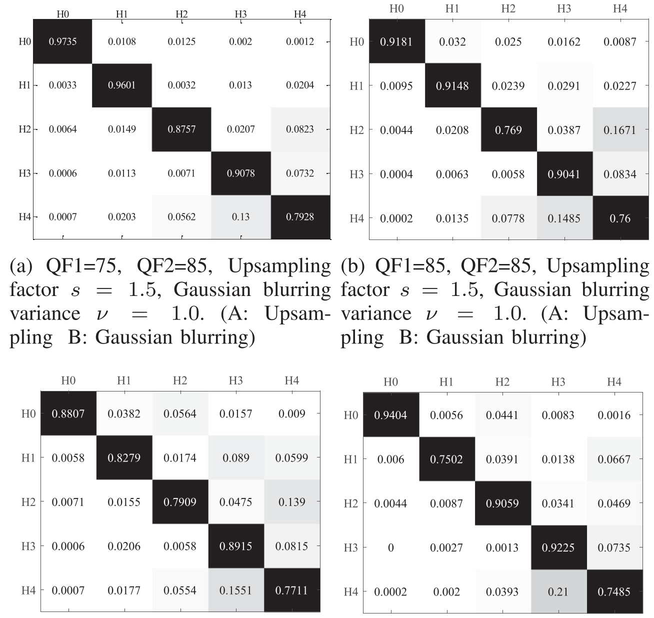
D. Image Operator Chain Detection Results of the Proposed CNN for JPEG Compressed Images

In this subsection, we investigate the case when the processed image is stored in the JPEG format as it is by far the most common scenario in practice. In our experiments, we apply the floating-point DCT implementation and standard quantization tables to obtain the JPEG image database. According Eq. (2), the manipulations are utilized to generate the tampered images, and the corresponding single and double JPEG compression are used to obtain the JPEG image blocks. We generate different sets of JPEG images with different compression quality factors. Table II gives the average detection accuracies of different JPEG image operator chains and quality factors. The corresponding confusion matrices are shown in Fig. B1 in the Appendix. The JPEG image operator chains could be determined with high accuracy.

It is worth mention that [22] failed to detect the JPEG image operator chain, when image upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$, and QF1 = 75, QF2 = 85. The five classes could not be distinguished based on p-map related features. On the contrary, Fig. 7(a) shows that our proposed CNN can successfully distinguish five classes, and thus the image operator chain can be detected in this case.

Fig. 7(b) shows the classification result for detecting JPEG image operator chain with upsampling factor $s = 1.5$ and Gaussian blurring variance $\nu = 1.0$, when two JPEG compressed quality factors are the same QF1 = QF2 = 85. The average accuracy is 85.32%, and it implicitly indicates that the proposed method could achieve good performance for detecting JPEG image operator chain, no matter whether two quality factors are different.

Fig. 7(c) shows the experimental results when the network is not trained for a specific QF1. The Gaussian blurring variance $\nu = 1.0$ and median filtering window size $w = 5 \times 5$. For the training JPEG images, two JPEG compressed quality factors QF1 = 85 and QF2 = 75. The testing images are double compressed by QF1 = 80 and QF2 = 75. The maximum element in each row locates at the diagonal line of the confusion matrices,



(a) QF1=75, QF2=85, Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$. (A: Upsampling B: Gaussian blurring)

(b) QF1=85, QF2=85, Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$. (A: Upsampling B: Gaussian blurring)

(c) Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$, training JPEG images: QF1=85, QF2=75, testing JPEG images: QF1=80, QF2=75. (A: Gaussian blurring B: Median filtering)

(d) The exact parameters are unknown, but inside a certain range. Upsampling factor $s \in (1.5, 1.8)$, Gaussian blurring variance $\nu \in (0.7, 1.0)$, QF1=80, QF2=90. (A: Upsampling B: Gaussian blurring)

Fig. 7. Confusion matrices for order forensics of JPEG image operator chain with different compression quality factors.

and the average classification accuracy is 83.24%. Although the detection performance compared with the matching cases is decreased, the proposed CNN is also able to distinguish the JPEG image operator chain, when the network is tested with non-matching QF1.

We also consider the condition where the setting parameters of operations are not exactly known and the images are JPEG compressed. It can be observed in Fig. 7(d) the average detection accuracy is 85.35% and the maximum element in each row locates at the diagonal line of the confusion matrix. Thus, the proposed CNN could achieve good performance for detecting JPEG image operator chain without prior information.

It should be pointed out that the reported results for detecting JPEG image operator chain is based on a "best case" assumption in which the detector is trained the images with the correct QF1. The comparison results in Section IV-F are also based on the same assumption, and the competing methods are trained on images with the same QF1 as the testing images. In practice, this would require estimating the QF1, and estimation errors will result in a lower performance with respect to the reported "best case" performance.

E. Ablation Study – Influence of Two-Stream Fusion

To evaluate the importance of two-stream components in our proposed network, we perform a full ablation study. We respectively remove the spatial convolution stream and transform-feature extraction stream, and see how that affects performance. Table III shows the comparison results among only spatial convolution stream, only transform-feature extraction stream, and two-stream network. The average classification accuracies obtained by the only spatial convolutional stream and only transform-feature extraction stream are lower, and there is an advantage to use a two-stream rather than single-stream CNN architecture. Specifically, for the image operator chain of Gaussian blurring (variance $\nu = 1.0$) and median filtering (window size $w = 5 \times 5$), compared with each individual stream, the average classification accuracy could be improved by 15.19% and 1.96% by using the proposed two-stream CNN network. Thus, both the spatial convolution stream and transform-feature extraction stream play a crucial role in our proposed network. By fusing the features of two streams, it could improve the detection accuracy.

Note that the network construction of the transform-feature extraction stream without preprocessing layer would become similar to the spatial convolution stream. If we remove the pre-processing from the transform-feature extraction stream, the average classification accuracies obtained by two-stream would be close to that obtained by the only spatial convolutional stream. In fact, as the most important component of the transform-feature extraction stream, the preprocessing operations are used to reveal co-occurrence based local features and capture the hidden residual information. We specially design the preprocessing operations based on the characteristics of two involved manipulations in the image operator chain, so as to extract low-level noise residual features. In this way, the fusion of two streams could reveal both evidence of high-level visual tampering artifacts and low-level noise residual features, contributing to better performance than a single-stream network.

Compared with spatial features, transform features are more discriminative, which are designed ad-hoc for each pair of possible processing operators. We have to confess that the flexibility and generality of the proposed method are not satisfying. For non-JPEG images, our paper designs specific preprocessing operations for detecting different image operator chains, and plans to investigate a universal preprocessing operation in the future.

F. Comparisons With State-of-the-Art Methods

In this subsection, we compare our proposed two-stream CNN method with two state-of-the-art order detection methods for image operator chains, i.e., Chu *et al.*'s method [22] and Bayar *et al.*'s constrained CNN method [33], [34].

Table IV shows all the average classification accuracies obtained by the proposed CNN method are higher. It is because Chu *et al.*'s method is based on a theoretical parametric model of image data, and may not be accurate enough to detect manipulation fingerprints left by an image operator chain. As a data-driven approach, the proposed CNN model tries to directly learn forensic traces induced by a chain of processing operations from image data.

Bayar *et al.* proposed a constrained CNN model with a constrained convolutional layer to perform order detection, and further combined the extremely randomized trees (ERT) classifier. Table V shows the detection performance comparisons

TABLE III
ABLATION STUDY – INFLUENCE OF TWO-STREAM FUSION

Parameter Settings in Image Operator Chain	Spatial Convolution Stream	Transform-feature Extraction Stream	Our Proposed Two-stream CNN
Gaussian blurring variance $\nu = 1.0$, median filter window size $w = 5 \times 5$	79.90%	93.13%	95.09%
Upsampling factor $s = 1.5$, median filter window size $w = 5 \times 5$	88.28%	91.02%	92.86%
Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$	92.71%	95.23%	96.70%

TABLE IV
COMPARISONS WITH CHU *ET AL.*'S METHOD [22]

Parameter Settings in Image Operator Chain	Chu et al.'s Method	Our Proposed Two-stream CNN
Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$	64.39%	96.23%
Upsampling factor $s = 1.2$, Gaussian blurring variance $\nu = 1.0$	63.41%	91.21%
Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 0.7$	64.42%	94.05%
Upsampling factor $s = 1.2$, Gaussian blurring variance $\nu = 0.7$	57.00%	93.77%

TABLE V
COMPARISONS WITH BAYAR *ET AL.*'S CONSTRAINED CNN METHOD [33], [34]

Parameter Settings in Image Operator Chain	Constrained CNN	ERT-based Constrained CNN	Our Proposed Two-stream CNN
Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$	93.00%	94.17%	96.23%
Upsampling factor $s = 1.2$, Gaussian blurring variance $\nu = 0.7$	85.50%	90.90%	93.77%
Gaussian blurring variance $\nu = 1.0$, USM sharpening radius $r = 3$	81.18%	81.21%	88.51%
Gaussian blurring variance $\nu = 0.7$, USM sharpening radius $r = 2$	82.02%	84.40%	86.69%
Median filtering window size $w = 5 \times 5$, USM sharpening radius $r = 3$	76.05%	76.33%	84.51%
Median filtering window size $w = 3 \times 3$, USM sharpening radius $r = 2$	70.23%	70.93%	85.76%
Upsampling factor $s = 1.5$, USM sharpening radius $r = 3$	76.44%	77.55%	89.16%
Upsampling factor $s = 1.2$, USM sharpening radius $r = 2$	77.42%	77.96%	85.06%
Upsampling factor $s \in (1.5, 1.8)$, Gaussian blurring variance $\nu \in (0.7, 1.0)$, without knowledge	91.74%	94.52%	95.13%
QF1=75, QF2=85, Upsampling factor $s = 1.5$, Gaussian blurring variance $\nu = 1.0$	78.85%	81.87%	90.20%
QF1=85, QF2=75, Upsampling factor $s = 1.5$, Median filtering window size $w = 5 \times 5$	69.45%	72.78%	78.25%
QF1=90, QF2=70, Gaussian blurring variance $\nu = 1.0$, Median filtering window size $w = 5 \times 5$	72.10%	78.91%	83.65%

among the constrained CNN, ERT-based constrained CNN, and our proposed two-stream CNN. Some corresponding confusion matrices are provided in Figs. C1–C6 in the Appendix. It is shown that the proposed two-stream CNN could obtain improvements in the detection performance. Furthermore, considering the scenario with no knowledge of processing parameters, we combine Bayar *et al.*'s constrained CNN method with our solution, i.e., training with multiple parameter values as anchor points and testing on a range of parameters. The experimental results show that the image operator chain could be detected, and our solution is applicable to the constrained CNN method.

We have to admit that we compare the performance of the proposed framework and other competing methods under the same assumption that two image operators A and B (as well as QF1 and QF2 in case of JPEG compression) are known to the analyst. In practice, an analyst would have to estimate these parameters, and since those estimations are prone to errors, it would not be possible to replicate the same performance. Generally, estimation errors would result in a lower performance with respect to the reported "best case" performance. In the future, we plan to conduct the evaluation experiments combining with parameters estimation.

V. DISCUSSIONS

In our proposed network, a forensic investigator is always required to train a specific network for each specific chain. Our detection framework is based on some assumptions: 1) Each operation is not applied more than once. 2) The kinds of manipulations in an image operator chain are assumed known to a forensic detector. In a realist scenario, we could first apply universal image forensic strategies to identify the existence of manipulation in the presence of image operator chains. According to these potential manipulations, our proposed CNN network equipped with corresponding preprocessing operations could be used for detecting image operator chains. We have to admit that our proposed method is still rather far away from the ultimate goal of fully automatizing the process of detecting image operator chains. Note that in [33], [34], a forensic investigator can apply a network trained to directly distinguish among multiple possible chains. The proposed CNN-based image operator chain detection framework is flexible and general-purpose, which provides promising views for constructing the practical CNN-based order forensics detectors. That is an important part of our future work.

We provide the computation complexity analysis of the proposed CNN model. The mult-adds of the preprocessing operations and convolutional layers in the proposed CNN model are given in Tables D1-D2 in the Appendix, respectively. The maximum computation of preprocessing operations for non-JPEG images is 0.91 million mult-adds, and the computation of the twenty-five DCT basis patterns for JPEG images is 22.73 million mult-adds. The whole computations of the proposed CNN model for detecting non-JPEG image operator chain and JPEG image operator chain are 754.28 and 853.92 million mult-adds. We can observe that the conv1/transform layers including preprocessing operations would not significantly increase the computational load.

In an image operator chain, the interaction among these manipulations would be more complicated and difficult to learn. The latter operations would weaken the traces left by the previous one, and the tampering evidence would be extremely weak. It is a challenging problem for a forensic investigator to determine the processing history, though there is actually inevitable in the process of creating a fake photograph in the real-world. Hence, we start our research on this problem from manipulation pairs. We believe that if we fully understand this scenario first, we can then investigate how to detect a chain composed of more than two image manipulations. In the future, we will try to extend two-stream idea to design a new CNN model for detecting an image operator chain consisting of more than two manipulations. Further, it would also be of very relevant investigation the limit of how many operations that forensic investigators can detect at most.

VI. Conclusion

Up to now, little attention has been paid to the forensic analysis of multiple heterogeneous manipulations chains, which are actually inevitable in the process of creating a fake photograph. In this paper, we focus on the more difficult and less addressed issue when the image operator chain is utilized. Our contributions can be summarized in the following three aspects.

1) A data-driven order forensic framework for detecting operator chain consisting of two heterogeneous image operations is presented, which can automatically learn and obtain manipulation fingerprints. The proposed two-stream CNN network could explicitly detect both tampering artifact evidence and local noise residual evidence.
2) Various well-designed preprocessing operations are skillfully proposed for different image operator chains, and the transfer learning strategy for image forensics performance enhancement is presented. Experimental results show that our proposed CNN-based method not only achieves significant detection performance but also can determine the order in some cases that previous works were unable to distinguish.
3) The robustness of the proposed image operator chain detection framework is further evaluated and validated in two practical scenarios in which forensic investigators have no access to the operating parameter settings, and the processed image is JPEG compressed.

References

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] 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.

[3] X. Feng, I. Cox, and G. Doërr, "Normalized energy density based forensic detection of resampled images," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 536–545, Jun. 2012.

[4] T. Qiao, R. Shi, X. Luo, M. Xu, N. Zheng, and Y. Wu, "Statistical model-based detector via texture weight map: Application in re-sampling authentication," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1077–1092, May 2019.

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

[6] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain based approach," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4699–4710, Dec. 2013.

[7] M. C. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 3112–3115.

[8] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3, pp. 515–525, Mar. 2014.

[9] V. Christlein, C. Riess, J. Jordan, and C. Riess, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[10] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 3, pp. 507–518, Mar. 2015.

[11] C. Pasquini, G. Boato, and F. Pérez-González, "Multiple JPEG compression detection by means of Benford-Fourier coefficients," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2014, pp. 113–118.

[12] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compressions using first digit features," *APSIPA Trans. Signal Inf. Process.*, vol. 3, pp. E19–1–E19–10, 2014.

[13] T. Bianchi and A. Piva, "Reverse engineering of double JPEG compression in the presence of image resizing," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2012, pp. 127–132.

[14] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Reverse engineering of double compressed images in the presence of contrast enhancement," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2013, pp. 141–146.

[15] V. Conotter, P. Comesaña, and F. Pérez-González, "Forensic detection of processing operator chains: Recovering the history of filtered JPEG images," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2267–2269, Nov. 2015.

[16] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensic strategy based on steganalytic model," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2014, pp. 165–170.

[17] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1–6.

[18] B. Mehdi and J. Fridrich, "Scalable processing history detector for JPEG images," in *Proc. IS&T Int. Symp. Electron. Imag., Media Watermarking, Secur., Forensics*, 2017, pp. 128–137.

[19] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 31–45, Jan. 2018.

[20] P. Comesaña, "Detection and information theoretic measures for quantifying the distinguishability between multimedia operator chains," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2012, pp. 211–216.

[21] M. C. Stamm, X. Chu, and K. J. R. Liu, "Forensically determining the order of signal processing operations," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2013, pp. 162–167.

[22] X. Chu, Y. Chen, and K. J. R. Liu, "Detectability of the order of operations: An information theoretic approach," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 823–836, Apr. 2016.

[23] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2014.

[24] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1849–1853, Nov. 2015.

[25] B. Bayar and M. C. Stamm, "On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2152–2156.

[26] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2016, pp. 1–6.

[27] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps towards camera model identification camera identification with deep convolutional networks," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 259–263, Mar. 2017.

[28] H. Li, S. Wang, and A. C. Kot, "Image recapture detection with convolutional and recurrent neural networks," in *Proc. IS&T Int. Symp. Electron. Imag., Media Watermarking, Secur., Forensics*, 2017, pp. 87–91.

[29] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2016, pp. 1–6.

[30] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 5–10.

[31] B. Mehdi and J. Fridrich, "Deep learning for detecting processing history of images," in *Proc. IS&T Int. Symp. Electron. Imag., Media Watermarking, Secur., Forensics*, 2018, pp. 213–1–213–9.

[32] V. Verma, N. Agarwal, and N. Khanna, "DCT-domain deep convolutional neural networks for multiple JPEG compression classification," *Signal Process.: Image Commun.*, vol. 67, pp. 22–33, 2018.

[33] B. Bayar and M. C. Stamm, "Towards order of processing operations detection in JPEG-compressed images," in *Proc. IS&T Int. Symp. Electron. Imag., Media Watermarking, Secur., Forensics*, 2018, pp. 211–1–211–9.

[34] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.

[35] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "CNN-based detection of generic contrast adjustment with JPEG post-processing," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 3803–3807.

[36] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–10.

[37] "NIST nimble 2016 datasets," [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/

[38] A. D. Ker and R. Böhme, "Revisiting weighted stego-image steganalysis," in *Proc. SPIE, Secur., Forensics, Steganography, Watermarking Multimedia Contents X*, 2008, pp. 1–17.

[39] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.

[40] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 5, pp. 1200–1214, May 2018.

[41] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

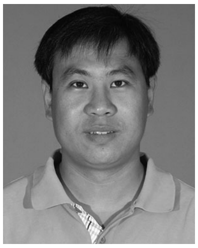
[42] G. Schaefer and M. Stich, "UCID-An uncompressed color image database," in *Prof. SPIE Storage Retrieval Methods Appl. Multimedia*, 2004, pp. 472–480.

[43] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: The ins and outs of organizing BOSS," in *Proc. Int. Workshop Inf. Hiding*, 2011, pp. 59–70.

[44] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

Xin Liao (Member, IEEE) received the B.E. degree and Ph.D. degree in information security from Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2012, respectively. He was a Visiting Scholar with University of Maryland, College Park, MD, USA, from 2016 to 2017. He is currently an Associate Professor with Hunan University, Changsha, China, where he joined in 2012. His current research interests include multimedia forensics, steganography, and watermarking.

Kaide Li received the B.E. degree in information and computing science from the Central South University of Foresty and Technology, Changsha, China, in 2016 and the M.S. degree in Computer Science and Technology from Hunan University, Changsha, China, in 2019. His current research interests include image forensics and information hiding.



Xinshan Zhu (Member, IEEE) received the B.E. degree and the M.E. degree in automation control from the Harbin Institute of Technology, Harbin, China, in 2000 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. His current research interests include deep learning, image processing, and multimedia security.

K. J. Ray Liu (Fellow, IEEE) is a Distinguished University Professor and a Distinguished Scholar-Teach of University of Maryland, College Park, where he is also Christine Kim Eminent Professor of Information Technology. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of information and communications technology with recent focus on wireless AI for indoor tracking and wireless sensing. Prof. Liu was the recipient of two IEEE Technical Field Awards: the 2021 IEEE Fourier Award for Signal Processing and 2016 IEEE Leon K. Kirchmayer Graduate Teaching Award, IEEE Signal Processing Society 2009 Technical Achievement Award, IEEE Signal Processing Society 2014 Society Award, and over a dozen of best paper/invention awards. Recognized by Web of Science as a Highly Cited Researcher, He is a fellow of AAAS and U.S. National Academy of Inventors. As the Founder of Origin Wireless, his invention won the 2017 CEATEC Grand Prix and CES 2020 Innovation Award. Dr. Liu was IEEE Vice President, Technical Activities, and a member of IEEE Board of Director as Division IX Director. He has also served as President of IEEE Signal Processing Society, where he was Vice President C Publications and Editor-in-Chief of *IEEE Signal Processing Magazine*.

He also received teaching and research recognitions from University of Maryland including university-level Invention of the Year Award; and college-level Poole and Kent Senior Faculty Teaching Award, Outstanding Faculty Research Award, and Outstanding Faculty Service Award, all from A. James Clark School of Engineering.