Image Manipulation Detection Using the Attention Mechanism and Faster R-CNN

Kang Tan, Linna Li and Qiongdan Huang

***Abstract*—With the advancement of image authoring software, the traces of image tampering operation have become increasingly difficult to detect. To enhance the performance of manipulation detection, we propose a novel end-to-end image tampering detection method using the attention mechanism and two-stream Faster R-CNN. In our model, we eliminate steganalysis rich model (SRM) filter, which suppresses image content, and employ a constrained convolution layer to adaptively extract image tamper features during model training. These features are then inputted into the backbone network in parallel with the RGB image. To emphasize image tamper features and restrict irrelevant features, we introduce the Convolutional Block Attention Module (CBAM) into the model, enabling better delivery of tamper-related information. Then, the output of backbone network is fed into the Region of Interest (RoI), where we replace max pooling operation with bilinear interpolation. This modification allows the model to retain more tamper information. After that, we use bilinear pooling for feature fusion of the two streams, and the fused results perform tamper classification operations. The RGB stream through Region Proposal Network (RPN) to achieve tamper localization. We evaluate our model on three publicly available standard image tampering datasets and demonstrate through experiments that our approach significantly improves the precision of manipulation detection and localization.**

***Index Terms*—image manipulation detection, Faster R-CNN, constrained convolution, attention.**

I. Introduction

IN recent years, the prevalence of edited images on social networking platforms has increased. While most photos are edited for aesthetic purposes such as life or art photography, there are also instances where images are maliciously altered to deceive viewers. Especially when such image manipulations are employed in political, military, cultural, economic, educational, and other domains, they can have serious consequences and even jeopardize social stability. Therefore, the detection of image manipulation has garnered significant attention.

Among various image manipulation techniques, image content tampering is the most common, involving alterations to certain areas or objects in images. Splicing (copying a part of an image and pasting it onto host image) [1], copy-move (duplicating a part of an image and pasting it into another region) [2], and removal (eliminating a region from an image) are the three most common image tampering operations. These manipulations are shown in Fig. 1. Additionally, after tampering, noise or JPEG compression [3] is sometimes added to further mask traces of manipulation, which poses greater challenges for image manipulation detection research.

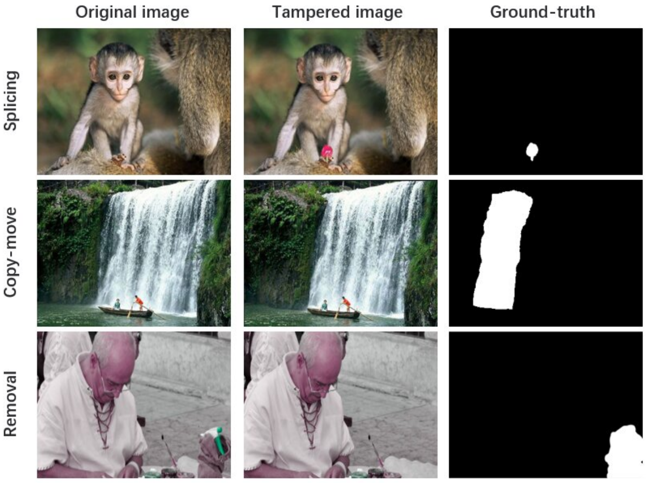

Fig. 1. Examples of different tampering methods.

Previously, image manipulation detection mainly focus on methods such as overlapping image patches and key points. In a typical study, Bianchi [5] used a probability model to predict the Discrete Cosine Transform (DCT) [6] and quantization factors of different areas in a JPEG image. The method then determined the probability of tempering in each DCT block. However, the precision of approach varied for different manipulation operations, and the robustness fell short of predetermined requirements. Recently, image manipulation detection algorithms have primarily concentrated on features related to image compression, edge inconsistency [4], local noise and camera filter array models. With rapid advancement of deep learning, research on image forgery detection has become more diversified. Among these methods, Faster R-CNN [7] has demonstrated promising performance in detecting image manipulation. The first stream utilizes the RGB stream to extract image features, such as tamper boundary inconsistency and contrast difference. The second stream employs SRM [8] filter to extract features related to local noise and detect noise inconsistency within tamper regions. However, the use of manual or predetermined features, such as the SRM filter and edge detection, limits the model's ability to generalize. Traditional convolutional neural networks (CNNs) typically extract features from image content, yet image tamper detection is to recognize and learn features that can effectively capture tampering traces. Given the aforementioned limitations, we have designed a model

Manuscript received April 17, 2023; revised September 6, 2023.

This work is supported by Xi'an Science and Technology Plan Project (Grant No. 22GXFW0124) and The Youth Innovation Team of Shaanxi Universities.

Kang Tan is a postgraduate student of the School of Control Science and Engineering, Xi'an University of Post and Telecommunications, Xi'an 710121, China. (e-mail: tankang0912@163.com).

Linna Li is a Senior Engineer of the Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Post and Telecommunications, Xi'an 710121, China. (e-mail: lilinna0808@163.com).

Qiongdan Huang is an Associate Professor of the School of Communication and Information Engineering, Xi'an University of Post and Telecommunications, Xi'an 710121, China. (e-mail: xuezhemail@163.com).

that combines a constrained CNN and attention module, enabling the adaptive extraction of tampering features during the training process.

In this research, we propose a tampering detection model based on two-stream Faster R-CNN, incorporating a constrained convolutional network and an attention module. The original model obtains image local noise features using specific SRM kernels, but this fixed set of features limits the model's ability to extract tampering information. To address this limitation, we introduce constrained convolution operations [9], which restrict the content of the image and enable the model to self-adaptively learn manipulation features. This approach avoids the problem of poor generalization caused by predetermined features. Furthermore, to better utilize the most relevant feature information, we incorporate the CBAM into the feature extraction network. The CBAM highlights the characteristic information of manipulation while suppressing irrelevant information. In the initial model, the max pooling operation is unbalanced in terms of retaining information from the image target and background, often ignoring background tampering features. To overcome this problem, we use bilinear interpolation instead of max pooling in the RoI. This modification prevents the model from disregarding excessive tampering information and improves the performance of image manipulation detection. The key contributions of this study are as follows:

(1) A constrained convolutional layer is added to two-stream Faster R-CNN to obtain tamper characteristics from the image, which is inputted to the model in parallel with RGB image for end-to-end training. Our model demonstrates different levels of performance improvement compared to the noise and edge streams on three standard datasets.

(2) We incorporate the CBAM [10] into the backbone network, serving as a simple and effective module to enhance expressiveness of the model. The CBAM enables model to effectively identify image tampering information without significantly increasing the computational burden.

(3) We conduct experiments on three publicly available standard image tampering datasets, the results show that our model is better than some existing algorithms. Moreover, our model also shows excellent robustness.

The article is structured as follows: Section 2 summarizes the research on image tampering detection at home and abroad. Section 3 introduces the construction of this method and provides a detailed description. Section 4 presents experimental results and analysis of the model. Finally, Section 5 summarizes the study and discusses future prospects.

II. Related Works

This section summarizes existing methods of image manipulation detection. The main categories are traditional methods and deep learning methods.

A. Traditional Algorithms

Traditional image manipulation detection approaches often relied on designing manual features specific to certain tampering types or leveraging statistical characteristics of images. For example, Lyu [11] devised a model that detected region stitching by identifying inconsistencies in local noise. The algorithm took advantage of the specific regularity of natural image kurtosis in the bandpass domain and noise signature versus kurtosis. Rhayma [12] explored self-certification of JPEG2000 images using a semi-fragile watermark method, where the watermark is generated by the perceptual hash function based on the discrete wavelet coefficients. During compression, the discrete wavelet transform used exponentially modulated quantization, which was stepped by the approximate subband coefficients of the five wavelet decomposition. The watermark was embedded and extracted during image decoding to ensure stability of watermark anti-compression effects generated by the JPEG2000 encoder. In 2017, Li [13] proposed a combined model incorporating statistical features and copy-move manipulation detection, using a threshold to identify tampered areas. This method enhanced precision of detection but also increased complexity. Wang [14] introduced a copy-move tamper detection using accelerated robust features and polar complex index transform, which effectively eliminated false matching points. The model had a low complexity of calculation and high precision for the detection of tampered areas. However, these methods were limited to detecting specific tampering types, and their performance on multi-tampered images was less favorable. Moreover, these approaches often involved complex and specific post-processing steps, which increased computational complexity and challenged the stability of the model.

B. Deep Learning Algorithms

In the development of manipulation detection algorithms using deep learning, CNNs have garnered significant attention. To address the limitations of traditional image tampering detection methods in terms of robustness, Bayar [15] proposed a general image manipulation detection method called constrained convolution neural network. This model could be trained to adaptively extract tampering features and effectively suppress image content. Zhou [7] designed a two-stream Faster R-CNN image tampering detection method, which enabled classification and localization of tampering images by simultaneously inputting noise stream features and RGB stream features into the model. Wei [16] developed an image tamper detection method using Faster R-CNN and edge processing. This approach involved extracting image edge features using Laplacian of Gaussian and Prewitt operators. Additionally, bilinear interpolation was employed to replace the max pooling of RoI to prevent the model from solely extracting high-frequency information. To mitigate the influence of image content on the acquisition of tampering features, Yang [9] introduced the constrained convolution layer to preprocess images. This ensured that the CNN paid more attention to extracting tampering features. Despite the success achieved by the two-stream Faster R-CNN in image tampering detection, the manual selection of the noise stream limited the detection performance of image tampering. In 2021, Chen [17] utilized a two-layer convolutional network to model weak features arising from image tampering. They established the weak feature backbone network using a multi-scale residual network to acquire tampering feature from tampered images.

In the aforementioned noise stream and edge detection stream, specific kernels are employed to extract features. In

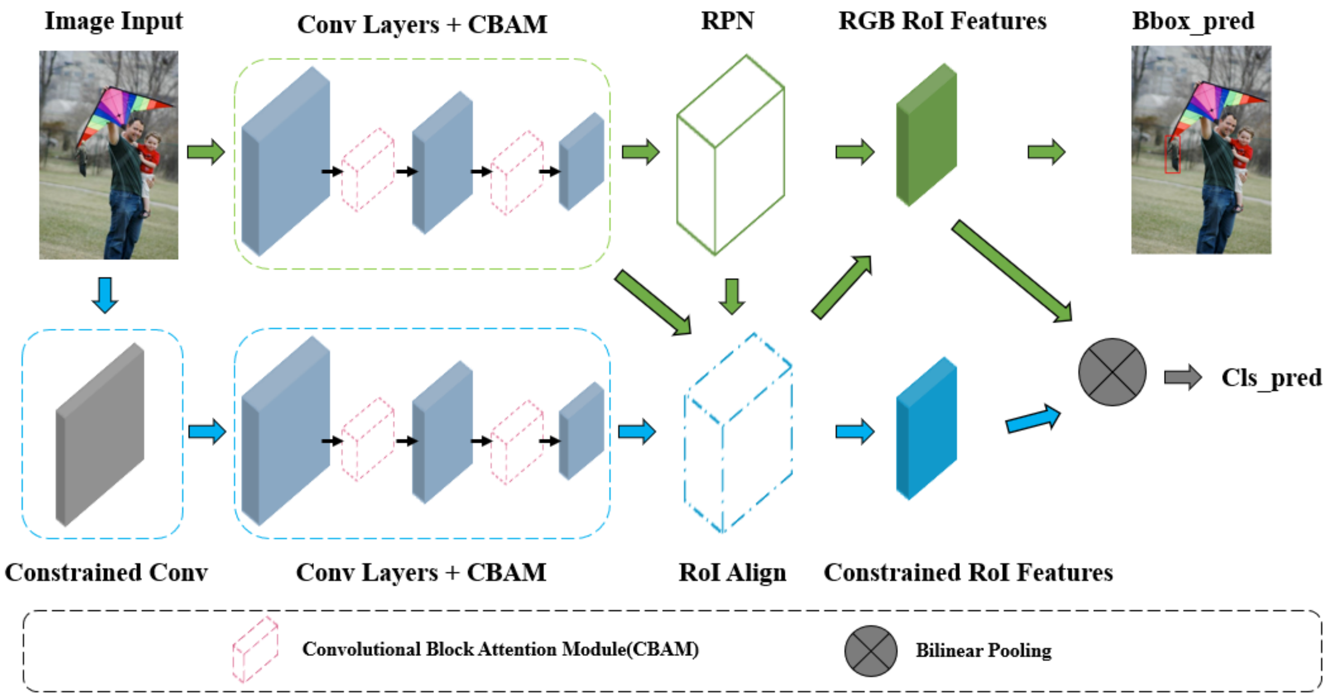

Fig. 2. An overview of image tamper detection model using Faster R-CNN.

this study, we remove manual features and instead utilize a constrained convolutional layer to adaptively extract tampering features. Furthermore, we replace max pooling with bilinear interpolation to maximize the utilization of higher resolution feature layers. Additionally, we introduce a simple, efficient, and lightweight attention module. This attention module enables our model to focus on image tampering features, leading to an overall improvement in performance without significantly increasing computational requirements.

III. Proposed Method

The Faster R-CNN model consists of feature extraction, RPN, and RoI pooling. In this paper, we introduce the constrained convolution layer and attention mechanism to enable manipulation classification and boundary box regression [18] within the Faster R-CNN framework. Fig. 2 illustrates the module utilizing Faster R-CNN. The RGB stream is utilized to detect edge inconsistencies introduced by tampering. The constrained convolution layer replaces the traditional SRM and adaptively extracts image tampering features, providing additional support for tampering classification. For feature extraction, we employ the residual network (ResNet) [19] as the feature extraction network and enhance it with CBAM to better capture tampering features. The RoI layer uses bilinear interpolation to retain more manipulation information, namely RoI Align [20]. Subsequently, bilinear pooling is employed to combine the characteristics of the two streams for the final classification. The confidence level is utilized to ascertain whether an image has undergone tampering, and the tampered area is selected for bounding box localization. The RPN utilizes only the RGB stream as input. The role of region proposal layer is to identify regions that may have been tampered with for subsequent manipulation localization. Anchors are employed to generate region candidate boxes and perform preliminary screening. The RPN network loss function is shown as follows:

$$\begin{aligned} L_{RPN}(p_i,t_i) &= \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,{p_i}^*) \\ &+ \lambda\frac{1}{N_{reg}}\sum_i {p_i}^* L_{reg}(t_i,{t_i}^*) \end{aligned} \tag{1}$$

Where, i represents each anchor point. p_i represents the probabilistic prediction of the region containing tampering for each anchor, while ${p_i}^*$ represents the corresponding ground-truth label for each anchor. t_i and ${t_i}^*$ represent the described values and the true values, respectively, for the boundary box of each anchor. L_{cls} represents cross-entropy loss of the RPN, N_{cls} denotes mini-batch size. L_{reg} represents the smoothing loss. N_{reg} represents total number of anchor positions. λ represents equilibrium parameter ($\lambda = 10$).

The loss of entire model include three components:

$$L_t = L_{RPN} + L_{tamper}(f_{RGB}, f_C) + L_{bbox}(f_{RGB}) \tag{2}$$

Where, L_t represents the overall loss of the entire model. L_{RPN} corresponds to the loss function of the RPN, while L_{tamper} represents subsequent cross-entropy classification loss. The loss is influenced by both the RGB channel features f_{RGB} and the constrained channel features f_C. On the other hand, L_{bbox} denotes the loss for boundary box regression loss, which is solely determined by the RGB channel features f_{RGB}.

A. Add Constrained Convolution Layer

Due to the post-processing techniques employed to conceal tampering features, it becomes challenging to detect subtle manipulation marks solely from the RGB stream. Traditional methods often rely on SRM convolution kernels and edge

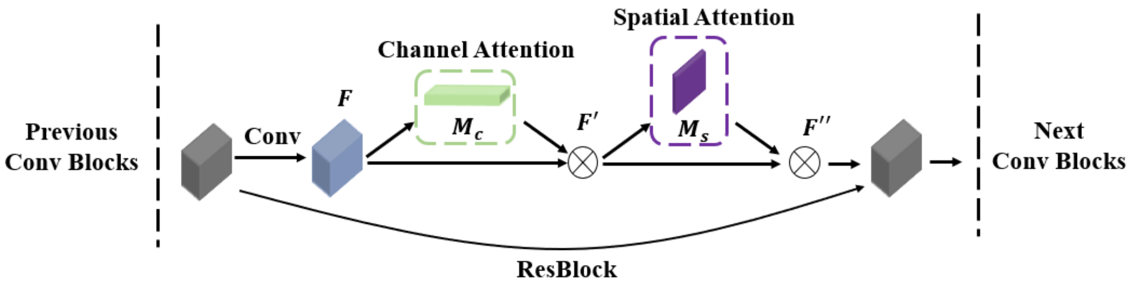

Fig. 3. CBAM and feature extraction network structure. "⊗" denotes element-wise multiplication.

detection techniques to identify tampering clues such as noise inconsistencies and edge differences. However, these manual methods are limited by their reliance on fixed convolution kernels for directly extracting tampering features from the tampered image. By applying constrained convolution layer to the images, we can extract rich tampering cues. The output features are fed into the ResNet. The constraint is described as follows:

$$\begin{cases} z_k(0,0) = -1 \\ \sum_{x,y\neq 0} z_k(x,y) = 1 \end{cases} \tag{3}$$

Where, $z_k(0,0)$ represents weight of the k^{th} convolution kernel at the center position, while $z_k(x,y)$ denotes the weight of the k^{th} convolution kernel at the x and y position. The initial weights of each convolution kernel are randomly selected, and the weights are subsequently updated through an iterative stochastic gradient descent method.

The three fixed filter kernels of SRM are replaced by constrained convolution layer with the size of 5×5 convolution kernels. Different from the Constrained R-CNN (CR-CNN) [9], we continue to employ the two streams model for manipulation detection. The RGB stream is utilized to capture boundary inconsistencies within the tampered region, while the constrained convolution provides an auxiliary role in analyzing local noise features of the image.

B. Add Convolutional Block Attention Module

Attention mechanisms [27] have gained significant attention and have been widely studied and applied in various tasks, including target detection. These mechanisms not only help determine where the focus of attention should be but also aid in adjusting the expression of characteristic information. In our study, we employ the CBAM framework to enhance the feature acquisition network. This allows us to concentrate on relevant tampering features while suppressing unnecessary ones. The CBAM framework consists of two components: channel attention module and spatial attention module. The specific placement of the CBAM model within convolutional block is depicted in Fig. 3.

Specifically, we input the feature maps $\mathbf{F}$ of size $M \times N \times C$ and obtain information of each channel through average pooling and max pooling. This result in $\mathbf{F_{avg}} \in \mathbb{R}^{1\times1\times C}$ and $\mathbf{F_{max}} \in \mathbb{R}^{1\times1\times C}$ after pooling operation. These pooled feature maps are then passed through a Multi-Layer Perceptron (MLP) is made up of two fully-connected layers. The output of the first layer represents $\frac{C}{r}$, where r represents the compression coefficient. After applying the ReLu, the output of second layer represents channel number as C. The output of the two fully-connected layers are added together and passed through a sigmoid function, resulting in $\mathbf{M_c}$. The spatial attention module takes the modified feature maps $\mathbf{F}^{'}$ as input, which is obtained by element-wise multiplication of the output $\mathbf{M_c}$ from channel attention module and original input feature maps. We perform two pooling operations, resulting in feature maps $\mathbf{F^s_{avg}}$ and $\mathbf{F^s_{max}}$ with reduced dimensions. Subsequently, a 7×7 convolution operation is applied, followed by sigmoid activation function, resulting in a spatial attention output. Finally, the CBAM framework produces the final output $\mathbf{F}^{''}$ by element-wise multiplication of the input features and output of spatial attention module. The specific formulas are shown as follows:

$$\begin{aligned} \mathbf{M_c}(\mathbf{F}) &= \sigma(MLP(avgpool(\mathbf{F})) + MLP(maxpool(\mathbf{F}))) \\ &= \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}}))) \end{aligned} \tag{4}$$

In the equation, $\mathbf{M_c}$ represents the output of channel attention module. σ denotes the Sigmoid. $\mathbf{W_0} \in \mathbb{R}^{\frac{C}{r}\times C}$, $\mathbf{W_1} \in \mathbb{R}^{C\times\frac{C}{r}}$ are weight matrices, where $\mathbf{W_0}$ and $\mathbf{W_1}$ share weights.

$$\begin{aligned} \mathbf{M_s}(\mathbf{F}) &= \sigma(f^{7\times7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})]) \\ &= \sigma(f^{7\times7}([\mathbf{F^s_{avg}}; \mathbf{F^s_{max}}])) \end{aligned} \tag{5}$$

Where, $\mathbf{M_s}$ represents the output of the spatial attention module. $f^{7\times7}$ denotes the convolution processing core with a size of 7×7.

We incorporate the CBAM into the output of each convolutional block in our approach. The placement of CBAM is determined based on the experimental results we obtain. Through rigorous analysis, we observe that adding the CBAM module in both streams yields better results compared to adding it in either stream alone.

C. Improved RoI Pooling

To mitigate the quantization effects in feature assemblies and maintain continuity, we adopt the bilinear interpolation method. This method allows us to handle pixels with floating-point coordinate values instead of quantizing them. In the case of Faster R-CNN, the model has two quantification processes: (x, y, w, h) of region proposal is a decimal number. To facilitate calculation, these values are usually quantized to integers. The processed image boundaries are evenly divided into $n \times n$ units, and the boundaries of the generated units are integer. However, this quantization process introduces a bias between the candidate box positions and the original regression. Such integer processing can negatively impact the accuracy of object detection. To address this issue, we introduce RoI Align, which avoids integer operations and preserves the decimal values. The improved process can be summarized as follows:

(1) For each candidate area in the image, the boundaries of the region use non-integer processed.

(2) Each candidate area is divided into a grid of $n \times n$ blocks, and the edge of these blocks are non-integer processed.

(3) Bilinear interpolation is utilized to calculate four fixed coordinate positions for each block, and pooling operations are applied to obtain the maximum values.

Through experimental comparisons, we demonstrate that the improved RoI pooling method improves the precision of our model. We evaluate the impact of max pooling and bilinear interpolation on the detection performance and compare it with two other models using the same methods. The results are shown in Table I. The use of our proposed model is denoted by "✓", while its absence is indicated by "×". Based on the experimental findings, it can be conclude that our model demonstrates superior detection performance under similar conditions.

TABLE I
COMPARISON OF ROI POOLING AND IMPROVED ROI POOLING MAP

Model	Max pooling	mAP
RGB-N	✓	0.7645
RGB-N	×	0.7837
RGB-N+Edge	✓	0.7459
RGB-N+Edge	×	0.8100
Ours	✓	0.8661
Ours	×	0.8873

IV. EXPERIMENTS

We evaluate our method on three standard datasets and compare it with various manipulation detection methods. The evaluation metrics employed in this study include Recall, Precision, F1 score and area under the curve (AUC).

A. Pre-Trained Model

Currently, availability of publicly accessible datasets for image manipulation detection is limited. In this study, we generate a tamper dataset by synthesizing label information from PASCAL VOC 2007. A total of 9963 tamper images are created for model training and testing purposes. To further expand the dataset, we employ image flipping as a data augmentation technique, resulting in a total of 15900 tamper images. The distribution of the final pre-trained model dataset is presented in Table II.

During training process, the parameters of our model are set to be consistent with those of the RGB-N model. We use 8, 16, 32, and 64 as the dimensions of the anchors, with aspect ratios of 1:2, 1:1, and 2:1. The initial learning rate is set to 0.001 and later changed to 0.0001 after 40k training iterations. The model's max number of training iterations is set to 60k. To reduce overlapping regions, we employ a non-maximum suppression method with a specific threshold set to 0.5. The output of pre-training model consists of a bounding box along with corresponding confidence values, indicating areas that have been tampered with.

TABLE II
TRAINING AND TESTING DIVISION OF DATASET

Dataset	Types	Number
PASCAL VOC 2007	Training	15000
	Testing	900

B. Standard Datasets

This paper conducts experiments and analysis using three available standard datasets. These datasets are as follows:

(1) CASIA dataset: This dataset consists of two types of tampered images, namely mosaic and copy-move. The images undergo post-processing operations such as filtering and blurring. The dataset is divided into CASIA 1.0 and CASIA 2.0. CASIA 1.0 is used for testing the model, while CASIA 2.0 is used for training.

(2) Columbia dataset [21]: The dataset provides ground-truth masks for uncompressed concatenation operations.

(3) NIST16 dataset [22]: The dataset contains three types of manipulation: splicing, copy-move, and removal. The images in this dataset have undergone post-processing operations to hide tampering, and it also provides ground-truth masks.

The training and testing for three standard datasets are divided as shown in Table III. The symbol "\" indicates that there is no raw data information available.

TABLE III
TRAINING AND TESTING DIVISION OF THREE DATASETS

Datasets	NIST16	Columbia	CASIA
Training	404	\	5123
Testing	106	180	921

To overcome the limited number of datasets, we employ data augmentation techniques to expand our dataset. This allows the trained model to exhibit improved detection performance. Specifically, we utilize image flipping as a form of data augmentation. Additionally, we compare performance of model using both flipping and no flipping datasets in terms of Recall, Precision, and F1 score. The results demonstrate that the use of image flipping significantly enhances the method's performance, as shown in Table IV.

C. Results and Analysis

We evaluate performance of our method on the standard datasets using various metrics, including Recall, Precision, F1 score, and AUC. In order to assess the manipulation detection performance of this model, we compare it with

TABLE IV
COMPARISON OF IMAGE FLIPPING RESULTS

Datasets	Flipping	Recall	Precision	F1 score
NIST16	×	0.9632	0.9654	0.9564
NIST16	✓	0.9701	0.9699	0.9637
Columbia	×	0.8011	0.7973	0.8225
Columbia	✓	0.8023	0.7978	0.8209
CASIA	×	0.6766	0.5012	0.6742
CASIA	✓	0.6913	0.5101	0.6958

other existing algorithms, namely BLK [23], CFA1 [24], RGB-N [7], and RGB-N+Edge [16]. The results of these comparisons are shown in Table V, VI and VII.

TABLE V
F1 SCORE COMPARISON OF THE METHODS

Method	NIST16	Columbia	CASIA
BLK	0.3019	0.5234	0.2312
CFA1	0.1743	0.4667	0.2073
RGB-N	0.7220	0.6970	0.5457
RGN-N+Edge	0.9533	0.7514	0.5794
Ours	0.9637	0.8225	0.6958

TABLE VI
RECALL COMPARISON OF THE METHODS

Method	NIST16	Columbia	CASIA
BLK	0.2562	0.4500	0.1705
CFA1	0.1500	0.6278	0.1857
RGB-N	0.9437	0.7944	0.6515
RGN-N+Edge	0.9563	0.7389	0.6608
Ours	0.9701	0.8023	0.6913

TABLE VII
PRECISION COMPARISON OF THE METHODS

Method	NIST16	Columbia	CASIA
BLK	0.3674	0.6254	0.3590
CFA1	0.2080	0.3714	0.2346
RGB-N	0.8830	0.7044	0.5096
RGN-N+Edge	0.9503	0.7644	0.5158
Ours	0.9699	0.7978	0.5101

Based on the results presented in the above Table, our model demonstrates superior performance compared to other image manipulation detection methods. However, in Table VI, the accuracy rate of our manipulation detection on the CASIA dataset is slightly lower than that of edge detection stream. This can be likely due to the fact that tampered images in the CASIA dataset undergo blurring and filtering. This results in a decrease in image resolution. As a result, the actual performance of our model is slightly affected.

To further validate the effectiveness of our model, we employ the AUC evaluation metric to compare it with other available manipulation detection algorithms. This comparison is presented in Table VIII.

TABLE VIII
COMPARISON OF AUC SCORES FOR MANIPULATION DETECTION BY DIFFERENT MODELS

Method	NIST16	Columbia	CASIA
RGB-N	0.9370	0.8580	0.7950
CR-CNN	0.9920	0.8610	0.7890
Fals-Unet	0.8325	\	0.8463
SPAN	0.9610	0.9360	0.8380
Ours	0.9693	0.9461	0.8491

Through comparative experiments, we demonstrate that our model, after incorporating the constrained convolutional layer and CBAM, surpasses existing methods in terms of detection performance on the NIST16 and Columbia datasets. Furthermore, compared to the Spatial Pyramid Attention Network (SPAN) [26], our model achieves comparable or slightly higher detection performance on the CASIA dataset. It is worth highlighting that on the NIST16 dataset, the CR-CNN algorithm achieves a higher AUC index, surpassing our model by 2%. This is mainly attributed to the fact that CR-CNN incorporates both low-level and high-level features, refining tamper detection process. The AUC data for the comparison models in the Table VIII are sourced from [24], [25], [26].

D. Robustness

To assess the robustness of our algorithm, we conducted experiments on the CASIA dataset. It is compared with the detection performance of RGB-N and RGB-N+Edge methods under various attacks. We use F1 score for evaluation. The attacks employed in these experiments primarily involve Gaussian white noise (mean value of 0, the variances of 5, 10, 15), JPEG compression (quality factors of 85, 70, 55) and Gaussian blurring (filter size 3×3 , variances are set to 0.5, 1.0, 1.5).

As depicted in Fig. 4, experimental results show that the method has better robustness than the other two methods. From left to right in the Fig. 4, we have Gaussian white noise, JPEG compression, and Gaussian blurring. As the noise variance increases, our method has a slight decrease. It is superior to RGB-N and RGB-N+Edge in noise resistance. It is worth noting that under JPEG compression, our method experiences a minimal and gradual decline. Our algorithm score only decreases by 1.56%. Additionally, under Gaussian blurring, our method exhibits a larger decline compared to the previous two attacks but still outperforms several existing methods. We conclude that the addition of Gaussian blurring has a slightly more pronounced effect on our model.

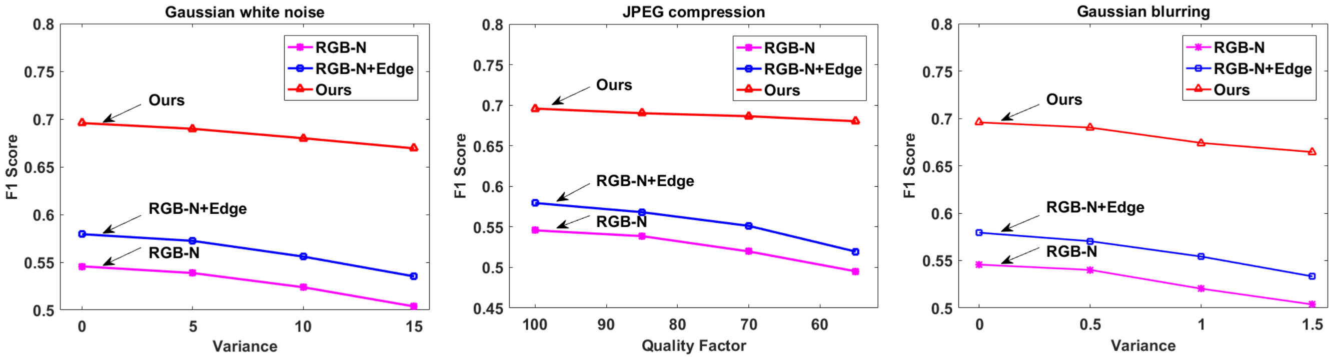

Fig. 4. Comparison of F1 score of three algorithms under different attacks.

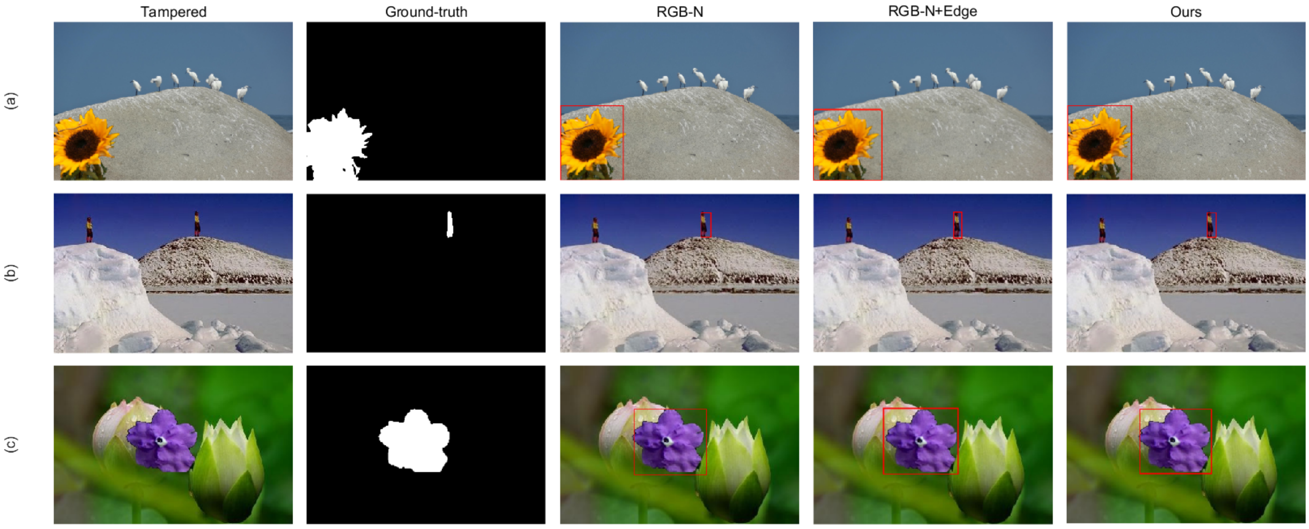

Fig. 5. Comparison of manipulation detection localization results.

E. Localization of the Tampered Region

In order to assess localization performance of the method, we conduct experiments and compare them with the original RGB-N and RGB-N+Edge methods using publicly available standard datasets. As shown in Fig. 5, our model demonstrates more accurate localization results compared to other model.

Specifically, in Fig. 5, we observe that for tampered images (a) and (c), the positioning effect of the RGB-N method tends to be biased towards the right. Moreover, the selected manipulation region is not enclosed within a complete bounding box. The RGB-N+Edge method, on the other hand, successfully locates the tampered region but also includes irrelevant areas, resulting in suboptimal positioning results. In contrast, our model accurately selects the manipulation region with a complete bounding box, adjusting its size to closely match the ground-truth mask. Furthermore, for image (b) involving copy-move tampering, our model outperforms both RGB-N and RGB-N+Edge in positioning accuracy. While the original RGB-N method selects the manipulation region, it includes an additional part in the bounding box to the right. In contrast, our model selects a more reasonable positioning area by reducing the frame size while maintaining proper boundary framing. Overall, our propose model demonstrates substantial improvements in localization of tampered region compared to the existing methods.

V. Conclusion

In this study, we propose an image manipulation detection method using two-stream Faster R-CNN architecture. To enhance model's ability to learn tampering features adaptively and overcome limitations of manual specified features, we introduce a constrained convolution layer in the preprocessing stage. Additionally, we incorporate CBAM to improve the attention towards tampered regions, allowing the model to focus on a larger manipulation area within the image. Furthermore, we employ bilinear interpolation at the RoI layer to increase the size of the RoI, enabling the model to retain more tampering information. Through experiments on three standard image tampering datasets, the method demonstrates preferable performance in both manipulation classification and localization tasks. At present, multi-scale feature fusion has shown remarkable performance in image manipulation detection. We intend to further investigate and study in our future work.

References

[1] H.W. Ding, L.Y. Chen, Q. Tan, Z.W. Fu, L. Dong and X.H. Cui, "DCU-Net: a dual-channel U-shaped network for image splicing forgery detection," *Neural Computing and Applications*, vol. 35, no. 7, pp. 5015-5031, 2023.

[2] B. Soni, P.K. Das and D.M. Thounaojam, “Dual System for Copy-move Forgery Detection using Block-based LBP-HF and FWHT Features,” *Engineering Letters*, vol. 26, no. 1, pp. 171-180, 2018.

[3] V. Holub and J. Fridrich, “Low-Complexity Features for JPEG Steganalysis Using Undecimated DCT,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219-228, 2015.

[4] K. Asghar, X.F. Sun, P.L. Rosin, M. Saddique, M. Hussain and Z. Habib, “Edge-texture feature-based image forgery detection with cross-dataset evaluation,” *Machine Vision and Applications*, vol. 30, no. 7-8, pp. 1243-1262, 2019.

[5] T. Bianchi, A. De Rosa and A. Piva, “Improved DCT coefficient analysis for forgery localization in JPEG images,” 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011, 22-27 May 2011, Prague, Czech Republic, pp. 2444-2447.

[6] W. Ahn, S.H. Nam, M. Son, H.K. Lee and S. Choi, “End-to-end double JPEG detection with a 3D convolutional network in the DCT domain,” *Electronics Letters*, vol. 56, no. 2, pp. 82-84, 2020.

[7] P. Zhou, X.T. Han, V.I. Morariu and L.S. Davis, “Learning Rich Features for Image Manipulation Detection,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018*, CVPR 2018, 18-23 June 2018, Salt Lake City, UT, USA, pp. 1053-1061.

[8] 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.

[9] C. Yang, H.Z. Li, F.T. Lin, B. Jiang and H. Zhao, “Constrained R-CNN: A General Image Manipulation Detection Model,” *IEEE International Conference on Multimedia and Expo 2020*, ICME 2020, London, UK, pp. 1-6.

[10] S. Woo, J. Park, J.Y. Lee, I.S. Kweon, “CBAM: Convolutional Block Attention Module,” *European Conference on Computer Vision 2018*, ECCV 2018, vol 11211, pp. 3-19.

[11] S.W. Lyu, X.Y. Pan and X. Zhang, “Exposing Region Splicing Forgeries with Blind Local Noise Estimation,” *International Journal of Computer Vision*, vol. 110, no. 2, pp. 202-221, 2014.

[12] H. Rhayma, A. Makhloufi, H. Hamam and A.B. Hamida, “Semi-fragile watermarking scheme based on perceptual hash function (PHF) for image tampering detection,” *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26813-26832, 2021.

[13] H.D. Li, W.Q. Luo, X.Q. Qiu and J.W. Huang, “Image Forgery Localization via Integrating Tampering Possibility Maps,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1240-1252, 2017.

[14] C.Y. Wang, Z. Zhang, Q.W. Li and X. Zhou, “An Image Copy-Move Forgery Detection Method Based on SURF and PCET,” *IEEE Access*, vol. 7, pp. 170032-170047, 2019.

[15] 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.

[16] X.Y. Wei, Y.R. Wu, F.M. Dong, J. Zhang and S.F. Sun, “Developing an Image Manipulation Detection Algorithm Based on Edge Detection and Faster R-CNN,” *Symmetry*, vol. 11, no. 10, pp. 1223, 2019.

[17] H. Chen, Q. Han, Q. Li, X.J. Tong, “Digital image manipulation detection with weak feature stream,” *The Visual Computer*, vol. 38, no. 8, pp. 2675-2689, 2022.

[18] K.S. Sim, F.F. Ting, J.W. Leong, and C.P. Tso, “Signal-to-noise Ratio Estimation for SEM Single Image using Cubic Spline Interpolation with Linear Least Square Regression,” *Engineering Letters*, vol. 27, no. 1, pp. 151-165, 2019.

[19] B.Q. Li and Y.Y. He, “An Improved ResNet Based on the Adjustable Shortcut Connections,” *IEEE Access*, vol. 6, pp. 18967-18974, 2018.

[20] X.Y. Wang, H. Wang, S.Z. Niu and J.W. Zhang, “Detection and localization of image forgeries using improved mask regional convolutional neural network,” *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 4581-4593, 2019.

[21] “Columbia image splicing detection evaluation dataset,” 2022. [Online]. Available: http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm,

[22] “Nist Nimble 2016 Datasets,” 2022. [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation.2018.

[23] W.H. Li, Y. Y and N.H. Y, “Passive detection of doctored JPEG image via block artifact grid extraction,” *Signal Processing*, vol. 89, no. 9, pp. 1821-1829, 2009.

[24] P. Ferrara, T. Bianchi, A.D. 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.

[25] F.Z. El Biach, I. Lala, H. Laanaya and K. Minaoui, “Encoder-decoder based convolutional neural networks for image forgery detection,” *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 22611-22628, 2021.

[26] X.F. Hu, Z.H. Zhang, Z.Y. Jiang, S. Chaudhuri, Z.H. Yang and R. Nevatia, “SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization,” *European Conference on Computer Vision 2020*, ECCV 2020, pp. 312–328.

[27] Z.F. Hu, W.H. Wang, K.L. Zhu, H.Y. Zhou and J.T Chen, “Loop Closure Detection Algorithm Based on Attention Mechanism,” *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 592-598, 2023.