TBFormer: Two-Branch Transformer for Image Forgery Localization

Yaqi Liu, Binbin Lv, Xin Jin, Xiaoyu Chen, and Xiaokun Zhang

Abstract-Image forgery localization aims to identify forged regions by capturing subtle traces from high-quality discriminative features. In this paper, we propose a Transformer-style network with two feature extraction branches for image forgery localization, and it is named as Two-Branch Transformer (TBFormer). Firstly, two feature extraction branches are elaborately designed, taking advantage of the discriminative stacked Transformer layers, for both RGB and noise domain features. Secondly, an Attention-aware Hierarchical-feature Fusion Module (AHFM) is proposed to effectively fuse hierarchical features from two different domains. Although the two feature extraction branches have the same architecture, their features have significant differences since they are extracted from different domains. We adopt position attention to embed them into a unified feature domain for hierarchical feature investigation. Finally, a Transformer decoder is constructed for feature reconstruction to generate the predicted mask. Extensive experiments on publicly available datasets demonstrate the effectiveness of the proposed model.

Index Terms—Image forgery localization, two-branch, Transformer, hierarchical-feature fusion.

I. INTRODUCTION

E DITING digital images may change the semantic content of original images, and the edited images are often too realistic to distinguish their authenticity. It poses a threat to the stability and harmony of the society if they are used illegally. Image forgery localization is a kind of image forensics task which aims at locating forged regions in investigated images, and it has attracted more and more attention in both research and industry [1]-[2].

Researchers have proposed many image forgery localization methods for specific forgery types, e.g., splicing [3]-[7], copymove [8]-[12], and removal [13]-[14]. In practice, the investigated image may contain multiple forgery types at the same time [15]-[16]. Some researchers [15]-[19] have also proposed methods applicable to multiple forgery types, while many of these methods extract features from the RGB domain [16]. Some researchers [14]-[15], [20]-[24] have also attempted to combine features extracted from different domains. Wu et al. [23] and Hu et al. [15] concated RGB image and its

Xiaoyu Chen is with the China North Industries Group Corporation Limited, Beijing 100089, China. corresponding noise map before the feature extractor. Zhou et al. [24] and Chen et al. [20] designed two parallel branches to extract RGB features and noise features. While the abovementioned methods are constructed based on convolutional neural networks.

In recent years, Transformer has been widely used in various vision tasks, e.g., object detection [25]-[27] and image segmentation [28]-[30], showing superior performance. Researchers also tried to apply Transformer to image forgery localization. Wang et al. [31] designed a multimodal Transformer framework. Instead of using images directly as the input, they used convolutional layers to extract feature maps for patch embedding. Sun et al. [32] adopted multiple Transformer layers to extract features only from the RGB domain, and constructed a convolutional decoder.

In this paper, we propose a Transformer-style image forgery localization network, namely TBFormer, and the architecture is shown in Fig. 1. The noise domain contains subtle forgery traces, which are visually invisible and difficult to capture from the RGB domain. Therefore, we develop two feature extraction branches with multiple Transformer layers to extract discriminative features from the RGB domain and the noise domain independently. The two branches have the same architecture and their weights are not shared. The consideration is that Transformer layers are powerful for discriminative feature representation, and the non-shared design makes them concentrate on their specific domains. However, the nonshared feature extraction branches provide feature maps from different domains with large differences. How to fuse these feature maps becomes a key problem. Thus, we design an Attention-aware Hierarchical-feature Fusion Module (AHFM) to effectively fuse hierarchical features from two different domains. RGB features and noise features from the same layer are gone through a position attention module, to integrate them into a unified feature domain. Then hierarchical features are combined by element-wise addition and a convolutional layer to get the final fused feature map with rich hierarchical information from both RGB and noise domains. Finally, we design a Transformer decoder to reconstruct the fused features and provide the predicted mask. Category embeddings are set in the decoder to further learn unified feature representations of authentic and forged classes, and they are interacted with fused feature map patch embeddings to produce the predicted masks. Last but not least, in order to train and test our Transformerstyle network, we generate a synthesized image dataset with 140432 images for training, 7787 images for validating, and 7787 images for testing. The synthesized dataset is made publicly available for further research.

Manuscript received February 25, 2023. This work was supported in part by the NSFC under Grant 62102010; in part by the Advanced Discipline Construction Project of Beijing Universities under Grant 20210037Z0401; and in part by the Information Center Project of China North Industries Group Corporation Limited under Grant 20220100H0113. (*Corresponding author: Xin Jin.*)

Yaqi Liu, Binbin Lv, Xin Jin, and Xiaokun Zhang are with the Beijing Electronic Science and Technology Institute, Beijing 100070, China (e-mail: liuyaqi@besti.edu.cn; lv-bin-bin@outlook.com; jinxin@besti.edu.cn; sam@besti.edu.cn).



Fig. 1. Overview of the proposed TBFormer. TBFormer consists of two feature extraction branches, an AHFM module, and a Transformer decoder.

The main contributions of this paper can be summarized as follows: (1) A novel Transformer-style network (TBFormer) with two feature extraction branches is proposed for image forgery localization. (2) An Attention-aware Hierarchicalfeature Fusion Module (AHFM) is proposed to effectively fuse hierarchical features from two different domains. (3) A Transformer decoder is constructed for feature reconstruction to generate the predicted mask. (4) All our codes, models and the generated dataset are available online (https://github.com/ free1dom1/TBFormer).

II. PROPOSED METHOD

A. Two-Branch Feature Extractor

To exploit the potential forgery cues in different domains, we design two feature extraction branches to extract discriminative features from the RGB domain and the noise domain. The two branches have the same architecture, and their weights are not shared which makes them concentrate on their specific domains. We adopt BayarConv [33] for converting the RGB domain to the noise domain. Transformer can overcome the shortcomings of convolutional neural networks with only limited receptive fields and has the powerful ability to model contextual global dependencies [34]-[35]. The rich contextual information is also crucial for locating forged regions, so Transformer is adopted for our feature extraction.

The input color RGB image $I_c \in \mathbb{R}^{H \times W \times 3}$ is first converted to the noise map $I_n \in \mathbb{R}^{H \times W \times 3}$ by BayarConv, where W and H denote the width and height of the input image. We divide I_c into image patches of size 16×16 to obtain the sequence $X_c = \left\{ x_c^{(1)}, x_c^{(2)}, \cdots, x_c^{(N)} \right\}$, where $x_c^{(i)} \in \mathbb{R}^{16 \times 16 \times 3}$ and $N = H/16 \times W/16$ is the number of image patches. Each image patch $x_c^{(i)}$ is reshaped into a one-dimensional vector, followed by a linear projection layer to obtain the image patch embedding sequence $P_c = \left\{ p_c^{(1)}, p_c^{(2)}, \cdots, p_c^{(N)} \right\} \in \mathbb{R}^{N \times L}$, where L denotes the feature dimension. The corresponding position embedding $pos_c^{(i)}$ is added to the image patch embedding $p_c^{(i)}$ to obtain the resulting input sequence $E_c = \left\{ e_c^{(1)}, e_c^{(2)}, \cdots, e_c^{(N)} \right\} \in \mathbb{R}^{N \times L}$, where $e_c^{(i)} = p_c^{(i)} + \mathbf{pos}_c^{(i)}$. Then E_c is fed into the feature extractor which is constructed based on 12 Transformer layers.

The feature maps of the 4th, 8th, and 12th layers (i.e., $T_c^{(4)}, T_c^{(8)}, T_c^{(12)}$) are output for further investigation:

$$\boldsymbol{T}_{c} = \left\{ \boldsymbol{T}_{c}^{(4)}, \boldsymbol{T}_{c}^{(8)}, \boldsymbol{T}_{c}^{(12)} \right\} = f_{c} \left(\boldsymbol{E}_{\boldsymbol{c}} \right)$$
(1)

where f_c denotes the feature extractor of the RGB branch. The Transformer layer consists of a Multi-Head Self-Attention (MSA) block and a Multi-Layer Perceptron (MLP) block, and the architecture of the *i*th layer can be represented as:

$$\boldsymbol{M}_{c}^{(i)} = \mathrm{MSA}_{c}^{(i)} \left(\mathrm{LN} \left(\boldsymbol{T}_{c}^{(i-1)} \right) \right) + \boldsymbol{T}_{c}^{(i-1)}$$
(2)

$$\boldsymbol{T}_{c}^{(i)} = \mathrm{MLP}_{c}^{(i)} \left(\mathrm{LN} \left(\boldsymbol{M}_{c}^{(i)} \right) \right) + \boldsymbol{M}_{c}^{(i)}$$
(3)

where LN represents layer norm. The $MSA_c^{(i)}$ block is constituted by the Self-Attention (SA) operation:

$$\mathrm{SA}_{c}^{(i)}\left(\boldsymbol{T}_{c}^{(i-1)}\right) = \mathrm{softmax}\left(\boldsymbol{Q}_{c}^{(i)}\left(\boldsymbol{K}_{c}^{(i)}\right)^{\mathrm{T}}/\sqrt{L}\right)\boldsymbol{V}_{c}^{(i)} \quad (4)$$

where query, key, value are computed as $Q_c^{(i)} = T_c^{(i-1)} W_{cQ}^{(i)}$, $K_c^{(i)} = T_c^{(i-1)} W_{cK}^{(i)}$, $V_c^{(i)} = T_c^{(i-1)} W_{cV}^{(i)}$, and $W_{cQ}^{(i)}$, $W_{cK}^{(i)}$, $W_{cV}^{(i)}$ are the learnable parameters of three linear projection layers in self-attention [36].

The same processes are performed on the noise map I_n to obtain $E_n \in \mathbb{R}^{N \times L}$. The noise features are obtained by feeding E_n into the feature extractor of the noise branch:

$$\boldsymbol{T}_{n} = \left\{ \boldsymbol{T}_{n}^{(4)}, \boldsymbol{T}_{n}^{(8)}, \boldsymbol{T}_{n}^{(12)} \right\} = f_{n} \left(\boldsymbol{E}_{\boldsymbol{n}} \right)$$
(5)

where f_n denotes the feature extractor of the noise branch, $T_n^{(4)}, T_n^{(8)}, T_n^{(12)} \in \mathbb{R}^{N \times L}$ denote the features output by the 4th, 8th, and 12th Transformer layers.

B. Attention-aware Hierarchical-feature Fusion Module

The feature maps of the two branches have significant differences for that they are extracted from different domains. A carefully designed decoder is helpful for mask reconstruction from different domains, and a well-designed feature fusion module is also an indispensable part of a network to investigate multi-domain information. We design an Attention-aware Hierarchical-feature Fusion Module (AHFM) to effectively fuse hierarchical features from two different domains.

For the RGB features and noise features from the same layer, we construct a position attention block [37]



Fig. 2. Computational procedure of Position Attention.

to investigate their correlation and fuse them into unified feature maps. Taking the 4th-layer features as examples, the matrixes $T_c^{(4)} \in \mathbb{R}^{N \times L}$ and $T_n^{(4)} \in \mathbb{R}^{N \times L}$ are transposed and reshaped to get the three-dimensional tensors $\hat{T}_c^{(4)} \in \mathbb{R}^{L \times h \times w}$ and $\hat{T}_n^{(4)} \in \mathbb{R}^{L \times h \times w}$, where $N = h \times w$, h = H/16, and w = W/16. Then, $\hat{T}_c^{(4)}$ and $\hat{T}_n^{(4)}$ are concatenated along the channel dimension to get $\bar{T}^{(4)} \in \mathbb{R}^{L \times h \times w}$. A convolution operation is performed on $\bar{T}^{(4)}$ to get $\hat{T}^{(4)} \in \mathbb{R}^{L \times h \times w}$, then three different convolutional layers are constructed for $\hat{T}^{(4)}$ to obtain $\hat{T}^{(4_{-1})} \in \mathbb{R}^{L/8 \times h \times w}$, $\hat{T}^{(4_{-2})} \in \mathbb{R}^{L/8 \times h \times w}$, and $\hat{T}^{(4_{-3})} \in \mathbb{R}^{L \times h \times w}$. Then, they are reshaped to $T^{(4_{-1})} \in \mathbb{R}^{L/8 \times N}$, $T^{(4_{-2})} \in \mathbb{R}^{L \times N}$. Position attention weights $A^{(4)} \in \mathbb{R}^{N \times N}$ can be computed as:

$$\boldsymbol{A}^{(4)} = \operatorname{softmax}\left(\left(\boldsymbol{T}^{(4_1)}\right)^{\mathrm{T}}\boldsymbol{T}^{(4_2)}\right)$$
(6)

Then, we conduct matrix multiplication between $T^{(4_3)}$ and $A^{(4)}$, and the computed result is reshaped to $\hat{Z}^{(4)} \in \mathbb{R}^{L \times h \times w}$. Then, $\hat{Z}^{(4)}$ is multiplied with a learnable weight $\alpha^{(4)}$, and we perform element-wise addition between the weighted $\hat{Z}^{(4)}$ and $\hat{T}^{(4)}$. A convolution operation is conducted to get the fused feature map $Z^{(4)} \in \mathbb{R}^{L \times h \times w}$ as follows:

$$\boldsymbol{Z}^{(4)} = \operatorname{Conv}^{(4)} \left(\alpha^{(4)} \left(\boldsymbol{T}^{(4_3)} \boldsymbol{A}^{(4)} \right)_{\text{reshape}} \oplus \hat{\boldsymbol{T}}^{(4)} \right) \quad (7)$$

where \oplus denotes element-wise addition. The detailed computational procedure is shown in Fig. 2. Following the same computational procedure, we can also get the fused feature maps $Z^{(8)}$ for the 8th layer and $Z^{(12)}$ for the 12th layer.

In order to sufficiently integrate the hierarchical features, we conduct element-wise addition followed by a convolution operation to get the final fused feature map $Z \in \mathbb{R}^{L \times h \times w}$:

$$\boldsymbol{Z} = \operatorname{Conv}\left(\boldsymbol{Z}^{(12)} \oplus \boldsymbol{Z}^{(8)} \oplus \boldsymbol{Z}^{(4)}\right)$$
(8)

The general framework of our AHFM module is shown in Fig. 1 (the bounding box of AHFM).

C. Transformer Decoder

Image forgery localization classifies each pixel in an image into two classes, i.e., authentic class and forged class. It can essentially be considered as a special image segmentation task. We set two learnable category embeddings in the decoder to further learn the feature representations of authentic and forged classes [38], and they are interacted with the patch embeddings of the fused feature map to produce the predicted masks. Our decoder mainly contains 2 Transformer layers.

Specifically, $Z \in \mathbb{R}^{L \times h \times w}$ is sequentially reshaped, transposed, and linearly projected to obtain the embedding sequence $\dot{Z} \in \mathbb{R}^{N \times L}$. Then \dot{Z} and the category embeddings

 $S \in \mathbb{R}^{2 \times L}$ are reconstructed by Transformer layers to obtain $\ddot{Z} \in \mathbb{R}^{N \times L}$ and $\ddot{S} \in \mathbb{R}^{2 \times L}$. After performing linear projection and L2 normalization on \ddot{Z} and \ddot{S} , respectively, the quantization value $\ddot{Y} \in \mathbb{R}^{N \times 2}$ can be obtained by the scalar product operation:

$$\ddot{\boldsymbol{Y}} = L_2 \left(f_{\text{proj}} \left(\ddot{\boldsymbol{Z}} \right) \right) \left(L_2 \left(f_{\text{proj}} \left(\ddot{\boldsymbol{S}} \right) \right) \right)^T \tag{9}$$

where L_2 denotes L2 normalization and f_{proj} denotes linear projection. The transpose and reshape operations are performed sequentially on \ddot{Y} to obtain $Y \in \mathbb{R}^{2 \times h \times w}$, and the predicted mask M is computed as:

$$M = \text{softmax}\left(\text{Upsample}\left(Y\right)\right) \tag{10}$$

where Upsample denotes the upsampling operation which can resize Y to the same size as the input image. Our model is trained using a pixel-level binary cross-entropy loss function.

III. EXPERIMENTS

A. Experimental Settings

1) Synthesized dataset: We generate a large amount of synthesized images to train our Transformer-style network. For splicing and copy-move operations, we enlarge the CASIA v2.0 dataset [39]-[40]. By learning the association between scenes and forged regions, we try to find the most concealed position for inserting forged regions. Specifically, we select the most suitable donor image based on the consistency of chromaticity and complexity between the donor image and the acceptor image. Using all forged regions as candidate donors, we select the most suitable one for each CASIA v2.0 image and insert it at the most concealed position. For enlarging copy-move images in CASIA v2.0, we first find the authentic image of the forged region, then further find the corresponding copy-move image synthesized from this authentic image, and insert the forged region again at the most hidden position in this copy-move image. Each image of our enlarged CASIA v2.0 contains multiple forged regions, which may come from different images at the same time (possibly both from other images and from that image itself). These characteristics make the enlarged dataset more adaptable to complex forgery scenarios in practical applications. For removal operation, we randomly remove an annotated region from each ADE20k [41] image and fill it using the SOTA inpainting method [42]. We have generated 156006 synthesized images (140432 for training, 7787 for validation, and 7787 for testing. Our dataset can be downloaded in https://github.com/free1dom1/TBFormer).

2) Testing data: We use four publicly available datasets, i.e., NIST16 [43], CASIA v1.0 [40], IMD20 [44], and Realistic [45], to evaluate the performance of our model. CASIA v1.0 contains splicing and copy-move images. NIST16, IMD20, and Realistic contain splicing, copy-move, and removal images.

3) Evaluation metrics: We use F1-score, IoU and AUC as evaluation metrics. 0.5 is chosen as the threshold for all images when binarizing the predicted masks.

4) Implementation details: All the input images are resized to 512×512 . The feature extractor is initialized using the ViT model provided in [46], and the Transformer layers in the decoder are initialized using random weights from a truncated normal distribution. We use the SGD optimizer with the learning rate adjusted by the polynomial decay strategy $lr = lr_0 (1 - iter_{current}/iter_{total})^{0.9}$, where $iter_{current}$ denotes the current number of iterations, $iter_{total}$ denotes the total number of iterations, and $lr_0 = 0.001$ denotes the initial learning rate. We set the batch size to 8 and conduct 15-epoch training, i.e., 263310 iterations.

B. Ablation Study and Robustness Analysis

TABLE I Ablation study on Synthesized Dataset								
Variants	Precision	Recall	F1	IoU				
RGB-Only	0.922	0.872	0.890	0.825				
RGB+Noise	0.924	0.875	0.892	0.828				
RGB+Noise+AHFM	0.917	0.885	0.893	0.830				

1) Ablation study: In order to verify the effectiveness of the main modules, we set up different variants and conduct a series of experiments on the testing set of the synthesized dataset. Table I reports the experimental results of different variants. "RGB-Only" indicates that only the features output by the last layer of the RGB branch are fed into the decoder, "RGB+Noise" means that a two-branch structure is used, but only the features output by the last layer of two branches are fed into the decoder after simply concatenation, and "RGB+Noise+AHFM" denotes the proposed method, i.e., TBFormer. The results can demonstrate that both the twobranch architecture and the AHFM module are helpful to improve the performance. The F1-score and IoU can be improved by adding each module. AHFM can improve the recall with precision sacrificing, which indicates that AHFM can reserve more information from multi-domain hierarchical features, while cause more inevitable false alarms.

TABLE II AUC scores of TBFormer on IMD20 under various distortions

Distortion	AUC
no distortion	0.863
$\text{Resize}(0.78 \times)$	0.855 -0.008
$\text{Resize}(0.25 \times)$	0.853 -0.010
GaussianBlur(k=3)	0.852 -0.011
GaussianBlur(k=15)	0.792 -0.071
JPEGCompress(q=100)	0.861 -0.002
JPEGCompress(q=50)	0.822 -0.041

2) Robustness analysis: We conduct various distortion transformations, e.g., resizing, JPEG compression and Gaussian blur on the IMD20 dataset to evaluate the robustness of the model, and the experimental results are shown in Table II. From Table II, we can see that the AUC scores do not significantly decrease under different distortions, which can demonstrate the robustness of our TBFormer.

C. Comparison With State-of-the-art Methods

TBFormer is compared with six state-of-the-art methods, i.e., RGB-N [24], ManTra-Net [23], SPAN [15], MVSS-Net [20], PSCC-Net [19], and ObjectFormer [31]. Table III reports

TABLE III Comparison With State-of-the-art Methods

Method	NIS AUC	T16 F1	CASL AUC	A v1.0 F1	IMD20 AUC	Realistic AUC
RGB-N	0.937	0.722	0.795	0.408	-	-
ManTra-Net	-	-	-	-	0.748	0.680
SPAN	0.961	0.582	0.838	0.382	0.750	-
MVSS-Net	-	-	0.887	0.539	0.814	0.641
PSCC-Net	0.996	0.819	0.875	0.554	0.806	0.542
ObjectFormer	0.996	0.824	0.882	0.579	0.821	-
TBFormer	0.997	0.834	0.955	0.696	0.863	0.738



Fig. 3. Visual comparisons with the state-of-the-art methods.

the compared results on four publicly available datasets, and Fig. 3 visualizes predicted masks of the methods with publicly available codes. On the NIST16 dataset, RGB-N, SPAN, PSCC-Net, and ObjectFormer are fine-tuned, and we follow the same training/testing splits for fine-tuning the model to make a fair comparison. On the CASIA dataset, RGB-N, SPAN, PSCC-Net, and ObjectFormer are fine-tuned on CASIA v2.0 and tested on CASIA v1.0. The training dataset of MVSS-Net and our synthesized dataset are generated from CASIA v2.0, so the results of MVSS-Net and TBFormer on CASIA v1.0 are generated by the models without fine-tuning. The results of MVSS-Net on all datasets, the scores of ManTra-Net and SPAN on the IMD20 dataset, and the results of compared methods on the Realistic dataset are obtained by the pretrained models released by the authors, and the rest of scores are borrowed from their original papers. Table III shows that TBFormer achieves the best performance on each dataset, and it also can be seen in Fig. 3 that TBFormer can locate forged regions more accurately.

IV. CONCLUSION

In this paper, we introduce a novel Transformer-based image forgery localization model, named as TBFormer, which can achieve superior performance. TBFormer uses two Transformer branches to extract RGB and noise features independently to fully explore the potential forgery cues. The Attention-aware Hierarchical-feature Fusion Module (AHFM) is proposed for effectively integrating hierarchical features extracted from RGB and noise domains. Finally, the predicted mask is reconstructed by the Transformer decoder. In the future, TBFormer can be further improved by considering edge artifacts or other potential forgery cues.

REFERENCES

- Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 10, pp. 2551-2566, 2019.
- [2] Y. Liu, C. Xia, X. Zhu, and S. Xu, "Two-stage copy-move forgery detection with self deep matching and proposal superglue," *IEEE Trans. Image Process.*, vol. 31, pp. 541-555, 2022.
- [3] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1-6.
- [4] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1480-1502.
- [5] B. Liu and C.-M. Pun, "Deep fusion network for splicing forgery localization," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 237-251.
- [6] Y. Liu, Q. Guan, X. Zhao, and Y. Cao, "Image forgery localization based on multi-scale convolutional neural networks," in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Innsbruck, Austria, 2018, pp. 85-90.
- [7] B. Liu and C.-M. Pun, "Exposing splicing forgery in realistic scenes using deep fusion network," *Inf. Sci.*, vol. 526, pp. 133-150, 2020.
- [8] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copymove forgery detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2284-2297, 2015.
- [9] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Busternet: Detecting copymove image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 168-184.
- [10] J. Zhong and C.-M. Pun, "An end-to-end dense-inceptionnet for image copy-move forgery detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2134-2146, 2020.
- [11] A. Islam, C. Long, A. Basharat, and A. Hoogs, "DOA-GAN: dual-order attentive generative adversarial network for image copy-move forgery detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 4675-4684.
- [12] M. Barni, Q.-T. Phan, and B. Tondi, "Copy move source-target disambiguation through multi-branch cnns," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1825-1840, 2021.
- [13] X. Zhu, Y. Qian, X. Zhao X, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Process. Image Commun.*, vol. 67, pp. 90-99, 2018.
- [14] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8301-8310.
- [15] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "SPAN: Spatial pyramid attention network for image manipulation localization," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, UK, 2020, pp. 312-328.
- [16] Z. Gao, C. Sun, Z. Cheng, W. Guan, A. Liu, and M. Wang, "TBNet: A Two-Stream Boundary-Aware Network for Generic Image Manipulation Localization," *IEEE Trans. Knowl. Data Eng.*, 2022.
- [17] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath, "Exploiting spatial structure for localizing manipulated image regions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 4970-4979.
- [18] P. Zhou, B.-C. Chen, X. Han, M. Najibi, and L. Davis, "Generate, segment and replace: Towards generic manipulation segmentation," in *Proc. 34th Conf. Artif. Intell.*, NY, USA, 2020, pp. 13058-13065.
- [19] X. Liu, Y. Liu, J. Chen, and X. Liu, "PSCC-Net: Progressive spatiochannel correlation network for image manipulation detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7505-7517, 2022.
- [20] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14185-14193.
- [21] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2016, pp. 1-6.
- [22] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, "Constrained R-CNN: A general image manipulation detection model," in *Proc. IEEE Int Conf. multimedia expo*, 2020, pp. 1-6.
- [23] Y. Wu, W. AbdAlmageed, and P. Natarajan, "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with

anomalous features," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9543-9552.

- [24] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1053-1061.
- [25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, UK, 2020, pp. 213-229.
- [26] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7463-7472.
- [27] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [28] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6881-6890.
- [29] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 12077-12090.
- [30] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L. C. Chen, "Max-deeplab: End-to-end panoptic segmentation with mask transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5463-5474.
- [31] J. Wang, Z. Wu, J. Chen, X. Han, A, Shrivastava, S. N. Lim, and Y. G. Jiang, "Objectformer for image manipulation detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2364-2373.
- [32] Y. Sun, R. Ni, and Y. Zhao, "ET: Edge-Enhanced Transformer for Image Splicing Detection," *IEEE Signal Process Lett.*, vol. 29, pp. 1232-1236, 2022.
- [33] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2691-2706, 2018.
- [34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, and N. Houlsby, "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [35] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," ACM Comput. Surv., vol. 55, no. 6, pp. 1-28, 2022.
- [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, "Attention is all you need," in *Neural Inf. Process. Syst.*, 2017.
- [37] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146-3154.
- [38] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7262-7272.
- [39] Y. Wei, J. Ma, Z. Wang, B. Xiao, and W. Zheng, "Image splicing forgery detection by combining synthetic adversarial networks and hybrid dense U-net based on multiple spaces," *Int. J. Intell. Syst.*, vol. 37, no. 11, pp. 8291-8308, 2022.
- [40] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *Proc. IEEE summit Int. Conf. signal Inf. Process.*, China, 2013, pp. 422-426.
- [41] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *Int. J. Comput. Vision*, vol. 127, pp. 302-321, 2019.
- [42] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7760-7768.
- [43] "NIST: Nimble 2016 Datasets," [Online]. Available: https://www.nist.gov/itl/iad/mig/
- [44] A. Novozamsky, B. Mahdian, and S. Saic, "IMD2020: A large-scale annotated dataset tailored for detecting manipulated images," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops*, 2020, pp. 71-80.
- [45] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 809-824, 2017.
- [46] A. Šteiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," *arXiv preprint arXiv:2106.10270*.