IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2020, accepted January 27, 2020, date of publication January 31, 2020, date of current version February 11, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2970735*

# Deep Learning Local Descriptor for Image Splicing Detection and Localization

**YUAN RAO[1], JIANGQUN NI[2,3], (Member, IEEE), AND HUIMIN ZHAO[4]**

[1]School of Electronic and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
[2]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[3]Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China
[4]School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China

Corresponding author: Jiangqun Ni (issjqni@mail.sysu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1736215 and Grant 61772573, in part by the Science and Technology Program of Guangzhou under Grant 201707010029 and Grant 201804010265, and in part by the Innovation Team Project (Natural Science) of the Education Department of Guangdong Province under Grant 2017KCXTD021.

**ABSTRACT** In this paper, a novel image splicing detection and localization scheme is proposed based on the local feature descriptor which is learned by deep convolutional neural network (CNN). A two-branch CNN, which serves as an expressive local descriptor is presented and applied to automatically learn hierarchical representations from the input RGB color or grayscale test images. The first layer of the proposed CNN model is used to suppress the effects of image contents and extract the diverse and expressive residual features, which is deliberately designed for image splicing detection applications. In specific, the kernels of the first convolutional layer are initialized with an optimized combination of the 30 linear high-pass filters used in calculation of residual maps in spatial rich model (SRM), and is fine-tuned through a constrained learning strategy to retain the high-pass filtering properties for the learned kernels. Both the contrastive loss and cross entropy loss are utilized to jointly improve the generalization ability of the proposed CNN model. With the block-wise dense features for a test image extracted by the pre-trained CNN-based local descriptor, an effective feature fusion strategy, known as block pooling, is adopted to obtain the final discriminative features for image splicing detection with SVM. Based on the pre-trained CNN model, an image splicing localization scheme is further developed by incorporating the fully connected conditional random field (CRF). Extensive experimental results on several public datasets show that the proposed CNN based scheme outperforms some state-of-the-art methods not only in image splicing detection and localization performance, but also in robustness against JPEG compression.

**INDEX TERMS** Image splicing detection, splicing localization, convolutional neural network, feature fusion, conditional random field (CRF).

## I. INTRODUCTION

Image forensic is the science and art to establish the image authenticity, locate the anomalies in an image and reveal the history of image manipulation. Accompanied with the progresses in digital image processing and multimedia communication techniques, image forensic technology develops extremely fast in the last decades and faces consistently growing challenges than ever before. These challenges derive from the popularity of high quality digital camera and the development of user friendly image processing software, e.g., Adobe Photoshop or GNU Gimp, relieving the difficulty of image editing, meanwhile, inevitably promising the easiness of image tampering. By taking elaborately care to guarantee coherent illumination, consistent perspective and proper geometry of objects, the forged image can be extremely realistic and hardly perceived by human perceptual system. Among the countless images uploaded to social media network every day, malicious tampered photographs are appearing with a growing frequency and sophistication, potentially leading to some negative financial, legal or even political consequences in our daily life. Therefore, in order to regain the public trust to digital images, the design of effective image forgery detection tools is of great significance for digital image forensics.

Image splicing, also known as photo composition, is the most common form of image forgery. It consists in inserting fragments of alien images into a source image, which is

The associate editor coordinating the review of this manuscript and approving it for publication was Zhu Han.

usually aimed at deceiving the viewer. In general, the invisible subtle alterations induced by splicing operation can be traced back through physics-based and statistics-based approaches. The former is based on the inconsistencies left at ''scene level'', e.g., motion blur, illumination, perspective and geometry of objects, which usually requires some user interactions to select the investigated regions. For instance, general perspective constraint is applied to splicing detection in [1], which requires user interactions to determine the vanishing line of the reference plane and target borders. In contrast to physics-based approach, the statistics-based one concentrates on artifacts at ''signal level'', e.g., sensor patter noise, demosaicing, compression artifact, in which some necessary prior knowledge are usually explored. For example, given sufficient photographs taken from a host camera, it is capable to estimate photo-response non-uniformity noise (PRNU) [2] that is unique for any camera sensor to detect or localize the forgeries. On the other hand, assuming untouched images from a digital camera are characterized by the presence of CFA demosaicing artifacts [3], [4], the absence of this digital fingerprint can be treated as a forensic cue to detect tampering operations. Alternatively, prior knowledge can also be derived from manipulation history of the pristine image and its fake portions. In particular, assuming the images are saved in compressed JPEG format, the original distribution of DCT coefficients will be changed owing to recompression within spliced regions [5], [6], which can be intuitively applied to detect possible forgeries. However, the prior knowledge are not always available in practical applications, which asks for the involved splicing detection schemes to be more general and less dependent on specific hypothesis. To this end, expressive local features for the subtle artifacts introduced by forgeries in image residual domain are always explored. Following this idea, the local binary pattern (LBP) is proposed for image splicing detection with excellent results [23].

In recent years, deep neural networks (DNNs), such as Convolutional Neural Network (CNN) [10], Recurrent Neural Network (RNN) [11] and Generative Adversarial Networks (GAN) [12] have shown to be capable of characterizing the complex statistical dependencies from high-dimensional sensory inputs and efficiently learning their hierarchical representations, allowing them generalize well across a wide variety of computer vision (CV) tasks, including image classification [13], object tracking [14] and etc. More recently, in light of the powerful feature representation capability of DNNs, they have been applied to massive interdisciplinary applications, such as finger-vein verification [15], short range weather prediction [16] and image steganalysis [17], and achieve superior performance that outperforms the conventional hand-crafted feature based approaches. Inspired by this, a deep learning based image forgery detection scheme was also proposed in our previous work [18], which works quite well on several public benchmark datasets.

### A. MOTIVATION AND CONTRIBUTIONS

In this paper, we substantially improve the scheme in our previous work [18] and generalize it to the application of image splicing localization by taking advantage of the conditional random field model (CRF). For the CNN model adopted in [18], the first convolutional layer of the CNN model is initialized with the 30 high-pass filters in SRM [7]. Although it is proved to be effective for image forgery detection, more than half of the kernels are duplicated, which may attenuate the diversity of the extracting residual features. Intuitively, it is expected that the learned kernels in the first convolutional layer should still retain high-pass filtering property after network training, which could not be ensured with the conventional optimization strategy in weight updating. In addition, the robustness performance against possible JPEG compression in practical applications is not taken into account in [18] as well. Keep this in mind, we develop a new image splicing detection and localization scheme that is capable of learning feature representation automatically based on the supervised CNN framework in this paper. The pre-trained CNN model is used as a local descriptor to represent the test image on a block-by-block basis. And then a block pooling based feature fusion strategy is incorporated to obtain the discriminative feature vector for binary classification with SVM classifier. As an extension of our previous work in [18], this paper includes several new contributions:

- We propose an improved initialization method for the first convolutional layer based on the SRM. Unlike that of [18], we further optimize the initialization strategy in [18] by removing duplicate kernels and assembling SRM filters of similar functionalities, leading to more diverse kernels for residual computation.
- In order to maintain the high-pass filtering property of the kernels in the first layer, we propose a constrained learning strategy by regularizing the elementwise sum of each learned filters to be zero, leading to more distinctive residual-based features for splicing detection.
- The contrastive loss function is employed to decrease the intra-class variation and increase the inter-class deviation, improving the generalization ability of our CNN model.
- Unlike the global 1-D pooling strategy in [18], we propose a new block pooling based feature fusion technique to improve the robustness against JPEG compression.
- Finally, we develop a splicing localization scheme based on the proposed CNN model and fully connected conditional random field (CRF).

The rest of the paper is organized as follows. In Section II, we present a brief review of some conventional and newly-developed deep learning based image forgery detection methods. The proposed image splicing detection and localization scheme are described in Section III and Section IV, respectively, which is followed by the experimental results and analysis in Section V. Finally, the concluding remarks are drawn in Section VI.

## II. RELATED WORKS

Nowadays, the most effective image forgery detection and manipulation detection schemes can be divided into three categories: (1) model based; (2) local descriptor based; and (3) deep learning based approaches. In the followings, the involved techniques, current status and development perspectives are briefly presented.

### A. MODEL BASED APPROACH FOR IMAGE FORGERY DETECTION

The essence of model based forgery detection approach lies in modeling the statistics for a class of images (typically natural images) to reveal the statistical dependency among image pixels. Based on this statistical model, the deviation from these statistics introduced by forgeries can be captured. In [19], Shi *et al*. [19] proposed a natural image model for image splicing detection, where the features extracted from statistical moments of characteristic functions of wavelet subbands are combined with the ones from the Markov transition probability matrixes in DCT domain to obtain the discriminative feature vectors for support vector machine (SVM) classification. The method in [19] was then extended to adopt discrete wavelet transform (DWT) features in [21], leading to a cross-domain feature used to train a SVM classifier. Later, Zhao *et al*. [20] improved the model in [21] by resorting to a 2-D noncausal Markov model to characterize the underlying relationship of adjacent pixels. In general, model based forgery detection approach may undergo expensive computational cost to obtain the statistical model with high-order statistics, and the resulting features are extremely high dimensional, thus the proper feature selection strategies are usually required. In [9] and [22], the spatial rich model (SRM), which is widely used in image steganalysis [7], is generalized to train the SVM classifier for image forgery detection. For forged images involving multiple kinds of tampering, Li *et al*. in [32] proposed an effective forgery detection scheme by taking advantage of the possibility maps obtained with the splicing and copy-move detectors, where the spatial color rich model (SCRM) is incorporated for splicing detection.

### B. LOCAL DESCRIPTOR BASED APPROACH FOR IMAGE FORGERY DETECTION

Tampering operations may inevitably induce the variations of visual elements in images, e.g., texture, illumination or color, and these subtle artifacts can be effectively captured by local feature descriptors for forgery detection. In this context, Muhammad *et al*. [23] employed a steerable pyramid transform (SPT) to the chrominance component of YCbCr images, then applied local binary pattern (LBP) to detect the distortions of texture units for forged images and achieved fairly good detection performances on CASIA v2.0 dataset [24]. Instead of using only one local descriptor, Carvalho *et al*. [25] employed several image descriptors, and color space models as well to expose the artifacts introduced by splicing in image illuminant map, achieving the state-of-the-art splicing detection performance in DSO-1 dataset [26].

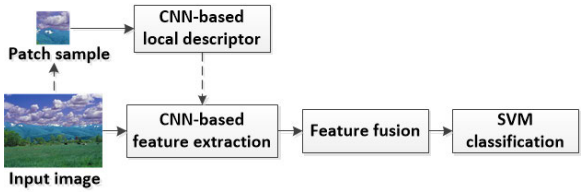



**FIGURE 1.** The framework of the proposed splicing detection approach.

### C. DEEP LEARNING BASED APPROACH FOR IMAGE FORGERY DETECTION

Unlike the arduous process of feature engineering to construct the hand-crafted features in model based and local descriptor based approaches, deep learning based approach can directly learn and optimize the hierarchical feature representations for image forgery detection, which allows end-to-end training and is independent from prior knowledge and human effort in feature design. However, directly applying conventional DNN architecture to image forensic tasks sometimes yields barely satisfactory performance. This is because, DNN tends to model some un-relevant objects, e.g., salient objects or complex textures when the domain-specific SNR (e.g., tampering signal to image content) is not high enough. In recognition of this fact, one intuitive solution is to take advantage of the domain knowledge of the forensic applications. In [27], Ying *et al*. adopted the wavelet features as input of the deep autoencoder for tampering localization. While in [28], by recasting the splicing localization in terms of anomaly detection, forgeries were exposed by autoencoder based on the local features used in [22]. Alternatively, a more systematic solution lies in integrating the domain knowledge into the DNN models. In our prior work [18], a new initialization strategy was applied to regularize the convolutional layer to learn more expressive features for forgery detection, which outperforms several state-of-the-art model based and local descriptor based approaches. Spatial and noise features were adopted in [44] where a two-stream Faster R-CNN had been exploited to detect manipulated regions. More recently, a hybrid LSTM (long short-term memory) and encoder-decoder was adopted in [45] for pixel-wise forgery localization based on resampling and spatial features.

## III. THE PROPOSED CONVOLUTIONAL NEURAL NETWORK (CNN) BASED METHOD FOR SPLICING DETECTION

In this Section, we first present the whole framework of the proposed CNN based image splicing detection approach, and then describe the architecture of the proposed CNN model that acts as a local descriptor for exposing the statistical artifacts caused by image splicing. Next, the customized design of the first convolutional layer for extracting the residual

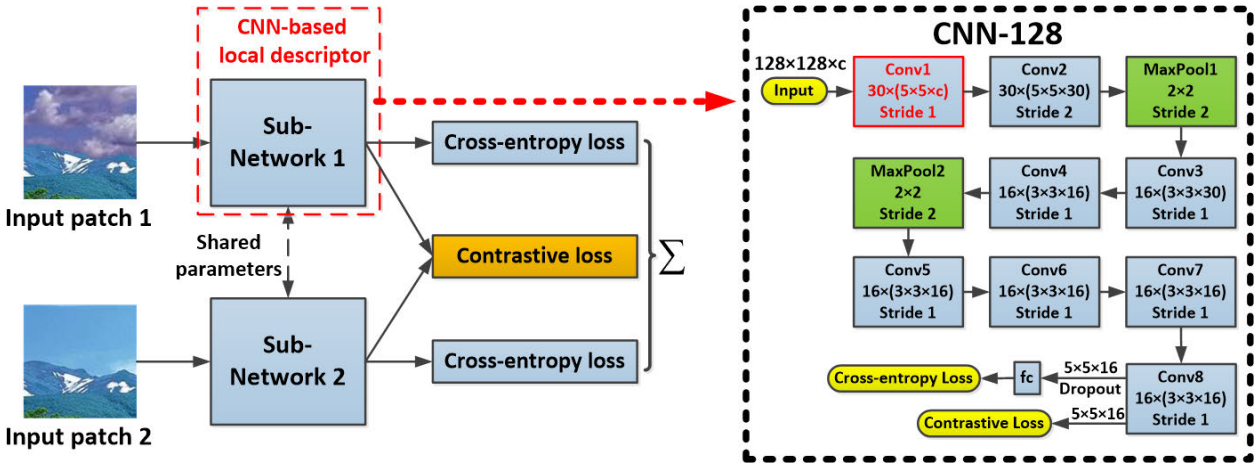



**FIGURE 2.** The architecture of the proposed two-branch CNN and its sub-network (in black dotted boxes). For the sub-network (CNN-128), ReLU and BN layers are not included for brevity. The size of kernels in each convolutional layer is specified as: (number of output feature maps)×height×width×(number of input feature maps). Note that, either of the two sub-networks can be used to validate the performance of pre-trained CNN model due to the parameters sharing.

based features and the contrastive loss function for improving the generalization ability of CNN model are illustrated, respectively. Finally, we show the feature extraction process and the feature fusion strategy to obtain the final discriminative feature vector for SVM classification.

### A. FRAMEWORK OF THE PROPOSED SPLICING DETECTION APPROACH

The framework of the proposed splicing detection approach is illustrated in Fig. 1, which consists of the following four major steps.

#### 1) CNN-BASED LOCAL DESCRIPTOR CONSTRUCTION

In the first step, the proposed CNN model (shown in Fig. 2) is pre-trained based on the labelled patch samples (spliced or pristine) extracted from images in the training set. The pre-trained CNN concentrates on the local statistical artifacts induced by image tampering operations and learns a hierarchical representation for spliced image patches, leading to a powerful local feature descriptor for splicing detection (refer to Section III-B, III-C and III-D for details).

#### 2) CNN-BASED FEATURE EXTRACTION

In this step, the image under investigation is firstly segmented into patch-sized image blocks. The pre-trained CNN-based local descriptor (sub-network of the proposed CNN model) is then applied to extract features for each block, in which, the feature maps of the last convolutional layer are adopted as an expressive feature for an image block (refer to Section III-E for details).

#### 3) FEATURE FUSION

With the CNN-based local descriptors for each block, they are aggregated into a global one to represent the test image. In specific, the extracted local features are integrated with the proposed feature fusion strategy, i.e., block pooling technique, leading to the final discriminative feature for SVM classification (refer to Section III-E for details).

#### 4) SVM CLASSIFICATION

In the final step, based on the discriminative feature vector obtained with the feature fusion strategy, a SVM classifier is trained to perform binary classification, i.e., splicing or authentication.

### B. ARCHITECTURE OF THE PROPOSED CNN

The architecture of the proposed two-branch CNN is illustrated in Fig. 2, where both sub-networks share the identical structure and weight parameters in all of the convolutional layers and fully-connected layers. With the two independent input patch streams, the proposed two-branch CNN is trained with an ensemble of multiple loss functions, i.e., two cross-entropy losses (intra-sub-network) and one scalable contrastive loss (cross-sub-network). Unlike the network of Siamese *et al.*'s work [29], which is used to compute the similarity between input pairs via a similarity function defined at the top of the network, the proposed CNN adopts two additional equal-weighted cross-entropy losses to perform binary classification for the input patches. This is because, for image splicing detection applications, the objective of CNN is not to pursue the similarity of input patch pairs, but to identify correctly the input patches (splicing or authentication) so as to obtain a representative local feature descriptor. Note that, the proposed two-branch CNN not only adopts the cross-entropy loss within branch to supervise the training, but also the contrastive loss with pairwise samples across branches to reduce the intra-class variation and highlight the inter-class deviation. We will show the effectiveness of contrastive loss function later in Section III-D. This "multi-loss" training strategy facilitates to improve the generalization ability of the

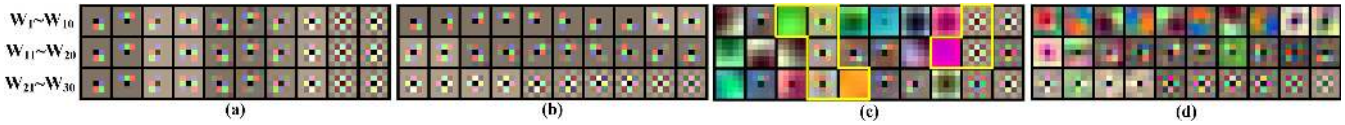



**FIGURE 3.** Visualizations of 30 kernels in the first convolutional layer. (a) Kernels initialized by the method in [18]. (b) Kernels initialized by the proposed improved initialization strategy. (c) Kernels in (a) fine-tuned by SRM-CNN. (d) Kernels in (b) fine-tuned by C_ISRM-CNN. Duplicated kernels and non-high-pass filtering kernels are marked in yellow boxes in (c).

**TABLE 1.** 30 High-pass filters and their corresponding residual classes in SRM.

| Class ($i$) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Class Name | 1a | 2a | 3a | E3a | E5a | S3a | S5a |
| Filter set ($c_i$) | $F_1 - F_8$ | $F_9 - F_{12}$ | $F_{13} - F_{20}$ | $F_{21} - F_{24}$ | $F_{25} - F_{28}$ | $F_{29}$ | $F_{30}$ |

CNN model so that a more distinct feature representation for input patch can be learned. Owning to the parameter sharing strategy across the sub-networks, either of the two pre-trained sub-networks (CNN) in the two-branch CNN model can serve as a deep learning based local descriptor for splicing detection.

In our work, patch (block) size of 128×128 is used for splicing detection and localization applications. A CNN model, i.e., CNN-128 is proposed, in which the first convolutional layer is adopted for residual computation, as illustrated in the doted boxes in Fig. 2. In addition to the first convolutional layer, CNN-128 consists of another 7 convolutional layers, 2 max-pooling layers and a fully-connected layer followed by a 2-way softmax classifier. Recent studies show the importance of batch normalization (BN) [30] to optimize CNN, which helps to reduce the internal-covariate-shift by normalizing the input distribution to the standard Gaussian. As a result, BN layers are applied to the output of each convolutional layer in the proposed CNN model. The activation function − ReLU is adopted after each BN layer to enforce sparsity in output feature maps [31]. Note that, equipped with only one necessary filly-connected layer with dropout technique [10], the proposed CNN model exhibits a ''lightweight'' structure in favor of effectively avoiding overfitting. Based on the pre-trained CNN-128, the input 128×128 patches are expressed with the local descriptors in terms of the feature maps of the size 5×5×16 (the output of ''Conv8'' in CNN-128).

We then proceed to discuss the naming conventions of the CNN based descriptors to simplify the performance evaluation. The CNN models, whose first convolutional layer is initialized with SRM [7] and the proposed improved initialization strategy, are referred to as SRM-CNN and ISRM-CNN, respectively. The ISRM-CNN with constrained learning, which will be shown in Section III-C, is known as C_ISRM-CNN, and it becomes C_ISRM_C-CNN when incorporating the contrastive loss function. In rest of this Section, we will elaborate the involved key techniques in our proposed CNN model.

### C. THE FIRST CONVOLUTIONAL LAYER

The residual based local descriptor has long been proved to be effective for image forgery detection in the prior arts [8], [9] and [22]. In [18] and [33], it is further shown that the local descriptor can be recasted as the CNN model. In our previous work [18], the first convolutional layer of the CNN based local descriptor is initialized with the 30 base high-pass filters in SRM. Although it is shown to be effective in residual computation, some of the implementation can still be improved. In this sub-section, we will propose an improved structure for the first convolutional layer with two improvements: (1) the improvement on the initialization strategy in [18]; (2) the constrained learning strategy for updating the kernel weights.

#### 1) IMPROVED INITIALIZATION STRATEGY

Let $\boldsymbol{W}_j = [W_{j1}\ W_{j2}\ W_{j3}]$ denote the filter assignment for R, G and B channels, i.e., convolution kernel of the $j^{th}$ output feature map ($j = 1, \cdots, 30$) in the first convolutional layer. $\boldsymbol{F} = [F_1, F_2, \cdots, F_{30}]$ denotes the filter bank with 30 high-pass filters in SRM, where $W_{ji} \in \boldsymbol{F}$. To generate a feature map for input RGB test images, three filters which are similar but not identical in $\boldsymbol{F}$ are required [18]. For the initialization strategy in [18], the filters are re-used in sequence of their arrangement in $\boldsymbol{F}$, leading to $\boldsymbol{W}_j = \boldsymbol{W}_{((j-1)\boldsymbol{od}10)+1}$. Under the circumstances, the kernels are resembled to each other as shown in Fig. 3(a), which may attenuate the diversity of the resulting residual signals, even though these kernels can be fine-tuned during network training. Besides, in [18], many kernels, e.g., $\boldsymbol{W}_{j=3,\ 7,\ 10}$, are composed of the filters belonging to different residual classes with distinct high-order statistics, which could be difficult to be modeled by CNN. In view of this, we propose an improved initialization strategy as follows. The 30 basic high-pass filters (linear filters and their rotated counterparts) used for initialization correspond to 7 residual classes of type ''spam'' in SRM, which include 8 filters in class ''1a'', 4 in class ''2a'', 8 in class ''3a'', 4 in class ''E3a'', 4 in class ''E5a'', 1 in class ''S3a''

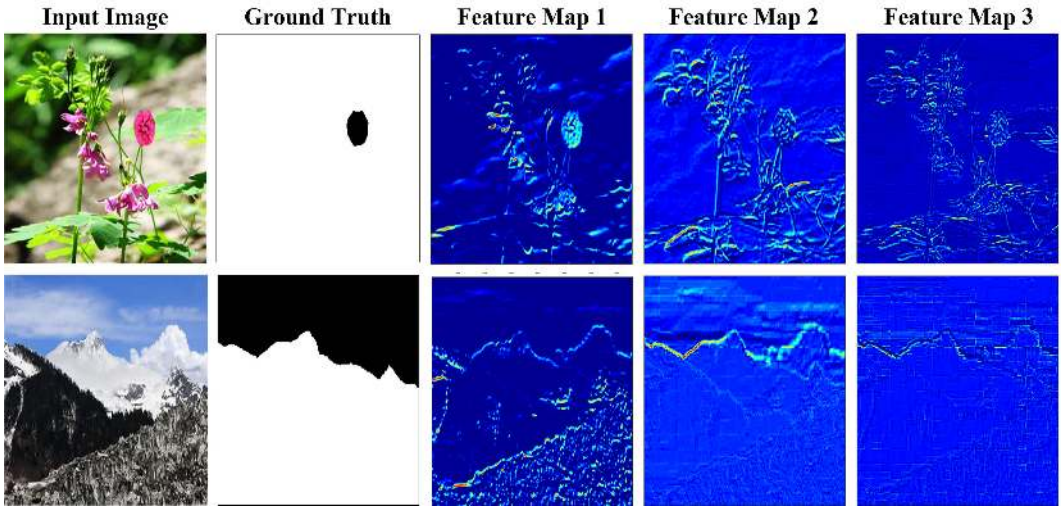



**FIGURE 4.** Output feature maps in the first convolutional layer of C_ISRM-CNN for two tampered images.

and 1 in class "S5a". If we denote the filter set in $i^{th}$ class by $c_i$ $(i = 1, \cdots, 7)$, as illustrated in Table 1, we have $c_1 = \{F_1, \cdots, F_8\}$, $c_2 = \{F_9, \cdots, F_{12}\}$, $\cdots$, $c_7 = \{F_{30}\}$ and $c = \{c_1, \cdots, c_7\}$. To initialize the convolution kernel $\boldsymbol{W}_j$ of size 5×5 for the R, G and B channels in the first convolutional layer, three filters in the same class are utilized by padding zeros around the high-pass filters in SRM. For $c_1$ to $c_5$, more than three SRM filters are included in each filter set, we initialize $\boldsymbol{W}_{j=1,\cdots,28}$ with 28 filters taken from $c_i = \{F_m, \cdots, F_n\}$ $(i = 1, \cdots, 5$ and $m \le j \le n)$ as follows:

$$\boldsymbol{W}_j = \begin{cases} [F_j\ F_{j+1}\ F_{j+2}], & if\ m \le j \le n-2 \\ [F_j\ F_n\ F_m], & if\ j = n-1 \\ [F_j\ F_m\ F_{m+1}], & otherwise \end{cases} \quad (1)$$

For the remaining two filter sets ($c_6$ and $c_7$), each with only one filter of different size, the involved filters are actually used to compute the submodels (residual class "S3a" and "S5a") of the same "SQUARE" class in SRM, and they are indeed the same type of square kernel. Therefore, we initialize $\boldsymbol{W}_{j=29,30}$ by combining these two filters symmetrically:

$$\boldsymbol{W}_{29} = [F_{29} \quad F_{30} \quad F_{29}], \quad (2)$$
$$\boldsymbol{W}_{30} = [F_{30} \quad F_{29} \quad F_{30}]. \quad (3)$$

The improved initialization strategy could help to: (1) generate 30 diverse kernels as illustrated in Fig. 3 (b); and (2) ensure that each kernel comes from the filters of the same residual class so as to obtain the residual with the same statistical characteristic, leading to a better local descriptor. To further show the superiority of the improved initialization strategy, we then quantitatively evaluate the detection performance of SRM-CNN and ISRM-CNN for CASIA v2.0 [24] and DSO-1 [26] datasets. As shown in Table 2, ISRM-CNN outperforms SRM-CNN by 0.5%∼1% in terms of detection accuracy in both datasets due to the better starting point for model training, indicating the effectiveness of the improved initialization strategy.

**TABLE 2.** The performance comparison in terms of detection accuracy for different network with 1-D pooling.

| Dataset | 1-D Pooling | Acc(%) | | | |
|---|---|---|---|---|---|
| | | SRM -CNN | ISRM -CNN | C_ISRM -CNN | C_ISRM _C-CNN |
| CASIA v2.0 | Max | 93.51 | 94.49 | 95.62 | 96.33 |
| | Mean | 94.00 | 94.49 | 95.14 | 96.70 |
| DSO-1 | Max | 94.50 | 95.50 | 96.50 | 96.50 |
| | Mean | 92.50 | 93.00 | 94.00 | 95.50 |
| DVMM | – | 96.38 | – | 96.73 | 97.04 |

### 2) CONSTRAINED LEARNING STRATEGY

The utilization of the improved initialization strategy could lead to the first convolutional layer to generate more diverse residual feature maps. However, such potentials could be somehow offset during model training, for the conventional learning algorithm, e.g., stochastic gradient descent, could not ensure retaining the high-pass filtering property of the kernels in the first convolutional layer when updating the kernel weights. To tackle this issue, a constrained learning strategy is adopted by forcing the resulting kernels in the first layer to be high-pass filtering in weight updating. Let $W_{jk}^n$ be the weight matrix of $n^{th}$ iteration in $\boldsymbol{W}_j$ for channel $k$ $(k = 1, 2, 3)$, we have:

$$\sum_{1 \le x,y \le 5} W_{jk}^n(x, y) = 0, \quad (4)$$

and (4) is initially held for each involved high-pass filters in SRM. And the weight updating process can be formulated as:

$$W_{jk}^n = W_{jk}^{n-1} + \Delta W_{jk}^{n-1}, \quad (5)$$

where $\Delta W_{jk}^n$ is the increment of $W_{jk}^n$. To enforce (4) in each iteration, it is intuitive to modify the learning strategy by redefining $\Delta W_{jk}^n$ as:

$$\Delta W_{jk}^n = \Delta W_{jk}^{n-1} - E(\Delta W_{jk}^{n-1}), \tag{6}$$

where $E(\cdot)$ denotes the expectation function. The constrained learning could maintain the high-pass filtering property of the learned kernels in the first layer consistently during model training, and help to generate more expressive residual feature maps.

To show the effectiveness of the constrained learning strategy (C_ISRM-CNN), we also evaluate it on CASIA v2.0 and DSO-1 for RGB images and DVMM for grayscale images as illustrated in Table 2. It is ready to see that, for RGB images, the C_ISRM-CNN increase the detection performance over ISRM-CNN by 0.65%∼1.13%. For the DVMM dataset with grayscale images, it is noted that the first convolutional layer is initialized with the method in [18], because the improved initialization strategy is designed specifically for RGB images. And the slight performance gain of C_SRM-CNN over SRM-CNN may attribute to the smaller sample size of DVMM. In addition, the superiority of the C_ISRM-CNN can also be better explained by the learned kernels in the first convolutional layer with visualization tool. As shown in Fig. 3 (c), for SRM-CNN, there exist many similar learned kernels (marked with yellow boxes), some filters exhibit in pure color and behave like smoothing filters. While the learned kernels for C_ISRM show much more diversity and retain the high-pass filtering property, indicating that more distinctive residual feature maps can be generated with the proposed C_ISRM-CNN. Finally, to show the effectiveness of the first convolutional layer in suppressing the interference of image contents, we depict three output feature maps in the first convolutional layer of C_ISRM-CNN for two forged images on CASIA v2.0 dataset. As illustrated in Fig. 4, image contents (backgrounds) are successfully suppressed and low-level forensic features (edge of objects) are extracted, which facilitate the subsequent deeper layers of C_ISRM-CNN to learn better high-level feature representation for splicing detection.

### D. CONTRASTIVE LOSS LAYER

An expressive CNN-based local descriptor should ensure features extracted from patches of the same identity to be similar and those of different identity to be distinct. In specific, this amounts to decrease the intra-class variation and increase the inter-class deviation for the proposed CNN. To this end, we take advantages of the contrastive loss function [35] and compute the contrastive loss $J_c$ as follows:

$$\begin{aligned} J_c &= \frac{1}{2N}\sum_{n=1}^{N}(yd^2 + (1-y)max(m-d,0)^2) \\ y &= \mathbb{I}_{y_n^1}(y_n^2), \end{aligned} \tag{7}$$

where $N$ is the batch size of input pairs, $d = \|f_n^1 - f_n^2\|$ is the L2 distance between two feature vectors $f_n^1$ and $f_n^2$ extracted from the $n^{th}$ input pair of patches labeled by $y_n^1$ and $y_n^2$ and $m$ represents the margin that controls $d$. The indicator function $\mathbb{I}_i(x) = 1$ if $x = i$, otherwise $\mathbb{I}_i(x) = 0$. Essentially, the contrastive loss function tends to minimize the $L_2$ norm of two feature vectors from the same class, while it requires the $L_2$ distance of two feature vectors from different class to be larger than a pre-defined margin $m$.

With two objective functions, i.e., the contrastive loss and cross-entropy loss, the proposed two-branch CNN is trained to minimize the sum of these two supervisory loss functions weighted by a hyper parameter $\lambda$ as follows:

$$J = 0.5 \cdot J_{s1} + 0.5 \cdot J_{s2} + \lambda \cdot J_c, \tag{8}$$

where $J_{s1}$ and $J_{s2}$ are the cross-entropy losses computed in the softmax layer of each sub-network.

To verify the effectiveness of the adopted contrastive loss function, we visually compare the extracted feature representations of the input patches with the pre-trained C_ISRM-CNN and C_ISRM_C-CNN models on CASIA v2.0 dataset. For fair comparison, the two involved models are trained on the same training set and are applied to extract features on the same validation set with same number of spliced and pristine patches. The extracted 400-D features ($5\times5\times16$, the output of "Conv8" layer of CNN-128 in Fig. 2) are projected to 2-D space for visualization as shown in Fig. 5. It is ready to see that, compared with the features for C_ISRM-CNN, the intra-class distance of features for C_ISRM_C-CNN is notably decreased and the inter-class variation is increased. As a result, the extracted features are more linearly separable, which helps to improve the classification accuracy for the subsequent SVM classifier. As shown in Table 2, C_ISRM_C-CNN increases the detection performance over C_ISRM-CNN by 0.71%∼1.56% on CASIA v2.0 dataset and 1.5% on DSO-1 dataset, respectively.

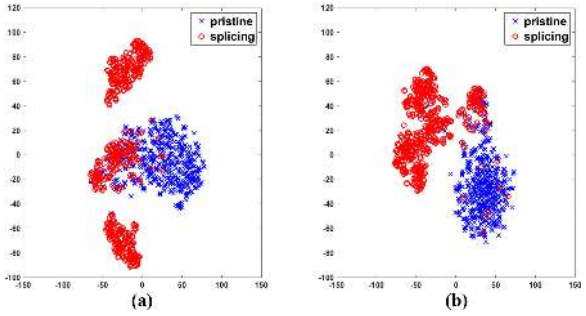



**FIGURE 5.** The visualization of the output feature maps in "Conv8" layer of CNN-128 extracted by (a) C_ISRM-CNN and (b) C_ISRM_C-CNN on CASIA v2.0 dataset, respectively.

### E. FEATURE EXTRACTION AND FEATURE FUSION

In this sub-section, we elaborate the detailed procedure to generate discriminative feature vectors for SVM classification. As illustrated in Fig. 6, this process is mainly composed of a 3-step pipeline:

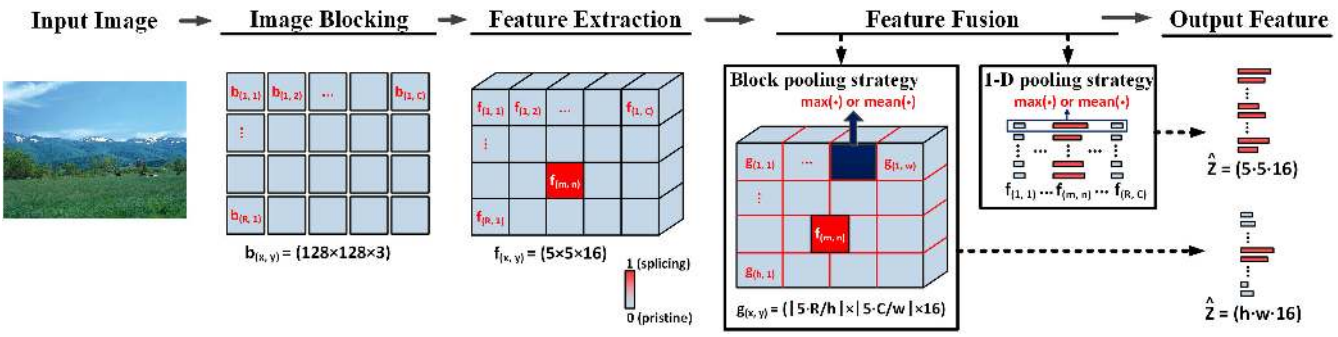



**FIGURE 6.** The process of generating a discriminative feature for SVM classification and the comparison between two feature fusion strategies. Note that the red blocks in the reconstructed feature map correspond to the suspicious splicing regions with activated values larger than the ones in blue blocks corresponding to the pristine regions.

#### 1) IMAGE BLOCKING

In the first step, we conduct image blocking for the input images. The block size, 128×128, is set to be consistent with the input patch size of the pre-trained CNN model in order to extract block-wise features. For an investigated image $\boldsymbol{I}$ of size $H \times W$, we partition it into $R \times C$ non-overlapping blocks, and denote each block as $b(x, y)$, where $(x, y)$ is the block index ($1 \le x \le R$ and $1 \le y \le C$).

#### 2) FEATURE EXTRACTION

In the second step, the block based features are extracted with the pre-trained CNN based local descriptor $Conv(\cdot)$, i.e., C_ISRM_C-CNN for RGB images. For input patch block $b(x, y)$, it is expressed as a much condensed feature representation $f(x, y) = Conv(b(x, y))$ with CNN based local descriptor, where $f(x, y)$ is of size 5×5×16 (5×5 matrix of 16 channels), i.e., the output feature map of layer "Conv8" for CNN-128. Then, all the extracted features $f(x, y)$ for each block are assembled to constitute the new representation $\hat{F}$ of size $(5{\cdot}R){\times}(5{\cdot}C){\times}16$ for input image $\boldsymbol{I}$, i.e., 16 feature images of size $(5{\cdot}R){\times}(5{\cdot}C)$.

#### 3) FEATURE FUSION

In the third step, we generate the final discriminative feature vector according to $\hat{F}$ with the proposed feature fusion strategy, known as block pooling. For block pooling, it re-divides each feature image of size $(5{\cdot}R){\times}(5{\cdot}C)$ into $h \times w$ blocks, and then fuse the block-wise features $g(x, y)$ ($1 \le x \le h$ and $1 \le y \le w$) of size $\lfloor \frac{5{\cdot}R}{h} \rfloor \times \lfloor \frac{5{\cdot}C}{w} \rfloor$ ($\lfloor \cdot \rfloor$ is a floor function) by applying pooling to the blocks of each feature image independently,

$$Z_k = [pool(g^k(1, 1)), \cdots pool(g^k(h, w))], \tag{9}$$

where $k \in [1, 16]$ and $pool(\cdot)$ represents the max or mean function. Finally, by concatenating all the $Z_k$ together, i.e.,

$$\hat{Z} = [Z_1, \cdots Z_{16}], \tag{10}$$

we obtain a $h{\cdot}w{\cdot}16$ dimensional discriminative feature vector for SVM classification. Note that, in our implementation, we take a fixed $h \times w$ mesh generation for feature images of various sizes, leading to a fixed number (i.e., $h \cdot w$) of blocks regardless of the feature image size so as to maintain the invariance of dimension for the final feature vector, which is known as the $h \times w$ block pooling.

Unlike the global 1-D pooling strategy in our previous work [18], which is devised to apply pooling operation on each dimension of the vectorization of $f(x, y)$ over $R \cdot C$ extracted features, the proposed one is a local pooling strategy. To show the superiority of the proposed block pooling strategy, we then further compare these two techniques as illustrated in Fig. 6. In the interest of simplicity, we take $k = 1$. Note that the normalized activation values corresponding to the pristine blocks are usually smaller than the spliced ones in the extracted feature map of our CNN. If block $b(m, n)$ of the input pristine image is misclassified as forgery, then $f(m, n)$ becomes the most dominant feature vector among the involved $f(x, y)$, $1 \le x \le R, 1 \le y \le C$, and it would spread over along each dimension of the output feature vector $\hat{Z}$ when the 1-D pooling strategy is adopted. However, at most the values of two dimensions of the output feature vector are affected for the block pooling strategy. This explains why the local block pooling strategy tends to be more robust in detection of pristine images, especially in the case of JPEG compression which will be verified later in Section V-C.

## IV. THE PROPOSED CONVOLUTIONAL NEURAL NETWORK (CNN) BASED SPLICING LOCALIZATION

Compared to detection, splicing localization is much more challenging, which requires pixel-wise predictions on the test image. As an extension of [18], in this Section, we propose a splicing localization method based on the pre-trained CNN descriptor and a fully connected conditional random field model (CRF) [36].

### A. CNN-BASED SLIDING WINDOW LOCALIZATION

In the training stage, based on the labelled patch of size 128 × 128 sampled from the training images, we pre-train the same supervised CNN model as the one for splicing detection. The softmax classifier at the top of the pre-trained CNN outputs the probability $l \in [0, 1]$ of being tampered

for each input patch. In the testing stage, we analyze the test image through the patch-sized sliding window with a stride of $s$ pixels. Let $L$ be the predicted binary label map with the identical size of the test image, the predicted label $L(x, y)$ at position $(x, y)$ of $L$ is computed as follows:

$$L(x, y) = [\frac{1}{K}\sum_{k=1}^{K} l_k \geq \tau], \tag{11}$$

where $[P]$ is the Iverson bracket ($[P] = 1$ if statement $P$ is true, otherwise $[P] = 0$), $l_k$ is the tampering probability for the $k^{th}$ block containing $(x, y)$, $K$ is the number of these blocks and $\tau$ is the threshold whose optimal value is obtained from training data. It is noted that $K = (p/s)^2$ for most $L(x, y)$ because the size of each block (sliding window) is $p \times p$ pixels, while $K$ is smaller for those $L(x, y)$ near the image boundary. Morphological operations, e.g., dilation and erosion, are then performed to erase tiny isolated regions and fill the holes. According to [32], the resulting binary map is less reliable than the continuous possibility map in tampering localization, we then further apply mean filtering with a window size of $64 \times 64$ to smooth the predicted label map $L$, leading to a continuous splicing possibility map $L^{MF}(0 \leq L^{MF} \leq 1)$ which indeed helps to boost the localization performance according to our experimental results in Section V-D.

### B. REFINING BY FULLY CONNECTED CRF

Although mean filtering eliminates the mosaic artifacts induced by sliding window, there still is room for refinement of the resulting splicing possibility map. To recover the splicing edges with a fine-grained predicted label map, we resort to fully-connected CRF, which has shown superior performance in semantic segmentation under the framework of deep learning [37]. Let $l_i$ be the label assignment for pixel $i$, we perform pixel-wise labeling on the splicing possibility map $L^{MF}$ by minimizing the following energy function based on mean field approximation [36]:

$$\min_{l_i} E(\boldsymbol{l}) = \min_{l_i}(\sum_i \psi_u(l_i) + \sum_{i<j} \psi_p(l_i, l_j)), \tag{12}$$

where the unary potential $\psi_u(l_i)$ is the data term based on class scores computed by the classifiers, and the pair-wise potential $\psi_p(l_i, l_j)$ is a smooth term based on the local interactions of pixels. In the same way to produce the distribution over two class labels in the softmax classifier of CNN, $\psi_u(l_i)$ is computed based on the class conditional probability as follows:

$$\psi_u(l_i) = -\log \frac{\exp(v_i)}{\sum_j \exp(v_i(j))}, \tag{13}$$

where $v_i = [L^{MF}(x, y), 1 - L^{MF}(x, y)]$ is a 2-D feature vector for pixel $i$. The pair-wise potential $\psi_p(l_i, l_j) = \mu(l_i, l_j)k(f_i, f_j)$ consists of a label compatibility function $\mu(l_i, l_j) = 1 - \mathbb{I}_{l_i}(l_j)$, which introduces a penalty for nearby similar pixels that are assigned different labels, and a Gaussian kernel $k(f_i, f_j)$, which depends on the extracted features $f$ (pixel intensity and position) for pixel $i$ and $j$, and is defined by the linear combination of bilateral filtering kernel (weighted by $w_1$) and spatial kernel (weighted by $w_2$) as follows:

$$k(f_i, f_j) = w_1 \exp(-\frac{\|p_i - p_j\|^2}{2\theta_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\theta_\beta^2}) + w_2 \exp(-\frac{\|p_i - p_j\|^2}{2\theta_\gamma^2}), \tag{14}$$

where $I$ and $p$ are the pixel color intensity and pixel position, respectively. Bilateral filtering term is designed for clustering the pixels with similar colors and nearby positions to be in the same class, and the spatial term is used for removing small isolated regions in the splicing localization results. The hyper parameters $\theta_\alpha$, $\theta_\beta$ and $\theta_\gamma$ in (14) are learned from data to control the degree of nearness and similarity. Note that we find that the localization performance would not increase significantly with CRF training based on extensive experiments, which is mainly due to the unary potential is only computed from a 2-D feature vector. Considering the trade-off between computational cost and localization performance, following [37], we only perform CRF inference with the default setting in [36] when refining the localization result in our implementation. We will evaluate the splicing localization performance of our CNN based scheme in Section V-D.

## V. EXPERIMENTAL RESULTS AND ANALYSIS

In this Section, extensive experiments are carried out to demonstrate the effectiveness of our CNN based method for image splicing detection and localization. We compare our method with several state-of-the-art image forgery detection and localization methods on some benchmark datasets.

### A. DATASETS AND DATA PREPARATION

In this paper, all of our experiments are conducted on 3 public datasets for forgery detection, i.e., CASIA v2.0 [24], Columbia gray DVMM [34] and DSO-1 [26]. The DVMM dataset consists of 933 authentic and 912 spliced images of size $128 \times 128$ pixels in BMP format without any post-processing. DSO-1 dataset is composed of 100 spliced images with pixel-wise ground truth and 100 pristine images in the resolution of $2{,}048 \times 1{,}536$ pixels. While the CASIA v2.0 database contains 7,491 authentic and 5,123 forged (spliced and copy-move) color images with the size ranging from $240 \times 160$ to $900 \times 600$ pixels in JPEG and TIFF formats. The forged images in both CASIA v2.0 and DSO-1 datasets are post-processed to increase detection difficulty. Note that the proposed CNN based method is specifically designed for image splicing detection and localization, thus only the splicing images and the same number of pristine images randomly selected from the authentic ones are utilized in CASIA v2.0 dataset. To learn an expressive CNN based local splicing descriptor, the statistical inconsistency near edges of spliced regions should be captured during network training, in other words, the training data for the CNN model should consist of massive authentic samples (negative) and

the ones that are partially tampered (positive). In light of this, we draw the patches randomly alone the boundaries of splicing regions in a forged image for the positive samples. In specific, a patch-sized sliding window with a fixed stride is applied to extract patches along the splicing boundaries. For negative samples, we randomly draw equal number of patches from the authentic images. Note that all the patches are only extracted from the training images, which are divided into a training set for training the CNN model and a validation set for testing its performance. To avoid overfitting in CNN training, some label-preserving transformations, e.g., transposing and rotating, are conducted on the training set, which increases the dataset by a factor of 8. For DVMM dataset, the images in it are of size 128×128 pixels, and can be directly used for CNN training after data augmentation. Therefore, the patch sampling process is only conducted on CASIA v2.0 and DSO-1 datasets.

### B. IMPLEMENTATION DETAILS

To evaluate the detection and localization performance of the our CNN based method and other involved methods, for each data set, we present the experimental results based on the average accuracy over 5-fold cross validation. For CASIA v2.0 and DSO-1 databases, we draw 10,064 and 29,430 patch samples of size 128×128 from the training images, respectively. Then 5/6 of the extracted patches are randomly picked out to train the proposed CNN model and the rests are used to validate the network performance.

The CNN model of C_ISRM_C-CNN for CNN-128 is trained on CASIA v2.0 and DSO-1 datasets for RGB images, while C_SRM_C-CNN is trained on DVMM dataset for grayscale images. The mini-batch size for CNN-128 is set to 128. Within each mini-batch, the same numbers of positive and negative samples are adopted. To incorporate the contrastive loss function, patch pairs across channels are sampled offline so as to generate equal numbers of positive (patches in the same class) and negative (patches in different classes) pairs. The training of the proposed CNN model (for local feature extraction) is implemented using Caffe [38] and its Matlab interface. For all the involved CNN models, we minimize the multi-loss function (8) based on SGD optimization with a momentum value of 0.99 and the initial learning rate is set to 0.01 with 10% of decrement every 10 epochs. To balance the convergence rate between softmax layer and contrastive loss layer, we set the margin $m$ to 10 and the weight $\lambda$ of contrastive loss function to 0.01. In addition, weight decays are fixed to $5 \times 10^{-3}$ and $1 \times 10^{-3}$ during training stages for CASIA v2.0 and DSO-1 datasets, respectively.

The output feature map of "Conv8" layer (5×5×16) for pre-trained C_ISRM_C-CNN is used to characterize the 128×128 RGB patches in CASIA v2.0 and DSO-1 datasets. Feature fusion is then conducted by adopting 5×5 ($h = w = 5$) block-pooling, leading to 400-D discriminative feature vectors for test images in CASIA v2.0 and DSO-1 datasets. In the interest of fair comparisons, the feature dimension obtained by the proposed block pooling strategy is the same as 1-D pooling. a C-support vector machine (SVM) [39] with non-linear RBF kernel is trained using the resulting discriminative features for splicing detection, where the optimal parameters ($C$, $g$) of the SVM classifier is determined by an exhaustive grid search strategy. For DVMM dataset with 128×128 grayscale images, we apply C_SRM_C-CNN to perform classification directly, and bypass the processes described above, e.g., feature fusion and SVM classification.

### C. SPLICING DETECTION PERFORMANCE

We first compare the splicing detection performance of our CNN based method with other state-of-the-art CNN and hand-crafted feature based methods. Recall that the block pooling strategy in our method is specifically designed to improve the robustness performance, we show its effectiveness by comparing our CNN based detector with the method in [18] (the same CNN model with 1-D pooling strategy) under JPEG attack. When JPEG compression is not applied, the proposed method with block pooling outperforms consistently the one in [18] with 1-D pooling on CASIA v2.0 and DSO-1 datasets as shown in Table 3. Although only a slight performance gain is achieved on CASIA v2.0, it is ready to see that the proposed method with block pooling increases the performance in terms of TNR (true negative rate) by 4% when compared with the one with 1-D pooling on DSO-1. To show the effects of JPEG attack on feature extraction process, we visually compare the extracted feature image $\hat{F}$ (refer to Section III-E) with and without JPEG compression. Due to the blurred boundaries of tampered regions and the blocking artifacts caused by JPEG compression, it not only attenuates the traces of splicing operation for the forged images, but also increases the false alarms as illustrated in Fig. 7. To evaluate the robustness against JPEG compression for the proposed CNN based detector and the competing methods, we compress the images in CASIA v2.0 and DSO-1 with QF 65, 75, 85 and 95. For CASIA v2.0, only the images in TIFF format are utilized to avoid double JPEG compression. Experiments are then carried out based on the following two types of settings:

- *Setting #1:* Splicing detection performance is tested on the compressed images while the CNN-based local descriptor and the SVM model are both trained on the uncompressed images.
- *Setting #2:* All the experimental settings are the same as Setting #1 except that the CNN-based local descriptor and the SVM classification model are both trained on the compressed dataset.

#### 1) COMPARISON WITH CNN BASED DETECTOR WITH 1-D POOLING FOR SETTING #1

In this circumstance, we show the effectiveness of the block pooling strategy in practical applications, where the CNN and SVM models are trained with uncompressed images and tested with JPEG compressed images. We compare the detection performance of our CNN based method (with block pooling) with the CNN based detector (with 1-D pooling) [18].

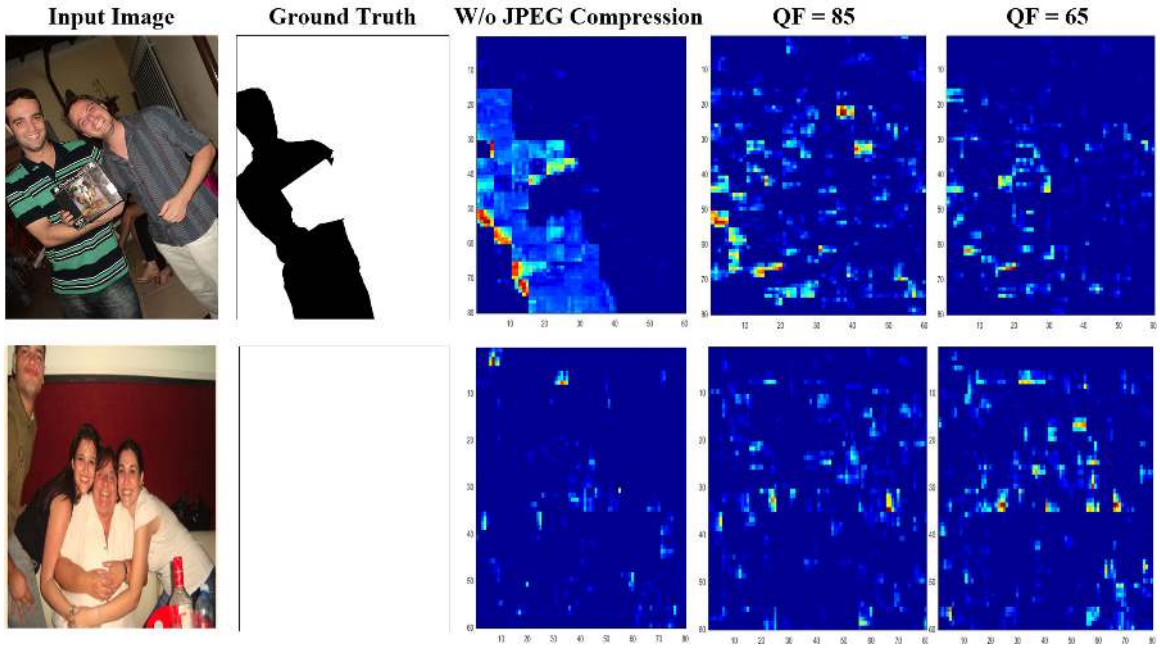



**FIGURE 7.** The heat maps of extracted feature image under different quality factors (QFs) of JPEG compression for the forged (the first row) and pristine (the second row) images, respectively. Note that we just illustrate the feature image with single channel for brevity.

**TABLE 3.** The performance comparisons of the two feature fusion strategies in terms of detection accuracy under JPEG attack for two experimental settings on different datasets. Note that QF=100 represents the original images in each dataset.

| Dataset | Pooling Method | | QF=100 (Acc(%)) | | | Setting #1 (Acc(%)) | | | | Setting #2 (Acc(%)) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Acc | TPR | TNR | QF=95 | QF=85 | QF=75 | QF=65 | QF=95 | QF=85 | QF=75 | QF=65 |
| CASIA v2.0 | Block pooling | max | 96.33 | 96.43 | **96.22** | **59.19** | **50.84** | **50.22** | **49.86** | **87.79** | **84.22** | **84.22** | **83.84** |
| | | mean | **96.97** | **96.76** | 97.19 | **58.70** | **51.62** | **51.97** | **50.78** | **87.95** | **84.22** | **84.22** | **84.22** |
| | 1-D pooling | max | 96.33 | 96.87 | 95.79 | 57.30 | 50.54 | 50.16 | 48.38 | 87.51 | 83.35 | 83.62 | 82.49 |
| | | mean | 96.70 | 96.22 | 97.19 | 54.97 | 50.65 | 49.97 | 49.51 | 86.97 | 82.60 | 83.35 | 82.49 |
| DSO-1 | Block pooling | max | **97.50** | 99.00 | **96.00** | **78.50** | **69.50** | **62.00** | **61.00** | **87.50** | **81.50** | **76.50** | **73.50** |
| | | mean | **96.50** | **99.00** | **94.00** | **73.00** | **64.50** | **58.50** | **56.50** | **88.00** | **78.50** | **72.00** | **70.50** |
| | 1-D pooling | max | 96.50 | 99.00 | 92.00 | 73.50 | 65.50 | 59.00 | 56.50 | 86.00 | 79.50 | 74.50 | 70.00 |
| | | mean | 95.50 | 98.00 | 93.00 | 65.00 | 53.00 | 51.50 | 51.50 | 85.50 | 75.50 | 67.50 | 65.50 |

Table 3 shows the performance comparison between the two methods in terms of detection accuracy. It is ready to see that although detection accuracy decreases consistently with stronger JPEG compression on both datasets, the proposed method outperforms the one in [18] by a clear margin on DSO-1. For CASIA v2.0 dataset, however, only slight performance gain is achieved with our method, although the performance improvement of 1.89%∼3.73% is observed for test images at QF=95. This is because the relationship of the spatial size between $f(x, y)$ and $g(x, y)$ (refer to Section III-E) would affect the performance of block pooling strategy. For DSO-1 dataset, the involved test images are typically of size 2,048×1,536, and the adopted block size (patch block) is of 128×128. Therefore, we have 16×12 blocks of $f(x, y)$ after feature extraction. For CNN-128, 5×5 block-pooling is adopted. Thus each $g(x, y)$ consists of multiple blocks of $f(x, y)$, and is only affected by these $f(x, y)$s. However, for CASIA v2.0 dataset, in which, 85% of images are smaller than 640×480 pixels and the block size is still of 128×128 pixels, each $f(x, y)$ relates to multiple $g(x, y)$ after 5×5 block pooling. As a result, each $f(x, y)$ would affect all the associated $g(x, y)$.

### 2) COMPARISON WITH CNN BASED DETECTOR WITH 1-D POOLING FOR SETTING #2

Based on this type of setting, both the training and testing stages of splicing detection are carried out on compressed images with the same QF. As shown in Table 3, by incorporating the block pooling strategy, the proposed CNN based method exhibits superiority over the one with 1-D pooling, especially on DSO-1 dataset, achieving the performance gains of 1.5%∼4.5% for different QFs. In addition, the

**TABLE 4.** The performance comparison of the proposed method with other state-of-the-arts traditional methods in terms of detection accuracy for two experimental settings on different datasets. Note that QF=100 represents the original images in each dataset.

| Dataset | Method | QF=100 (Acc(%)) | Setting #1 (Acc(%)) | | | | Setting #2 (Acc(%)) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | QF=95 | QF=85 | QF=75 | QF=65 | QF=95 | QF=85 | QF=75 | QF=65 |
| CASIA v2.0 | Proposed | **96.97** | **59.19** | **51.62** | **51.97** | 50.78 | **87.95** | **84.22** | **84.22** | **84.22** |
| | Muhammad [23] | 93.51 | 51.35 | 50.16 | 50.54 | **54.05** | 50.00 | 50.00 | 50.00 | 50.00 |
| | He [21] | 84.91 | 50.00 | 50.00 | 50.00 | 50.00 | 82.91 | 82.64 | 82.64 | 82.16 |
| DVMM | Proposed | **97.04** | **94.41** | **78.46** | **72.31** | **68.59** | **96.38** | **94.74** | **93.75** | **92.76** |
| | He [21] | 93.55 | 50.10 | 50.00 | 50.00 | 50.00 | 64.80 | 64.47 | 63.49 | 64.80 |
| DSO-1 | Proposed | **97.50** | **78.50** | **69.50** | **62.00** | **61.00** | **88.00** | **81.50** | **76.50** | **73.50** |
| | Carvalho [25] | 94.00 | 62.00 | 63.00 | 59.00 | 55.00 | 68.50 | 65.00 | 63.50 | 63.00 |

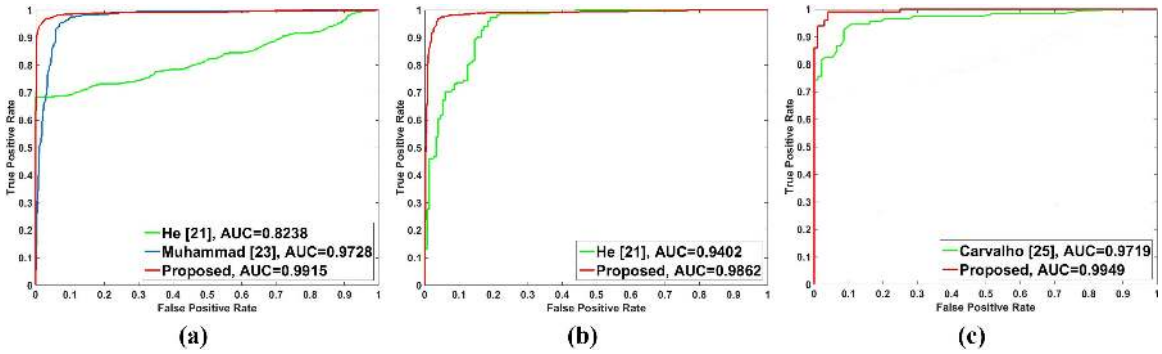



**FIGURE 8.** ROC curves and AUC comparisons on (a) CASIA v2.0, (b) DVMM and (c) DSO-1 datasets, respectively. Note that the performances are tested on the datasets which contains original uncompressed images.

detection performance with block pooling on CASIA v2.0 dataset is much better than the ones obtained in Setting #1, indicating the capability of the proposed method to detect splicing under JPEG compression. It is also observed that the block pooling strategy outperforms 1-D pooling by 0.28%∼1.35% and 0.87%∼1.73% when max and mean pooling operations are adopted on CASIA v2.0 dataset, respectively. The performance improvement in both Setting #1 and #2 shows that the block pooling strategy could alleviate the effects of JPEG compression to some extent, especially when the spatial resolution of $g(x, y)$ is much larger than $f(x, y)$.

### 3) COMPARISON WITH OTHER STATE-OF-THE-ART SPLICING DETECTION METHODS

We compare our CNN based method with several other state-of-the-art hand-crafted feature based image splicing detection methods, which includes He [21], Muhammad [23] and Carvalho [25], in the same two types of experimental settings. For a fair comparison, all the competing methods are evaluated using their default settings with the same five-fold cross validation protocol as our method. Table 4 shows the performance comparison results. It is observed that, except the performance obtained in Setting #1 on CASIA v2.0 with $QF = 65$, our CNN based method outperforms other involved methods at all tested QFs and on all the adopted datasets. To the best of our knowledge, the methods in [21], [23] and [25] are so far the state-of-the-arts hand-crafted feature based approaches for uncompressed images on CASIA v2.0, DVMM and DSO-1 datasets, respectively. The artifacts left by splicing operations tends to be more difficult to be captured by the LBP descriptor in [23] due to JPEG compression, even when it is trained on the compressed images for CASIA v2.0 dataset. Our CNN based method gains the advantage over the one in [23] by 3.46% for the original CASIA v2.0 dataset ($QF = 100$) and achieves the best detection performance in Setting #2. For DVMM dataset with grayscale images, our method outperforms convincingly the one in [21] in terms of both accuracy and robustness performance. Note that the method in [23] cannot be applied to grayscale images since it is designed to detect forgeries in YCbCr color space. In this respect, our method is more preferable for its extensibility to grayscale images. For DSO-1 dataset, our method also shows superiority over the one in [25] for the involved two experimental settings, achieving the performance gains of 3.5% and at least 10.5% for uncompressed images and Setting #2, respectively. Fig. 8 shows the ROC curves of our method and other competing methods for uncompressed images on the three datasets. It is observed that the ROC performance of our method also outperforms other methods for all the tested datasets.

### 4) COMPARISON WITH OTHER STATE-OF-THE-ART DEEP LEARNING BASED SPLICING DETECTION METHODS

We also compare our method with state-of-the-art deep learning based splicing detection methods, which includes, Huh [41] and Pomari [42] on DSO-1 dataset. As illustrated in Table 5, although a ResNet [13] with 50 layers is adopted

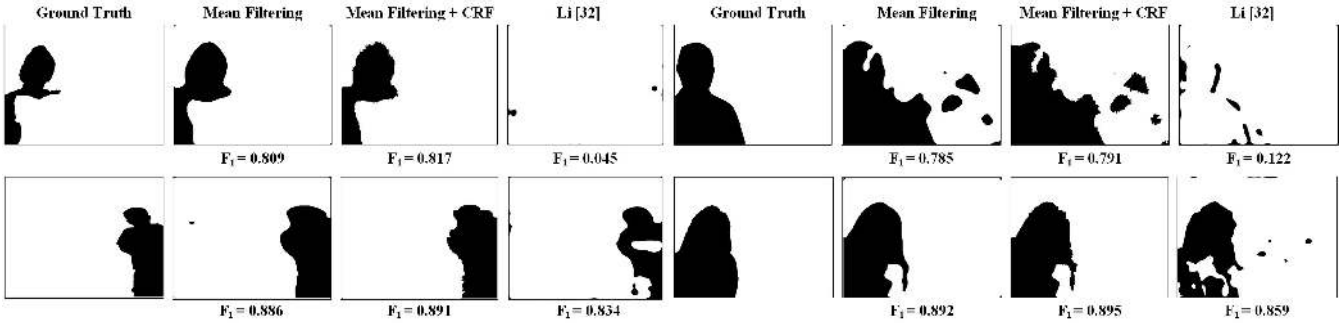



**FIGURE 9.** Splicing localization results for four forged images with the proposed method and Li's method. The black and white pixels in images correspond to the splicing and authentic pixels, respectively.

**TABLE 5.** The performance comparison of the proposed method with other state-of-the-arts deep learning based methods in terms of detection accuracy on DSO-1 dataset.

| Method | **Proposed** | Huh [41] | Pomari [42] |
|---|---|---|---|
| Accuracy (%) | **97.50** | 87.00 | 96.00 |

in [41] and [42], our method shows superiority over them in terms of detection accuracy with much fewer parameters.

### D. SPLICING LOCALIZATION PERFORMANCE

In this sub-section, we evaluate the splicing localization performance of the proposed CNN based method with fully connected CRF on DSO-1 dataset. We set the stride of the sliding window to be $s = 8$ and measure the localization performance with the $F_1$-score defined as follows:

$$F_1 = \frac{2TP}{2TP + FN + FP}, \tag{15}$$

where $TP$, $FN$, $FP$ are true positive, false negative and false positive, respectively. We then compare the performance of our proposed method with other state-of-the-art methods based on Setting #1, i.e., the involved models pre-trained on uncompressed images are directly used to localize the spliced forgeries in images with JPEG compression.

#### 1) COMPARISON BETWEEN CNN BASED METHODS

We first compare the performance of the proposed CNN model (C_ISRM_C-CNN) and the one in [18] (SRM-CNN) when incorporating with mean filtering (MF) and fully connected CRF as shown in Table 6. For uncompressed images (QF=100), the application of post-processing (MF+CRF) could increase the $F_1$-score by at least 2.5% for both the CNN based methods. However, the post-processing could not improve the localization performance for images with relatively strong JPEG compression (QF≤85 for C_ISRM_C-CNN and QF≤95 for SRM-CNN). This is because: (1) JPEG compression tends to prevent the CNN models from correctly localizing the tampered pixels (true positive), leading to the increase of the wrongly detected pristine pixels (false negative); (2) Mean filtering and CRF would also refine the edges of those wrongly predicted forged regions, which increases the value of $FN$. In addition, although similar localization performances are achieved by both networks for uncompressed images, the proposed CNN model consistently outperforms the one in [18] by 5.93%∼13.9% for JPEG compressed images owing to the improved design of CNN model as shown in Table 6.

**TABLE 6.** The localization performance comparison of the proposed method with the state-of-the-art methods in terms of $F_1$-score under JPEG attack on DSO-1 dataset. Note that MF stands for mean filtering.

| Method | Operation | QF=100 | QF=95 | QF=85 | QF=75 |
|---|---|---|---|---|---|
| Li [32] | MF | 0.7454 | 0.0857 | 0.0336 | 0.0146 |
| C_ISRM _C-CNN | – | 0.7969 | 0.4668 | **0.2306** | **0.1546** |
| | MF | 0.8120 | 0.4703 | 0.2223 | 0.1474 |
| | MF+CRF | **0.8276** | **0.4771** | 0.2196 | 0.1431 |
| SRM-CNN [32] | – | 0.7943 | 0.4195 | 0.0975 | 0.0384 |
| | MF | 0.8096 | 0.4151 | 0.0871 | 0.0301 |
| | MF+CRF | 0.8268 | 0.4178 | 0.0804 | 0.0252 |

#### 2) COMPARISON WITH OTHER STATE-OF-THE-ART SPLICING LOCALIZATION METHOD

We then compare the localization performance of our CNN based method with Li's method [32], which is the state-of-the-art hand-crafted feature based image forgery localization method. Note that only the statistical feature in Li's method is adopted for our CNN based method is specifically designed for detection of image splicing forgery. In the interest of fairness, both the proposed CNN and Li's statistical model are trained on the same training set and the same step size of 8 pixels is employed for the sliding window. As shown in Table 6, our CNN based methods significantly improve the localization performance in terms of $F_1$-score for all tested QFs, when compared with Li's method. Fig. 9 shows the localization results for two splicing images without JPEG compression, it is ready to see that the utilization of CRF significantly reduces the numbers of the misclassified pixels (false positives and false negatives), leading to more accurate predicted label maps with higher $F_1$-score than Li's method.

**TABLE 7. The localization performance comparison of the proposed method with other state-of-the-art deep learning based methods in terms of $F_1$-score on DSO-1 dataset. Note that, following the competing methods, we train our CNN on CASIA v2.0 dataset and conduct cross-dataset evaluation on DSO-1 dataset.**

| Method | **Proposed** | Salloum [43] | Huh [41] | Shi [40] |
|---|---|---|---|---|
| $F_1$-score | **0.5813** | 0.4790 | 0.5200 | 0.5800 |

### 3) COMPARISON WITH THE STATE-OF-THE-ART DEEP LEARNING BASED SPLICING LOCALIZATION METHOD

We finally compare the localization performance of our method with several state-of-the-art deep learning based methods in Shi [40], Huh [41] and Salloum [43]. In the interest of fairness, following [40] and [43], we train our method on CASIA v2.0 dataset while evaluate the localization performance on DSO-1 dataset. We use the results reported in [40], [41] and [43] for comparison. As illustrated in Table 7, the proposed method outperforms Huh [41] and Salloum [43] and achieves comparable performance to [40] in terms of $F_1$-score.

## VI. CONCLUSION

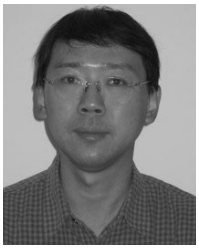
In this paper, a novel image splicing detection and localization scheme based on deep convolutional neural network (CNN) is proposed. To suppress the effects of image contents and extract more diverse and expressive residual features for RGB color images, the first layer of the CNN model is initialized with the optimized combination of the 30 basic high-pass filters used in spatial rich model (SRM) for image steganalysis. A constrained learning strategy is then applied in the first convolutional layer to retain the high-pass properties for the learned kernels. In addition to the cross-entropy loss, the contrastive loss function is also adopted to improve the generalization ability of the proposed CNN model, which consists of two symmetric sub-networks with shared parameters. In our method, the CNN model serves as a local feature descriptor, which is trained based on the labelled patches sampled from the training images. The pre-trained CNN based local descriptor is then used to extract block-wise features from the input test images, and a feature fusion strategy, known as block pooling, is incorporated to obtain the final discriminative features for image splicing detection with SVM classifier. Compared to the 1-D pooling in our previous work, the proposed block pooling strategy shows to improve the robustness performance in splicing detection of JPEG compressed images. The proposed CNN model is then further generalized to the task of image splicing localization by incorporating with the fully-connected CRF model. Extensive experiments are carried out on several public datasets, which demonstrates the superior performance of the proposed CNN based method over other state-of-the-art methods in image splicing detection and localization, especially for JPEG compressed images, which is more preferable for practical applications.

## REFERENCES

[1] M. Iuliani, G. Fabbri, and A. Piva, "Image splicing detection based on general perspective constraints," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Rome, Italy, Nov. 2015, pp. 1–6.
[2] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF approach for PRNU-based image forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 554–567, Apr. 2014.
[3] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.
[4] 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.
[5] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.
[6] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.
[7] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
[8] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 149–154.
[9] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5297–5301.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[11] J. Kuen, Z. Wang, and G. Wang, "Recurrent attentional networks for saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3668–3677.
[12] J. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1–9.
[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.
[14] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.
[15] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.
[16] B. Klein, L. Wolf, and Y. Afek, "A dynamic convolutional layer for short rangeweather prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4840–4848.
[17] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.
[18] 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. (WIFS)*, Abu Dhabi, UAE, Dec. 2016, pp. 1–6.
[19] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proc. 9th Workshop Multimedia Secur.*, Dallas, TX, USA, 2007, pp. 51–62.
[20] X. Zhao, S. Wang, S. Li, and J. Li, "Passive image-splicing detection by a 2-D noncausal Markov model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 185–199, Feb. 2015.
[21] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, Dec. 2012.
[22] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Rome, Italy, Nov. 2015, pp. 1–6.
[23] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Mach. Vis. Appl.*, vol. 25, no. 4, pp. 985–995, May 2014.
[24] J. Dong and W. Wang. (2011). *CASIA Tampered Image Detection Evaluation (TIDE) Database, v1.0 and v2.0*. [Online]. Available: http://forensics.idealtest.org/

[25] T. Carvalho, F. A. Faria, H. Pedrini, R. Da S. Torres, and A. Rocha, "Illuminant-based transformed spaces for image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 720–733, Apr. 2016.
[26] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1182–1194, Jul. 2013.
[27] Z. Ying, J. Goha, L. Wina, and V. Thinga, "Image region forgery detection: A deep learning approach," in *Proc. Singapore Cyber-Secur. Conf. (SG-CRC)*, vol. 14, 2016, pp. 1–11.
[28] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Abu Dhabi, UAE, Dec. 2016, pp. 1–6.
[29] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah, "Signature verification using a siamese time delay neural network," *Adv. Neural Inf. Process. Syst.*, 1993, pp. 737–744.
[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 448–456.
[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Jun. 2010, pp. 807–814.
[32] H. Li, W. Luo, X. Qiu, and J. Huang, "Image forgery localization via integrating tampering possibility maps," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1240–1252, May 2017.
[33] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec)*, New York, NY, USA, 2017, pp. 159–164.
[34] T.-T. Ng, J. Hsu, and S.-F. Chang. *Columbia Image Splicing Detection Evaluation Dataset*. Accessed: Sep. 19, 2019. [Online]. Available: http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm
[35] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1988–1996.
[36] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 109–117.
[37] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1520–1528.
[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[39] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
[40] Z. Shi, X. Shen, H. Kang, and Y. Lv, "Image manipulation detection and localization based on the dual-domain convolutional neural networks," *IEEE Access*, vol. 6, pp. 76437–76453, 2018.
[41] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proc. 15th Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 106–124, doi: 10.1007/978-3-030-01252-6_7.
[42] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3788–3792.
[43] R. Salloum, Y. Ren, and C.-C. Jay Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018, doi: 10.1016/j.jvcir.2018.01.010.
[44] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1053–1061.
[45] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.

**YUAN RAO** received the B.S. degree from the Beijing University of Posts and Telecommunications, in 2011, and the M.S. degree from the College of Information Science and Technology, Jinan University, in 2014. He is currently pursuing the Ph.D. degree with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. His current research interests include image forensics, image anti-forensics, and deep learning.



**JIANGQUN NI** received the Ph.D. degree in electronic engineering from The University of Hong Kong, in 1998. He then worked as a Postdoctoral Fellow for a joint program between the Sun Yat-sen University, China, and the Guangdong Institute of Telecommunication Research from 1998 to 2000. Since 2001, he has been with the School of Data and Computer Science, Sun Yat-sen University, where he is currently a Professor. He is also currently with Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests include data hiding, digital forensics, and image/video processing. He has published more than 50 articles in these areas.

**HUIMIN ZHAO** received the B.Sc. and M.Sc. degrees in signal processing from Northwestern Polytechnical University, Xian, China, in 1992 and 1997, respectively, and the Ph.D. degree in electrical engineering from Sun Yat-sen University, in 2001. He is currently a Professor with Guangdong Polytechnic Normal University. His research interest includes image, video, and information security technology.