Hindawi
Security and Communication Networks
Volume 2022, Article ID 1090307, 13 pages
https://doi.org/10.1155/2022/1090307

Research Article

Image Forgery Detection Using Tamper-Guided Dual Self-Attention Network with Multiresolution Hybrid Feature

Fengyong Li[1,2] **Zhenjia Pei**[1] **Weimin Wei**[1] **Jing Li**[1] **and Chuan Qin**[3]

[1] *College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 201306, China*
[2] *Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China*
[3] *School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*

Correspondence should be addressed to Fengyong Li; fyli@shiep.edu.cn

Received 8 July 2022; Revised 12 August 2022; Accepted 23 September 2022; Published 10 October 2022

Academic Editor: Feng Ding

Image forgery detection can efficiently capture the difference between the tampered area and the nontampered area. However, existing work usually overemphasizes pixel-level localization, ignoring image-level detection. As a result, false detection for tampered image maybe cause a large number of false positives. To address this problem, we propose an end-to-end fully convolutional neural network. In this framework, multiresolution hybrid features from RGB stream and noise stream are firstly fused to learn visual artifacts and compression inconsistency artifacts, which can efficiently identify the tampered images. Furthermore, a tamper-guided dual self-attention (TDSA) module is designed, which can focus the network's attention on the tampered areas and segment them from the image by capturing the difference between the tampered area and the nontampered area. Extensive experiments demonstrate that compared to existing schemes, our scheme can simultaneously effectively achieve pixel-level forgery localization and image-level forgery detection while maintaining higher detection accuracy and stronger robustness.

1. Introduction

Image forgery refers to pasting a region (an object or multiple objects) in a real image to a certain position in another real image. In the tampered area, postprocessing operations, such as blurring, smoothing, retouching, and fusion, are used to cover up the tampering traces. Accordingly, the tampered image looks more realistic and natural so that the purpose of tampering with image content can be achieved successfully. Illegal persons may tamper with the content to make the images/videos convey incorrect or misleading information [1]. This is unacceptable in some application scenarios that are extremely sensitive to image content, such as military communications, political news photos, legal forensics, and electronic bill. For this reason, image forgery detection technology is needed to discriminate the authenticity of the image. In general, image forgery detection technology includes image-level forgery detection and pixel-level forgery region localization, which usually contain two main issues. First, for image-level detection, authenticity discrimination should be a prerequisite for the practical application of image tampering detection technology. However, according to the observation and testing of mainstream methods, we find an interesting phenomenon that most of existing methods usually ignore the capability to distinguish between true and false. As shown in Figures 1(c) and 1(d), with an authentic image (untampered image) as input, the model still outputs a fake area. This flaw makes these models nearly unusable. On the other hand, for pixel-level localization, pixel-level manipulation region localization is the mainstream solution in this field, and the work in this area looks more like a simplified semantic segmentation problem. A series of methods are also adjusted on mature semantic segmentation models. However, the localization of pixel-level manipulation regions by simply using the semantic segmentation [2, 3] model is not ideal. As shown in

Figure 1, the pixel-level localization results of the current model are also inaccurate.

Regarding the first question, almost all image forgery detection methods use datasets that only include tampered images in the training phase and only include tampered images in the testing phase for evaluation, as shown in Table 1. Considering the practical value of image forgery detection, only being able to locate the forged image is unqualified. In order to solve this problem, authentic images (untampered images) should be also added in the dataset during the training phase. However, with the addition of authentic images, it will inevitably have a negative impact on the localization performance and can make it difficult for the model to converge. For the second question, some researchers tried to utilize the semantic segmentation models to enhance the localization accuracy of pixel-level forgery, which mainly involve the semantic features [12, 13]. Nevertheless, unlike semantic segmentation, image forgery detection focuses more on tampering artifacts [8] rather than image content. Correspondingly, apart from semantic features, inconsistent features [4–6] should also be considered for image forgery detection. To this end, we try to add the noise stream and design a noise inconsistency multiresolution feature extractor based on constrained convolution to discover noise inconsistency between authentic and tampered regions. The union of semantic features and noise inconsistency features is called hybrid features, such multiresolution hybrid feature can provide more evidence for image forgery detection.

This paper designs an end-to-end fully convolutional neural network to detect and localize tampered regions. The network includes an RGB stream, noise stream, and a multiresolution hybrid feature fusion process. In this framework, the RGB stream is used to learn visual artifacts, while the noise stream is used to learn noise inconsistency artifacts. During the fusion stage, multiple resolution features from the two streams are utilized to generate the final mask. Furthermore, a tamper-guided dual self-attention (TDSA) module is designed to enhance the feature representation and reduce false positives from natural regions. Moreover, to reduce false positives for authentic images (Figure 1), authentic images are considered during the training phase to address the drift problem that easily occurs in other attention mechanisms. Our proposed scheme can simultaneously achieve pixel-level forgery localization and image-level forgery detection, which greatly improves the practical applicability of the network framework.

Our main contributions are summarized as follows:

(i) We propose a novel end-to-end deep network model, which builds a hybrid feature model by combining semantic features and multi-resolution noise-inconsistent features, and meanwhile designs a tamper-guided dual self-attention (TDSA) module to enhance the fine-grained discriminative capability from the multiresolution hybrid feature.

(ii) Our proposed scheme can simultaneously effectively achieve pixel-level forgery localization and image-level forgery detection. With the blessing of authentic images, the capability of the proposed model to distinguish between true and false is significantly improved, even do not generate any false positives.

(iii) Extensive experiments implemented over four public datasets demonstrate that our method outperforms state-of-the-art methods in terms of detection accuracy and robustness, and shows an overwhelming advantage on the image-level forgery detection experiments.

The rest of this paper is organized as follows: Section 2 introduces the related work. The detailed procedure of our proposed scheme is shown in Section 3. We perform comprehensive experiments to evaluate the performance of the proposed scheme and present the results and corresponding discussions in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Image Forgery Detection. Image forgery detection aims to distinguish between tampered images and real images. If the image is a tampered image, the tampered region should be accurately located. Table 1 summarizes some recent forgery detection works. Most of the early work [4–6, 14] were based on unsupervised algorithms and did not involve neural networks. For instance, Krawetz et al. [4] used error level analysis (ELA) to find compression differences for image forgery detection. Mahdian et al. [5] exploited image noise inconsistency to detect image forgery. Ferrara et al. [6] utilized the CFA of nearby pixels to predict a pixel. These unsupervised methods performed forgery detection by a statistical manner, which has the advantage of supporting image-level detection, but the disadvantage is that its pixel-level localization can only be accurate to the block and can only support specific forgery types, e.g., splicing, copy-move, and removal.

With the widespread use of neural networks, DNN-based image forgery detection [3, 7–11, 15, 16] schemes have become popular. The performance of pixel-level forgery localization is greatly improved, and it can also handle many types of tampered images. For example, J-LSTM [7] employed a hybrid CNN-LSTM model to capture discriminative features between manipulated and nonmanipulated regions. J-LSTM adopted the Patch-LSTM network as the backbone network, but the size of the patch will limit the localization of the tampered area. RGB-N [8] used a Bilinear pooling method to fuse forensic cues from RGB features and noise-inconsistent features. However, due to the use of the steganalysis [17, 18] rich model and Faster R-CNN, it only provides bounding boxes, which are object-level localization rather than pixel-level localization. Mantra-Net [9] treated tamper detection as anomaly detection. It detects manipulated pixels by identifying local anomalous features. FCN [3] was a network for semantic segmentation tasks and can support inputs of any size. With the network training, the image-level detection performance of this network can be greatly increased. This also proves that the

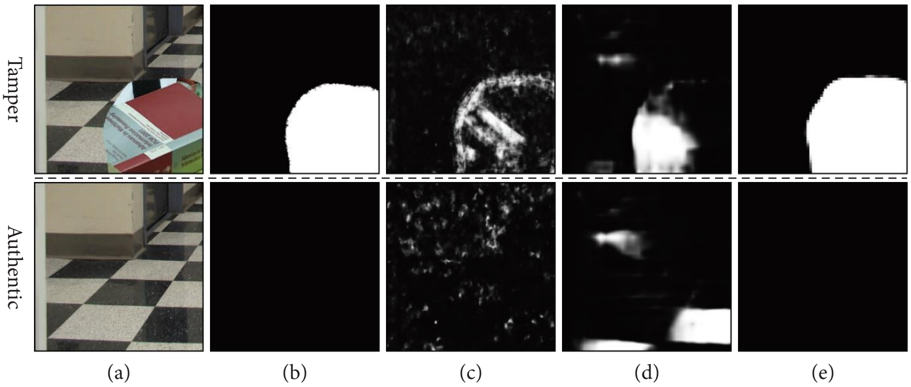

Figure 1: Detection results before and after tampering for the same image. In this figure, the images from left to right belong to tampered/authentic image, ground-truth, detection results of Mantra-Net, detection results of FCN, and detection results of our scheme.

Table 1: Summary of existing image forgery detection methods.

Method	Backbone	Forensic clue	Fusion method	Training data	Localization
ELA [4]	—	Error level analysis	—	Authentic, tamper	Block-level
NOI [5]	—	Noise-inconsistency	—	Authentic, tamper	Block-level
CFA [6]	—	Local CFA inconsistency	—	Authentic, tamper	Block-level
J-LSTM [7]	Patch-LSTM	RGB	—	Tamper	Pixel-level
RGB-N [8]	Faster R-CNN	RGB, noise-inconsistency	Bilinear pooling	Tamper	Object-level
ManTra-net [9]	Wider VGG	RGB, noise-inconsistency	Feature concatenation	Tamper	Pixel-level
FCN [3]	-	RGB	—	Tamper	Pixel-level
CR-CNN [10]	Mask R-CNN	Noise-inconsistency	—	Tamper	Pixel-level
GSR-net [11]	Deeplabv2	RGB	—	Tamper	Pixel-level
Ours	HRNet-48	RGB, noise-inconsistency	Multi-resolution concatenation, tamper-guided dual self-attention	Authentic, tamper	Image-level Pixel-level

semantic feature based training strategy is effective. CR-CNN [10] used constrained convolution [19] to extract noisy inconsistency features and accurately performed image forensics with a coarse-to-fine architecture. We believe that this model using only noise-inconsistent features is insufficient. GSR-Net [11] localized manipulation regions by learning to spot boundary artifacts. In general, combining with Table 1, it can observe that the above-given DNN-based methods [3, 7–11] only used tampered images in the training phase, they can only perform pixel-level image localization but cannot perform image-level forgery detection.

2.2. Attention Mechanism. Attention mechanism [20–22], as an efficient low-cost way to enhance features, has been widely used in various visual tasks [23–26]. The early attention mechanism in the visual field is the squeeze-and-excitation module (SE) proposed by SENet [20]. It can simultaneously extract the spatial and channel information of the feature map. On the basis of the SE module, CBAM (convolutional block attention module) [21] extracted the channel and spatial information by using the tandem structure. DA (Dual Attention) module [22] modelled the semantic interdependencies in spatial and channel dimensions, respectively. The position attention module selectively aggregates the feature at each position by a weighted sum of the features at all positions, while the channel attention module selectively emphasizes interdependent channel maps by integrating associated features among all channel maps.

However, compared to other vision tasks, image forgery detection features are more difficult to identify because man-made tampered areas may appear anywhere in the image so that spatial attention is hard to be captured. We try to design a tamper-guided dual self-attention (TDSA) mechanism to improve the feature expression ability. TDSA mechanism enables the network to decide the right to use the attention mechanism through learnable parameters and therefore makes the use of the TDSA module more flexible. Also, the traditional attention mechanism inevitably has a drift problem [27]. For image tampering detection, the network attention is focused on the nontampering area, which greatly affects the localization results of the network. To alleviate the attention drift problem, we use tampered regions for supervision and make the attention mechanism focus on the forged regions.

3. Proposed Method

The proposed network model is an end-to-end tamper detection and localization framework, that is also to say, inputting a tampered image, the proposed network model can locate the tampered area, and meanwhile output the

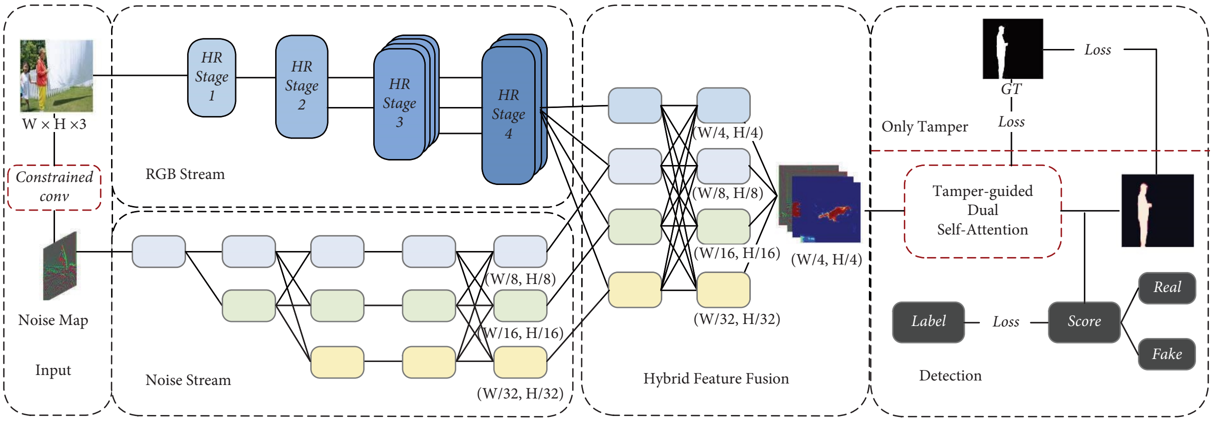

Figure 2: Overall framework of proposed method.

confidence level of image tampering. Figure 2 shows that the network structure, which consists of an RGB stream, a noise stream, a fusion stage, and a detection stage. In this model, the RGB stream focuses on visual cues, while the noise stream mainly detects noise inconsistency artifacts. The RGB stream uses HRNet-48 [28] as the backbone network, which maintains high-resolution outputs throughout the feature extraction process. To match the multi-resolution output of the RGB stream, we follow the style of the noise stream of HRNet-48 [28] and design a multiresolution noise feature extractor. Furthermore, we introduce a novel fusion method that does not lose the details of features at any resolution to fuse the features of the two stages as the hybrid features, which can be further enhanced by the TDSA module. The network model finally outputs the predicted mask and predicted score.

3.1. RGB Stream. Most of the previous work on extracting visual artifacts from RGB streams uses traditional Resnet [29], whose disadvantage is that the image will be deeper with the number increasing of layers of the Resnet, and the resolution of the output image will be lower. This shortcoming does not meet the requirements of image forensics, because a large number of upsampling operations are performed in the network output stage to restore the original resolution of the image, which may cause the generated prediction to mask not clear enough. We use HRNet-48 [28] as the backbone network for RGB streams. As a high-resolution network, HRNet-48 can capture detailed features well and avoid the loss of details due to continuous upsampling like other backbone. As shown in Figure 2, the backbone network consists of four stages. The output resolution of each stage is gradually reduced to half and the number of channels is doubled, the high-resolution output of the previous stage is yet preserved. The output of the HRStage module can be shown in the following formula:

$$\{f_1, f_2, \cdots, f_i\} \leftarrow \text{HR Stage}[i](x). \tag{1}$$

3.2. Noise Stream. RGB streams are not sufficient for all types of tampered images, especially for the case that the tampered images are processed to hide splicing boundaries and reduce contrast differences. However, many tampered images are manipulated so that the splice boundary are not recognized, making the tampered area look more natural and thus fooling the algorithm. The noise stream is not affected by the above-given problem. Accordingly, we add a noise stream to solve this problem. The SRM filter [30] are widely used to capture noise-inconsistent cues. Nevertheless, these noise-inconsistent cues cannot be learned by the network, and they thus are more vulnerable to robust attacks. In contrast, we use constrained conv [19] to directly adaptively learn noise-inconsistent features, obtaining better generality and robustness. Specifically, the constraint is applied as follows:

$$\begin{cases} w_k(0,0) = -1, \\ \sum_{m,n\neq 0} w_k(m,n) = 1, \end{cases} \tag{2}$$

where w_k denotes the k_{th} convolution kernel, and $(0,0)$ is the center coordinate of w_k. w_k is updated with the entire model, and then the above constraint process is performed. Through formula (2), the noise map is fed into our designed noise stream. As shown in Figure 2, the structure of the noise stream extracts noise-inconsistent features of the forgery image by connecting high-resolution to low-resolution convolutions in parallel. On this basis, information exchange is carried out between the individual resolution features. The output feature map of the noise stream includes three resolutions $(1/8, 1/16, 1/32)$. These noise features will be fused with the features extracted from the RGB stream, and then are input to the hybrid feature fusion stage.

3.3. Multiresolution Hybrid Feature Fusion. We combine the RGB stream with the noise stream for manipulation detection in the hybrid feature fusion stage. Most methods [9, 10] only perform simply feature combining, e.g., bilinear pooling [8, 31]. However, bilinear pooling is not learnable, and the pooling operation is easy to lose details, which is not conducive to manipulation localization. Therefore, the current manipulation detection tasks [11, 32] mostly adopt a no-pooling method.

To solve the above-given problem, we design a multiresolution hybrid feature fusion module. We firstly utilize a channel to concatenate the features of the same resolution from the two streams. The features of different resolutions can be fully information exchanged through a hybrid feature fusion unit. As shown in Figure 3, the four cells on the left are fused to generate the rightmost cell. Specifically, for low-resolution features, they are firstly processed by an upsampling operation (bilinear interpolation) to achieve the same size as the high-resolution feature. Then, the processed features pass through a 1×1 convolution and perform elementwise addition with the high-resolution features. While for high-resolution features, they are downsampled to the size of the low-resolution features through a 3×3 convolution with stride 2 (one or more). Correspondingly, the high-resolution features and low-resolution features are fused using elementwise addition. After performing hybrid feature fusion, the network can output multiresolution hybrid features. Finally, the low-resolution features are upsampled and added into the highest-resolution features.

3.4. Tamper-Guided Dual Self-Attention Module. In the previous stage, multi-resolution features from the RGB stream and Noise stream are sequentially fused. However, multiresolution feature fusion inevitably introduces some redundant information, that is to say, some redundant features that may affect the localization accuracy are also fused, which should be further removed to preserve clues related to forged regions. Considering that the attention mechanism [20–22] can effectively implement feature filtering and select important features from the input information, we design a tamper-guided dual self-attention (TDSA) module to enhance the feature representation about forgery cues. Our TDSA module is built on DA attention [22] and consists of channel attention and tamper-guided position attention. TDSA module uses a self-attention mechanism to capture interchannel or interposition dependencies, where channel attention uses all channel attention map weights to update each channel attention map and tamper-guided position attention aggregates all position features to update the attention map, in which the weights are determined by the feature similarity between two positions. In the TDSA module, any two positions may be associated, and the spatial distance is no longer limited by the receptive field. Moreover, the tamper areas may be in nonsalient areas of the image, and the attention network may cause an attention drift problem [27] to make it focus on nontampered areas. To address the attention drift problem, we add tamper regions supervision in the position attention map. In this way, the position attention can be forced to locate to the tampered region, which can enhance the expressive capability of the feature, and further improve the overall accuracy of the network.

3.4.1. Channel Attention Module. The pipeline of the channel attention module is shown in Figure 4(left side). We calculate the channel attention map $\mathbf{M^C} \in \mathbb{R}^{C\times C}$ from the hybrid features $\mathbf{F} \in \mathbb{R}^{C\times W\times H}$, where C, W, H represent channel, width, and height dimensions, respectively. First, the input $\mathbf{F} \in \mathbb{R}^{C\times W\times H}$ is reshaped to $\mathbf{F}^1 \in \mathbb{R}^{C\times (W\times H)}$. Then, the reshaped feature $\mathbf{F}^1$ is transposed to get $\mathbf{F}^2 \in \mathbb{R}^{(W\times H)\times C}$ and a matrix multiplication is performed between $\mathbf{F}^1$ and $\mathbf{F}^2$. Finally, a softmax layer is applied to obtain the channel attention map $\mathbf{M^C} \in \mathbb{R}^{C\times C}$.

$$m^c_{j,i} = \frac{\exp\left(F^1_i \cdot F^2_j\right)}{\sum_{i=1}^{C} \exp\left(F^1_i \cdot F^2_j\right)}, \tag{3}$$

where $m^c_{j,i}$ stands for the impact of the i^{th} channel on the j^{th}. Furthermore,we perform a matrix multiplication between $\mathbf{F}^1$ and $\mathbf{M^C}$ and result to $F_{CAM,j} \in \mathbb{R}^{C\times W\times H}$.

$$F_{\mathrm{CAM},j} = \lambda_c \sum_{i=1}^{C} m^c_{j,i} F^1_j + F_j, \tag{4}$$

where λ_c controls the importance of the channel attention map over the input feature map $\mathbf{F}$. λ_c gradually learns a weight from 0. This formula takes the weighted channel features into the original hybrid features. The channel attention module selectively emphasizes interconnected channel maps by integrating relevant features in all channel maps.

3.4.2. Tamper-Guided Position Attention Module. The pipeline of the tamper-guided position attention module is shown on the right side of Figure 4. We calculate the position attention map $\mathbf{M^P} \in \mathbb{R}^{(W\times H)\times (W\times H)}$ from the hybrid features $\mathbf{F} \in \mathbb{R}^{C\times W\times H}$. Specifically, $\mathbf{F}$ firstly passes through three parallel 1×1 convolution blocks, resulting in $\mathbf{F}^1 \in \mathbb{R}^{C'\times W\times H}$, $\mathbf{F}^2 \in \mathbb{R}^{C'\times W\times H}$, $\mathbf{F}^3 \in \mathbb{R}^{C\times W\times H}$, respectively, where C' is equal to $C/8$. Then, $\mathbf{F}^1$, $\mathbf{F}^2$ is reshaped to $\mathbf{F}^1 \in \mathbb{R}^{C'\times (W\times H)}$, $\mathbf{F}^2 \in \mathbb{R}^{C'\times (W\times H)}$, respectively. Then, the reshaped feature $\mathbf{F}^1$ is transposed to get $\mathbf{F}^1 \in \mathbb{R}^{(W\times H)\times C'}$ and a matrix multiplication is performed between $\mathbf{F}^1$ and $\mathbf{F}^2$. We finally apply a softmax layer to obtain the Position attention map $\mathbf{M^P} \in \mathbb{R}^{(W\times H)\times (W\times H)}$.

$$m^p_{j,i} = \frac{\exp\left(F^1_i \cdot F^2_j\right)}{\sum_{i=1}^{W\times H} \exp\left(F^1_i \cdot F^2_j\right)}, \tag{5}$$

where $m^p_{j,i}$ evaluates the impact of the i^{th} position on the j^{th} position. Meanwhile, $\mathbf{F}^3$ is reshaped to $\mathbf{F}^3 \in \mathbb{R}^{C\times (W\times H)}$ and performed a matrix multiplication between $\mathbf{F}^3$ and the transpose of $\mathbf{M^P}$ and result to $F_{PAM,j} \in \mathbb{R}^{C\times W\times H}$.

$$F_{\mathrm{PAM},j} = \lambda_p \sum_{i=1}^{W\times H} m^p_{j,i} F^3_i + F_j, \tag{6}$$

where λ_p is initialized as 0 and gradually learns to assign more weight. This formula takes the weighted position features into the original hybrid features. Therefore, similar features are related to each other regardless of distance.

Notably, since the features of tamper forensics are more difficult to identify, our proposed supervision mechanism in the attention map can efficiently alleviate the problem of

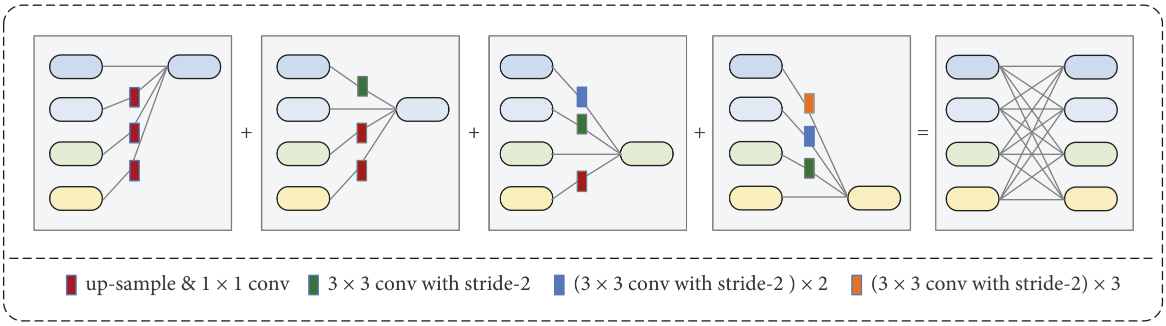
+
+
+
=
🟥 up-sample & 1 × 1 conv
🟩 3 × 3 conv with stride-2
🟦 (3 × 3 conv with stride-2) × 2
🟧 (3 × 3 conv with stride-2) × 3

Figure 3: Multiresolution hybrid feature fusion.

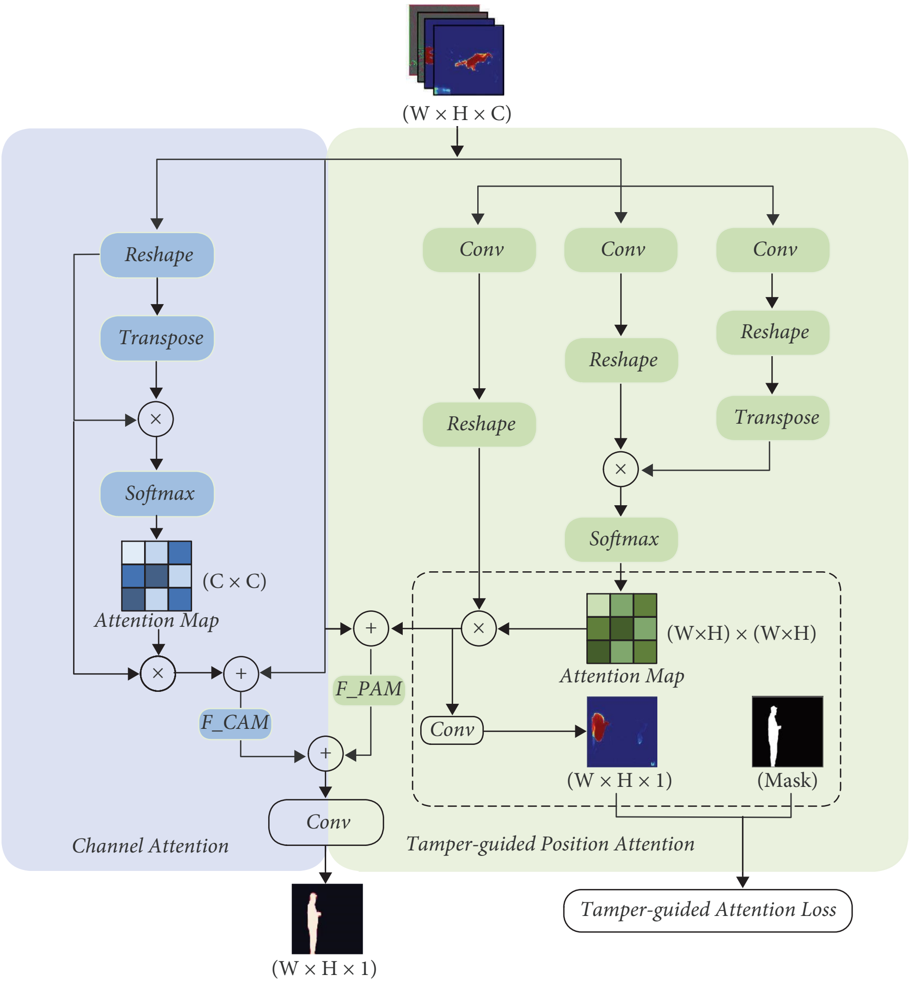

Figure 4: The structure of TDSA. ⊗ represents the matrix multiplication and ⊕ is the elementwise addition.

attention drift and make the attention mechanism focus on the forged regions. This allows our attention mechanism to force localization to the forged regions, making the attention map closer to ground-truth. To be specific, F_{PAM} gets the attention map through a convolutional layer, and calculates tamper-guided attention loss with the ground truth. This effectively removes the negative properties of the attention mechanism and focuses attention on the forged regions.

3.5. Loss Function. Our network model involves three loss functions, including pixel-level loss, tamper-guided attention loss, and image-level loss, where pixel-level loss is used to improve pixel-level manipulation localization, tamper-guided attention loss for reducing attention drift, and image-level loss for improving image-level forgery detection. Since the whole network is similar to a binary classification problem, we use BCE loss [33] as our loss function.

3.5.1. Pixel-Level Loss. Let M be the ground-truth mask, $\widetilde{M}$ be the predicted mask. Since the forged regions in the forgery images are generally small and imperceptible, we increase the weights for unbalanced data during training. The pixel-level loss can be computed as follows:

$$\mathscr{L}_{\text{pixel}} = -\sum_{i,j} M_{i,j} \cdot \log\left(\widetilde{M}_{i,j}\right) - \sum_{i,j}\left(1 - M_{i,j}\right) \cdot \log\left(1 - \widetilde{M}_{i,j}\right). \quad (7)$$

3.5.2. Tamper-Guided Attention Loss. A tamper mask is used to guide the training of attention. We reduce the dimension of the feature map enhanced by position attention and let it be A. M is the ground-truth mask. Accordingly, the output of the attention can be normalized as follows:

$$A^{\text{norm}} = \frac{A - \min(A)}{\max(A) - \min(A)}, \quad (8)$$

where A represents the attention map and A^{norm} is the normalized attention map. Then, the tamper-guided attention loss can be calculated:

$$\begin{aligned}\mathscr{L}_{\text{atte}} = &-\sum_{i,j} M_{i,j} \cdot \log\left(A^{\text{norm}}_{i,j}\right)\\ &-\sum_{i,j}\left(1 - M_{i,j}\right) \cdot \log\left(1 - A^{\text{norm}}_{i,j}\right).\end{aligned} \quad (9)$$

Note that, to accelerate network convergence, the above two functions only compute the loss on forgery images.

3.5.3. Image-Level Loss. Previous works only use fake images for the training phase, ignoring image-level loss. In our network framework, we incorporate real images in the training phase. Nevertheless, we found that applying pixel-level loss directly on real images may cause a lack of localization details, since there are no tampered areas in the real images, and all pixels in the real image are true. In other words, the real image contains only one class and such data are bad for a binary classification problem. It is difficult for the entire network to converge quickly. Therefore, we design a label-based image-level loss function. Since image-level detection is based on pixel-level prediction mask $\widetilde{M}$, image-level confidence scores can be obtained by using a function of global max pooling on $\widetilde{M}$.

$$G(\widetilde{M}) = \text{Global Max Pooling}(\widetilde{M}), \quad (10)$$

where $G(\widetilde{M})$ represents the image-level confidence scores, and $G(\widetilde{M}) \in (0,1)$. Specifically, the higher the $G(\widetilde{M})$ value, the less realistic the image. Furthermore, we combine this score with the image-level label to calculate the loss.

$$\begin{aligned}\mathscr{L}_{\text{image}} = &-(\text{label}) \cdot \log(G(\widetilde{M}))\\ &-(1 - \text{label}) \cdot \log(1 - G(\widetilde{M})),\end{aligned} \quad (11)$$

where label = $\{0, 1\}$, the label of the authentic image is 0, and the label of the tampered image is 1. Notably, this loss can work well for both tamper and authentic images.

Finally, the total loss for the entire network can be presented as follows:

$$\begin{aligned}\mathscr{L}_{\text{total}} = &(\text{lable}) \cdot \left(\alpha \cdot \mathscr{L}_{\text{pixel}} + \beta \cdot \mathscr{L}_{\text{atte}}\right)\\ &+ (1 - \alpha - \beta) \cdot \mathscr{L}_{\text{image}},\end{aligned} \quad (12)$$

where label = $\{0, 1\}$,and $\alpha, \beta \in (0, 1)$ are weight parameters.

4. Experimental Results

4.1. Experimental Datasets. Table 2 lists the image datasets used in the experiments. In our experiments, we add a large number of authentic images for training and evaluation. Note that, since NIST16 and IMD2020 datasets do not provide authentic images, only image-level evaluations are performed on CASIA and COLUMBIA.

(i) CASIA [34, 35]. CASIA contains CASIA v1.0 and CASIA v2.0. CASIA v1.0 has 920 tampered images and 800 authentic images, which are mainly manipulated by splicing and copy-move. CASIA v2.0 contains more images, a total of about 13000 images. Overall, CASIA v2.0 is mainly used for network training, while CASIA v1.0 is mainly used for testing.

(ii) COLUMBIA [36]. Columbia has a total of 363 images, including 180 tampered images and 183 authentic images. The tampered images are created using only the splicing operation, e.g., copying and pasting visually salient objects in Adobe PhotoShop to the authentic ones.

(iii) NIST16 [37]. NIST16 is an authoritative high-resolution dataset, which includes 564 images that mainly cover three manipulation types: copy-move, splicing, and removal. It is worth noting that there is no corresponding authentic image provided in NIST16, and only the tampered image and the ground mask are provided. Our evaluations on this dataset only involve metrics for tampered images.

(iv) IMD2020 [38]. IMD2020 is a novel dataset consisting of real-life manipulated images as well as manually created ground-truth masks. IMD2020 includes a total of 2151 images. Since the image material is collected on the Internet, it is more convincing than the image generated by the virtual environment. Note that, this dataset does not provide any real images corresponding to the tampered images.

4.2. Implementation Details. We initialize the weights of the network by pretraining on ImageNet classification [39] for the network. Our network is a fully convolutional network and thus supports any resolution image input. The network is implemented with PyTorch and trained on an NVIDIA Tesla P100 GPU, using stochastic gradient descent with a momentum of 0.9 for the optimizer. The learning rate started from 0.005 and decayed exponentially and the batch size is

Table 2: The details of training set and testing set (number of images) for four datasets.

Datasets	Training set		Testing set		Total
	Authentic	Tamper	Authentic	Tamper	Authentic + tamper
CASIA [34, 35]	7791	5405	800	920	14916
COLUMBIA [36]	133	130	50	50	363
NIST16 [37]	0	414	0	150	564
IMD2020 [38]	0	2010	0	141	2151

13. Moreover, to address the class imbalance, the weight of pixel-level loss is increased sixfold on tampered class.

4.3. Evaluation Metrics. For pixel-level forgery detection, following previous work [8], we compute pixel-level precision and recall and then report their F_1. AUC (Area Under Curve) is defined as the area under the ROC curve enclosed by the coordinate axes, as a decision threshold free metric, is also reported. For image-level forgery detection, in order to measure the miss detection rate and false alarm rate, we also report TNR (specificity), their F_1 and AUC, respectively.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\%. \tag{13}$$

Notably, since there are no authentic images in the evaluation of other methods, for fair comparison with other methods, the authentic images in each testing set are only used for image-level detection. Meanwhile, to measure the equalization performance of the model, we report combine-F_1 in image-level detection, which is an average of pixel-level F_1 and image-level F_1.

$$\text{TNR} = \frac{\text{TN}}{\text{FP} + \text{TN}} \times 100\%. \tag{14}$$

4.4. Comparison with State-of-the-Arts. We compare our proposed method with several state-of-the-art baseline methods, which belong to two different categories: classical unsupervised methods, e.g., ELA [4], NOI [5], CFA [6], and fine-tuned models, e.g., FCN [3], J-LSTM [7], RGB-N [8], ManTra-Net [9], CR-CNN [10], GSR-Net [11]. We compare pixel-level localization capabilities with these methods and provide prediction masks with Mantra-Net [9] and FCN [3]. In this experiment, we retrain the FCN by adding real images and using our training strategy. Moreover, we also compare image-level detection capabilities with three existing schemes [3, 10, 11] and provide their prediction masks. Note that, image-level detection can only be performed on two datasets (Columbia, CASIA) that provide authentic images.

4.4.1. Pixel-Level Localization. We perform pixel-level image manipulation localization in four standard datasets. Since IMD2020 dataset is just created recently, it has not been tested with many methods. The performance of different models is shown in Table 3. Our model leads the F_1 scores on Columbia, NIST16, and IMD20 and obtains the second on CASIA. In addition, GSR-Net [11] has a slight advantage over CASIA. This may be because GSR-Net [11] performs boundary refinement. Nevertheless, our scheme obtains a significant advantage comparing with GSR-Net over the other three datasets. Specifically, our F_1 score on NIST16 is more than double it. We can explain this phenomenon that the NIST16 dataset has a more natural transition between the real area and the fake area, leading to little difference in contrast, while GSR-Net is a specially designed network for boundary artifacts, so it does not work well on this dataset.

In terms of AUC scores, our model achieves the best performance over four datasets. To be specific, for unsupervised methods, our improvement is between 24% and 56% and with an average of 40% improvement. For the fine-tuned methods, the average improvements are 10.8% in NIST16, 9.9% in COLUMBIA, and 6.6% in CASIA, respectively. IMD2020 contains incomplete data, but the improvement can still get 22.9% through rough calculation. This illustrates that our method is applicable to various types of forgery. Although adding real images may be not conducive to pixel-level localization, our training strategy makes up for this deficiency, enabling pixel-level localization to achieve a leading performance.

In order to more intuitively express the localization capability of our model, we provide prediction masks on four datasets, which are shown in Figure 5. It can be clearly seen that our method can generate accurate predicted masks, which are very close to the ground truth. Also, it can be seen that Mantra-Net [9] can only superficially see the outline of the forged area, but, accompanied by a large number of false positives. Moreover, in contrast to FCN [3], a relatively accurate mask can be obtained. However, we can find an interesting phenomenon that FCN has false positives in real (nontamper) regions, while our prediction mask does not suffer from the problems of the above two methods. This is mainly because our TDSA module can force the network's attention to the tampered area, effectively reducing the false positives of the real area.

4.4.2. Image-Level Detection. We believe that it is necessary to design an image-level evaluation for image forgery detection because our purpose is not only to accurately locate a known tampered image but also to identify the authenticity of an unknown image. Otherwise, once some real images are added to the dataset, the localization capability of the network will be greatly weakened. However, most of the existing

Table 3: Pixel-level localization AUC and F_1 are reported. The symbol "—" indicates that the experimental results are not provided in the corresponding literature. The significance of bold emphasis is to show that the value is the maximum in current column.

Method	Type	NIST16		COLUMBIA		CASIA		IMD2020	
		AUC	F_1	AUC	F_1	AUC	F_1	AUC	F_1
ELA [4]	Unsupervised	42.9%	23.6%	58.1%	47.0%	61.3%	21.4%	—	—
NOI [5]	Unsupervised	48.7%	28.5%	54.6%	57.4%	61.2%	26.3%	—	—
CFA [6]	Unsupervised	50.1%	17.4%	72.0%	46.7%	52.2%	20.7%	58.6%	—
J-LSTM [7]	Fine-tuned	76.4%	—	—	—	—	—	48.7%	—
RGB-N [8]	Fine-tuned	93.7%	72.2%	85.8%	69.7%	79.5%	40.8%	—	—
ManTra-net [9]	Fine-tuned	79.5%	73.7%	82.4%	70.3%	81.7%	45.2%	74.8%	—
CR-CNN [10]	Fine-tuned	99.2%	92.7%	86.1%	79.0%	78.9%	47.5%	—	—
GSR-net [11]	Fine-tuned	94.5%	45.6%	-	62.2%	79.6%	**57.4%**	—	—
OURS	Fine-tuned	**99.4%**	**93.8%**	**94.8%**	**90.8%**	**86.4%**	56.9%	**83.7%**	**63.3%**

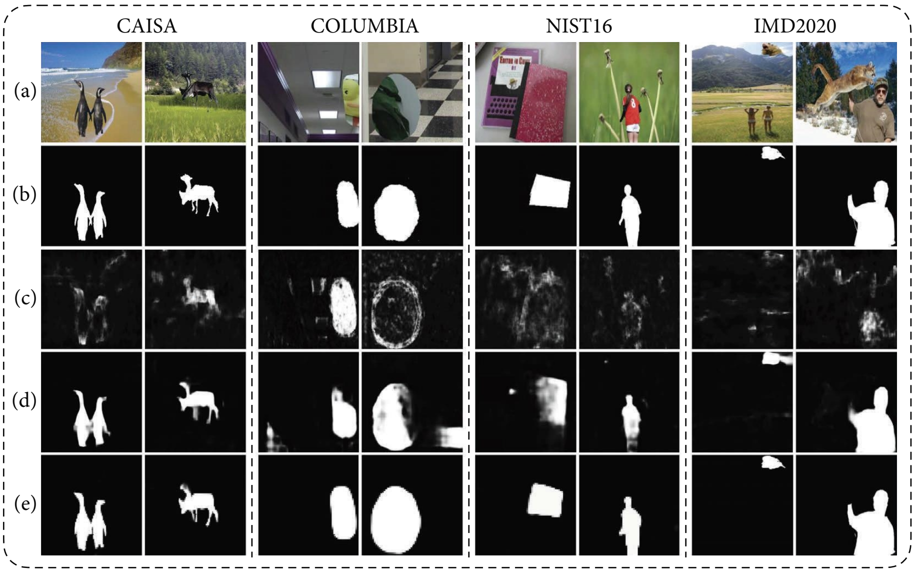

Figure 5: Comparison of pixel-level localization prediction results of different methods. From top to bottom, the figures sequentially represent (a) the tampered images, (b) the ground-truth, (c) Mantra-Net, (d) FCN, and (e) our method.

Table 4: Image-level detection evaluation over COLUMBIA and CASIA. TNR, Image-F_1, AUC, and Combine-F_1 are reported in this experiment, respectively. The significance of bold emphasis is to show that the value is the maximum in the current column.

Method	COLUMBIA				CASIA			
	TNR (%)	Image- F_1 (%)	AUC (%)	Combine-F_1 (%)	TNR (%)	Image- F_1 (%)	AUC (%)	Combine-F_1 (%)
FCN [3]	28.7	42.3	68.9	—	57.5	59.5	80.9	—
CR-CNN [10]	24.6	39.2	78.3	59.1	22.4	36.1	76.6	41.8
GSR-net [11]	1.1	2.2	50.6	32.2	1.1	2.2	50.2	29.8
OURS	**99.9**	**83.2**	**99.1**	**87.0**	**98.4**	**78.5**	**85.7**	**67.7**

works always ignore this detection. Table 4 presents our results comparing with other methods. In each experiment, we retrain the FCN [3] and add authentic images to the training set. The other two methods use the trained model provided in the paper. TNR, also known as specificity, represents the ratio of the predicted true image to the actual true image, which can accurately describe the detection at the image level.

From the results, our method can obtain the best performance in several schemes. This is mainly because other methods always generate a large number of false positives on authentic images, making them almost unusable. We summarize the reasons for the false positives shown in Table 1. Almost all DNN-based methods are trained without authentic images. Among them, the "Combine-F_1" in Table 4 represents the average value of pixel-level F_1 and image-level F_1. It can be seen that the performance of our model can achieve more than 40% improvement on the COLUMBIA dataset, and about 30% on the CASIA dataset compared to other methods. In Figure 6, we present some visualization results of real images. It is not difficult to see that our method produces hardly any false positives, while

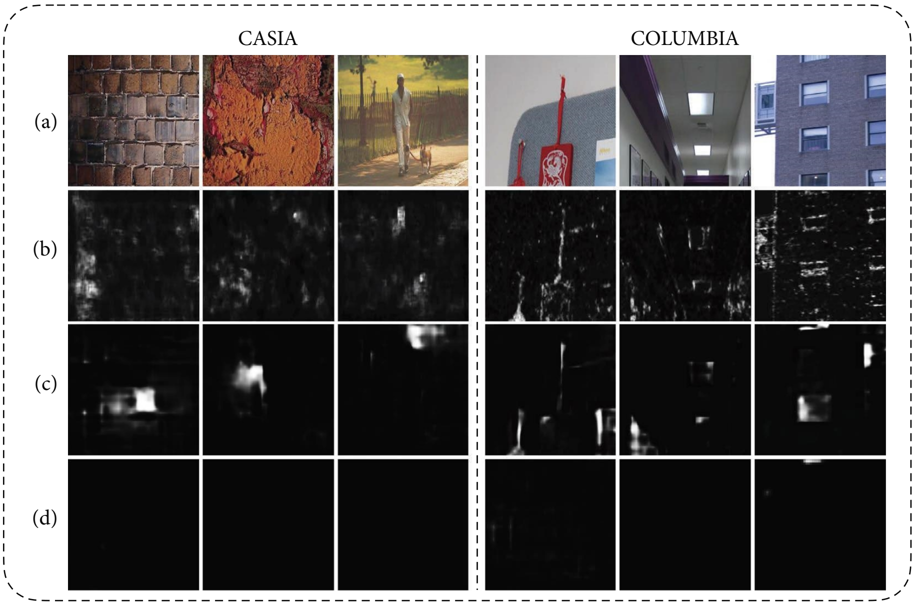

Figure 6: Comparison of image-level detection results of different methods. From top to bottom, the figures are sequentially: (a) the tampered images, (b) Mantra-Net, (c) FCN, and (d) our method.

Table 5: Ablation experiment evaluation on NIST16 and COLUMBIA. Accuracy, F_1, and AUC are reported in this experiment, respectively.

Network	NIST16			COLUMBIA		
	Accuracy (%)	F_1	AUC (%)	Accuracy (%)	F_1	AUC (%)
Proposed	99.7	93.8%	99.4	98.9	90.8%	94.8
Proposed without TDSA	98.4	90.7%	98.5	97.7	89.5%	92.1
RGB stream only	99.2	91.6%	96.5	97.5	86.4%	90.1
Noise stream only	97.9	89.3%	94.5	98.6	88.7%	92.5

the other two produce a large number of false positives. The reason for this achievement is that we add an image-level loss function, which can greatly increase the image-level detection capability of the network without affecting the pixel-level localization capability. Overall, our models are more adaptable in the real world.

4.4.3. Ablation Experiment. To verify the effectiveness of different modules in our network, we compare our full network with three combination models (combinations of different modules). Experimental results are shown in Table 5, where "Proposed" represents our proposed full model, "Proposed without TDSA" stands for a model that does not include the TDSA attention module, "RGB Stream only" indicates detection using only RGB features in the hybrid features, and "Noise Stream only" represents detection using only Noise features in the hybrid features. Our ablation experiments are evaluated over two standard datasets, NIST16 and COLUMBIA. As can be seen from Table 5, the performance of our full model is better than other combination models in the three evaluation indicators (Accuracy, F_1, AUC). The first two rows in the table demonstrate that our network achieves better performance after adding the TDSA module. The reason for this phenomenon is that our designed TDSA module can focus the network's attention to the tampered regions. Correspondingly, the TDSA module eliminates the redundancy generated in the multi-resolution fusion stage and improves the representation of features in tampered regions.

Apart from the second row, the other three rows of data also verify that our model can easily achieve the best performance with hybrid features. Specifically, "RGB Stream only" performs better on the NIST16 dataset, and "Noise Stream only" performs better on COLUMBIA dataset. Since the tampered images are diverse, using only one feature for image forgery detection may be unstable. Accordingly, we combine the two features with multiple resolutions to finally achieve the best performance.

4.4.4. Robustness Analysis. To analyze the robustness of pixel localization capability, we follow the settings in SPAN [32], and conduct a series of experimental evaluations over the NIST16 dataset and COLUMBIA dataset. In this experiment, we apply the standard OpenCV [40] built-in functions for image processing, image processing operations (attack) include image resizing with different scales, Gaussian blur with kernel size k, Gaussian noise with standard deviation σ, and compressing images with different compression factors QF. The above-mentioned image

Table 6: Robustness evaluation using AUC score over NIST16 and COLUMBIA. Different parameters are tested and Mantra-Net and SPAN are compared to show the performance evaluation.

Image processing	Parameter	NIST16			COLUMBIA		
		Mantra-net [9] (%)	S PAN [32] (%)	O URS (%)	Mantra-net [9] (%)	S PAN [32] (%)	O URS (%)
No manipulation	—	78.0	83.9	99.4	77.9	93.6	94.8
Resize	0.78x	77.4	83.2	89.8	69.0	89.9	90.4
Resize	0.25x	75.5	80.3	81.4	68.6	69.0	78.2
Gaussian blur	$k = 3$	77.4	83.1	87.7	67.7	78.9	84.1
Gaussian blur	$k = 15$	74.5	79.1	79.8	62.8	67.7	73.3
Gaussian noise	$\sigma = 3$	67.4	75.1	82.6	68.2	75.1	80.5
Gaussian noise	$\sigma = 5$	58.5	67.2	70.3	54.9	65.8	67.2
JPEG compression	QF = 100	77.9	83.5	97.0	75.0	93.3	92.9
JPEG compression	QF = 50	74.3	80.6	88.2	59.3	74.6	87.5

processing operations are common operations for image dissemination in social networks.

Table 6 presents the AUC results of the robustness analysis of several models under pixel-level localization. Combining the data in Table 6, we observe that our model can obtain more robustness in all respects than Mantra-Net [9] and SPAN [32]. To be specific, under the image resizing operation, when the image is resized to $0.25x$ size, our model only achieves a slight advantage comparing with SPAN over the NIST16 dataset. But, for other cases, our model has a significant advantage compared to others, and the image resize operation has minimal impact on our model. This is because we extract features with different resolutions, which can achieve better performance under image resizing operations. In addition, since our training set contains a large number of real images, the identification model can be fully trained by combining authentic images and tampered images. Therefore, our model also obtains better robustness when applying different levels of Gaussian blur, Gaussian noise, and JPEG Compression.

5. Conclusions

In this paper, a novel end-to-end network framework was proposed to meet the challenge of image forgery detection. We designed the TDSA module by self-attention mechanism to capture interchannel or interposition dependencies, which can precisely locate the tampered regions. Then, the image-level training strategy was designed, where the authentic images were added into the training set to improve the robustness of the detection model. By introducing an image-level training strategy, the ratio of the predicted true image to the actual true image (TNR) was greatly improved. Our network is validated over four different datasets. We experimentally demonstrate the importance of multi-resolution hybrid features, which can cope with more complex situations.

Although our scheme can achieve a superior performance comparing with a series of existing works, we should note that our scheme is actually not sensitive to smaller tampered regions. If the tampered region is too small, the extracted features are hard to reflect the unusual statistics information of the operation region, leading to more false positives. In the future, we will further strengthen hybrid features to enhance their anti-interference capability. We will also study the balance between image-level detection and pixel-level detection under the same framework.

Data Availability

The hyperlinks of experimental image datasets used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Shanghai under Grant no. 20ZR1421600, in part by Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security under Grant no. MIMS21-M-02, and in part by the STCSM Capability Construction Project for Shanghai Municipal Universities under Grant no. 20060502300.

References

[1] R. Salloum, Y. Ren, and C. C. Jay Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201–209, 2018.

[2] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528, Santiago, Chile, December 2015.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, May 2015.

[4] N. Krawetz and H. F. Solutions, "A picture's worth," *Hacker Factor Solutions*, vol. 6, no. 2, p. 2, 2007.

[5] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, no. 10, pp. 1497–1503, 2009.

[6] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[7] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, “Exploiting spatial structure for localizing manipulated image regions,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4970–4979, Venice, Italy, October 2017.

[8] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Learning rich features for image manipulation detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1053–1061, Salt Lake City, UT, USA, June 2018.

[9] Y. Wu, W. AbdAlmageed, and P. Natarajan, “Mantra-Net: manipulation tracing network for detection and localization of image forgeries with anomalous features,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9543–9552, Long Beach, CA, USA, June 2019.

[10] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, “Constrained R-CNN: a general image manipulation detection model,” in *Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, London, UK, July 2020.

[11] P. Zhou, B.-C. Chen, X. Han et al., “Generate, segment, and refine: towards generic manipulation segmentation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 13058–13065, 2020.

[12] Y. Shi and P. Sheng, “J-Net: asymmetric encoder-decoder for medical semantic segmentation,” *Security and Communication Networks*, vol. 2021, Article ID 2139024, 8 pages, 2021.

[13] Z. Wang, G. Li, Z. Zhuo, X. Ren, Y. Lin, and J. Gu, “A deep learning method for android application classification using semantic features,” *Security and Communication Networks*, vol. 2022, Article ID 1289175, 16 pages, 2022.

[14] M. H. Siddiqi, K. Asghar, U. Draz et al., “Image splicing-based forgery detection using discrete wavelet transform and edge weighted local binary patterns,” *Security and Communication Networks*, vol. 2021, Article ID 4270776, 10 pages, 2021.

[15] H. Wang and H. Wang, “Perceptual hashing-based image copy-move forgery detection,” *Security and Communication Networks*, vol. 2018, Article ID 6853696, 11 pages, 2018.

[16] Z. Xue, X. Jiang, and Q. Liu, “Semantic modeling and pixel discrimination for image manipulation detection,” *Security and Communication Networks*, vol. 2022, Article ID 9755509, 10 pages, 2022.

[17] F. Li, K. Wu, C. Qin, and J. Lei, “Anti-compression JPEG steganography over repetitive compression networks,” *Signal Processing*, vol. 170, Article ID 107454, 2020.

[18] F. Li, Z. Yu, and C. Qin, “Gan-based spatial image steganography with cross feedback mechanism,” *Signal Processing*, vol. 190, Article ID 108341, 2022.

[19] B. Bayar and M. C. Stamm, “Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, 2018.

[20] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[21] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: convolutional block attention module,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Munich, Germany, September 2018.

[22] J. Fu, J. Liu, H. Tian et al., “Dual attention network for scene segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, Long Beach, CA, USA, June 2019.

[23] Q. Zuo, S. Chen, and Z. Wang, “R2AU-Net: attention recurrent residual convolutional neural network for multimodal medical image segmentation,” *Security and Communication Networks*, vol. 2021, Article ID 6625688, 10 pages, 2021.

[24] X. Bi, X. Yang, C. Wang, and J. Liu, “High-capacity image steganography algorithm based on image style transfer,” *Security and Communication Networks*, vol. 2021, Article ID 4179340, 14 pages, 2021.

[25] Y. Zhou, B. Li, Z. Wang, and H. Li, “Integrating temporal and spatial attention for video action recognition,” *Security and Communication Networks*, vol. 2022, Article ID 5094801, 8 pages, 2022.

[26] F. Li, Y. Zeng, X. Zhang, and C. Qin, “Ensemble stego selection for enhancing image steganography,” *IEEE Signal Processing Letters*, vol. 29, pp. 702–706, 2022.

[27] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, “Focusing attention: towards accurate text recognition in natural images,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5076–5084, Venice, Italy, October 2017.

[28] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, Long Beach, CA, USA, June 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, Nevada, June 2016.

[30] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[31] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457, Washington, DC, USA, December 2015.

[32] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, “SPAN: spatial pyramid attention network for image manipulation localization,” in *European Conference on Computer Vision*, pp. 312–328, Springer, Cham, Switzerland, 2020.

[33] S. Jadon, “A survey of loss functions for semantic segmentation,” in *Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–7, IEEE, Vina del Mar, Chile, October 2020.

[34] J. Dong, W. Wang, and T. Tan, *Casia image tampering detection evaluation database*, in *Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing*, pp. 422–426, Beijing, China, June 2013.

[35] J. Dong, W. Wang, and T. Tan, “Casia image tampering detection evaluation database,” in *Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing*, pp. 422–426, Beijing, China, June 2013.

[36] T.-T. Ng, J. Hsu, and S.-F. Chang, *Columbia Image Splicing Detection Evaluation Dataset*, DVMM lab. Columbia Univ CalPhotos Digit Libr, Columbia, 2009.

[37] H. Guan, M. Kozak, E. Robertson et al., “MFC datasets: large-scale benchmark datasets for media forensic challenge evaluation,” in *Proceedings of the 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 63–72, Waikoloa Village, HI, USA, January 2019.

[38] A. Novozamsky, B. Mahdian, and S. Saic, "IMD2020: a large-scale annotated dataset tailored for detecting manipulated images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pp. 71–80, Waikoloa, HI, USA, January 2020.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, p. 1, Miami, FL, USA, June 2009.

[40] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–123, 2000.