CNN-Transformer Based Generative Adversarial Network for Copy-Move Source/ Target Distinguishment

Yulan Zhang, Guopu Zhu, *Senior Member, IEEE*, Xing Wang, Xiangyang Luo, Yicong Zhou, *Senior Member, IEEE*, Hongli Zhang, *Member, IEEE*, and Ligang Wu, *Fellow, IEEE*

***Abstract*— Copy-move forgery can be used for hiding certain objects or duplicating meaningful objects in images. Although copy-move forgery detection has been studied extensively in recent years, it is still a challenging task to distinguish between the source and the target regions in copy-move forgery images. In this paper, a convolutional neural network-transformer based generative adversarial network (CNN-T GAN) is proposed to distinguish the source and target regions in a copy-move forged image. A generator is first utilized to generate a mask that is similar to the groundtruth mask. Then, a discriminator is trained to discriminate the true image pairs from the false ones. When the discriminator cannot discriminate the true/false image pairs accurately, the generator can be used to obtain the final localization maps of copy-move forgery. In the generator, convolutional neural network (CNN) and transformer are exploited to extract the local features and global representations in copy-move forgery images, respectively. In addition, feature coupling layers are designed to integrate the features in CNN branch and transformer branch in an interactive way. Finally, a new Pearson correlation layer is introduced to match the similarity features in source and target regions, which can improve the performance of copy-move forgery localization, especially the localization performance on source regions. To the best of our knowledge, this is the first work to utilize transformer for feature extraction in copy-move forgery localization. The proposed method can not only detect the copy-move regions, but also distinguish the source and target regions. Extensive experimental results on several commonly used copy-move datasets have shown that the proposed method outperforms the state-of-the-art methods for copy-move detection.**

***Index Terms*— Image forensics, copy-move source/target distinguishment, convolutional neural network, transformer.**

Manuscript received 31 July 2022; revised 16 October 2022; accepted 27 October 2022. Date of publication 7 November 2022; date of current version 5 May 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102900; in part by the National Natural Science Foundation of China under Grant 62172402, Grant 62033005, Grant 62172435, Grant U1804263, and Grant 61872350; in part by the Zhongyuan Science and Technology Innovation Leading Talent Project 214200510019; in part by the Science and Technology Development Fund, Macau, under Grant 0049/2022/A1; in part by the University of Macau under Grant MYRG2022-00072-FST; and in part by the Fundamental Research Funds for the Central Universities under Grant FRFCU5710011322. This article was recommended by Associate Editor S. Wang. *(Corresponding author: Guopu Zhu.)*

Yulan Zhang is with the School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China, and also with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: xyzyl@foxmail.com).

Guopu Zhu is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: guopu.zhu@hit.edu.cn).

Xing Wang and Hongli Zhang are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: wxhit@hit.edu.cn; zhanghongli@hit.edu.cn).

Xiangyang Luo is with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China, and also with the Key Laboratory of Cyberspace Situation Awareness of Henan Province, Zhengzhou 450001, China (e-mail: luoxy_ieu@sina.com).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Ligang Wu is with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: ligangwu@hit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3220630.

Digital Object Identifier 10.1109/TCSVT.2022.3220630

I. Introduction

NOWADAYS, digital images on the Internet are increasing rapidly as social media becomes more diverse. Digital images can be easily manipulated due to the high availability of image editing tools such as Photoshop, Meitu, and GIMP. Copy-move forgery is one of the most commonly and easily used image tampering techniques, in which a region (called the source region) in an image is duplicated, then preprocessed with scaling, rotating, or color adjusting, and finally pasted to another region (called the target region) in the same image. Copy-move forgery can be used to hide or duplicate objects in an image for malicious purposes. For example, fake news with copy-move forgery images in politics will confuse the public and cause political biases. If malicious manipulations are made on evidence in court or reported experimental results in academic papers, this may lead to serious judicial injustice or academic misconduct. Hence, it is important to develop image forensic methods for copy-move forgery detection. Moreover, in some cases, it is of great significance to distinguish between the source and target regions. For example, in an image used as evidence in court, two guys have the same gun in their hands, one of which is generated by copy-move forgery. In this case, we wonder which gun is the original one. The original and the clone guns can be distinguished from each other with the help of the technique of copy-move source/target distinguishment (CMSTD).

In the past few years, a number of studies including traditional and deep learning based methods have been proposed for the detection of copy-move forgery. The traditional

methods can be classified into the block-based methods [1], [2], [3], [4] and the keypoint-based methods [5], [6], [7], [8], [9], [10]. The block-based methods first divide images into overlapping or non-overlapping blocks, then extract features from these blocks, and finally perform feature matching. The feature extraction algorithms of the block-based methods can be implemented by principal component analysis [11], discrete cosine transform [4], local binary pattern [12], discrete wavelet transform [13], Fourier transform [14], and so on. The keypoint-based methods have a similar procedure with the block-based methods, but these methods select image features from high entropy regions to determine the local extreme points. Scale-invariant feature transform [5], [9] and speeded up robust features (SURF) [6] are two commonly-used local features in the keypoint-based algorithms due to their geometric scale invariance. The keypoint-based methods turned out to be more efficient than the block-based methods for the reason that the former focuses on the sparse extreme points. Other works [8], [10] integrate the block-based and the keypoint-based methods simultaneously. Pun et al. [8] proposed to segment images adaptively and then extract the block features by feature point matching so as to indicate the forgery regions. Manu et al. [10] proposed a copy-move forgery detection method by first segmenting an image into blocks and then exploiting SURF to extract features. In general, these traditional methods that extract features manually may have some serious shortcomings. For example, the block-based methods cannot detect the region with large-scaling distortion, and the keypoint-based methods have difficulty in dealing with the smoothing forgery regions.

In recent years, with the explosive progress of deep learning in computer vision [15], [16], some deep learning based methods have been proposed for image forgery detection [17], [18], [19], [20], [21] and copy-move forgery localization (CMFL) [22], [23], [24], [25]. To suppress the effect of image contents and extract artifacts introduced by copy-move forgery, Rao et al. [22] proposed a convolutional neural network (CNN) to detect whether an image is copy-moved or not by utilizing high-pass filters as the initialized layer. To extract block features and the self-correlation between feature pixels, Wu et al. [23] proposed an end-to-end deep neural network to obtain the similar regions in copy-move forgery images. Later, Wu et al. [24] proposed a novel long short-term memory (LSTM) network to detect local anomalies left by 385 image manipulation types including copy-move forgery. It may fail in images that are intentionally contaminated with highly correlated noise. To localize the copy-move regions, Zhong et al. [25] proposed a dense-inception network consisting of pyramid feature extractors, correlation matching blocks and hierarchical post-processing modules. Liu et al. [26] proposed a CMFL method with a self-deep matching network and Proposal SuperGlue. Then, post-processing operations, including integrated score map generation and refinement methods are designed to obtain better localization results.

As far as we know, the existing traditional methods and the above-mentioned deep learning based methods can only detect and localize the copy-move forgery, but cannot distinguish between the source and the target regions. These methods focus only on localizing the duplicated regions in images. However, it is of great importance to distinguish between the source and the target regions in copy-move forgery images. Wu et al. [27] proposed a network named BusterNet to distinguish between the source and the target regions in copy-move forgery images for the first time. BusterNet consists of two parallel branches: one is the similarity detection branch, the other is the manipulation detection branch. The similarity detection branch relies on VGG16 [28] and Pearson correlation coefficient to detect similar regions in images. The manipulation detection branch exploits VGG16 and BN-Inception layers to detect the traces of manipulations. These features extracted by the similarity detection branch in BusterNet are single-level and of low resolution. Moreover, to obtain the final localization maps correctly, both of the two branches should localize the target regions correctly. In order to improve BusterNet, Chen et al. [29] proposed a cascaded network consisting of two subnetworks, *i.e.,* a copy-move similarity detection network (CMSDNet) and a source/target region distinguishment network (STRDNet). The authors introduce the double-level self-correlation, atrous spatial pyramid pooling, and attention mechanism to extract multi-scale features. Due to that the STRDNet utilizes the detection map of the CMSDNet, the whole network should be trained separately. Islam et al. [30] proposed a dual-order attentive generative adversarial network (DOA-GAN) for CMSTD. DOA-GAN utilizes a dual-order attention module to extract location-aware and co-occurrence features, and then extracts the global features with the atrous spatial pyramid pooling blocks. DOA-GAN may not perform well when the scale of the target regions is changed significantly. Barni et al. [31] designed a multi-branch CNN to differentiate the source and target regions between two nearly duplicated regions with a hypothesis testing framework given the binary localization mask (*i.e.,* the similar regions in a copy-move forgery image). This method relies on the known binary localization mask; it just identifies which region is the forged one. In the end-to-end scenario, the authors adopted the method in [2] to obtain the binary localization map.

In conclusion, most of the prior works were proposed to solve CMFL tasks, and only a few works [27], [29], [30], [31] focused on CMSTD. CMSTD is a very challenging task since the source and the target regions are difficult to distinguish. The existing methods cannot obtain satisfactory localization results due to the following reasons: first, if the size of the target regions is significantly different from that the source regions, the source and target regions are difficult to co-localize; second, since the inconsistencies between the target regions and their neighbor regions are too weak to detect, it is a hard task to identify the target regions as forged regions, which leads to the difficulty of distinguishing the target from the source regions.

To solve the above issues, we propose a CNN-transformer based generative adversarial network (CNN-T GAN) for CMSTD. The generator of the proposed GAN consists of a transformer branch and a CNN branch. The transformer branch is designed to extract the global features. The extraction of global features can improve the co-localization of the source/target regions in the whole image. The CNN branch

is designed to extract the local features from edge neighborhoods, which is good for the detection of the inconsistencies between the forged regions and their neighbor regions. Note that local features refer to the details extracted in local image neighborhoods; whereas global features include, but are not limited to, contour representations, shape descriptors, and object typologies at long distance [32], [33], [34]. Besides, feature coupling layers (FCLs) are introduced to fuse the global features from the transformer branch and the local features from the CNN branch. FCLs can enhance the global representation of the CNN branch and the local representation of the transformer branch. The main contributions of this work are as follows:

1) We propose an end-to-end CNN-transformer based generative adversarial network for copy-move forgery detection. In the proposed GAN, the generator is used to produce a three classified mask according to the copy-move forgery image; then, a discriminator is utilized to identify if the image pairs are real or fake. Once the generator can confuse the discriminator, the generator can be used to obtain the localization map of the copy-move forgery.
2) Transformer is introduced to the forensics of copy-move forgery for the first time. CNN and transformer are used to extract the local and the global features in images, respectively. The feature coupling layers are utilized to integrate the global features in the transformer branch and the local feature in the CNN branch in an interactive way. In this way, the representation in global feature and local feature are strengthened.
3) A similarity loss and a mask loss are introduced to the copy-move forgery detection, which can improve the capability of source and target distinguishment. To better obtain the final CMSTD localization maps, a new Pearson correlation layer is introduced to extract the similarity feature from images.

The remaining part of this paper proceeds as follows: Section II discusses the preliminaries about CMFL and CMSTD. Section III proposes a CNN-T GAN based network for copy-move source/target distinguishment. Section IV describes the experimental settings and then analyzes the experimental results. Finally, Section V concludes this work.

II. Preliminaries

A. Copy-Move Forgery Localization

Copy-move forgery is a commonly used and easily implemented image tampering method. Fig. 1 shows two examples of copy-move forgery. The first row shows the hiding of a dark purple flower; and the second row shows the duplications of a flying seagull. CMFL attempts to determine whether there are cloned regions in a query image and localize the forged regions. Traditional CMFL methods have a similar framework, including a feature extraction block, a feature matching block, and a refining block. By extracting the patch-based or keypoint-based features and designing feature matching algorithms, the works in [23], [24], and [27] only detect the similar regions in images as shown in the third column in Fig. 1, where we cannot distinguish the source and target regions.

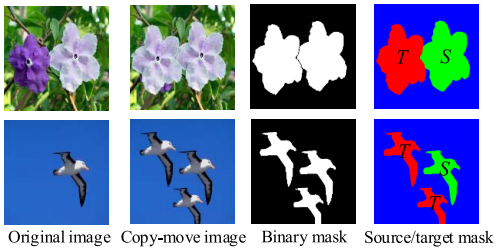

Fig. 1. Two examples of copy-move forgery.

B. Copy-Move Source/Target Distinguishment

The objective of CMSTD is to obtain the source/target masks, as shown in the fourth column of Fig. 1. The methods [24], [25] learned the traces of image manipulations by CNNs. Since the source regions (shown in green) of copy-move forgery images are not forged at all, it is more challenging to localize these source regions. Moreover, the CNNs [27], [29], [30] that can distinguish the source/target regions utilize only the local features extracted by the CNNs, while the global features are neglected for feature extraction.

C. CNN and Transformer

CNN is a kind of feed-forward neural network with deep structure and convolutional computation, and is one of the classical algorithms of deep learning. Various CNNs have been proposed for vision tasks, such as image classification [34], [35] and semantic segmentation [15], [36]. CNNs are good at extracting local features, but have difficulty in learning global representations. In CNNs, the global features can be extracted by enlarging the receptive field, compared with the case of local feature extraction, which would require more pooling operations [37], [38], [39] and lower the spatial resolution of features. Besides, the global attention mechanisms [40], [41] can also be used to capture the long-distance dependencies. However, if the convolutional operations are not properly fused with the attention mechanisms, the representation of the local features may be deteriorated.

Transformer is an encoder-decoder network with global self-attention, and can extract the global representation of an image. Vision transformers aggregate global representations among the compressed patch embeddings by the cascaded self-attention modules. Transformers are based solely on attention mechanisms, and dispense with recurrence and convolutions [42]. The original transformer [42] is proposed for natural language processing. Then, transformer blocks are introduced to CNNs for vision tasks [33], [43]. Dosovitskiy et al. [44] applied the standard transformers to images by splitting images into patches and providing the linear embeddings into transformer. By using self-attention mechanism and multi-layer perception structure, the vision transformer can achieve better results compared to the state-of-the-art CNNs [33], [45], [46]. However, the local and global

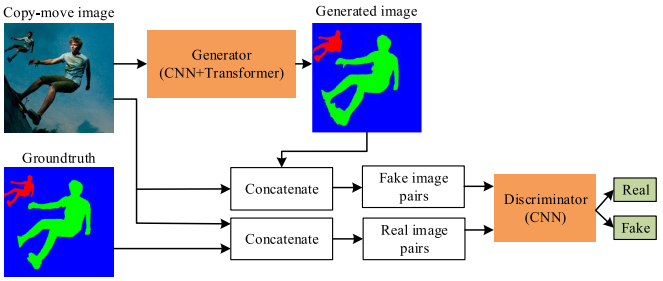

Fig. 2. Framework of CNN-T GAN.

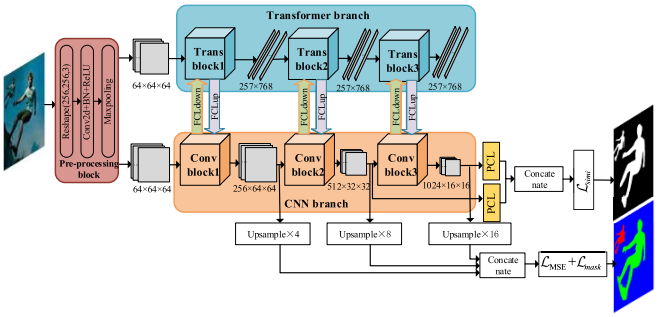

Fig. 3. Architecture of the generator in CNN-T GAN.

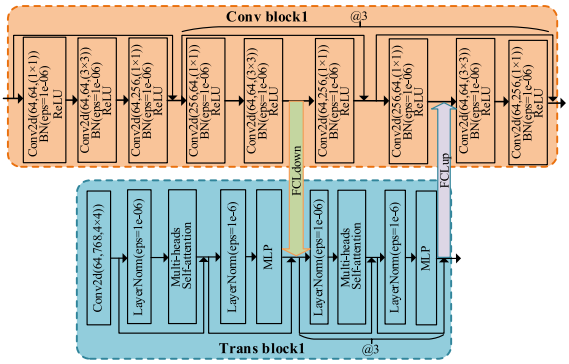

Fig. 4. Connection between the transformer branch and CNN branch.

features cannot be embedded into each other well. Hence, it is critical to develop a method to fuse the local and global features.

In this paper, to extract more discriminative features for CMSTD, CNN and transformer are used to extract the local and global features, respectively. The local features extracted by CNN and global representations extracted from transformer are fused in an interactive way. In this way, the proposed network can not only inherits the structure advantages of both CNN and transformer, but also retains the representation capability of local and global features to the maximum extent.

D. Evaluation Metrics

The pixel-level metrics, including $precision$, $recall$, and $F1\text{-}score$, are exploited to report the performance on the CMFL task of the proposed method. Both the source and the target regions are regarded as the forged regions. The pixel-level metrics including $precision$, $recall$, $F1\text{-}score$ in the source, target, and pristine regions are utilized to report the performance in CMSTD task. In addition, the overall accuracy is also used to evaluate the CMSTD performance. For a test image, $precision$, $recall$, $F1\text{-}score$, and $accuracy$ can be obtained by

$$precision = \frac{TP}{TP + FP}, \tag{1}$$

$$recall = \frac{TP}{TP + FN}, \tag{2}$$

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall}, \tag{3}$$

and

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}, \tag{4}$$

respectively, where TP, FP, FN, and TN are the true positive, false positive, false negative, and true negative, respectively.

III. Proposed Method

A. Overview of CNN-T GAN

To overcome the shortcomings of the existing CMSTD methods mentioned in Section I, a GAN with CNN-transformer (called CNN-T GAN) is proposed for source-target distinguishment of copy-move forgery. The overall framework of CNN-T GAN is shown in Fig. 2. In CNN-T GAN, the generator is utilized to generate a binary mask and an RGB mask by extracting both local and global features. The discriminator is used to discriminate whether the generated image pairs are real or fake. When the discriminator cannot differentiate the generated image pairs from the real image pairs, the generator can be used to obtain the localization maps from copy-move forgery images.

In the following, we will introduce the structures of the generator and the discriminator of the proposed CNN-T GAN in Sections III-B and III-C, respectively. Finally, Section III-D describes the loss function.

B. Generator in CNN-T GAN

The generator is composed of a pre-processing block, a transformer branch that extracts local features, a CNN branch that extracts global features, and feature coupling layers that interact with the transformer branch and the CNN branch. The architecture of the generator is shown in Fig. 3.

The pre-processing block first reshapes the images into the size of $256 \times 256 \times 3$, and then filters the images with a convolutional layer with kernel size of 7×7 and stride of 2, followed by a batch normalization (BN) layer and a rectified linear units (ReLU) layer. Finally, a max pooling layer with stride 2 is used to halve the size of the output feature. The size of the output feature of the pre-processing block is $64 \times 64 \times 64$. Then, the extracted initial local features are put into the transformer branch and the CNN branch separately. The architectures of the transformer block and CNN block are shown in Fig. 4.

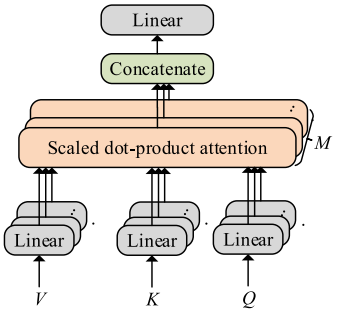

Fig. 5. Architecture of multi-heads self-attention block.

In the following, we briefly introduce the design of the transformer branch and CNN branch.

1) Transformer Branch: The main architecture of the transformer branch is designed as done in [44], and is composed of N_{tb} transformer blocks. In this paper, we set $N_{tb} = 3$. As shown in the upper half part of Fig. 4, the features output from the pre-processing block are first put into a convolutional layer with kernel size of 4×4 and stride 4. Then four pairs of multi-heads self-attention (MHSA) [42] blocks and multilayer perceptron (MLP) layers are followed to extract the global features. A layernorm layer is added before the MHSA blocks and the MLP layers to normalize the feature maps. Residual connections are added in both the self-attention layer and MLP layers.

An attention function maps a query Q and a set of key-value pairs (K, V) to an output. The output is calculated by

$$\text{Attention}(Q, K, V) = softmax(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})V, \tag{5}$$

where $softmax$ denotes the activation function implemented by softmax, the query Q, key K is of dimension d_k, and value V is of dimension d_v.

Then MHSA blocks are utilized to attend information from different representation subspaces jointly. The MHSA blocks have N attention layers running in parallel which can linearly project the queries, keys, and values for N times with different linear projections to d_k, d_k, and d_v dimensions, respectively. The architecture of MHSA is shown in Fig. 5.

The output of the MHSA can be obtained by

$$\text{MultiHeads}(Q,K,V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_i, \ldots, \text{head}_N), \tag{6}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, $i \in [1, N]$, $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^Q \in \mathbb{R}^{hd_v \times d_{model}}$. In this work, N is set to 12, $d_{model} = 768$, $d_k = d_v = d_{model}/N = 64$.

The MLP layers introduce one or more hidden layers based on the basic network. The output of the hidden layers are transformed by activation functions. An MLP layer consists of an up-projection fully connected layer, a Gaussian error leaky unit (GELU) layer [47], a down-projection layer, and a dropout layer. The output size of the first transformer block is 257×768. The second transformer block is similar to the first one, which also consists of four pairs of MHSA blocks and MLP layers. The third block consists of three pairs of MHSA blocks and MLP layers. The output sizes of all features from these three blocks are 257×768.

2) CNN Branch: As shown in the bottom half of Fig. 3, the CNN branch in our proposed method also consists of 3 blocks. The CNN branch has a feature pyramid structure [48], in which the size of feature maps decreases with the depth of the network increases. As shown in Fig. 4, the first convolutional block is composed of 7 bottlenecks. Each bottleneck consists of a 1×1 down-projection convolutional layer, a 3×3 spatial convolutional layer, and a 1×1 up-projection convolutional layer, each convolutional layer is followed by a BN layer and a ReLU layer. A residual connection is added between the input and the output of the bottleneck. Except for the first bottleneck, the following 6 bottlenecks can be divided into 3 pairs, each pair corresponds to a stage in Trans block1 in transformer branch. The output feature size of the first Con block is $256 \times 64 \times 64$. The second Conv block consists of 8 bottlenecks, where 2 sequential bottlenecks constitute a stage. The output feature size of the second Conv block is $512 \times 32 \times 32$. The third Conv block consists of 6 bottlenecks, where 2 sequential bottlenecks constitute a stage. The output feature size of the second Conv block is $1024 \times 16 \times 16$.

The features from the CNN branch and the transformer branch interact with each other through feature coupling layers (FCLs), which will be introduced in Section III-B.3. The FCLs provide the possibility to preserve fine-detailed features. In this way, the CNN branch consecutively provides local feature details for the transformer branch. Finally, the feature maps from the Conv block1, Conv block2, and Conv block3 are up-sampled to the input image size 256×256 and then fused by concatenation. Then, the fused feature maps are put into a tanh activation function. With a mean squared error (MSE) loss and a L_{mask} loss function, the final source-target distinguishment maps can be obtained. The feature maps of Conv block2 and Conv block3 are put into a Pearson correlation layer (PCL) to obtain the similarity information in the copy-move forgery images. Taking the feature maps from Conv block3 as an example, we describe the process of feature matching. The size of the feature map from Conv block3 is $1024 \times 16 \times 16$, which can be regarded as 1024 patch-like features of size 16×16, *i.e.*,

$$F^X = \begin{bmatrix} f^X(0,0) & f^X(0,1) & \ldots & f^X(0,i_c) \\ f^X(1,0) & f^X(1,1) & \ldots & f^X(1,i_c) \\ \ldots & \ldots & \ldots & \ldots \\ f^X(i_r,0) & f^X(i_r,1) & \ldots & f^X(i_r,i_c) \end{bmatrix}, \tag{7}$$

where $i_r, i_c \in [0, 15]$, the patch-like feature has 1024 dimensions. Then we compute the feature similarity score with self-correlation to extract the useful information and decide the matched similar regions, which may be the potential copy-move regions. With F^X, the Pearson correlation coefficient p is exploited to quantify the similarity between two patch-like features $f^X[i]$ and $f^X[j]$, where $i = (i_r, i_c)$ and $j = (j_r, j_c)$. First, the feature maps are normalized by

$$\tilde{f}^X[i] = (f^X[i] - \mu^X[i])/\sigma^X[i], \tag{8}$$

where $\mu^X[i]$ and $\sigma^X[i]$ are the mean and the standard deviation of X, respectively. Then, p can be obtained by

$$p(i, j) = (\tilde{f}^X[i])^{\mathrm{T}} \tilde{f}^X[j]/1024, \tag{9}$$

where $(\cdot)^{\mathrm{T}}$ denotes the transpose operator. For a given location $i = (i_r, i_c)$, we calculate the Pearson correlation coefficient for all possible $j = (j_r, j_c)$, the similarity between $f^X[i]$ and $f^X[j]$ can be measured by a score vector $S^X[i, j]$, *i.e.*,

$$S^X[i][j] = [p(i, 0), \ldots, p(i, j), \ldots, p(i, 255)]. \tag{10}$$

If the feature map $f^X[i]$ is matched, the score $S^X[i][j](j \neq i)$ should be significantly greater than other scores $S^X[i][k](k \notin \{i, j\})$. So the scores are sorted in a descending order, which is shown as

$$S'^X[i][j] = \mathrm{sort}(S^X[i][j]). \tag{11}$$

The sorted vector of scores contains sufficient information for selecting the matched feature. Regardless the length L of the sorted vector of scores, the top K scores are picked to form a new pooled score vector

$$P^X[i][k] = S'^X[i][k'], \tag{12}$$

where $k \in [0, K-1]$, and k' is the index of raw sorted vector S'^X. Finally, the patches corresponding to the top-K maximal correlation scores are utilized to localize the similar regions in copy-move forgery images. For the features output from the Conv block2, the same method is used to calculate the pooled score vector. The features from these two PCLs are fused by concatenation, and finally a binary cross-entropy loss is utilized to acquire the similar regions in images. Through the PCLs, the proposed network can obtain the potential copy-move maps.

3) Feature Coupling Layer: The features from the CNN branch are of size $C \times H \times W$, where C, H, and W represent the channels, height, and width of the features, respectively; while the features from the transformer branch are of size $(K+1) \times E$, where K and E denote the number of image patches and the embedding dimension, respectively. In other words, the local features from the CNN branch and the global features from the transformer branch are inconsistent. Hence, it is necessary to design FCLs to eliminate the misalignment between these two features. We perform the feature fusion using a feature pyramid network (FPN) architecture [48]. FCLs are inserted into every block to consecutively eliminate the semantic gap between the transformer branch and the CNN branch. The architecture of FPN can take full advantages of the features of different scales. The structures of FCLs are shown in Fig. 6, Figs. 6 (a) and (b) are the structures of feature coupling down-sampling layer and feature coupling up-sampling layer, respectively.

On the one hand, in order to feed the global features from the transformer branch to the CNN branch, a feature coupling up-sampling layer is built between the MLP layers in transformer branch and the 3×3 convolutional layers in the CNN branch. The patch embeddings are first arranged by the localization information of the patch to align the spatial scale $S \times S \times E$, where $S \times S = E$. Then, the channel dimension is aligned with that of CNN feature maps through a 1×1 convolutional layer. Finally, up-sampling is performed on these features to align with those from the CNN branch. Meanwhile, batch normalization and LayerNorm are used to regularize features. In this way, the dimension of the features from the transformer branch can be aligned with that of the features from the CNN branch.

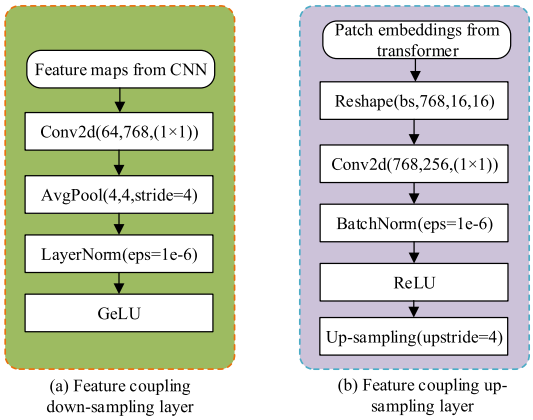

Fig. 6. Structures of feature coupling layers.

On the other hand, to align the dimension of the features from the CNN branch with that from the transformer branch, a feature coupling down-sampling layer is designed after the 3 convolutional residual blocks in the CNN branch and the MHSA blocks in the transformer branch. The feature maps from the CNN branch are first put into a 1×1 convolutional layer to align the channel numbers of the patch embeddings. Then down-sampling is implemented to complete the spatial dimension alignment through an average pooling layer with stride 4. The features are regularized by LayerNorm and GeLU layer. In this way, the dimension of the features from the CNN branch is aligned with that of the features from the transformer branch.

The feature coupling layers can not only reinforce the global representations in CNN branch, but also enrich the local details of the transformer branch.

C. Discriminator in CNN-T GAN

The generator is used to generate a mask that can distinguish the source and target regions in copy-move forgery images. The generated mask and the copy-move forgery image compose an image pair, the ground-truth mask and the copy-move forgery image compose another image pair. Then a discriminator is designed to predict whether these image pairs are real or fake, while the specially designed generator tries to fool the discriminator. Until the discriminator cannot distinguish the fake image pairs from the real image pairs, the generator can be utilized to obtain the final localization maps. The structure of the discriminator is designed based on the discriminator of Patch-GAN [49], as shown in Fig. 7.

From Fig. 7, it can be observed that the discriminator consists of 4 convolutional blocks and a fully connected layer. Each convolutional layer is followed by a BN layer and a

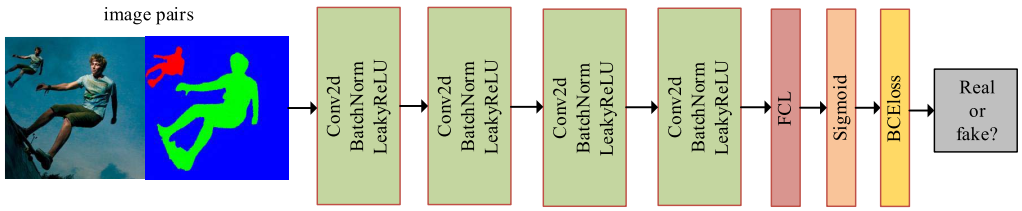

Fig. 7. Architecture of the discriminator in CNN-T GAN.

LeakyReLU layer. The kernel sizes of all former 4 convolutional blocks are 5×5. Except that the last convolutional layer is of stride of 1, the former 3 convolutional layers are of a stride of 2. The output channels of the 4 convolutional layers are 64, 128, 256, and 512, respectively. The output channel of the linear layer is 1. Finally, a sigmoid activation function and BCE loss are followed to predict whether the image pair is real or fake.

D. Loss Function

Considering the task of copy-move source-target distinguishment, the total loss function, which consists of the adversarial learning loss, minimal square error (MSE) loss, mask loss, and similarity loss, is formulated with

$$\mathcal{L}_{\text{total}}(\text{G}, \text{D}) = \mathcal{L}_{adv}(\text{G}, \text{D}) + \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \mathcal{L}_{\text{mask}} + \lambda_3 \mathcal{L}_{simi}, \tag{13}$$

where λ_1, λ_2, and λ_3 are the weights on $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{mask}}$, and $\mathcal{L}_{simi}$, respectively. In our experiments, we find that the values of $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{mask}}$, and $\mathcal{L}_{simi}$ are much smaller than that of $\mathcal{L}_{adv}$ when the training converges. To ensure the four losses have the same order of magnitude, we empirically set $\lambda_1 = 100$, $\lambda_2 = 50$, and $\lambda_3 = 20$.

1) Adversarial Loss: The adversarial loss $\mathcal{L}_{adv}$ is defined as:

$$\mathcal{L}_{adv}(\text{G}, \text{D}) = E_{(X,Y)}[\log \text{D}(X, Y) + \log(1 - \text{D}(X, \text{G}(X)))], \tag{14}$$

where G and D represent the generator and the discriminator, respectively. E denotes the expectation of a specified distribution, X denotes the input image to be queried, Y denotes the ground-truth mask, $\text{D}(X, Y)$ denotes the probability that the discriminator predict the real image pairs as real, and $\text{D}(X, \text{G}(X))$ denotes the probability that the discriminator predict the generated image pairs as real.

2) MSE Loss: To obtain a more precise localization map from the copy-move forgery images, the generated mask should be as close to the ground-truth mask as possible. So the MSE loss, *i.e.*, the L_2 distance between the ground-truth mask and the generate mask, is calculated by

$$\mathcal{L}_{\text{MSE}} = E_{(Y,G(X))}[(Y - \text{G}(X))^2], \tag{15}$$

where Y denotes the ground-truth mask that corresponds to the source and target regions, $\text{G}(X)$ is the image generated by G.

3) Mask Loss: To make the network pay more attention to the copy-move forgery regions, *i.e.*, the red and green regions shown in Fig. 1, we reweight the MSE loss $\mathcal{L}_{\text{MSE}}$ in Subsection III-D.2 with $mask = [0, 1, 1]$. The first, second and third channels of $mask$ correspond to the B, G, and R channels of the RGB mask, respectively. That is to say, only the channels G and R are taken into consideration. Hence, the mask loss $\mathcal{L}_{mask}$ is obtained by

$$\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{MSE}} \odot mask, \tag{16}$$

where $\odot$ denotes the matrix multiplication. The reweighting on color channels can gain more attention on source and target regions of copy-move forged images, which improves the performance on CMSTD.

4) Similarity Loss: The similarity loss is exploited to measure the similarity between the predicted binary mask and the binary ground-truth mask, which is calculated by

$$\mathcal{L}_{simi} = -\sum_{j=1}^{W}\sum_{i=1}^{H} S_{i,j} \log(P_{i,j}) + (1 - S_{i,j}) \log(1 - P_{i,j}), \tag{17}$$

where W and H denote the width and the height of the input images, respectively; $S = [S_{i,j}]$ represents the predicted copy-move forgery regions, and pixel (i, j) has undergone copy-move forgery when $S_{i,j} = 1$; in other words, the pixel (i, j) is the pristine region when $S_{i,j} = 0$. $P_{i,j}$ represents the possibility that the pixel (i, j) has been modified by copy-move forgery. The $\mathcal{L}_{\text{simi}}$ loss is applied in the CNN branch following the PCLs introduced in Section III-B.2. The features output from the PCLs contain lots of similarity information, so the $\mathcal{L}_{simi}$ loss can be used to localize the similar regions that are suspected as the copy-move forged regions.

During training, the proposed network attempts to maximize the loss of the discriminator and minimize the loss of the generator. Hence, the training for the proposed network can be described as

$$(\text{G}^*, \text{D}^*) = \arg \min_{\text{G}} \max_{\text{D}} \mathcal{L}_{\text{total}}(\text{G}, \text{D}), \tag{18}$$

where G^* and D^* are the optimized solutions of G and D, respectively.

IV. Experiments

A. Experimental Setup

The experiments are implemented with the Pytorch framework on NVIDIA GeForce RTX 2080 Ti GPU. During training, the adaptive moment estimation optimizer [50] is adopted to optimize the generator and discriminator. To further

improve the performance of the generator, the discriminator and the generator are trained alternatively. In an iteration, the discriminator is updated once and then the generator is updated twice. The learning rates of the generator and the discriminator are set to 1×10^{-4} and 1×10^{-5}, respectively. The batch size is set to 24, and the network is trained for 100 epochs. The model with the highest F1-score on test data is selected for evaluation.

1) Datasets: The number of samples in the existing publicly available datasets is small for copy-move forgery detection. The authors in [27] collected a synthetic CMFD dataset[1] called USCISI. USCISI dataset contains 100k samples, each of which has a binary mask for copy-move forgery detecion, and a three-class mask that can distinguish the source and target regions for copy-move source-target distinguishment. In our experiments, 80k, 10k, and 10k samples are randomly selected from the USCUSI dataset for training, validation, and testing, respectively. Then two commonly used datasets for copy-move forgery detection, *i.e.*, the CASIA2.0 dataset[2] and the CoMoFoD dataset[3] [51], are used to evaluate the generalization of the proposed network. The CASIA2.0 dataset contains 1,313 copy-move forgery images; and the CoMoFoD dataset contains 200 basic copy-move forged images and some augmented forged images that are generated by six distortion methods including JPEG compression (JC), noise adding (NA), image blurring (IB), brightness change (BC), color reduction (CR), and contrast adjustments (CA), summing to 5k samples.

For the robustness analysis, the distortion methods mentioned in [51] are applied to the test data in USCISI dataset. The quality factor (QF) of JC is set to $QF \in \{70, 80, 90, 100\}$, the standard deviation σ of Gaussian noise in NA is set to $\sigma \in \{0.1, 0.2, 0.3, 0.4\}$, the size W of Gaussian filter is set to $W \in \{3 \times 3, 5 \times 5, 7 \times 7\}$ for IB, the intensity S_bc of BC is set to $S\_bc \in \{0.8, 0.9, 1.1, 1.2\}$, the intensity S_cr of CR is set to $S\_cr \in \{8, 16, 32, 64\}$, and the lower bound B_lower and upper bound B_upper of CA are set to $(B\_lower, B\_upper) \in \{(0.01, 0.95), (0.01, 0.9), (0.01, 0.85), (0.01, 0.8)\}$.

2) Compared Methods: The performance of the proposed method is validated on CMFL and CMSTD. The goal of CMFL is to localize the copy-move regions in images without distinguishing the source and target regions. To evaluate the performance on CMFL task, some advanced methods on CMFL are considered for comparison: (1) BusterNet [27]; (2) DenseInceptionNet [25]; (3) Mantra-Net [24]; (4) DOA-GAN [30]; (5) CMSD_STRD [29]; (6) DenseFCN [18]; and (7) CMFD_FMPS [26]. The CMSTD methods can not only localize the copy-move regions, but also can distinguish the source/target regions. For the CMSTD task, several existing methods are considered for comparison: (1) BusterNet [27]; (2) CMSD_STRD [29]; (3) DOA-GAN [30]; and (4) Multi-branch CMSTD [31].

[1]https://github.com/isi-vista/BusterNet/tree/master/Data/USCISI-CMFD-Small

[2]http://forensics.idealtest.org/casiav2

[3]https://www.vcl.fer.hr/comofod/comofod.html

B. Experimental Results

In this subsection, we study the effectiveness of the proposed CNN-T GAN on three benchmark datasets. The experimental results of the proposed method on CMFL and CMSTD are compared with those of the state-of-the-art methods. Then, several ablation experiments on the backbone network, feature selection, and loss functions are studied. Finally, the robustness against several post-processing operations is analyzed.

1) Comparisons With the State-of-the-Art CMFL Methods: For the comparisons on CMFL, the released codes and pre-trained models of BusterNet,[4] Mantra-Net,[5] CMSD_STRD,[6] and DOA-GAN[7] are utilized for testing with our test data. Note that for CMSD_STRD [29], the pre-trained single-channel CMSDNet is used for CMSD task. For Dense-FCN [18], the network is retrained based on the model released in DenseFCN.[8] For DenseInceptionNet [25], the training and testing are implemented based on the model.[9] For CMFD_FMPS [26], the results attained by SelfDM-SA-MobileNetV3+PS+CRF reported in [26] are used for comparison.

For the CMFL task, similar copy-move regions are both considered as the forged regions. The precision, recall, and F1-score are first calculated for each image in pixel level. Then, the average precision, recall, and F1-score on the whole dataset are calculated. The comparative CMFL results on three benchmark datasets, *i.e.*, USCISI dataset, CASIA2 dataset, and CoMoFoD dataset, are shown in Table I.

From Table I, it can be found that the proposed CNN-T GAN obtains the highest precision and F1-score on USCISI dataset. The Mantra-Net [24] obtains higher recall than the proposed method. On CASIA2 dataset, Mantra-Net achieves the highest performance in both precision and recall. Since the threshold for Mantra-Net is set to a very small value (threshold=0.02), most of the positive regions can be correctly localized, which leads to a high true positive (TP) rate. Hence, true positive samples account for a high proportion of the predicted positive samples, and this gives a high precision value. In this situation, Mantra-Net obtains a higher precision than the proposed method. However, the low threshold value tends to result in misclassifying negative pixels as positive pixels. Therefore, by a more comprehensive performance metric, *i.e.*, F1-score, the proposed method achieves better results than Mantra-Net. On CoMoFoD dataset, the proposed method achieves the highest precision and F1-score, the Mantra-Net obtains the best recall. The high recall achieved by Mantra-Net on three datasets may be due to that Mantra-Net rarely misidentified the positive samples as negative, *i.e.*, the Mantra-Net has a lower false negative rate. The comprehensive metrics, *i.e.*, F1-scores, of the proposed method are 6.88%, 0.62%, and 6.0% higher than those of the second best method

[4]https://github.com/isi-vista/BusterNet

[5]https://github.com/ISICV/ManTraNet

[6]https://github.com/imagecbj/A-serial-image-copy-move-forgery-localization-scheme-with-source-target-distinguishment

[7]https://github.com/asrafulashiq/doagan_clean

[8]https://github.com/ZhuangPeiyu/Dense-FCN-for-tampering-localization

[9]https://github.com/HilbertXu/CMFD-Dense-InceptionNet

TABLE I

CMFL RESULTS OF PRECISION, RECALL, AND F1-SCORE (%) WITH COMPARED METHODS ON DIFFERENT DATASETS; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Datasets	Methods	presicion	recall	F1-score
USCISI	BusterNet [27]	60.56	59.13	34.84
	DenseInceptionNet [25]	93.83	94.70	72.79
	Mantra-Net [24]	95.89	**98.22**	65.94
	DOA-GAN [30]	94.79	94.75	81.55
	CMSD_STRD [29]	93.62	92.83	74.74
	DenseFCN [18]	92.73	94.74	74.09
	CMFD_FMPS [26]	88.48	82.81	83.94
	Proposed	**96.28**	96.23	**90.82**
CASIA2	BusterNet [27]	49.83	48.69	30.00
	DenseInceptionNet [25]	69.18	71.06	41.91
	Mantra-Net [24]	**87.45**	**93.87**	53.49
	DOA-GAN [30]	38.41	35.27	15.10
	CMSD_STRD [29]	79.63	78.51	62.80
	DenseFCN [18]	68.45	71.13	43.58
	CMFD_FMPS [26]	63.62	47.52	49.18
	Proposed	70.91	70.12	**63.42**
CoMoFoD	BusterNet [27]	55.33	55.18	30.63
	DenseInceptionNet [25]	74.83	78.01	49.87
	Mantra-Net [24]	89.45	**96.45**	65.20
	DOA-GAN [30]	56.04	55.17	28.48
	CMSD_STRD [29]	78.86	78.86	61.68
	DenseFCN [18]	70.57	73.52	54.22
	CMFD_FMPS [26]	78.85	82.02	77.42
	Proposed	**91.06**	91.20	**83.42**

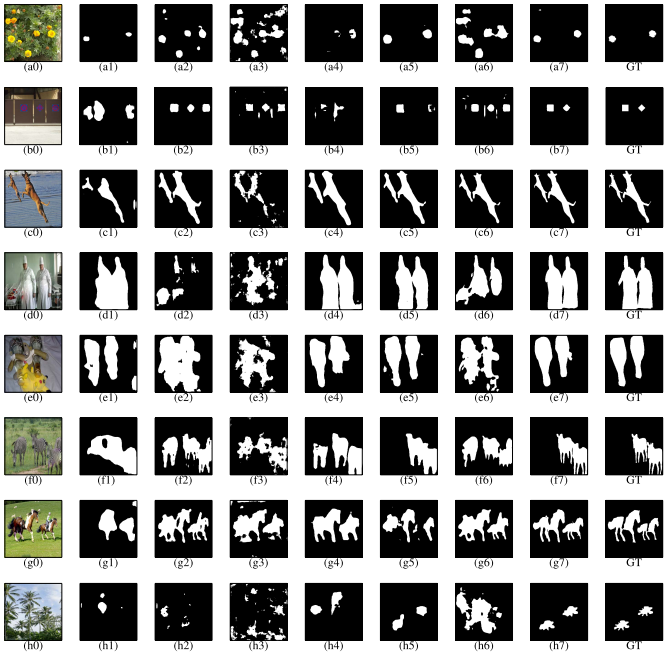

Fig. 8. CMFL localization maps of compared methods. The first column is the copy-move forged images; the second to eighth columns represent the localization maps obtained by BusterNet [27], DenseInceptionNet [25], Mantra-Net [24], DOA-GAN [30], CMSD_STRD [29], DenseFCN [18], and the proposed method, respectively; the last column is the corresponding ground-truth masks of the first column.

on USCISI dataset, CASIA2 dataset, and CoMoFoD dataset, respectively.

Fig. 8 shows some CMFL maps of the compared methods. In Fig. 8, (a0)-(b0) are from CoMoFoD dataset, (c0)-(g0) are from USCISI dataset, and (h0) is from CASIA2 dataset. From the second column of Fig. 8, it can be seen that BusterNet can localize some of the similar regions, whereas, most of the duplicated regions cannot be recognized. The fourth column shows that MantraNet cannot detect complete objects in images. DenseInceptionNet, DOA-GAN, CMSD_STRD, and DenseFCN can roughly obtain similar regions in copy-move forgery images with almost complete outlines. However, the details of the edges of the targets cannot be well localized, for example, the legs of the zebras in the sixth row of Fig. 8. Our proposed method can localize the similar regions with more texture details. This may be due to that the proposed CNN-T GAN extracts both the global structure features and the local texture features in images. In Section IV-B.3, we will validate the effectiveness of the integration of the CNN and transformer.

2) Comparisons With the State-of-the-Art CMSTD Methods: For the comparisons on CMSTD, the released codes and pre-trained models of BusterNet, CMSD_STRD, and DOA-GAN are adopted for testing. For CMSD_STRD, we first utilize the pre-trained CMSDNet to localize the copy-move regions, and then utilize the pre-trained STRDNet to distinguish the source and the target regions. Since the Multi-branch CMSTD is designed for source-target disambiguation on the condition that the copy-move regions are known, the authors recommend that the method DeseInceptionNet [25] is first utilized to obtain the copy-move regions; then, the pre-trained model of Multi-branch CMSTD[10] is used to obtain the CMSTD maps.

Table II shows the comparative CMSTD results on three benchmark datasets. Since the pristine regions account for a large amount of the images, the evaluation metrics in pristine region will be a relatively large number even though a method cannot correctly detect the copy-move regions. Therefore, the metrics in the source and target regions are more significant in evaluating the performance of CMSTD. Table II shows that the proposed CNN-T GAN obtains the best results on all the evaluation metrics on USCISI dataset. The overall F1-scores of the proposed method are 1.46%, 24.16%, and 11.57% higher than those of the second best method, *i.e.*, DOA-GAN, on pristine, source, and target regions, respectively. Except for the precision and the recall on pristine regions on CASIA2 dataset, the proposed method obtained the best metrics on CASIA2 dataset. The overall F1-scores of CNN-T GAN are 0.97%, 14.74%, and 22.69% higher than those of the second best method, *i.e.*, CMSD_STRD, on pristine, source, and target regions, respectively. On CoMoFoD dataset, the proposed method obtains the best metrics except the recall in pristine regions. The F1-scores of CNN-T GAN are 0.73%, 33.22%, and 33.28% higher than those of the second best method, *i.e.*, CMSD_STRD, on pristine, source, and target regions, respectively. The proposed method has achieved great improvement in localization performance on source and target regions.

[10]https://github.com/andreacos/MultiBranch_CNNCopyMove_Disambiguation

TABLE II

CMSTD RESULTS OF PRECISION, RECALL, F1-SCORE (%) ON PRISTINE, SOURCE, AND TARGET REGIONS OF COMPARED METHODS ON DIFFERENT DATASETS; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Datasets	Methods	precision			recall			F1-score			average accuracy
		pristine	source	target	pristine	source	target	pristine	source	target	
USCISI	BusterNet [27]	91.01	23.83	10.19	98.72	11.44	2.91	94.51	12.84	3.99	93.04
	CMSD_STRD [29]	97.02	50.68	26.24	97.84	45.48	42.32	97.37	45.35	30.40	95.03
	DOA-GAN [30]	96.68	68.65	74.64	98.33	58.89	78.65	97.46	60.29	74.79	97.04
	Multi-branch CMSTD [31]	95.55	31.87	24.58	90.50	43.57	30.69	92.64	33.62	25.41	90.39
	Proposed	**98.71**	**86.12**	**87.10**	**99.14**	**84.37**	**87.44**	**98.92**	**84.45**	**86.36**	**98.66**
CASIA2	BusterNet [27]	94.45	20.93	6.36	98.01	22.51	1.64	96.00	18.58	2.14	94.21
	CMSD_STRD [29]	**96.43**	31.88	22.86	97.39	36.74	26.91	96.75	30.86	22.71	94.09
	DOA-GAN [30]	91.02	9.86	12.52	**99.56**	4.76	6.16	94.76	5.30	6.92	93.76
	Multi-branch CMSTD [31]	92.50	14.58	12.32	88.03	24.93	10.29	89.36	14.07	9.12	87.63
	Proposed	96.19	**53.56**	**57.57**	99.55	**45.46**	**44.24**	**97.72**	**45.60**	**45.40**	**97.15**
CoMoFoD	BusterNet [27]	96.92	15.88	5.81	94.87	25.03	1.64	95.64	14.38	2.18	94.42
	CMSD_STRD [29]	97.95	32.05	27.52	98.35	29.53	29.63	98.13	28.15	26.58	96.35
	DOA-GAN [30]	94.22	15.47	12.60	**99.63**	6.09	4.71	96.69	7.25	5.63	95.71
	Multi-branch CMSTD [31]	94.83	17.99	20.21	92.03	15.30	5.65	92.84	9.56	7.72	91.04
	Proposed	**98.20**	**80.58**	**78.88**	99.56	**60.73**	**58.99**	**98.86**	**61.37**	**59.86**	**98.44**

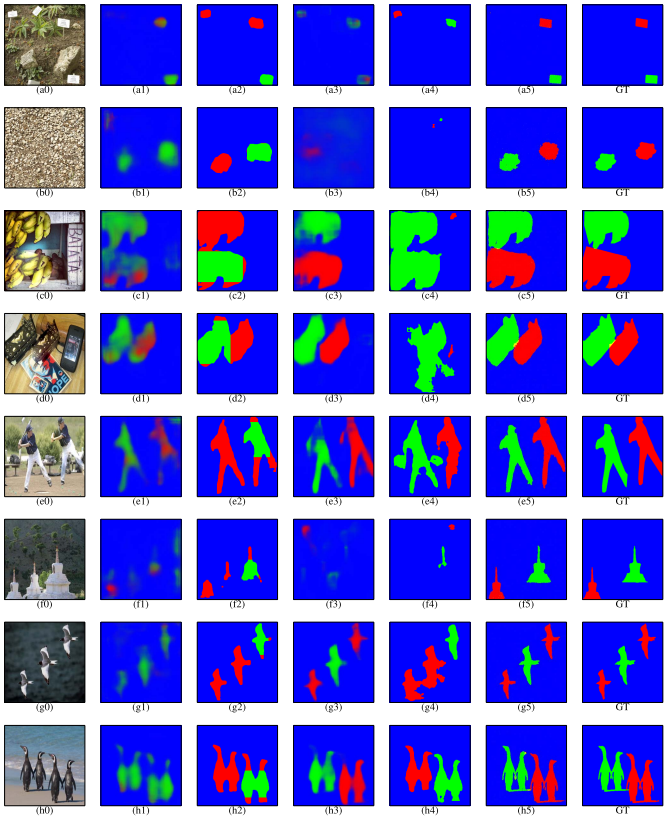

Fig. 9. CMSTD localization maps of compared methods. The first column is the copy-move forged images; the second to the fifth columns represent the localization maps obtained by BusterNet [27], CMSD_STRD [29], DOA-GAN [30], Multi-branch CMSTD [31], and the proposed method, respectively; the last column is the corresponding ground-truth masks of the first column.

Fig. 9 shows the CMSTD localization maps of the compared methods. In Fig. 9, (a0)-(b0) are from CoMoFoD dataset, (c0)-(e0) are from USCISI dataset, and (f0)-(h0) are from CASIA2 dataset. From Fig. 9, it can be seen that the proposed CNN-T GAN can not only localize the copy-move regions in images, but also can distinguish the source and the target regions well. BusterNet obtains rough localization maps of the copy-move forged regions, but cannot distinguish the source and target regions. CMSD_STRD can obtain accurate copy-move regions, but has difficulty in distinguishing the soure/target regions. DOA-GAN can distinguish the source/target regions, but the localization maps are too rough to show the edge details on copy-move regions. Multi-branch CMSTD distinguishes the soure/target regions based on the binary mask obtained by DenseInceptionNet [25]. The accuracy on source/target distinguishment is dependent on the performance of DenseInceptionNet. Sometimes, Multi-branch CMSTD incorrectly discriminates the source and target regions even when DenseInceptionNet obtains proper copy-move regions, as shown in Figs. 9(g4) and (h4). The proposed method can obtain satisfactory source/target localization maps with accurate edge information, as shown in Figs. 9(f5), (h5). In addition, the proposed method can distinguish the multi-target duplications, as shown in Fig. 9(g5).

3) Ablation Study: We first investigate the performance of different backbones of the generator in GAN. The U-Net [15], Deeplab-V3 [16] and the proposed CNN-T network are considered to generate the three-class masks. The discriminators of the three networks are identical to that described in Section III-C. Table III shows the CMSTD performance of these backbones on USCISI copy-move dataset. From Table III, it can be observed that Deeplab-V3 obtains better CMSTD results than U-Net. This is due to that Deeplab-V3 exploits the atrous convolution and the atrous spatial pyramid pooling modules to extract multi-scale features. Our proposed CNN-T Network can further improve the CMSTD performance due to that the CNN-T Network extracts both the global and the local features. It also can be found that all these three backbones obtain better localization results on the target regions than on the source regions. This is due to that the target regions are the actually forged regions, in which the duplicated traces may be left on the edges. Meanwhile, the source regions are not forged at all, thus, it is more challenging to localize the source regions.

TABLE III

CMSTD RESULTS OF PRECISION, RECALL, AND F1-SCORE (%) ON PRISTINE, SOURCE, AND TARGET REGIONS WITH DIFFERENT BACKBONES ON USCISI COPY-MOVE DATASET; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Backbone	precision			recall			F1-score			average accuracy
	pristine	source	target	pristine	source	target	pristine	source	target	
U-Net [15]	94.85	42.81	68.89	96.37	42.54	68.65	95.49	40.61	65.25	94.63
Deeplab-V3 [16]	96.33	55.42	77.65	97.54	54.79	76.53	96.87	53.27	75.48	96.34
Proposed	**98.71**	**86.12**	**87.10**	**99.14**	**84.37**	**87.44**	**98.92**	**84.45**	**86.36**	**98.66**

Then, ablation studies are conducted to validate the effectiveness of the design of feature coupling layers. These ablation studies are performed in five cases: 1) only the CNN branch is used as the generator; 2) only the transformer branch is used as the generator; 3) the CNN-transformer only with the feature coupling down-sampling layers is used as the generator; 4) the CNN-transformer only with the feature coupling up-sampling layers is used as the generator; 5) the CNN-transformer with both the feature coupling down-sampling layers and up-sampling layers is used as the generator. "Our-CNN", "Our-transformer", "CNN-T with FCLdown", "CNN-T with FCLup", and "Proposed" denotes the above five cases, respectively. The experimental results are shown in Table IV.

From the fourth and fifth rows in Table IV, it can be found that the feature coupling down-sampling layers can improve the CMSTD performance of transformer a bit. Comparing the third and sixth rows in Table IV, we see that the feature coupling up-sampling layers can improve the CMSTD performance of CNN a lot, especially for the source/target regions. The CNN-T with both FCLup and FCLdown obtains the best performance compared with the other schemes. The reason for these results may be explained as follows. On the one hand, the feature coupling down-sampling layers can supply the transformer with local features, and thus enrich the local details of the transformer. On the other hand, the feature coupling up-sampling layers can transmit global representations from the transformer branch to the CNN branch, which reinforces the global perception capability of the CNN branch. The CNN-transformer GAN can not only maintain the advantages of both CNN and transformer, but also retain the representation capability of local and global features to a large extent.

In the following, we discuss the effects of feature selection and loss function. We denote the feature from the Transblock1 and Convblock1 as $f_1 \in \Re^{256\times64\times64}$, the feature from the Transblock2 and Convblock2 as $f_2 \in \Re^{256\times32\times32}$, and the feature from the Transblock3 and Convblock3 as $f_3 \in \Re^{256\times16\times16}$. Each of these three level features can be used alone to predict. Moreover, the four components of $\mathcal{L}_{\text{total}}$ are investigated. Thus, six schemes shown in Table V are tested.

The following observations can be obtained from Table V:

- *Feature Level Selection.* Schemes 1, 2, and 3 are designed to validate the selection of feature levels, and are all trained with $\mathcal{L}_{adv} + \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{mask}}$. Comparing the three schemes, it can be found that multi-level features can improve the localization performance on pristine, source and target regions dramatically. This is due to that all of f_1, f_2, and f_3 can supply features of different scales, which is beneficial to the localization of objects in all sizes.
- *Loss Function.* The results of Scheme 4 given in Table V show that the network could not obtain satisfactory results on CMSTD when only using $\mathcal{L}_{adv}$ loss. With using $\mathcal{L}_{\text{MSE}}$, Scheme 5 can greatly improve the performance of source/target distinguishment. By adding $\mathcal{L}_{\text{mask}}$, Scheme 3 can further improve the localization accuracy of the soure/target regions. This is due to that $\mathcal{L}_{\text{mask}}$ loss focuses on the green and red regions that correspond to the source and target regions, respectively. Finally, compared with Scheme 3, the use of $\mathcal{L}_{simi}$ in the proposed scheme can further improve the CMSTD results, especially the localization results of the source regions. The $\mathcal{L}_{simi}$ loss improves the detection performance of source regions by taking the similarity between the predicted binary mask and the binary ground-truth mask into consideration.

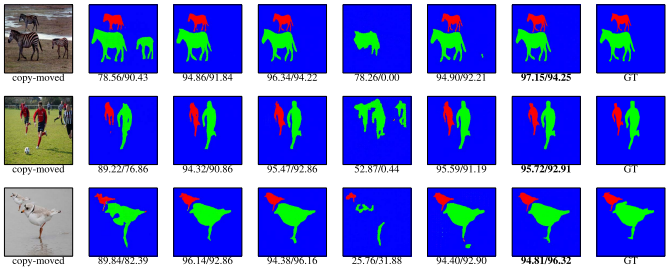

Fig. 10. CMSTD results of different schemes (F1-scores on source/target regions (%)). The first column is copy-move forged images; the second to the seventh columns represent the localization maps obtained by Scheme 1, Scheme 2, Scheme 3, Scheme 4, Scheme 5, and the proposed method, respectively; the last column corresponds to the ground-truth mask of the first column. The best results are highlighted in bold.

Fig. 10 further presents several CMSTD maps with the above-mentioned six schemes. It can be seen from Fig. 10 that the schemes with f_2 and f_3 can obtain better localization results than those only with f_1. The Scheme 4 only with $\mathcal{L}_{adv}$ could hardly localize the target regions. The $\mathcal{L}_{\text{MSE}}$ loss can distinguish the source and target regions with extra false positive rates. The $\mathcal{L}_{\text{mask}}$ loss concentrates on the source and target regions, and obtains better performance. Finally, the $\mathcal{L}_{simi}$ loss takes advantages of the similarity in source and target regions, and improves the localization results in source regions.

4) Robustness Analysis: The copy-move forged images may undergo some post-processing operations mentioned in Section IV-A.1, such as JC, NA, IB, BC, CR, and CA.

TABLE IV

ABLATION STUDIES ON TRANSFORMER NETWORK AND FEATURE COUPLING. THE CMSTD RESULTS ARE EVALUATED BY PRECISION, RECALL, AND F1-SCORE (%) ON PRISTINE, SOURCE, AND TARGET REGIONS ON USCISI COPY-MOVE DATASET; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

types	precision			recall			F1-score			average accuracy
	pristine	source	target	pristine	source	target	pristine	source	target	
Our-CNN	97.93	77.95	84.05	98.78	76.37	80.20	98.33	76.04	81.09	98.05
Our-transformer	96.32	69.39	74.00	98.73	63.51	68.33	97.45	63.66	69.39	96.97
CNN-T with FCLdown	97.02	69.63	77.83	98.44	67.25	74.43	97.68	66.94	74.56	97.29
CNN-T with FCLup	98.28	81.88	85.61	98.89	79.71	83.72	98.57	79.96	83.80	98.34
Proposed	**98.71**	**86.12**	**87.10**	**99.14**	**84.37**	**87.44**	**98.92**	**84.45**	**86.36**	**98.66**

TABLE V

EFFECT OF FEATURE SELECTION AND LOSS FUNCTION ON F1-SCORES (%) OF CMSTD ON USCISI DATASET; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

		Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Proposed
Feature level	f_1	✓	✓	✓	✓	✓	✓
	f_2		✓	✓	✓	✓	✓
	f_3			✓	✓	✓	✓
Loss function	$\mathcal{L}_{adv}$	✓	✓	✓	✓	✓	✓
	$\mathcal{L}_{\mathrm{MSE}}$	✓	✓	✓		✓	✓
	$\mathcal{L}_{\mathrm{mask}}$	✓	✓	✓			✓
	$\mathcal{L}_{simi}$						✓
F1-score	pristine	94.80	97.74	98.82	89.54	98.20	**98.92**
	source	46.28	68.25	83.38	23.14	73.73	**84.45**
	target	60.27	75.00	86.09	4.94	79.32	**86.36**

TABLE VI

F1-SCORES (%) ON PRISTINE, SOURCE, AND TARGET REGIONS AGAINST DIFFERENT POST-PROCESSING OPERATIONS ON USCISI COPY-MOVE DATASET; THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Post-processing operations		pristine	source	target
Without post-processing		98.92	84.45	86.36
QF	100	98.91	84.08	86.23
	90	98.82	82.53	84.36
	80	98.70	80.52	82.47
	70	98.61	79.20	81.04
σ	0.1	97.51	61.87	62.62
	0.2	95.67	33.70	32.91
	0.3	94.69	16.81	16.12
	0.4	94.15	7.91	7.34
W	3×3	98.47	75.84	79.05
	5×5	97.64	62.20	64.55
	7×7	96.66	48.34	48.68
S_bc	0.8	98.94	84.50	86.37
	0.9	**98.94**	**84.69**	**86.69**
	1.1	98.91	84.19	86.25
	1.2	98.88	83.82	85.80
S_cr	8	98.88	83.52	85.41
	16	98.80	82.33	84.06
	32	98.58	78.89	80.83
	64	98.00	70.55	72.27
(B_lower, B_upper)	$(0.01, 0.95)$	98.93	84.45	86.22
	$(0.01, 0.9)$	98.92	84.28	86.00
	$(0.01, 0.85)$	98.91	84.07	85.81
	$(0.01, 0.8)$	98.90	83.91	85.66

An image forensic method should be robust against commonly used post-processing operations. In this subsection, we validate the robustness against several post-processing operations of the proposed method.

Table VI shows the F1-scores obtained by the proposed method against several post-processing operations. From Table VI, it can be seen that the proposed method is less robust against NA and IB. The reasons are as follows: first, NA will add noise in images, which may cause interference for GAN; second, IB will lead to loss of textures in images when blurring the entire image. In addition, even though the performance on CMSTD of the proposed method decreases to some degree, the proposed method is more robust against JC, BC, CR, and CA than against NA and IB. This is due to that JC, BC, CR, and CA change the intensity of the images, but do not change the contents of images. It should be noted that for BC, when $S\_bc < 1$, *i.e.*, the images are weighted towards brighter outputs, the performance can even be improved by BC. This may be due to that brighting images properly can improve the saliency of objects in images, which is beneficial to object detection.

V. CONCLUSION

In this paper, a novel CNN-T GAN is proposed for copy-move forgery localization and source/target distinguishment. The generator consists of a CNN branch, a transformer branch, and several feature coupling layers. The CNN branch and transformer branch extract local features and global representations of the copy-move regions, respectively. Feature coupling layers are designed to integrate the features in these two branches. To enhance the performance of generator, the generator and the discriminator are alternatively trained, and the training will not stop until the discriminator cannot discriminate the generated masks correctly. Moreover, a new Pearson correlation layer is introduced to extract the similarity in copy-move forgery images. Finally, a mask loss and a similarity loss are designed to focus on the source and target regions. Ablation experiments have verified the effectiveness of the designing of network architecture, the feature selection, and loss function. Extensive experimental results have shown that the proposed method outperforms the state-of-the-art methods in both CMFL and CMSTD on three commonly used copy-move forgery datasets.

REFERENCES

[1] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy–move forgery," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1053–1056.

[2] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.

[3] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A patchmatch-based dense-field algorithm for video copy–move detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 669–682, Feb. 2019.

[4] E. A. A. Vega, E. G. Fernández, A. L. S. Orozco, and L. J. G. Villalba, "Copy–move forgery detection technique based on discrete cosine transform blocks features," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4713–4727, May 2021.

[5] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A sift-based forensic method for copy–move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[6] B. L. Shivakumar and S. S. Baboo, "Detection of region duplication forgery in digital images using SURF," *Int. J. Comput. Sci.*, vol. 8, no. 4, pp. 199–205, 2011.

[7] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy–move forgery detection scheme," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.

[8] C. Pun, X. Yuan, and X. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1705–1716, Aug. 2015.

[9] B. Yang, X. Sun, H. Guo, Z. Xia, and X. Chen, "A copy–move forgery detection method based on CMFD-SIFT," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 837–855, Jan. 2018.

[10] V. Manu and B. M. Mehtre, "Detection of copy–move forgery in images using segmentation and SURF," in *Advances in Signal Processing and Intelligent Recognition Systems*. Cham, Switzerland: Springer, 2016, pp. 645–654.

[11] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Comput. Sci., Dartmouth College, Hanover, NH, USA, Tech. Rep., TR2004-515, 2004.

[12] L. Li, S. Li, H. Zhu, S.-C. Chu, J. F. Roddick, and J.-S. Pan, "An efficient scheme for detecting copy–move forged images by local binary patterns," *J. Inf. Hiding Multimedia Signal Process.*, vol. 4, no. 1, pp. 46–56, 2013.

[13] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring duplicated regions in natural images," *IEEE Trans. Image Process.*, early access, Mar. 25, 2019, doi: 10.1109/TIP.2010.2046599.

[14] L. Su, C. Li, Y. Lai, and J. Yang, "A fast forgery detection algorithm based on exponential-Fourier moments for video region duplication," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 825–840, Apr. 2018.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[16] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[17] J. Yang, S. Xiao, A. Li, W. Lu, X. Gao, and Y. Li, "MSTA-Net: Forgery detection by generating manipulation trace based on multi-scale self-texture attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4854–4866, Jul. 2022, doi: 10.1109/TCSVT.2021.3133859.

[18] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image tampering localization using a dense fully convolutional network," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2986–2999, 2021.

[19] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang, and Y. Zhou, "Multi-task SE-network for image splicing localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4828–4840, Jul. 2022, doi: 10.1109/TCSVT.2021.3123829.

[20] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets," *IEEE Consum. Electron. Mag.*, vol. 11, no. 2, pp. 42–50, Mar. 2022.

[21] H. Wu and J. Zhou, "IID-Net: Image inpainting detection network via neural architecture search and attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1172–1185, Mar. 2022.

[22] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy–move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[23] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Image copy–move forgery detection via an end-to-end deep neural network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1907–1915.

[24] 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. (CVPR)*, Jun. 2019, pp. 9543–9552.

[25] J.-L. Zhong and C.-M. Pun, "An end-to-end dense-InceptionNet for image copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2134–2146, 2020.

[26] 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.

[27] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy–move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 168–184.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[29] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y.-Q. Shi, "A serial image copy–move forgery localization scheme with source/target distinguishment," *IEEE Trans. Multimedia*, vol. 23, pp. 3506–3517, 2021.

[30] 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. (CVPR)*, Jun. 2020, pp. 4676–4685.

[31] M. Barni, Q.-T. Phan, and B. Tondi, "Copy move source-target disambiguation through multi-branch CNNs," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1825–1840, 2021.

[32] D. A. Lisin, M. A. Mattar, M. B. Blaschko, E. G. Learned-Miller, and M. C. Benfield, "Combining local and global image features for object class recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops*, Jun. 2005, pp. 47–55.

[33] 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.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[34] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 367–376.

[35] F. Wang et al., "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 3156–3164.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[37] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[38] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[39] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[40] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3588–3597.

[41] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3286–3295.

[42] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[43] B. Wu et al., "Visual transformers: Token-based image representation and processing for computer vision," 2020, *arXiv:2006.03677*.

[44] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[45] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[46] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 558–567.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[48] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[51] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "CoMoFoD: New database for copy–move forgery detection," in *Proc. Int. Symp. Electron.*, Mar. 2013, pp. 49–54.

Yulan Zhang received the M.S. degree from Wuyi University, Jiangmen, China, in 2017, and the Ph.D. degree in computer application technology from the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, China, in 2022. She is currently a Lecturer with Huizhou University. Her research interests mainly include multimedia security, deep learning, and image processing.

Guopu Zhu (Senior Member, IEEE) received the B.S. degree in transportation from Jilin University, China, in 2002, and the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, China, in 2004 and 2007, respectively. He is currently a Professor with the Harbin Institute of Technology. He has authored or coauthored more than 50 articles in peer-reviewed international journals. His main research areas are multimedia security, image processing, and control theory. He serves as an Associate Editor for several journals, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE SYSTEMS JOURNAL, *Journal of Information Security and Applications*, and *Electronics Letters*.

Xing Wang received the B.S. and M.S. degrees in computer science from Northwest University, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from the Harbin Institute of Technology, China, in 2015. He is currently an Assistant Researcher with the Harbin Institute of Technology. His main research areas are machine learning, ocial network data mining, and network security.

Xiangyang Luo received the B.S., M.S., and Ph.D. degrees from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2001, 2004, and 2010, respectively. He is the author or coauthor of more than 200 refereed international journals and conference papers. He is currently a Professor with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research interests include network security and multimedia security.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of "Highly Cited Researchers" in 2020 and 2021. He received the Third Price of Macao Natural Science Award as a Sole Winner in 2020 and a co-recipient in 2014. He serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Hongli Zhang (Member, IEEE) received the B.Sc. degree in computer science from Sichuan University, Chengdu, China, in 1994, and the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1999. She is currently a Professor with the School of Cyberspace Science, Harbin Institute of Technology. Her research interests include network and computer security, network modeling, and parallel processing.

Ligang Wu (Fellow, IEEE) received the B.S. degree in automation from the Harbin University of Science and Technology, China, in 2001, and the M.E. degree in navigation guidance and control and the Ph.D. degree in control theory and control engineering from the Harbin Institute of Technology, China, in 2003 and 2006, respectively. He was a Research Associate/Senior Research Associate with The University of Hong Kong, the City University of Hong Kong, and the Imperial College London. He is currently a Professor with the Harbin Institute of Technology. He has published seven research monographs and more than 200 research articles in internationally refereed journals. His current research interests include analysis and design for cyber-physical systems, robotic and autonomous systems, intelligent systems, and power electronic systems.

His awards and recognitions include the National Science Fund for Distinguished Young Scholar, the China Young Five-Four Medal, has been the Distinguished Professor of Chang Jiang Scholar, and the Highly Cited Researcher since 2015. He also serves as an Associate Editor for a number of journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, and *IET Control Theory and Applications*.