MB-Net: multiscale boundary interaction learning for image manipulation localization

Jintong Gao[a] **and Yongping Huang**[b,*]
[a]Jilin University, College of Software, Changchun, China
[b]Jilin University, College of Computer Science and Technology, Changchun, China

Abstract. Image editing techniques can modify the content of images indiscriminately, which causes a grave threat to the security of society. Hence, the localization of manipulated images is inevitable. A serious challenge for image manipulation detection is the lack of strategies for perceiving global features and refining edges. In this paper, we present a multiscale boundary interaction learning network for image manipulation localization to solve both problems. This network contains an adjacent-scale mutual module to enrich the global perception domain by interactively learning adjacent scale features. It avoids the tremendous noise interference caused by the direct fusion of all scale features. To effectively suppress semantic content segmentation, the boundary pixel disparity module computes interpixel differences at specific angles to enhance boundary artifact recognition between tampered and real regions. The fusion attention module is proposed to combine scale and edge messages, integrating spatial and channel correlations in a compatible way. Extensive experimental results indicate that our proposed method is significantly superior to current state-of-the-art methods on public standard datasets.
 [DOI: 10.1117/1.JEI.31.6.063008]

Keywords: image manipulation localization; multiscale learning; boundary artifact localization; cross-attention fusion.

Paper 220458G received Apr. 30, 2022; accepted for publication Oct. 6, 2022; published online Nov. 8, 2022.

1 Introduction

The rapid development of image editing software allows attackers to quickly forge visually unrecognizable tampered images. Widespread dissemination of these images can cause irreversible and tremendous harm to society. At present, it is a significant challenge to determine the image authenticity as well as the tampered regions in a short period of time. Common image manipulation techniques include splicing (copying and pasting elements from one image to another), copy-move (copying and pasting elements from an image to other areas of the same image), and removal (removing elements from an image). Figure 1 illustrates instances of manipulated images. These instances are produced by the three manipulation methods, namely splicing, copy-move, and removal, on NIST16,[1] Columbia,[2] and COVERAGE.[3] The operations can severely interfere with the semantic information of the image without creating a great sense of separation from the source image. The ground-truth indicates the tampered regions of the images.

Deep learning is extensively employed to detect image manipulation. The fully convolutional network for semantic segmentation (FCN)[4] is one of the most popular deep learning models. Most investigations[5–7] take the last layer result of FCN as the prediction mask. But because the FCN contains multilayer downsampling, the model loses the majority of details in the learning process. This suggests that focusing solely on the decision layer is detrimental. For further optimization, multiscale feature fusion[8,9] ameliorates this problem. It integrates information by concatenating features of all resolutions. Nevertheless, directly connecting multiresolution features can trigger uncontrollable excessive noise interference,[10] which is fatal for manipulated

*Address all correspondence to Yongping Huang, hyp@jlu.edu.cn

1017-9909/2022/$28.00

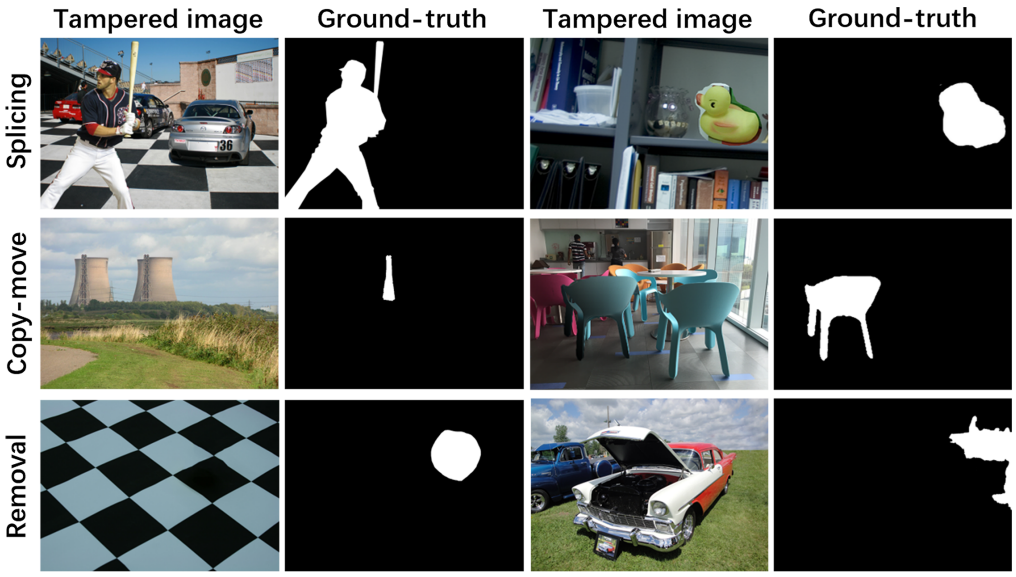

Fig. 1 Instances of manipulated images. These examples are derived from splicing, copy-move, and removal manipulations, respectively, on NIST16, Columbia, and COVERAGE. The ground-truth indicates the tampered region of the image.

image detection. Noise obscures the presence of tampered regions. Therefore, the method of blending multiscale features is one of the most urgent demands for addressing this issue.

Moreover, the FCN network is more concerned with capturing the image semantic information as an earlier semantic segmentation structure. Due to the unrestricted means of tampering, the manipulated regions are not necessarily complete objects. Purely semantic-based information segmentation is likely to result in loss of details, classification errors, or recognition difficulties. The semantic content in manipulation detection should be inhibited as much as possible. Meanwhile, attention mechanisms[11,12] are extensively exploited in tamper localization. Most of them depend on only one type of characteristic for enhancement. This affects the combination of local features with global relevance. The attention paid to the means of integration is indispensable.

To address the above issues, we establish a multiscale boundary interaction learning network, named MB-Net. The proposed network involves an adjacent-scale mutual module (ASMM) that enriches the perceptual domain at neighboring resolutions. In addition, the boundary pixel disparity module (BPDM) is introduced to enhance boundary recognition between tampered and real regions. Ultimately, the fusion attention module (FAM) integrates scale and edge messages in a compatible way.

The contributions of our work are as follows:

1. We build a multiscale boundary interaction learning network (MB-Net) to efficiently localize manipulated regions under multiresolution integration feature feedback.
2. ASMM is designed to obtain contextual information exchange at neighboring scales while avoiding generating redundant noise due to large resolution differences. BPDM calculates the differences between pixel points from specific directions to inhibit the semantic content of the image. The FAM interactively studies the spatial and channel information of adjacent-scale features and edge features.
3. Extensive experimental results indicate that our proposed method can precisely identify the manipulated regions. The performance on three public standard datasets is excellent and significantly superior to the current state-of-the-art methods.

The remainder of the paper is organized as follows. Section 2 discusses related work on the manipulated direction. Section 3 presents an overview of MB-Net and the corresponding module. The experimental procedure, comparisons, and visualization results are given in Sec. 4. We conclude this paper in Sec. 5.

2 Related Work

2.1 *Manipulation Detection*

The majority of current approaches are focused on a single sort of operation. For example, Bi et al.[13] proposed the Ringed Residual U-Net (RRU-Net) to identify splicing forgeries using residual propagation and convolutional feedback loops. Salloum et al.[5] also employed a multitask full convolutional network (MFCN) with two output branches for joint learning. However, it is difficult to determine the specific form of tampering in real-world circumstances, especially because photographs may have numerous types of overlay manipulation activities. Consequently, structures that can detect generic processes are essential.

Bappy et al.[14] combined elements of the long short-term network (LSTM) and convolutional layers to reinforce the network and highlight the disagreement between tampered and non-tampered regions. The method had trouble acquiring a large amount of contextual data, and the analysis took longer. By merging RGB images and noise features extracted through steganalysis rich model (SRM) filters, learning rich features network (RGB-N) highlights the importance of dual-stream feature fusion for manipulation detection.[15] Meanwhile, Zhou et al.[16] exploited boundary fusion to forecast tampered regions and refine branching, concentrating on operational artifacts rather than semantic content. Multiview multiscale supervision network (MVSS-Net) captures universal characteristics by employing numerous perspectives and focusing on the boundary distribution at various scales.[9] Hence, our proposed method focuses on the boundary details of different scale contents and tampered regions by interactively learning two branch features. It promotes the full communication between local perception and global perception.

2.2 *Attention Mechanism*

Attention mechanism has been increasingly popular in deep learning applications such as image manipulation, natural language processing, and audio recognition in recent years. The self-adversarial training model (SAT) provided a self-attentive technique for image tampering detection that incorporates internal dependencies in the spatial dimension as well as exterior relationships between channels.[17] The spatial-channel correlation module in progressive spatio-channel correlation network (PSCC) utilized a progressive process to improve multiscale feature representation by capturing spatial and channel correlations in a bottom-up approach. It provided holistic information to make the network more generalized to manipulation attacks.[18]

To reinforce the focus on contextual information, attention mechanisms such as coordinate attention (CA),[8] convolutional block attention module (CBAM),[11] and efficient channel attention (ECA)[19] boost the operational effect in both the channel and spatial dimensions. Their presence is common in tamper detection. In our method, the FAM combines RGB and border features to consolidate global information, resulting in improved inference.

3 Proposed Method

We propose a multiscale boundary interaction learning network to detect manipulated images. The general framework of MB-Net is shown in Fig. 2.

We denote the input image as $I \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are the height, width, and the number of channels of the image, respectively. Image I is input into the backbone of ResNet50,[20] where the features acquired are F_i, $i \in \{0,1,2,3,4\}$. These features are passed to ASMM and BPDM to obtain the feature maps, respectively. Then the multiscale feature map F_M and the new boundary map E_M at the same scale are connected by jumping to obtain the first prediction mask through the FAM. At the same time, the second prediction mask is obtained by combining the boundary maps at different scales. In the end, we calculate the loss of prediction maps and ground-truths to train the model.

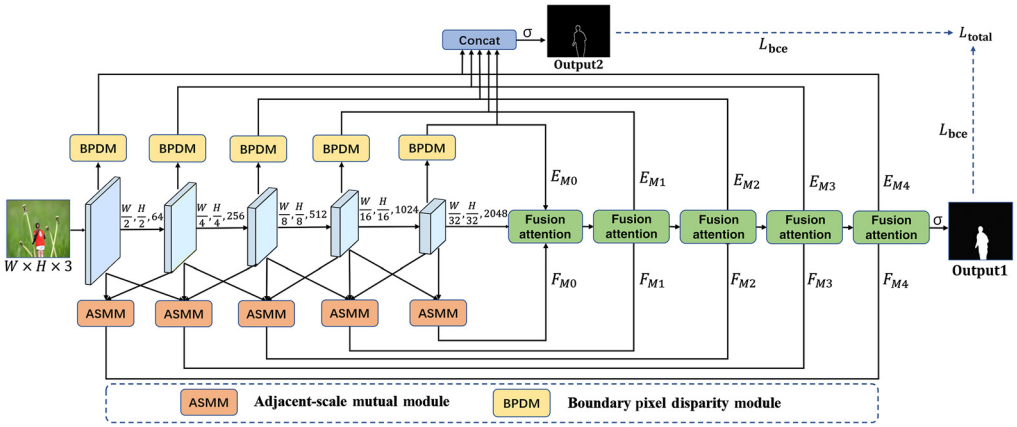

Fig. 2 MB-Net architecture.

3.1 *Adjacent-Scale Mutual Module*

Multiscale feature extraction is often applied in image tampering detection and localization. The method is the integration of shallower features layer by layer and the integration of multiple layers of features in a fully concatenated or heuristic manner. However, single-layer features can only represent information on a specific scale, and shallow features also lack deeper details. Although maximizing the integration of features at different scales, the heuristic suffers from difficulties due to the presence of different resolutions and noise at each scale. Influenced by the aggregated interaction strategy,[10] we propose the adjacent-scale mutual module to avoid the interference of large resolution differences caused by multiscale information fusion. It fully learns the contextual information of neighboring scales.

As illustrated in Fig. 3, ASMM is divided into two blocks. Of these, two pairs, F_0 and F_1 and F_3 and F_4 are entered into a two-layer adjacent-scale mutual block (2AS), and the rest are put into a three-layer adjacent-scale mutual block (3AS).

Consider the case of 3AS, the feature maps F_{i-1}, F_i, and $F_{i+1} \in \mathbb{R}^{H\times W\times C}$, $i \in \{1,2,3\}$ are fed into the 3×3 convolution layer to obtain F'_{i-1}, F'_i, and $F'_{i+1} \in \mathbb{R}^{H\times W\times \frac{C}{2}}$, respectively,

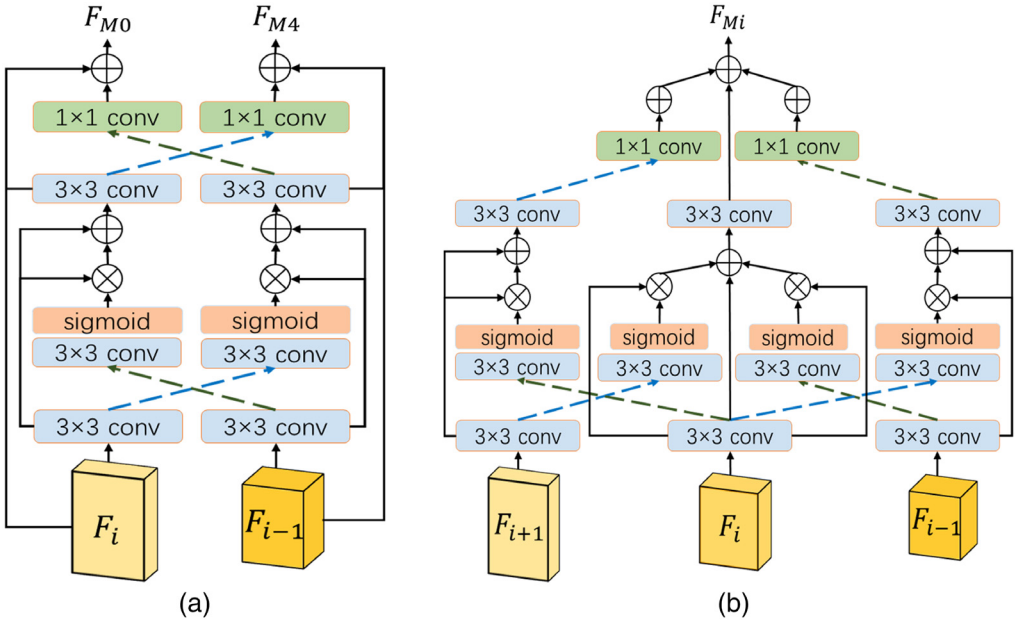

Fig. 3 Details of the adjacent-scale mutual module: (a) two-layer adjacent-scale mutual block (2AS) and (b) three-layer adjacent-scale mutual block (3AS).

by reducing the number of channels. The computational processes of interactively extracting features F''_{i-1}, F''_i, and F''_{i+1} are expressed as

$$F''_{i-1} = \sigma(C(\mathrm{Up}(F'_i))) \otimes F'_{i-1} \oplus F'_{i-1}, \tag{1}$$

$$F''_i = (\sigma(C(\mathrm{Down}(F'_{i-1}))) \otimes F'_i \oplus F'_i) \oplus (\sigma(C(\mathrm{Up}(F'_{i+1}))) \otimes F'_i \oplus F'_i), \tag{2}$$

$$F''_{i+1} = \sigma(C(\mathrm{Down}(F'_i))) \otimes F'_{i+1} \oplus F'_{i+1}, \tag{3}$$

where Up and Down are the samples, C denotes the 3×3 convolution layer with batch normalization and ReLU, and $\oplus$ and $\otimes$ denote dot product and addition, respectively. σ is the sigmoid function. The sigmoid function is applied to acquire the weight values of the feature. The prediction range of the pixels in the manipulation localization task is between 0 and 1. Therefore, we chose this function to implement cross-learning for multiplying weights at adjacent scales. At the same time, feature fusion at neighboring resolutions avoids the huge computational effort and noise disturbance associated with large resolution gaps while taking into account contextual information. This will optimize the feature extraction capability of the module. F_{M1}, F_{M2}, and F_{M3} are the final output resolution features.

In the same way, 2AS calculates two pairs of neighboring features to access F_{M0} and F_{M4}.

3.2 *Boundary Pixel Disparity Module*

To detect boundaries effectively, pixel difference convolution[21] is selected. It executes the convolution operation using pixel differences to replace the original pixels in the feature map covered by the convolution kernel. The BPDM consists of normal convolution, central convolution, angular convolution, skip convolution, and radial convolution, as shown in Fig. 4. In addition to normal convolution, other convolutions capture differences between pixels in specific directions, thereby reinforcing the pixel perceptual domain.

The calculations of pixel disparity convolutions are expressed as follows:

$$y_{\mathrm{nor}} = \sum_{i=1}^{N} \omega_i \cdot x_i, \tag{4}$$

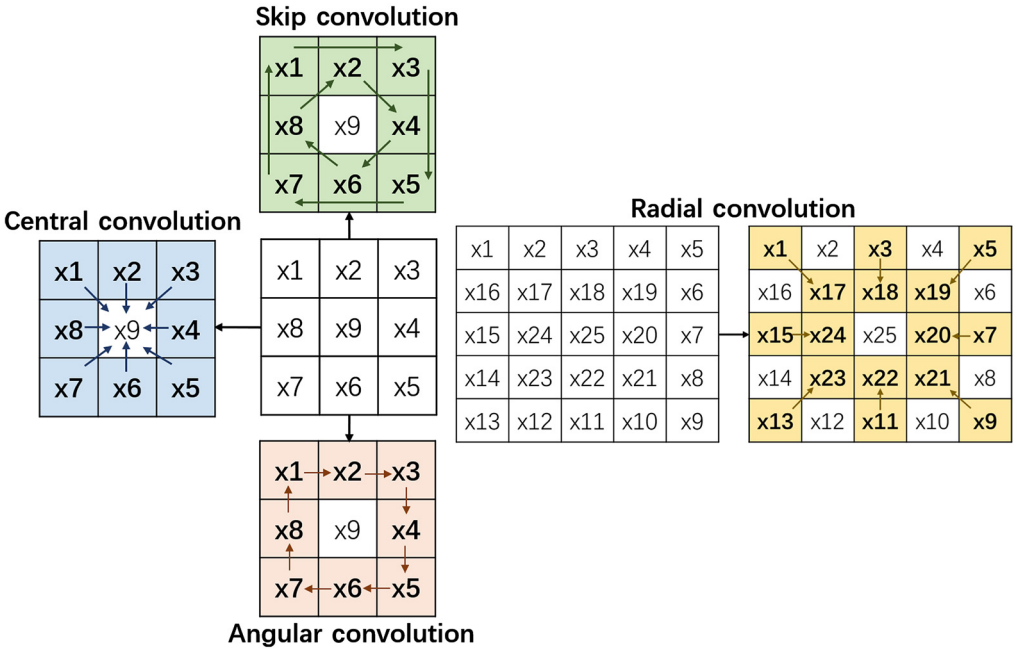

Fig. 4 Various convolutions of boundary pixel disparity module.

$$y_{\text{cen}} = \sum_{i=1}^{N-1} \omega_i \cdot (x_i - x_9), \tag{5}$$

$$y_{\text{ang}} = \sum_{i=1}^{N-2} \omega_i \cdot (x_i - x_{i+1}) + \omega_8 \cdot (x_8 - x_1), \tag{6}$$

$$y_{\text{skip}} = \sum_{i=1}^{N-3} \omega_i \cdot (x_i - x_{i+2}) + \omega_7 \cdot (x_7 - x_1) + \omega_8 \cdot (x_8 - x_2), \tag{7}$$

$$y_{\text{rad}} = \sum_{i=0}^{N-2} \omega_i \cdot (x_{2i+1} - x_{i+17}), \tag{8}$$

where x_i is the input pixels and ω_i is the weight in the 3×3 or 5×5 convolution kernel.

The ability to perceive the surrounding region is reduced when only a single pixel is computed. As a result, a suitable pixel disparity must be chosen. First, the central convolution expands the perception of the surrounding area by the central pixel. Second, according to the extended local binary pattern (ELBP),[22] both the angular and radial directions are useful for computer vision recognition. Thus, angular and radial convolution are born. In addition, we suggest skip convolution, which is similar to angular convolution but differs in that the pixel pairs are the different values from two neighboring pixels, as inspired by a jump junction. Except for the radial convolution, which selects eight pairs of pixels in this local block, all convolution kernels are 3×3, as shown in Fig. 4. Their differences are calculated as weights. The radial convolution kernel is of size 5×5. Eight pairs of pixels are created in the radial direction, and together with the middle pixel difference of 0, they are again treated as weights.

We sequentially input F_i, $i \in \{0,1,2,3,4\}$ to radial convolution, skip convolution, angular convolution, central convolution, and normal convolution to obtain multiscale boundary maps E_M. Each convolution contains a group normalization and rectified linear unit as postprocessing. They are merged as the edge prediction output, namely output2.

3.3 *Fusion Attention Module*

ASMM and BPDM can inform each other to capture the abundant contextual dependencies of the intrinsic inconsistencies extracted from the tampered region. Therefore, we propose a fusion attention module to bridge and channel the exchange of the two information streams and to facilitate qualitative feature fusion. The module consists of two parts, shown in Fig. 5, the cross-attention learning block and the receptive domain enhancement block. The output feature of FAM is represented as F_{output}.

The cross-attention learning block is divided into three parts: spatial feature extraction, channel feature extraction, and feature fusion. Spatial feature extraction A_s and channel feature extraction A_c are denoted as

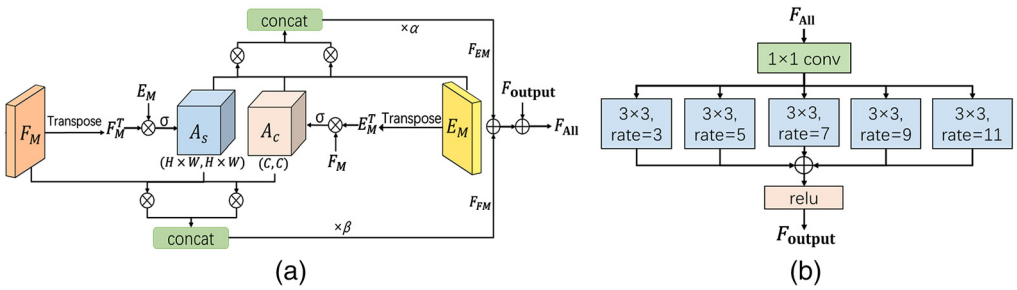

Fig. 5 Details of the fusion attention module: (a) cross-attention learning block and (b) receptive domain enhancement block.

$$A_s = \sigma(F_M^T \otimes E_M), \tag{9}$$

$$A_c = \sigma(F_M \otimes E_M^T), \tag{10}$$

where F_M^T and E_M^T are the transposed features of F_M and E_M, respectively, $\otimes$ denotes the dot product, and σ is attention weighting obtained by the sigmoid function. Subsequently, the feature maps are input into the extraction blocks for interactive learning as

$$F_{\mathrm{EM}} = \alpha \times \mathrm{concat}((E_M \otimes A_s), (A_c \otimes E_M)), \tag{11}$$

$$F_{\mathrm{FM}} = \beta \times \mathrm{concat}((F_M \otimes A_s), (A_c \otimes F_M)), \tag{12}$$

where α and β denote the adaptive training parameters, which are used to emphasize the importance of complementary information. The computation of the feature fusion is presented as

$$F_{\mathrm{All}} = F_{\mathrm{EM}} \oplus F_{\mathrm{FM}} \oplus F_{\mathrm{output}}, \tag{13}$$

where $\oplus$ denotes addition.

The cross-attention learning block extracts mappings in the channel and spatial dimensions under multiple features. It adaptively trains the model structure using learnable factors that reflect long-term contextual information and effectively exploits weighting properties. Specifically, boundary extraction aims to capture tampered boundaries at full strength. However, it tends to focus on all of the item boundaries of images in real scenarios. Thus, it requires the introduction of adaptive parameters for boundary screening guided by a multiscale module. In the end, we successfully interactively fused the features.

To better refine the feature map, we present a receptive domain enhancement block. It takes F_{All} as the input and reduces the number of channels through a 1×1 convolution layer. It then convolves through dilated convolution of specified dilatation rates to capture the information and resize the feature map. The final prediction mask is generated with a channel of 1 by the sigmoid function.

3.4 *Loss Function*

To better train and evaluate the functionality of the model, MB-Net has two outputs: loss at pixel scale for improving sensitivity to detection of pixel-level operations and loss at boundary for learning semantically irrelevant features. In the training process, binary cross-entropy loss (L_{bce}) is used as the training loss function. The expression of the loss function is formulated as follows:

$$L_{\mathrm{bce}} = -(y \log x) + (1 - y)\log(1 - x), \tag{14}$$

where y represents the ground-truth and x represents the output of the model. The total loss L_{total} is expressed as

$$L_{\mathrm{total}} = L_{\mathrm{bce}}(y_{\mathrm{gt}}, y_{\mathrm{output1}}) + L_{\mathrm{bce}}(y_{\mathrm{edge}}, y_{\mathrm{output2}}), \tag{15}$$

where y_{gt} denotes ground-truth, y_{output1} denotes prediction mask, and y_{edge} and y_{output2} denote boundary ground-truth and boundary-prediction masks, respectively.

4 Experiments

4.1 *Experimental Setup*

4.1.1 *Datasets*

In our investigation, we conducted comprehensive experiments, demonstrated our proposed method in four public standard image manipulation localization datasets (DEFACTO,[23]

Table 1 Training methods and dataset distribution.

Training method	Training	Testing
Ablation study	DEFACTO (54000)	DEFACTO (5400), NIST16 (564)
Benchmark	COVERAGE (75), Columbia (126), and NIST16 (404)	COVERAGE (25), Columbia (54), and NIST16 (160)
Fine-tuning	DEFACTO (54000), COVERAGE (75), Columbia (126), and NIST16 (404)	COVERAGE (25), Columbia (54), and NIST16 (160)

COVERAGE,[3] NIST16,[1] and Columbia[2]), and compared the results with state-of-the-art methods. DEFACTO is a synthesized dataset generated from MSCOCO.[24] We select 54,000 photos for training (excluding morphing). Due to memory limitations, we randomly select 8000 images per epoch from the synthetic dataset for training. To ensure that the size of the testing dataset does not surpass the number of actual training images, the ratio of training to testing in the ablation experiment is adjusted to 10:1. It is worth noting that the photos that we dealt with are far fewer than Mantra-Net[25] and SPAN.[12] In the comparison experiment, we employed benchmark training and fine-tuning as our model training regimes for optimal performance. According to benchmark training, the model was directly trained and tested on COVERAGE, Columbia, and NIST16. Fine-tuning entails training the model on DEFACTO initially, followed by additional training and testing on COVERAGE, Columbia, and NIST16. Specifically, COVERAGE[3] generates 100 images by copy-move tactics, which are derived from real images and were 75:25 for training and testing, respectively. NIST16[1] consists of 564 tampered images by three operations (splicing, copy-move, and removal). It has a training to testing ratio of 404:160. With the exception of ground-truth and boundary-masks, Columbia[2] offers 180 splicing pictures. A 7:3 training/testing division is provided for fine-tuning. Table 1 gives details of the training approach and the distribution of the datasets.

4.1.2 *Metrics*

For pixel-level manipulation detection, pixel-level AUC (the area under the receiver operating characteristic curve) and F1 scores are employed for comparison experiments. When F1 is calculated from pixel-level precision and recall, we follow previous work to acquire the best F1 scores obtained by adjusting the prediction thresholds.

4.1.3 *Implementation details*

MB-Net is implemented in PyTorch[26] and trained on an NVIDIA GeForce RTX 3090. The dimensions of the entered model are 320×320. Our backbone is initialized by ImageNet pretrained parameters. Adam[27] is deployed to optimize the whole model. The batch size in one epoch is 10, and the initial learning rate is fixed at 1×10^{-3}. If the validation loss recorded in each epoch is not decreased within 10 epochs, the learning rate is divided by 10 until it reaches 1×10^{-8}. In terms of processing speed, MB-Net takes ~0.21 s per image (320×320).

4.2 *Ablation Study*

This section discusses the effectiveness of each module and different boundary convolutions.

4.2.1 *Effectiveness of modules*

To evaluate the effectiveness of the proposed ASMM, BPDM, and FAM, we quantitatively evaluate MB-Net and its components:

ResNet50: As the backbone of the model, ResNet50[20] is selected.

Table 2 Quantitative results of MB-Net and its components on DEFACTO.

Component	AUC	F1
ResNet50	97.2	71.4
ASMM	97.4	74.2
ASMM + BPDM	97.9	76.8
ASMM + BPDM + FAM (MB-Net)	98.5	78.1

ASMM: It is constructed on the backbone of ResNet50 and consists of a pair of two-layer adjacent feature interaction blocks and many three-layer adjacent feature interaction blocks.

BPDM: This module contains blocks of interpixel disparity convolutions at different angles.

FAM: This module applies attention mechanisms to fuse multiscale interaction streams and boundary extraction streams after invoking skip connections for self-attentive feature enhancement.

Our proposed components are beneficial on DEFACTO, as demonstrated in Table 2, with enhancements in both AUC and F1. ASMM is optimized over backbone in terms of AUC and F1 by 0.2% and 2.8%, respectively. It demonstrates that incorporating adjacent scale features significantly amplifies the extraction strength of the module. The accurately derived characteristics are essential for manipulation localization. Based on that, there is a further enhancement of BPDM from the experimental results. The module is sensitive to surrounding pixels and expands the perceived domain. Local edge information is therefore purposefully refined. Compared with the former, MB-Net with the introduction of FAM improves by 0.6% and 1.3% on AUC and F1, respectively. The cross-attention learning block provides the majority of the optimization support. Unlike most attention mechanisms that focus only on themselves, it cross-multiplies the spatial and channel information of scale features and edge features separately. The final result, therefore, contains the advantages of both features.

4.2.2 *Effectiveness of pixel disparity convolutions*

Table 3 demonstrates the effectiveness of the different boundary convolutions on NIST16 for benchmark training. Normal convolution (also called vanilla convolution), central convolution, angular convolution, skip convolution, and radial convolution are mainly included.

From the table, the module is somewhat ameliorated by replacing the vanilla convolution with the last four convolutions. For example, in comparison with setup2, setup3 enhanced the AUC by 0.19% and F1 by 5.84%. The AUC for setup5 represents a 0.19% increase over setup4 and a 5.84% increase for F1. This means that convolutions with interpixel differences at specific orientations are estimated optimally. Although setup2 is lower than setup1 in F1, it is 0.25% higher in AUC than the latter. We speculate that the main reason for the drop may be the lack

Table 3 Quantitative results of pixel disparity convolutions on NIST16.

Setup	Component	AUC	F1
1	Normal convolution (NC) × 5	97.79	81.52
2	Central convolution (CC) + NC × 4	98.04	77.94
3	Angular convolution (AC) + CC + NC × 3	98.23	83.78
4	Skip convolution (SC) + AC + CC + NC × 2	98.31	84.21
5	Radial convolution (RC) + SC + AC + CC + NC	98.68	86.90

of adequate sensory fields. Central convolution has difficulties with complementary pixel values in the same row or column. Subsequently, other convolutions have compensated for this problem.

4.3 *Quantitative Results Compared Against State-of-the-Art Methods*

We select two training settings (benchmark/fine-tuning) and evaluate them on three public datasets (COVERAGE,[3] Columbia,[2] and NIST16[1]), comparing the current more state-of-the-art methods. The differences between these settings are presented in Sec. 4.1. The compared detection methods are the classical unsupervised method (ELA,[28] NOI1,[29] and CFA1[30]), the detection method implemented by deep learning networks (J-LSTM,[31] H-LSTM,[14] RGB-N,[15] GSR-Net,[16] SPAN,[12] Mantra-Net,[12] and SAT[17]), in which the training settings of the last two methods are pre-training and benchmark training, respectively. Table 4 shows the results of the AUC/F1 comparison between our method and other baseline methods. Specifically, the performance of our method appears to be fairly outstanding on public datasets and the average values.

Compared with typical unsupervised methods, deep learning network structure (incorporating MB-Net) largely avoids the errors associated with manual manipulation analysis. As evidenced by the table, the overall results for the network with the supervised system are better than the unsupervised methods, with the AUC and F1 being far superior to the latter.

MB-Net achieves the best performance on Columbia, NIST16, and the average compared with the supervised approach deep learning method. Although the AUC metric did not achieve the best value on COVERAGE, it is only 1% lower than the AUC of SPAN, and F1 outperformed the other networks. It demonstrates the excellence of MB-Net. Meanwhile, when using benchmark training with few training images, e.g., Columbia with 126 forged images and NIST16 with 404 forged images, MB-Net surpasses SAT in terms of performance. When faced with high-value evaluation results, we outperformed AUC and F1 of SPAN by 3.9% (97.5% - 93.6%) and 8.1% (89.6% - 81.5%) on Columbia. Our method does not rely on large-scale training data to complete the objectives.

Analysis of the models and tables reveals that the J-LSTM and H-LSTM split the image into patches for input, limiting the network to locating contiguous tampered regions at a single scale. They apply the LSTM for linear processing but struggle to acquire large areas of spatial and

Table 4 Quantitative results compared against state-of-the-art methods.

Method	Training	COVERAGE	Columbia	NIST16	Average
ELA[28]	Unsupervised	58.3/22.2	58.1/47.0	42.9/23.6	53.1/30.9
NOI1[29]	Unsupervised	58.7/26.9	54.6/57.4	48.7/28.5	54.0/37.6
CFA1[30]	Unsupervised	48.5/19.0	72.0/46.7	50.1/17.4	56.9/27.7
J-LSTM[31]	Fine-tuning	61.4/—	—/—	76.4/—	68.9/—
H-LSTM[14]	Fine-tuning	71.2/—	—/—	79.4/—	75.3/—
RGB-N[15]	Fine-tuning	81.7/43.7	85.8/69.7	93.7/72.2	87.1/61.9
GSR-Net[16]	Fine-tuning	76.8/47.7	—/—	94.5/73.6	85.7/60.7
ManTra-Net[25]	Pretraining	81.9/—	82.4/—	79.5/—	81.3/—
SPAN[12]	Fine-tuning	93.7/55.8	93.6/81.5	96.1/58.2	94.5/65.2
SAT[12]	Benchmark	85.6/52.6	91.7/89.1	94.3/62.2	90.5/68.0
MB-Net	Benchmark	79.7/43.1	96.0/90.5	96.4/71.5	90.7/68.4
MB-Net	Fine-tuning	92.7/ 59.6	97.5/89.6	98.7/86.9	96.3/78.7

Pixel-level AUC/F1 (%) on public datasets. "—" denotes that the result is not available in the literature.

contextual information. RGB-N improved performance substantially by introducing noise stream enhancement features. Generative networks are employed by GSR-Net for various operations. There is the necessity for information interaction. The AUC of GSR-Net exceeded that of H-LSTM by 15.1% on NIST16. However, further improvements are needed to the operation segmentation problem. SPAN constructs a pyramid of local self-attentive blocks and effectively models the relationships between multiscale image blocks, with quantitative results far exceeding those of the previously mentioned models. ManTra-Net learns operation traces by classifying 385 image operation types and proposes a long-term and short-term memory solution to evaluate local anomalies. The network is pretrained using over 300,000 images and has a high AUC on common datasets. It reveals that the larger the dataset is, the more beneficial the model training is.

SAT utilizes a channeled high-pass filter block (CW-HPF) to enhance noise inconsistency between the original and tampered regions. Although MB-Net clearly outperformed SAT on other datasets and the average results, the predictions on COVERAGE were less promising. We suspect that this is because COVERAGE has too few images. We have difficulty refining the performance of the model with a small amount of data in benchmark training. It is therefore insensitive to this dataset. However, MB-Net has compensated for the data volume, and the evaluation results are satisfactory with fine-tuning. We also consider that SAT introduces noise blocks to complement the frequency information. Frequency is additive to manipulation localization. But frequencies are not utilized in MB-Net. In future research, we will investigate the high- and low-frequency features in further depth.

4.4 *Robustness Evaluation*

To evaluate the robustness of MB-Net under various attacks, we performed the following attacks on NIST16 and Columbia, respectively: noise attacks with Gaussian, Uniform, and Poisson; JPEG and WEBP compression; and Box, Gaussian, and Median blur. Figures 6 and 7 show the performance curves of MB-Net for pixel-level AUC under benchmark training, with each

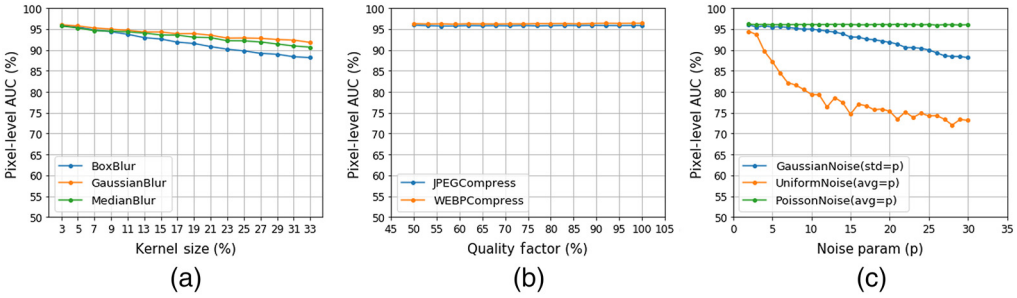

Fig. 6 Robustness results on NIST16 under various attacks: (a) blur attacks, (b) compression attacks, and (c) noise attacks.

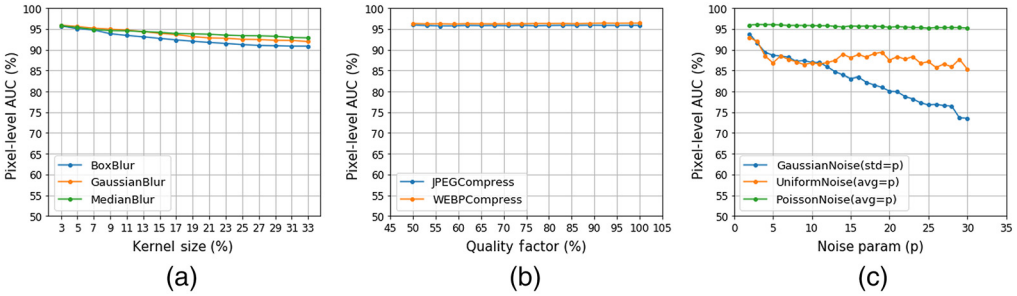

Fig. 7 Robustness results on Columbia under various attacks: (a) blur attacks, (b) compression attacks, and (c) noise attacks.

Table 5 Robustness comparison with respect to various distortions on NIST16.

Attack	ManTra-Net	SPAN	Ours
None	78.05	83.95	96.40
Gaussian blur (kernel size = 3)	77.46	83.10	96.06
Gaussian blur (kernel size = 15)	74.55	79.15	94.31
JPEG compress (quality = 100)	77.91	83.59	95.89
JPEG compress (quality = 50)	74.38	80.68	95.97
Gaussian noise (sigma = 3)	67.41	75.17	95.48
Gaussian noise (sigma = 15)	58.55	67.28	93.10

Results are reported as pixel-level AUC (%).

Table 6 Robustness comparison with respect to various distortions on Columbia.

Attack	ManTra-Net	SPAN	Ours
None	77.95	96.60	96.00
Gaussian blur (kernel size = 3)	67.72	78.97	95.92
Gaussian blur (kernel size = 15)	62.88	67.70	93.94
JPEG compress (quality = 100)	75.00	93.32	95.88
JPEG compress (quality = 50)	59.37	74.62	95.75
Gaussian noise (sigma = 3)	68.22	75.11	91.56
Gaussian noise (sigma = 15)	54.97	65.80	83.08

Results are reported as pixel-level AUC (%).

data point representing the performance under a fixed distortion. MB-Net is stable under the compression attacks. Under the blurring attacks, the model performance degrades, but the degradation is almost negligible. In addition, under the attacks of noise, especially UniformNoise and PossionNoise, the performance of the MB-Net degrades even more, almost linearly. The reason is that the model does not have the introduction of a high-pass filter for noise feature extraction. We will investigate this aspect in the future. In addition, Tables 5 and 6 include results for comparison with other models on NIST16 and Columbia. The performance of ManTra-Net and SPAN was severely affected by the attack, with a much higher degradation trend than MB-Net. This indicates the robustness of MB-Net.

4.5 *Visualization Results*

Figure 8 shows the visualization results of the MB-Net on NIST16 and Columbia. The samples contain the tampered image, the ground-truth mask, and the predicted binary mask. From top to bottom, the image processing methods are as follows: splicing, copy-move, and removal on NIST16 on the left. The Columbia dataset contains only splicing operations on the right. For different operations, our model still detects forgery regions accurately. Reinforcing the details of the edge-optimized prediction maps by suppressing semantic information ensures more accurate results.

In Fig. 9, MB-Net compares the qualitative comparative visualization of localization predictions against state-of-the-art methods from NIST16, COVERAGE, and Columbia. MB-Net is

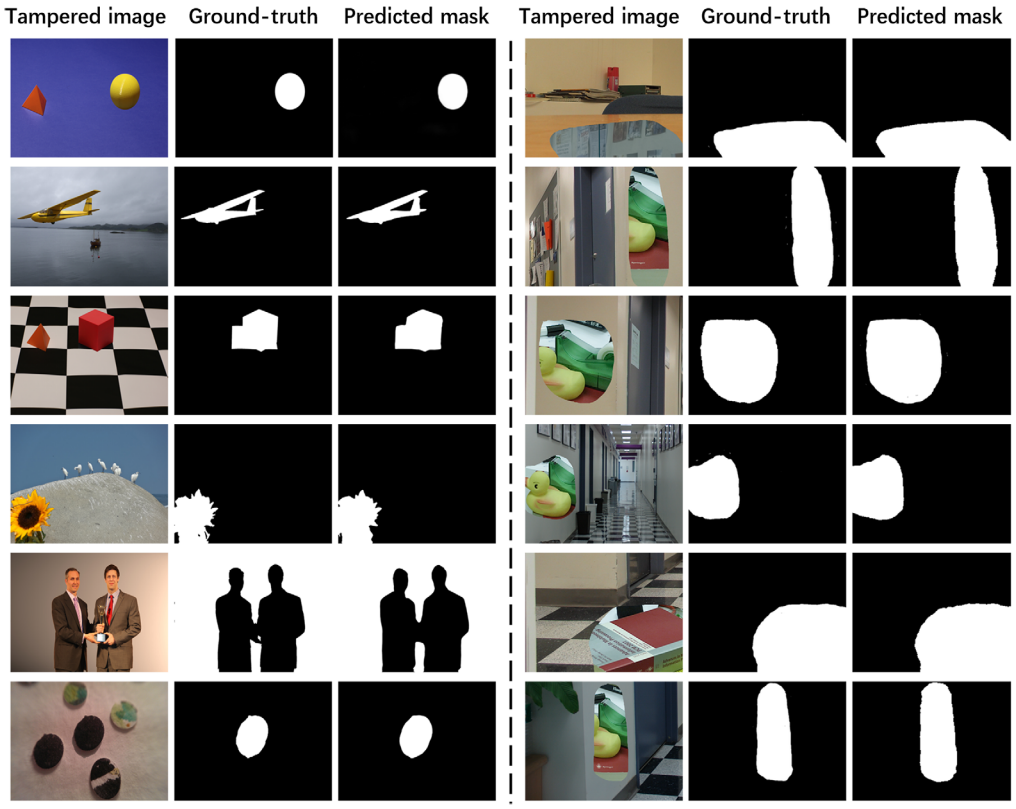

Fig. 8 Examples of segmentation results are shown on NIST16 from the left and the Columbia dataset from the right. The instances indicate tampered images, ground-truth, and predicted binary masks. From the top to the bottom are the samples showing manipulations of splicing, copy-move, and removal on NIST16. The Columbia contains only splicing operations.

more effective at localization for all three operations than the vaguely localized ManTra-Net, SPAN, and less precise MVSS-Net. Multiscale interactive learning facilitates our model to focus on the tampered region, suppressing misclassification due to semantic information and ensuring more accurate segmentation. The prediction masks for the splicing images are extremely similar to ground-truth. The boundaries are smoother. The prediction masks for copy-move images were approximated to the ground-truth. Even reflections in water can be accurately identified. However, there are still misjudgments in terms of contour detail, and complete objects cannot be accurately identified. The predictions from the removal images are good, and basic shapes can already be discerned. However, further improvements are needed to achieve complete agreement. In the future, we will supplement more information by adding training data and adversarial training.

5 Conclusion

We proposed a multiscale boundary interaction learning network for image manipulation. It incorporated scale messages and boundary artifacts through fusion attention mechanisms. In this regard, the adjacent-scale mutual module facilitated the fusion of feature information at neighboring scales. Global awareness was further strengthened. The influence of semantic content was then substantially avoided by calculating the disparity between pixels in multiple directions. In addition, we introduced an attention mechanism with learnable parameters to dramatically promote the capability of integrating contextual components. Extensive experimental results demonstrated that MB-Net outperformed advanced image manipulation detection methods. In the future, we will explore even more superior high-frequency information to complement the texture-focused CNN streams to address the challenge of manipulation.

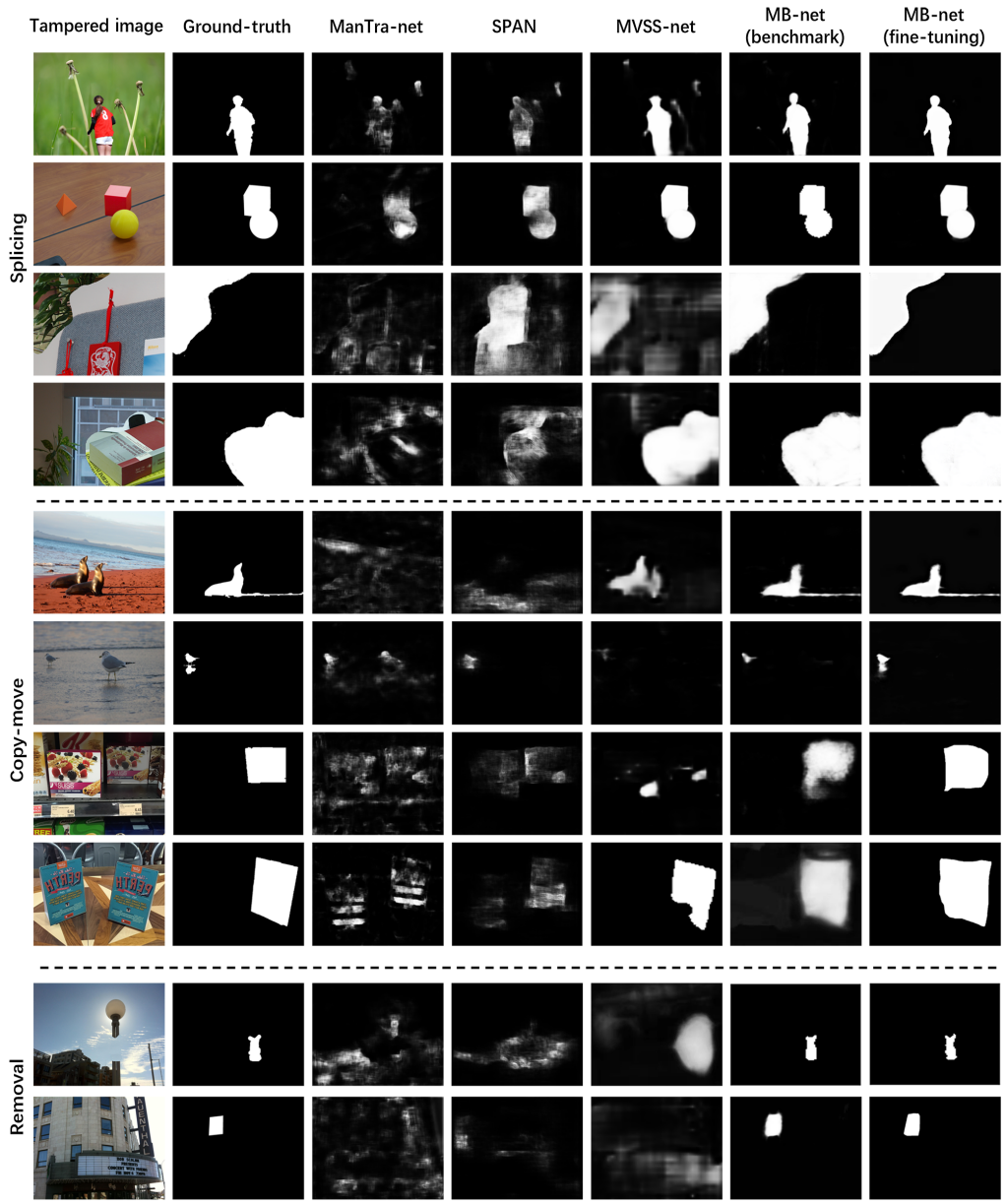

Fig. 9 Qualitative comparative visualization of localization predictions against state-of-the-art methods on NIST16, COVERAGE, and Columbia. It contains splicing, copy-move, and removal manipulations.

Acknowledgments

This research was supported by the Regional Joint Fund of NSFC (U19A2057), the National Natural Science Foundation of China (61876070).

References

1. X. Liu et al., "NIST: NIST nimble 2016 datasets," (2016). https://www.nist.gov/itl/iad/mig/.
2. T. T. Ng, J. Hsu, and S. F. Chang, "Columbia image splicing detection evaluation dataset," in *Proc. AAAI Conf. Artif. Intell.* (2009). http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/photographers.htm.
3. B. Wen et al., "Coverage-a novel database for copy-move forgery detection," in *IEEE Int. Conf. Image Process. (ICIP)* (2016).

4. E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017).
5. R. Salloum, Y. Ren, and C.-C. Jay Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Visual Commun. Image Represent.* **51**, 201–209 (2018).
6. J. B. Liu et al., "A holistically-guided decoder for deep representation learning with applications to semantic segmentation and object detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **PP**, 1–1 (2021).
7. Y. W. Li et al., "Fully convolutional networks for panoptic segmentation," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 214–223 (2021).
8. Q. B. Hou, D. Q. Zhou, and J. S. Feng, "Coordinate attention for efficient mobile network design," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 13708–13717 (2021).
9. X. R. Chen et al., "Image manipulation detection by multi-view multi-scale supervision," in *IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, pp. 14165–14173 (2021).
10. Y. W. Pang et al., "Multi-scale interactive network for salient object detection," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 9410–9419 (2020).
11. S. Woo et al., *CBAM: Convolutional Block Attention Module*, Springer, Cham (2018).
12. X. Hu et al., "SPAN: spatial pyramid attention network for image manipulation localization," *Lect. Notes Comput. Sci.* **12366**, 312–328 (2020).
13. X. L. Bi et al., "RRU-Net: the ringed residual U-Net for image splicing forgery detection," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. Workshops (CVPRW)*, pp. 30–39 (2019).
14. J. H. Bappy et al., "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.* **28**(7), 3286–3300 (2019).
15. P. Zhou et al., "Learning rich features for image manipulation detection," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 1053–1061 (2018).
16. P. Zhou et al., "Generate, segment, and refine: towards generic manipulation segmentation," in *Proc. AAAI Conf. Artif. Intell.*, Vol. 34, pp. 13058–13065 (2020).
17. L. Zhuo et al., "Self-adversarial training incorporating forgery attention for image forgery localization," *IEEE Trans. Inf. Forensics Security* **17**, 819–834 (2022).
18. X. Liu et al., "PSCC-Net: progressive spatio-channel correlation network for image manipulation detection and localization," (2021).
19. Q. L. Wang et al., "ECA-Net: efficient channel attention for deep convolutional neural networks," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 11531–11539 (2020).
20. K. He et al., "Deep residual learning for image recognition," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 770–778 (2016).
21. Z. Su et al., "Pixel difference networks for efficient edge detection," in *IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, pp. 5097–5107 (2021).
22. L. Liu et al., "Extended local binary patterns for texture classification," *Image Vis. Comput.* **30**(2), 86–99 (2012).
23. G. Mahfoudi, B. Tajini, and F. Retraint, "Defacto: image and face manipulation dataset," in *27th Eur. Signal Process. Conf. (EUSIPCO)*, pp. 1–5 (2019).
24. T. Y. Lin et al., "Microsoft COCO: common objects in context," *Lect. Notes Comput. Sci.* **8693**, 740–755 (2014).
25. Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-Net: manipulation tracing network for detection and localization of image forgeries with anomalous features," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, pp. 9535–9544 (2019).
26. A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, p. 32 (2019).
27. D. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2014).
28. N. Krawetz and H. F. Solutions, *A Picture's Worth*, Hacker Factor Solutions (2017).
29. B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image Vis. Comput.* **27**(10), 1497–1503 (2009).

30. P. Ferrara et al., "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security* **7**(5), 1566–1577 (2012).
31. J. H. Bappy et al., "Exploiting spatial structure for localizing manipulated image regions," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 4980–4989 (2017).

Jintong Gao is currently pursuing an ME degree with the College of Software from Jilin University, China. Her research interests include multimedia forensics and pattern recognition, especially image manipulation detection and localization.

Yongping Huang received his PhD from CIOMP of Chinese Academy of Sciences and is an associate professor at Jilin University. His research interests include intelligent measurement and control systems, embedded software architecture, cyber physical systems, and complex self-adaptive systems.