Reality Transform Adversarial Generators for Image Splicing Forgery Detection and Localization

Xiuli Bi Zhipeng Zhang Bin Xiao*
Chongqing University of Posts and Telecommunications
Chongqing, China
bixl@cqupt.edu.cn S190201087@stu.cqupt.edu.cn xiaobin@cqupt.edu.cn

Abstract

When many forgery images become more and more realistic with help of image editing tools and convolutional neural networks (CNNs), authenticators need to improve their ability to verify these forgery images. The process of generating and detecting forgery images is the same as the principle of Generative Adversarial Networks (GANs). In this paper, since the retouching progress of forgery images requires to suppress the tampering artifacts and to keep the structural information, we consider this retouching progress as an image style transform, and then propose a fake-to-realistic transform generator G_T. For detecting the tampered regions, a localization generator G_M is proposed too, which is based on a multi-decoder-single-task strategy. By adversarial training two generators, the proposed α-learnable whitening and coloring transform (α-learnable WCT) block in G_T automatically suppress the tampering artifacts in the forgery images. Meanwhile, the detection and localization abilities of G_M will be improved by learning the forgery images retouched by G_T. The experiment results demonstrate that the proposed two generators in GAN can simulate confrontation between the faker and the authenticator well; the localization generator G_M outperforms the state-of-the-art methods in splicing forgery detection and localization on four public datasets.

1. Introduction

Cyberspace has experienced explosive growth, and countless images are uploaded to the Internet every day, which includes a lot of forgery images. Since forgery images can be easily produced by user-friendly image editing tools and used to create fake news and rumors, it is necessary to develop more effective methods for image forgery detection and localization. For the image forgeries, copy-move and removal forgery require a single source image, but splicing forgery copies and pastes regions from one or more source images onto the target image. Fig. 1-(a) demonstrates the two examples of splicing forgery images. In this paper, our work focuses on detecting the splicing forgery images and then locate the tampered regions of these detected images.

The image splicing forgery detection methods can be summarized into two main categories, methods based on conventional features extraction[19, 6, 14, 21] and methods based on convolutional neural networks (CNNs)[26, 28, 9, 1, 25, 24, 2, 13]. Most conventional methods focus on a particular image fingerprint that is caused by imaging processing and post-processing. Because the particular image fingerprint is easy to be influenced by post-processing, such as JPEG compression, down-sampling, and mean filtering, many conventional methods are easy to fail. Fig. 1-(c) shows the experiments results of a conventional method[19].

CNN-based methods can be further divided into patch-based nethods and end-to-end methods. For patch-based methods, since the final detection result is derived from the decisions of image patches, the detected results are generally composed of square white blocks, or only the patches on boundaries of the tampered regions are detected. For end-to-end methods, if the tampering artifacts are suppressed and reduced by the faker, it is difficult for end-to-end methods to detect tampered regions. Fig. 1-(d) shows the experimental results of a CNN-based method[1].

To solve these problems, V. Kniaz et al.[13] introduced a GAN-based method named Mixed Adversarial Generators (MAG) for image splicing forgery detection and localization. However, MAG requires class segmentations to retouch splicing forgery images, which consumes a host of computational resources. Furthermore, since the prediction of both tampered region and class segmentation is generated in a single decoder network, some untampered semantic regions, who are similar to the tampered regions in the ground truth, will be easily detected as the tampered regions, as the experiment results demonstrated in Fig. 1-(e)

* Corresponding Author

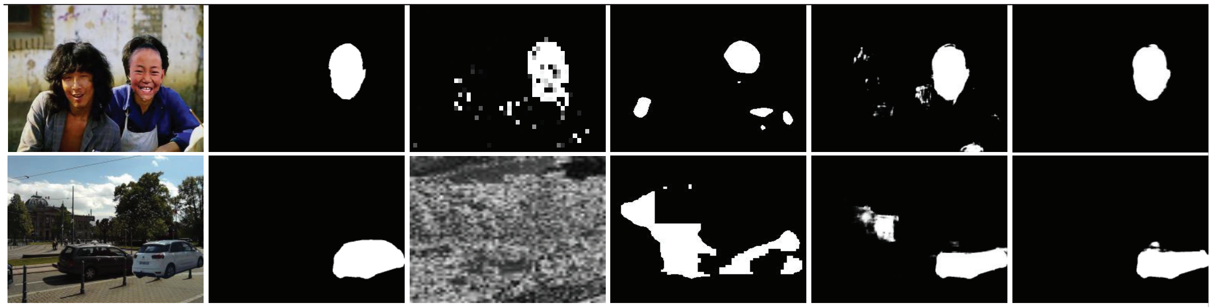


Figure 1. Two examples of splicing forgery images and corresponding localization results of four types of detection methods. (a) The splicing forgery images. (b) Ground truths of tampered regions. (c) The detection result of a conventional detection method ADQ[19]. (d) The detection result of a CNN-based detection method LSTM[1]. (e) The detection result of a GANs-based detection method MAG[13]. (f) The detection result of the proposed RTAG.

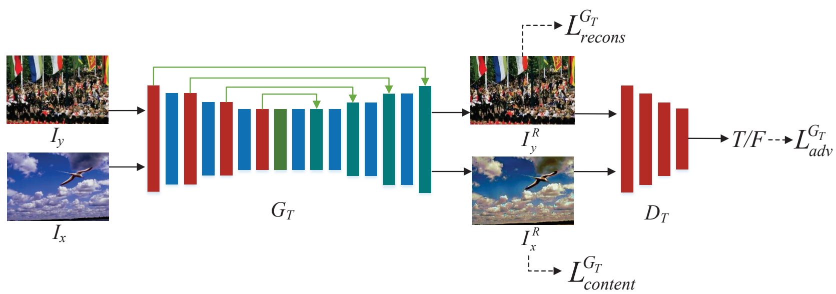
In this work, we rethink the principle of generating and detecting forgery images. When image fakers retouch the forgery images more realistic, they need to hide the tampering artifacts, while keeping the structural information of the forgery image unchanged. The retouching progress of forgery images is the same as the task of image style transform. Thus, we consider the retouching process of forgery image as the image style transform, which transforms splicing forgery images from a 'fake style' to a 'real style'. Based on this insight, we propose the fake-to-realistic transform generator G_T to simulate the faker. In contrast, the authenticators need to detect the tampered regions from these more 'real style' splicing forgery images, so a localization generator G_M with the multi-decoder-single-task (MDST) strategy is proposed. In the adversarial training between G_T and G_M, for progressively suppressing the tampering artifacts of the splicing forgery image, we propose α-learnable whitening and coloring transform blocks (α0.-learnable WCT) based on WCT[16] in G_T. While, through the multi-decoder-single-task strategy (MDST), G_M will improve its detection and localization ability by learning fewer tampering artifacts from the retouched images. Moreover, the discriminators D_T and D_M will qualify the outputs of G_T and G_M. The GAN framework for adversarial training G_M and G_T is named Reality Transform Adversarial Generators (RTAG), the two examples of detection results are presented in Fig. 1-(f).

The main contribution of this work can be summarized as follows: (1) The retouching progress of forgery images is considered as the image style transform in this paper. Based on this insight, a fake-to-realistic transform generator G_T is proposed, which applies the α-learnable WCT blocks to automatically progressively retouch the splicing forgery images more realistic; (2) For detecting the tampered regions by fewer tampering artifacts, a localization generator G_M is proposed according to the multi-decoder-single-task strategy; (3) By adversarial training G_T and G_M in the GAN framework, the localization generator G_M will detect and locate the tampered regions even the splicing forgery images has fewer tampering artifacts.

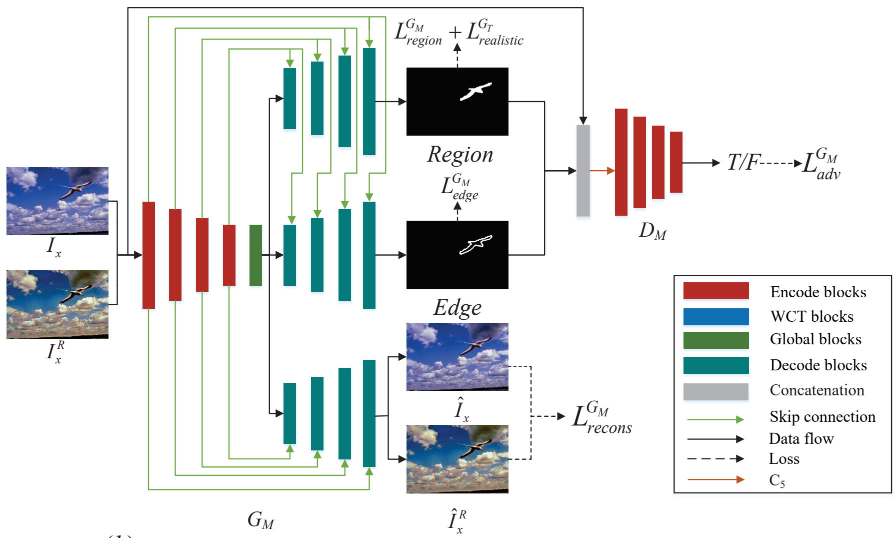
2. Related Work

Most existing image splicing forgery detection methods can be divided into the methods based on conventional features extraction and the methods based on CNN. For conventional methods, Liu et al.[20] proposed aligned double quantization detection (ADQ), which utilizes the distribution of the image discrete cosine transform (DCT) coefficients to distinguish the tampered regions. Krawetz et al.[14] proposed an error level analysis method (ELA), which is intended to find the compression error difference between the forgery regions and the real regions. Cozzolino et al. [4] proposed a method for blind detection and localization of splicing that uses cooccurrence based featrues, it requires no prior knowledge of the host camera, the splicing, or their processing history.

For CNN-based methods, many CNN-based methods mainly learn the differences between image patches and then determine whether an image patch was manipulated or not. For instance, Bappy et al.[1] proposed a network that contains a long short-term memory network (LSTM) and an encoder-decoder architecture network. This proposed network exploits resampling features from image patches to detect tampered regions. Xiao et al.[26] proposed a two-stage detection network, which learns the differences of the image properties between un-tampered and tampered regions from image patches with different scales. To directly learn from the whole images and locate the tampered regions, some end-to-end splicing forgery detection methods are proposed. Wu et al.[25] presented ManTra-Net, which contains an image manipulation trace feature extraction network and a local anomaly detection network. Bi et al.[2] proposed a ringed residual structure U-Net (RRU-Net), which is an end-to-end image essence attribute segmentation network without any pre-processing and post-processing. The end-to-end methods can detect tampered regions by learning various tampering artifacts



(a) Fake-to-realistic transform generator G_T and the corresponding Discriminator D_T.



(b)Localization generator G_M and the corresponding discriminator D_M.

Figure 2. The pipeline of RTAG. I_x and I_y denote randomly paired splicing forgery image and authentic image, I_x^R is the retouched image, I_y^R is the reconstruction of I_y, $\hat{I}_x$ and $\hat{I}_x^R$ denote the reconstructions of I_x and I_x^R. L denotes the loss function, the superscript and subscript of L illustrate the constraints of the network.

directly from the whole images. Hu et al.[9] proposed a spatial pyramid attention network(SPAN) architecture that compares patches through the local self-attention block on multiple scales.

GAN is a special framework of CNN, although recent researches[11, 15, 29, 22] have revealed that GANs can achieve amazing success in multiple tasks, GAN-based image splicing forgery detection is still rare. V. Kniaz et al.[13] introduced MAG for image splicing forgery detection and localization. MAG adversarial trains a retoucher to retouch the fake images and an annotator to predict the tampered regions. MAG requires class segmentations to reconstruct and retouch splicing forgery images, which consumes a host of computational resources and the qulity of retouched images are not realistic enough.

3. Proposed Method

In the proposed RTAG framework, generating and detecting splicing forgery image is considered as an adversarial game between a fake-to-realistic transform generator G_T and a localization generator G_M. G_T progressively retouches splicing forgery images from a 'fake style' to a 'real style', then G_M needs to detect the tampered regions by learning the images retouched by G_T, these retouched images have fewer tampering artifacts. By adversarial training of G_T and G_M, the detection and localization abilities of G_M will be enhanced. This RTAG framework is shown in Fig. 2. Here, G_T and G_M follow the objective function $V(G_M)$ and $V(G_T)$:

$$\begin{aligned}\min_{G_M} V(G_M) &= \frac{1}{3}\mathbb{E}_{x\sim\mathcal{X}}[(G_M(x)-m)^2] \\ &+ \frac{1}{3}\mathbb{E}_{x\sim\mathcal{X}}[(G_M(G_T(x))-m)^2] \\ &+ \frac{1}{3}\mathbb{E}_{y\sim\mathcal{Y}}[(G_M(y)-0_{W,H})^2], \\ \min_{G_T} V(G_T) &= \mathbb{E}_{x\sim\mathcal{X}}[(G_M(G_T(x))-0_{W,H})^2].\end{aligned} \tag{1}$$

Where x denotes the values in a splicing forgery image I_x; y denotes the values of in an authentic image I_y; $\mathcal{X}$ and $\mathcal{Y}$ denote the forgery domain and the authentic domain separately; m represents the ground truth of splicing forgery image I_x; $0_{W,H}$ is a black image that represents the authentic image I_y does not have any tampered regions.

3.1. Fake-to-realistic Transform Generator G_T

In MAG[13], the annotator generator makes sure the retouched images recognizable by generating the prediction of tampered regions and class segmentations. Generating class segmentations not only need additional computational sources but also may disturb the task of localization. Thus, in this paper, we consider the retouching progress as an image style transform. The fake-to-realistic transform generator G_T is expected to transform splicing forgery images I_x to realistic images while keeping the structural information of I_x unchanged. As shown in Fig. 2-(a), we propose the fake-to-realistic transform generator G_T that applies WCT[16] block between certain layers of the U-Net[23], and a global block is inserted between encoder and decoder. The structure of corresponding discriminator D_T is a conditional discriminator same as PatchGAN[10] architecture.

In generator G_T, a splicing forgery image I_x and an authentic image I_y are randomly paired, and they input the first encoding block to generate feature map f_x and f_y. WCT block directly matches feature map f_x to the covariance matrix of feature map f_y. WCT firstly peels off the style features in f_x, such as colors, contrast, etc. Then the transform feature map f_{xy} will be obtained by filling the peeled feature map f_x with the style features in f_y. Finally, the transform feature map f_{xy} is blended with feature map f_x by Eq. (2).

$$\hat{f}_{xy} = \alpha f_{xy} + (1 - \alpha) f_x \tag{2}$$

Where $\hat{f}_{xy}$ denotes the output feature of the first WCT block. $\alpha \in [0, 1]$ denotes the weight that controls the degree of retouching. Then, $\hat{f}_{xy}$ will be the input feature f_x of the next block.

The previous works[16, 17, 27] only manually set the value of α. However, if α is too high, the structural information in the retouched images may be lost, the retouched images always carry black plaques and the edges of the retouched image are blurred with a color halo. Moreover, if α is near to 1, the features of the splicing forgery image I_x are almost replaced by the authentic image, G_M will learn nothing to distinguish the tampered regions. On the other hand, if α is too low, the WCT will lose its function. Therefore, it's difficult to find a suitable value of α manually. To address this issue, we propose α-learnable WCT block, the structure of this block is shown in Fig. 3. α-learnable WCT block can determine the best value of α by learning the feature map f_x and f_y. The qualitative result of α-learnable WCT block is shown in Fig. 4-(f). Based on the experiment results for evaluating α-learnable WCT block, we believe it can be further used in other end-to-end style transform networks.

Because the features of the forgery image I_x should be

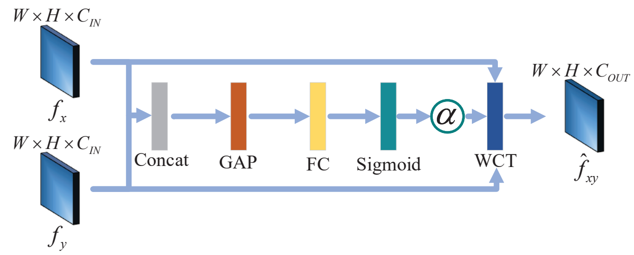


Figure 3. The structure of α-learnable WCT block. GAP denotes the global average pooling[18], FC denotes the fully connected layer.

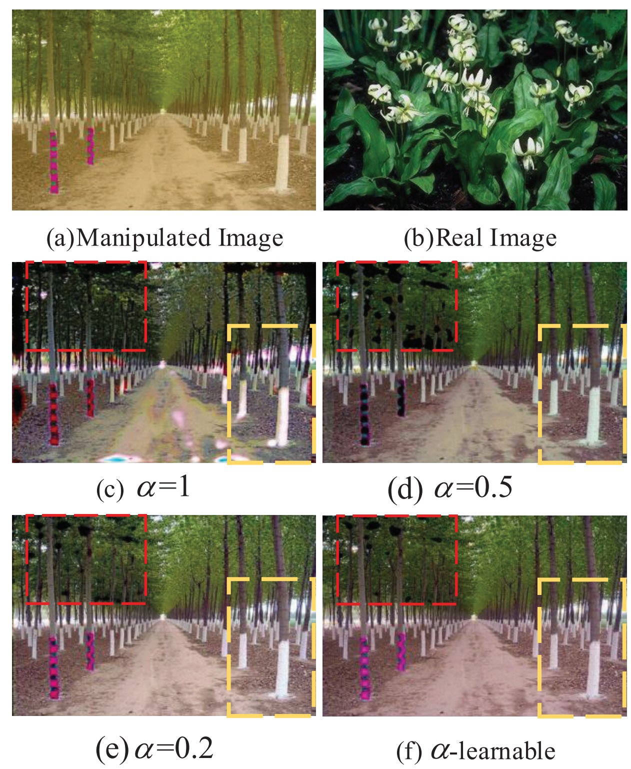


Figure 4. Qualitative results for α-learnable blocks and α-fixed blocks. The red and yellow dotted boxes demonstrate that black plaques and color halo blur are significantly reduced in the outputs of α-learnable WCT blocks.

replaced by the features of the authentic image I_y from a global view. Based on the global block proposed in[3], we modify this global block by applying the convolutions of different receptive fields to extract multi-level features, the modified global block can get more comprehensive global features. It is inserted between encoder and decoder to extract global features to enhance the realistic transform efficiency. The structure of the modified global block is shown in Fig. 5.

For implementing multi-task in G_T, G_T uses a mixed loss function, which consists of four parts: $L^{G_T}_{content}$, $L^{G_T}_{recons}$, $L^{G_T}_{realistic}$ and $L^{G_T}_{adv}$. G_T should not change the

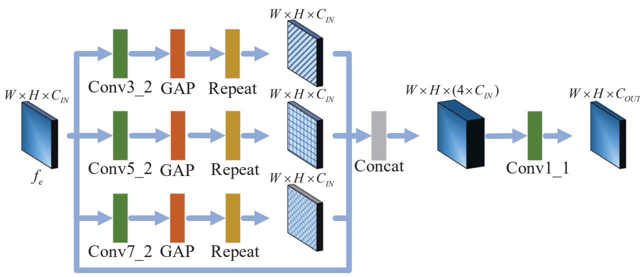


Figure 5. The structure of the modified global block. f_e denotes the feature map output by the encoder, the convolutional layer is denoted as Conv(kernel size)_(strides).

structural information of the forgery images while retouching the forgery images, so the content loss function $L^{G_T}_{content}$ is defined as Eq. (3).

$$L^{G_T}_{content} = \mathbb{E}_{x \sim \mathcal{X}}[\|x - G_T(x)\|_1] \tag{3}$$

In Eq. (3), x denotes the values of a splicing forgery image I_x, $\|\cdot\|_1$ denotes the ℓ_1 norm. Since G_T needs to reconstruct the authentic image I_y, and keep that the reconstructed image I^R_y is the same as the authentic image I_y. So, the loss function of reconstruction $L^{G_T}_{recons}$ is applied to reinforce the reconstruction ability of G_T, $L^{G_T}_{recons}$ is defined in Eq. (4).

$$L^{G_T}_{recons} = \mathbb{E}_{y \sim \mathcal{Y}}[\|y - G_T(y)\|_1] \tag{4}$$

Where y denotes the values of the authentic image I_y. Since G_T is adversarial trained against G_M, when the output retouched image I^R_x is more realistic, the prediction of G_M is harder. Therefore, we use a realistic loss function $L^{G_T}_{realistic}$ conducted by Eq. (5)

$$L^{G_T}_{realistic} = \mathbb{E}_{x \sim \mathcal{X}}[\|0_{W,H} - G_M(G_T(x))\|_1] \tag{5}$$

Finally, we use the least-squares equation, which is defined in Eq. (6), as adversarial loss function of corresponding discriminator D_T. The adversarial loss function $L^{G_T}_{adv}$ will make the output retouched image I^R_x more realistic.

$$L^{G_T}_{adv} = \frac{1}{2}\mathbb{E}_{x \sim \mathcal{X}}[(D_T(G_T(x)) - 1)^2]. \tag{6}$$

The final loss function can be summarized as:

$$\begin{aligned} L_T &= \lambda^{G_T}_{content} L^{G_T}_{content} + \lambda^{G_T}_{recons} L^{G_T}_{recons} \\ &+ \lambda^{G_T}_{realistic} L^{G_T}_{realistic} + \lambda^{G_T}_{adv} L^{G_T}_{adv}, \\ \lambda^{G_T}_{content} &= 1, \lambda^{G_T}_{recons} = 0.5, \lambda^{G_T}_{realistic} = 0.5, \lambda^{G_T}_{adv} = 1, \end{aligned} \tag{7}$$

Where λ denotes the weights of each loss function, the values of each weight is set by the experience of the experiments.

3.2. Localization Generator G_M

MAG[13] used U-Net[23] to generate the detected region, the detected edges, and class segmentation, as shown in Fig. 1-(e), a single encoder works for multiple tasks (SDMT) will cause low *precision* of detection result. Because the untampered semantic regions, who are similar to semantic classes of the tampered regions, will be detected as the tampered regions. Thus, we replace the SDMT with the multi-decoder-single-task strategy. MDST can make each decoder of the network to focus on a single task and avoid the interference between tasks.

As shown in Fig. 2-(b), G_M's structure is a modified U-Net, which has three encoders. The structure of D_M is the same as the structure of D_T. While the forgery image I_x is retouched by G_T, it's more difficult to distinguish the tampered regions by image properties, such as colors, contrast, etc. Thus, an edge decoder is needed to make G_M focus more on the edges between the tampered regions and untampered regions. To make sure the hide code output by the encoder is comprehensive and meaningful, a reconstruction decoder is used to regularize the shared encoder. Because the localization task is a global classification problem that needs to compare the features of different regions globally, the modified global block, which is used in G_T, is used between encoder and decoders too.

G_M is also trained by a mixed loss function that consists of four parts: region loss $L^{G_M}_{region}$, edge loss $L^{G_M}_{edge}$, reconstruction loss $L^{G_M}_{recons}$, and adversarial loss $L^{G_M}_{adv}$. The region loss $L^{G_M}_{region}$ is the loss function of the decoder who predicts the tampered regions in the ground truth. $L^{G_M}_{region}$ calculates the distance between the detected regions and the real tampered regions by a binary cross-entropy function, it is conducted by Eq. (8).

$$\begin{aligned} L^{G_M}_{region} = &-\frac{1}{2 \times W \times H}\mathbb{E}_{x \sim \mathcal{X}}[GT_{region} \cdot \log(G_M(x)) \\ &+ (1 - GT_{region}) \cdot \log(1 - G_M(x)) \\ &+ GT_{region} \cdot \log(G_M(G_T(x))) \\ &+ (1 - GT_{region}) \cdot \log(1 - G_M(G_T(x)))] \end{aligned} \tag{8}$$

In Eq. (8), W and H denote the width and height of the ground truth, GT_{region} is the tampered regions in the ground truth of forgery image I_x. $G_T\,(x)$ is the retouched image I^R_x. $G_M\,(x)$ is the detected regions of forgery image I_x. $G_M(G_T(x))$ is the detected regions of retouched image I^R_x. The edge loss function $L^{G_M}_{edge}$ can be calculated as the formula $L^{G_M}_{region}$. Since the edge of tampered regions contains few pixels, which will cause the loss result unstable and feedback insufficient. To address this issue, $L^{G_M}_{edge}$ is particularly defined as Eq. (9). We add external weights to the binary cross-entropy function to regularize the edge loss.

$$L_{edge}^{G_M} = -\frac{1}{2 \times W \times H}\mathbb{E}_{x\sim\mathcal{X}}[\omega_{pos} \cdot GT_{edge} \cdot \log(G_M(x)) + \omega_{neg} \cdot (1 - GT_{edge}) \cdot \log(1 - G_M(x)) + \omega_{pos} \cdot GT_{edge} \cdot \log(G_M(G_T(x))) + \omega_{neg} \cdot (1 - GT_{edge}) \cdot \log(1 - G_M(G_T(x)))] \tag{9}$$

Where, $G_M(x)$ is the detected edges of the tampered regions in forgery image I_x, $G_M(G_T(x))$ is the detected edges of the tampered regions in the retouched image I_x^R. GT_{edge} denotes the edges of tampered regions in the ground truth. ω_{pos} and ω_{neg} are the weights that make G_M focus more on the edges of the tampered regions. In the experiments below, we set $\omega_{pos} = 1.5$ and $\omega_{neg} = 0.5$. For reconstruction decoder in G_M, the loss function $L_{recons}^{G_M}$ is calculated by Eq. (10).

$$L_{recons}^{G_M} = \frac{1}{2}\mathbb{E}_{x\sim\mathcal{X}}[\|x - G_M(x)\|_1 + \|G_T(x) - G_M(G_T(x))\|_1] \tag{10}$$

In Eq. (10), $G_M(x)$ denotes the reconstructed image $\hat{I}_x$, $G_M(G_T(x))$ is the reconstructed image $\hat{I}_x^R$. Finally, an adversarial loss $L_{adv}^{G_M}$ is proposed to avoid blurry outputs, which is defined as Eq. (11).

$$L_{adv}^{G_M} = \frac{1}{2}(D_M(C_5) - 1)^2, \tag{11}$$

Where, C_5 is a concatenation input of three parts: the detected regions of forgery image I_x, the detected edges of the tampered regions in forgery image I_x, the splicing forgery image I_x or the retouched image I_x^R. The subscript of C_5 is the channel number of the concatenation. Finally, the The final loss function of G_M is computed as follow:

$$L_M = \lambda_{mask}^{G_M} L_{mask}^{G_M} + \lambda_{edge}^{G_M} L_{edge}^{G_M} + \lambda_{recons}^{G_M} L_{recons}^{G_M} + \lambda_{adv}^{G_M} L_{adv}^{G_M}, \quad \lambda_{mask}^{G_M} = 1, \lambda_{edge}^{G_M} = 1, \lambda_{recons}^{G_M} = 0.1, \lambda_{adv}^{G_M} = 0.1. \tag{12}$$

Each weight λ in L_M is set by the experience of the experiments.

4. Experiments

4.1. Datasets

For the fair comparison, we perform evaluations on four public splicing forgery image datasets: CASIA v2.0[5], Columbia[8], NIST 2016[7] and FantasticReality[13]. The details of each dataset are illustrated in Table 1. CASIA v2.0 contains three types of image forgeries: splicing, copy-move, and removal, the forgery images are post-processed by methods such as filtering and blurring. The Columbia dataset only contains splicing forgery and the tampered regions are the large meaningless smooth regions which is not post-processed. The image forgery types of NIST 2016 include splicing, copy-move, and removal, all the forgery images in the dataset are post-processed to hide visible traces of manipulation. FantasticReality contains a large number of forgery images but only splicing forgery is included, the splicing forgery images are not post-processed by any method. Because we aim to detect splicing forgery, only the splicing forgery images are selected in each dataset.

Dataset	Characteristics		
	Image Format	Forgery/Authebtic Images	Train/Test Images
CASIA v2.0	TIFF, JPEG	5123/7491	715/100
Columbia	JPEG	180/183	125/45
NIST 2016	JPEG	564/875	184/50
FantasticReality	JPEG	19422/16592	12000/1000

Table 1. Characteristics of the image splicing forgery datasets.

4.2. Evaluation Metrics

The performance of splicing localization is evaluated by mean average precision(mAP), Area Under Curve(AUC), and *F rate* defined by the following equations:

$$Precison = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \quad F = \frac{2 \times Precison \times Recall}{Precison + Recall}, \tag{13}$$

where, *TP* and *FP* denote the numbers of correctly detected and erroneously detected pixels, and *FN* is the number of falsely missed pixels.

4.3. Setup

In our experiments, RTAG is trained using an Adam[12] training optimizer with a batch size of 8, an initial learning rate of 3e-4, a decay rate of 0, and an epoch of 300. Note that, in our observation, the performance of G_T drops sharply when batch size is more than 1, and G_M's performance drops sharply when batch size is too low. Therefore, G_T is trained with one splicing forgery image and one authentic image each time, while G_M is trained with 8 splicing forgery images or 8 retouched images each time. To avoid G_T takes excessive advantage in the adversarial training, G_T is updated every 8 batches. For data augmentation, all the images are resized to 512×512. All training processes are implemented on a NVIDIA Tesla V100 (32G) GPU.

4.4. Comparisons

We compare RTAG with four state-of-the-art deep learning splicing forgery detection methods: ManTra[25],

Dataset	CASIA v2.0			Columbia			NIST			FantasticReality		
Metric	mAP	AUC	F	mAP	AUC	F	mAP	AUC	F	mAP	AUC	F
ADQ	0.293	0.698	0.476	0.344	0.637	0.536	0.096	0.319	0.296	0.221	0.511	0.409
ELA	0.054	0.306	0.158	0.302	0.595	0.475	0.081	0.301	0.243	0.267	0.587	0.398
ManTra	0.569	0.777	0.651	0.468	0.681	0.621	0.085	0.312	0.275	0.329	0.719	0.484
LSTM	0.526	0.758	0.617	0.488	0.723	0.622	0.112	0.552	0.366	0.388	0.757	0.530
C2Rnet	0.572	0.793	0.676	0.507	0.807	0.695	0.097	0.523	0.196	0.493	0.712	0.606
MAG	-	-	-	-	-	-	-	-	-	0.780	0.903	0.824
RTAG	0.707	0.888	0.815	0.796	0.860	0.823	0.531	0.776	0.623	0.910	0.965	0.936

Table 2. Experimental results of plain splicing forgery.

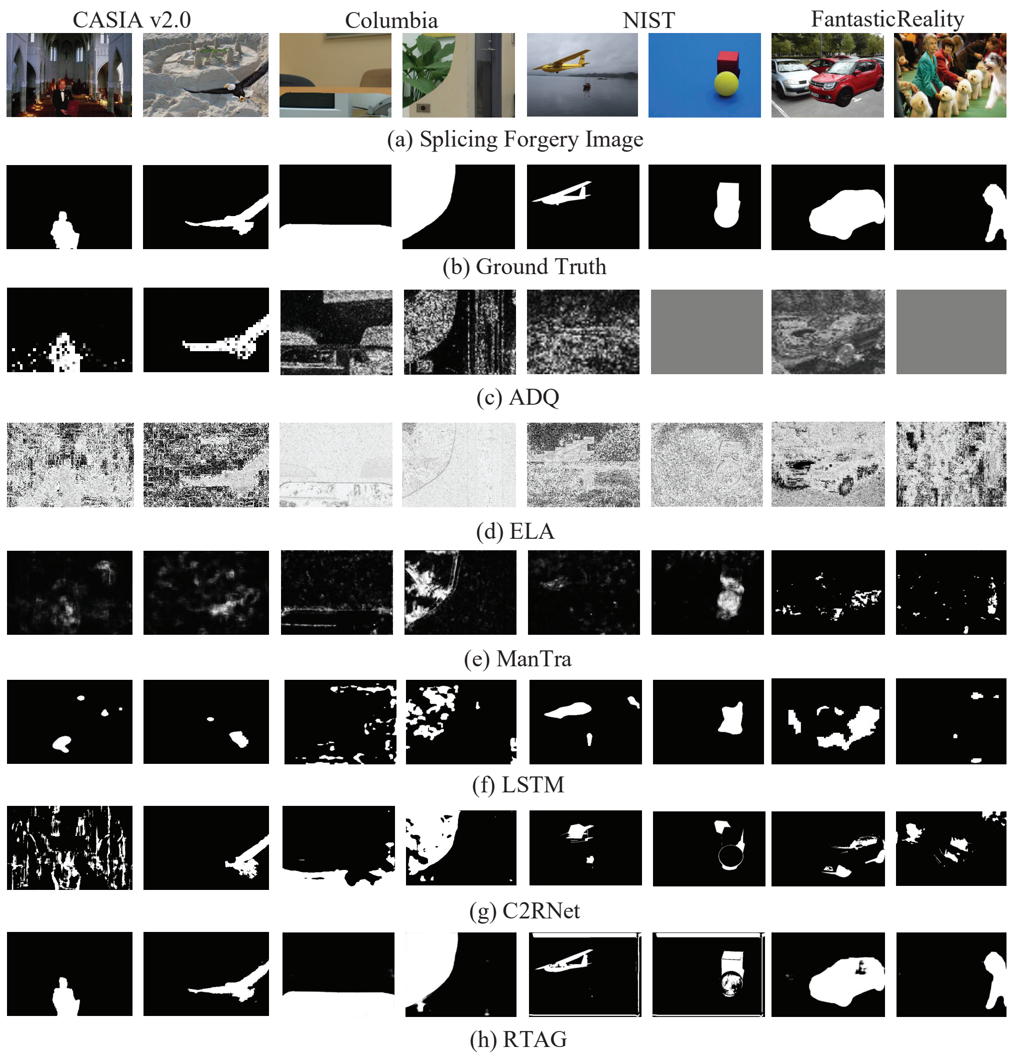


Figure 6. Qualitative results of RTAG and other state-of-the-art methods. 1^{st} and 2^{nd} columns are the results for CASIA v2.0; 3^{rd} and 4^{th} columns are the results for Columbia; 5^{th} and 6^{th} columns are the results for NIST 2016; 7^{th} and 8^{th} columns are the results for FantasticReality.

MAG[13], LSTM[1], C2RNet[26], and two conventional methods: ADQ[19] and ELA[14]. Moreover, we especially compare our method with MAG on the FantasticReality dataset only, because MAG needs class segmentations which are only provided in the FantasticReality dataset. All the methods we compared are implemented with the code and parameters proposed in the original papers.

We evaluate the performance of RTAG and comparative methods at pixel-level. The evaluation results are present in Table 2. The conventional methods always detect the whole image as a tampered region, so these methods contain very high *Recall* but low *Precision*. The training set of NIST 2016 is very small, and the tampered images are proper post-processed to hide tampering artifacts, so many methods fail on this dataset. But our model learns to detect tampered regions by fewer tampering artifacts and outperforms other methods on NIST 2016. The results shown in Fig. 6 and Fig. 7 indicate that the performance of our method is better than the state-of-the-art methods.

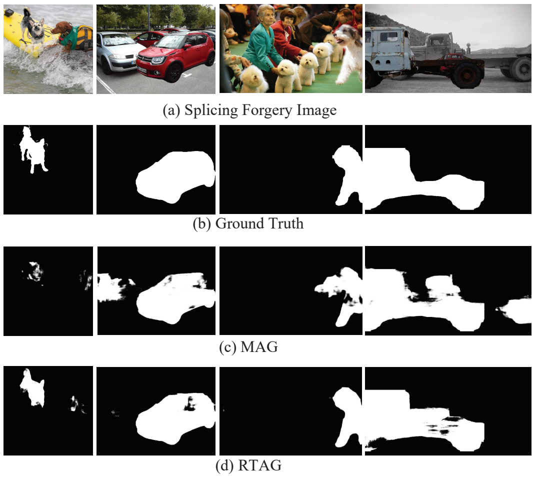


Figure 7. Qualitative results of MAG and RTAG on FantasticReality.

4.5. Ablation Study

To evaluate the necessity of each component of RTAG, we compare the splicing forgery detection performance of several ablated versions of RTAG on CASIA v2.0. The detection results are presented in Table 3 and Fig. 8. We first evaluate the performance of G_M that is only trained by splicing forgery images without G_T. The result demonstrates that the adversarial training between G_M and G_T is critical for RTAG. Then we evaluate G_M in SDMT strategy, which means all the outputs of G_M are generated by a single decoder. The result proves that the MDST strategy significantly improves the performance of the model.

Method	Components			Metrics	
	G_T	G_M Edge	G_M Recons	mAP	F
Without G_T		✓	✓	0.606	0.704
SDMT	✓			0.622	0.731
Without Recons	✓	✓		0.684	0.772
Without Edge	✓		✓	0.689	0.775
RTAG	✓	✓	✓	0.707	0.818

Table 3. Evaluation results for ablated versions of RTAG.

5. Conclusion

In this paper, we present a novel generative adversarial network framework ((RTAG))for splicing detection and localization. RTAG adversarial trains a fake-to-realistic translation generator G_T and a localization generator G_M to simulate the image fakers and the image authenticators. A novel α-learnable whitening and coloring transform block is proposed in G_T to automatically and progressively suppress the tampering artifacts of the forgery images. Meanwhile, the multi-decoder-single-task strategy of G_M will push G_M to improve its detection and localization abilities by learning the retouched images with less tampering artifacts multi-decoder-single-task strategy, and G_M can learn to detect tampered regions from fewer tampering artifacts by adversarial training against G_T. Experimental results demonstrate that the proposed method outperforms the state-of-the-art methods on image splicing forgery detection and localization.

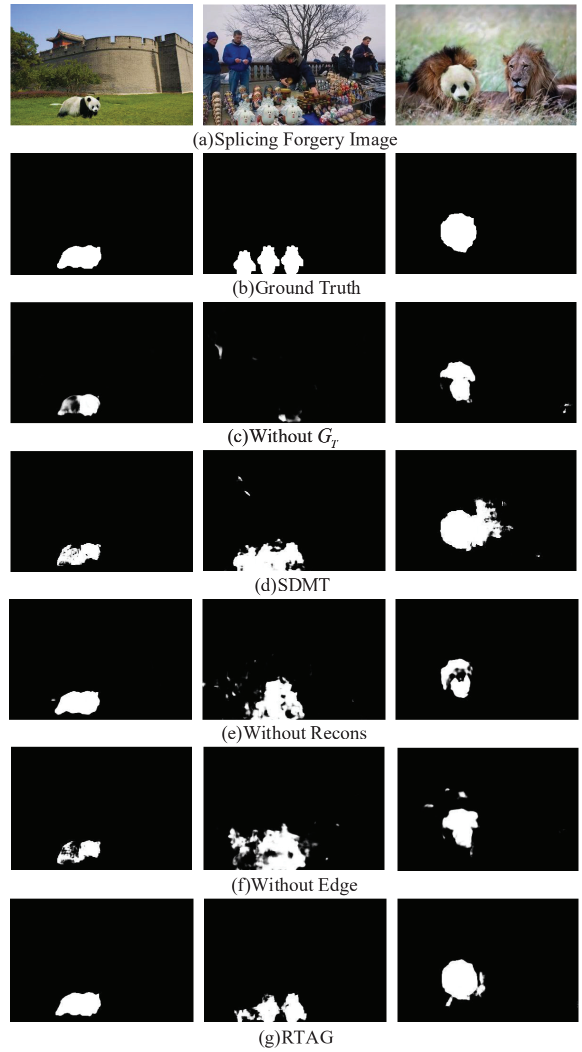


Figure 8. Qualitative results of ablated versions of RTAG on CASIA v2.0.

Acknowledgements

This work was supported in part by the National Key Research and Development Project under Grant 2019YFE0110800 and Grant 2016YFC1000307-3, in part by the National Natural Science Foundation of China under Grant 61806032 and Grant 61976031, in part by the National Major Scientific Research Instrument Development Project of China under Grant 62027827, in part by the Scientific and Technological Key Research Program of Chongqing Municipal Education Commission under Grant KJZD-K201800601. Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.

References

[1] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019. 1, 2, 7

[2] Xiuli Bi, Yang Wei, Bin Xiao, and Weisheng Li. Rru-net: The ringed residual u-net for image splicing forgery detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2

[3] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6306–6314, 2018. 4

[4] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2015. 2

[5] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426. IEEE, 2013. 6

[6] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012. 1

[7] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE, 2019. 6

[8] Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552. IEEE, 2006. 6

[9] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 1, 3

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. In *Advances in Neural Information Processing Systems*, pages 215–226, 2019. 1, 2, 3, 4, 5, 6, 7

[14] Neal Krawetz and Hacker Factor Solutions. A picture's worth. *Hacker Factor Solutions*, 6(2):2, 2007. 1, 2, 7

[15] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018. 3

[16] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, pages 386–396, 2017. 2, 4

[17] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 4

[18] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 4

[19] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009. 1, 2, 7

[20] Yu Liu, Lei Wang, Juan Cheng, Chang Li, and Xun Chen. Multi-focus image fusion: A survey of the state of the art. *Information Fusion*, 64:71–91, 2020. 2

[21] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009. 1

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 3

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4, 5

[24] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018. 1

[25] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019. 1, 2, 7

[26] Bin Xiao, Yang Wei, Xiuli Bi, Weisheng Li, and Jianfeng Ma. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. *Information Sciences*, 511:172–191, 2020. 1, 2, 7

[27] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9036–9045, 2019. 4

[28] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018. 1

[29] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3