Yu *et al. EURASIP Journal on Image and Video Processing* (2021) 2021:10
https://doi.org/10.1186/s13640-021-00550-w

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Stacked generative adversarial networks for image compositing

Bing Yu[1,2*], Youdong Ding[1,2], Zhifeng Xie[1,2] and Dongjin Huang[1,2]

*Correspondence:
yubing1989@shu.edu.cn
[1]Shanghai Film Academy, Shanghai University, Yanchang Road, 200072 Shanghai, China
[2]Shanghai Engineering Research Center of Motion Picture Special Effects, Shanghai University, Yanchang Road, 200072 Shanghai, China

Abstract

Perfect image compositing can harmonize the appearance between the foreground and background effectively so that the composite result looks seamless and natural. However, the traditional convolutional neural network (CNN)-based methods often fail to yield highly realistic composite results due to overdependence on scene parsing while ignoring the coherence of semantic and structural between foreground and background. In this paper, we propose a framework to solve this problem by training a stacked generative adversarial network with attention guidance, which can efficiently create a high-resolution, realistic-looking composite. To this end, we develop a diverse adversarial loss in addition to perceptual and guidance loss to train the proposed generative network. Moreover, we construct a multi-scenario dataset for high-resolution image compositing, which contains high-quality images with different styles and object masks. Experiments on the synthesized and real images demonstrate the efficiency and effectiveness of our network in producing seamless, natural, and realistic results. Ablation studies show that our proposed network can improve the visual performance of composite results compared with the application of existing methods.

Keywords: Image compositing, Generative adversarial networks, Deep neural network

1 Introduction

Image compositing is a fundamental technique in image editing that focuses on seamlessly integrating the foreground region of the source image into another target background. Ideally, a seamless composite result can trick humans into believing that it is not a fake image. However, as shown in Fig. 1a, some differences in appearance between the foreground and background, including illumination, lighting, white balance, and shading, severely reduce the fidelity of image composition. Therefore, to achieve highly realistic compositing, it is necessary to eliminate differences in appearance between the original foreground region and the target background as much as possible.

Early techniques performed gradient-domain blending [1, 5] or alpha matting [6] operations to refine the foreground region for seamless compositing. However, as shown in Fig. 1b, they ignored some essential consistency constraints; thus, their composite results often appear unrealistic. Subsequently, some harmonization methods [7, 8] attempted to yield seamless and realistic results by transferring the visual appearance, texture, and even

Fig. 1 Comparison of compositing methods on real cut-and-paste image. **a** Cut-and-paste. **b** MVCC [1]. **c** RC [2]. **d** DIH [3]. **e** DPH [4]. **f** Our proposed network

noise patterns between images before gradient-domain compositing [5]. Unfortunately, they did not take into account global semantic and structure information and produced unrealistic composite results when the foreground region and target background were very different.

As a powerful learning method, the deep neural network has been successfully applied to various fields of image processing, including image compositing. However, traditional convolutional neural network (CNN)-based methods [2–4, 9] are still tentative and imperfect for high-fidelity compositing. As shown in Fig. 1c, the realism CNN method [2] generates image composite results with unsatisfactory appearance through simple color parameter optimization. Deep image harmonization [3] was subsequently able to capture both the context and semantic information from images through a joint training scheme in which the scene parsing decoders can control semantic organization and generate sharp content efficiently in the process of compositing. However, if scene understanding fails, this method cannot produce a realistic composite result. As shown in Fig. 1d, due to some semantic errors, the composite effects of the deep image harmonization method are not sufficiently harmonized between the foreground and background. In addition, as shown in Fig. 1e, the recent deep painterly harmonization method [4] does not seem to work well for the adjustment of the appearance of nature images.

Recently, several generative adversarial networks (GANs) [10, 11] have been introduced to achieve image compositing. Although these GAN models have the ability to harmonize composite images, they cannot solve all compositing issues, including appearance artifacts and high resolution. Especially for high-resolution compositing, the GAN models that take context encoders as the generative network can only output composite results with a low resolution of 64×64. Thus, they cannot directly generate high-resolution composites, and gradient-based optimization is needed as a post-processing step to create high-resolution images.

In this paper, we propose a stacked generative adversarial network that can create realistic-looking image composites through an end-to-end network. As shown in Fig. 2, our model includes two generators, three discriminators, and multiple loss terms. The inputs to this network are a cut-and-paste composite image and its corresponding mask, and the output is a harmonized high-resolution composite result. Our new model can construct stacked generators and discriminators to harmonize the composite image and determine whether a composite image looks realistic and natural. The generators are essential components for training and testing, while the discriminators are auxiliary components used only for training. Furthermore, after building a multi-scenario high-resolution dataset, our new network can achieve stable training and faster convergence

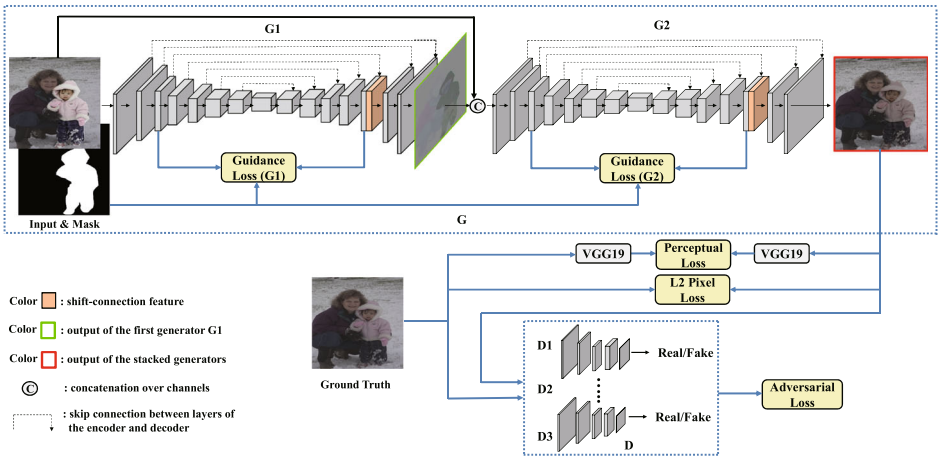

Fig. 2 The overview of the proposed stacked generative adversarial network. It consists of two generators and three discriminators: the output feature map of G_1 is concatenated with the input image and serves as the input of G_2. The discriminators D_1 and D_2 have an identical network architecture but operate at different image scales. D_2 and D_3 operate at the same image scales but have distinct architecture

solving in three steps: (1) train generator G_1 and all discriminators; (2) fix the parameters of generator G_1, and then train generator G_2; and (3) jointly fine-tune the whole network.

Briefly, to reduce appearance differences between the foreground and background, our end-to-end network can fully consider the texture and structure information of the background while effectively preserving the semantic consistency of the composite result. As shown in Fig. 1f, some appearance artifacts (e.g., illumination, contrast, noise, and texture) can be effectively eliminated by our new model; thus, the composite result is seamless and realistic. This paper makes four main contributions, summarized as follows:

- We propose a novel stacked generative adversarial network for high-resolution image compositing. It explores the cascade attention guidance generation strategy and aims to achieve a realistic-looking composite result. Unlike the state-of-the-art GAN-based methods, our network generates a harmonized image in an end-to-end manner.
- We introduce the shift-connection layer [12] to the image compositing task. The layer can utilize long-range and multilevel dependencies across different features to guide generation, improving the structure and texture consistency of the composite image. By doing so, we can take into account the advantages of learning-based and exemplar-based methods and obtain a more realistic composite result compared with the state-of-the-art methods.
- We propose a specialized discriminator for high-resolution image compositing that can employ diverse adversarial strategies at different scales to strengthen the ability of detail discrimination.
- We build a multi-scenario dataset for high-resolution image compositing that mainly contains indoor and outdoor scenes with different styles. To our knowledge, this is the first high-resolution publicly available dataset for image compositing.

The organization of this paper is as follows. Section 2 briefly reviews the existing relevant works. Section 3 describes the proposed network and implementation details. Section 4 verifies the proposed method through a number of comparisons and describes

ablation studies through experiments. Section 5 briefly summarizes this work and discusses possible future work.

2 Related works

In this section, we briefly introduce three subdomains, namely, image compositing, learning-based image editing, and image synthesis using GANs, with particular attention to related works.

2.1 Image compositing

Gradient-domain compositing [1, 5] can adjust the foreground region and the background region to be consistent in terms of illumination by blending the transition region between them. To make the composite image look more realistic, Sunkavalli et al. [7] proposed transferring the appearance of the target image to the source image before blending them. Darabi et al. [8] proposed combining Poisson blending [5] with patch-based synthesis in a unified framework (image melding) to produce a realistic composite. To avoid inconsistent colors and sacrificing texture sharpness, Darabi et al.'s work introduced an extra gradual transition operation between the foreground and background. Xue et al. [13] proposed using statistics and machine learning to study the realism of composites. Recently, deep neural networks have further improved image realism by learning context and semantic information. Zhu et al. [2] proposed a CNN-based model to distinguish composite images from realistic photographs. Tsai et al. [9] proposed using a scene parsing deep network to replace the sky background in a given photograph. These authors further proposed an end-to-end CNN method [3] for image appearance harmonization that could automatically learn both the context and semantic information of the input image and could be trained for both compositing and scene parsing tasks. Wu et al. [11] proposed a two-step method for high-resolution image compositing by combining Wasserstein GAN with multiscale gradient-based methods. Tan et al. [14] proposed a model that learns to predict foreground objects from source images before dealing with appearance compatibility. In contrast to the abovementioned methods, our GAN-based model can take into account the advantages of both exemplar-based and learning-based methods for high-fidelity image compositing.

2.2 Learning-based image editing

Many researchers have leveraged deep learning for image editing with the goal of modifying an image using given image pairs as training data. Zhang et al. [15] proposed a CNN-based image colorization method in which a color recommender system was used to help users interactively use the trained model to translate a gray image to a image. Wang et al. [16] proposed a learning-based image super-resolution method that uses an improved deep CNN to reconstruct a high-resolution image from a given low-resolution image. A deep reinforcement learning-based image enhancement method was proposed by Park et al. [17] that used the MIT-Adobe FiveK dataset [18] to model the stepwise nature of the human retouching process. Yan et al. [12] introduced a novel image inpainting model that uses attention-guided U-net [19] as the generator that fills in marked missing regions with suitable structure and texture. Our method shares a similar concept with learning-based methods and incorporates the advantages of multiple editing models to propose a novel trainable GAN architecture for image compositing.

2.3 Image synthesis using GANs

While GANs [20] can generate photorealistic images from random noise, the generated results might not be in accordance with the user's requirements. It is worth emphasizing some recent works on deep image synthesis using GANs. Conditional GANs [21, 22] are new models that generate images based on particular inputs other than simple noise, thus providing user-controllable results. Isola et al. [23] proposed a pix2pix method that explores conditional GANs to translate semantic label maps into photorealistic images. To solve the pix2pix model's unstable performance during adversarial training for high-resolution synthesis tasks, Wang et al. [24] synthesized 2048 × 1024 resolution realistic-looking photos through a robust training objective together with coarse-to-fine generators and multiscale discriminators. Recently, Xian et al. [25] introduced local texture loss to train a generative adversarial network that can take the texture patches and sketches as inputs and output a shoe or bag. Our method is inspired by the above successful work and is within the framework of image-to-image translation GANs. With our adversarial training objective as well as stacked generators and diverse discriminators, we can not only realize automatic image compositing but also achieve better results compared to existing methods.

3 Proposed method

In this section, we first introduce the attention-guided cascaded generative network and multiple losses. We then describe the training scheme that jointly fine-tunes all the networks together after two separate training processes. Finally, we introduce the multi-scenario synthesized dataset collection method.

3.1 Stacked generators

Given a source image y_{src} and a target image y_{trg}, the cut-and-paste composite image y can be given as follows:

$$y = y_{src} \odot M + y_{trg} \odot (1 - M) \tag{1}$$

where $\odot$ is element-wise multiplication. M is a binary mask corresponding to the foreground region with a value of 1 and 0 for the background region. Our goal is to generate a natural-looking composite result $\hat{y}$ in which the contents are the same as the cut-and-paste input but the appearance is more natural and realistic.

Similar to the pix2pix network [23], our generator is based on the U-net architecture and leverages the property of skip connections between each layer of the encoder and those of the corresponding layer of the decoder. This architecture maintains the texture and details of the image that are lost during the compression process in the encoder, which is important for image compositing [3] and other image editing tasks [26, 27]. Given a U-net of n layers, we denote $\Phi_l(y)$ as the encoder feature of the lth layer and $\Phi_{n-l}(y)$ as the decoder feature of the $(n-l)$th layer. In addition, we denote $\Psi_l(M)$ as a binary mask corresponding to the foreground region in both the encoder feature $\Phi_l(y)$ and the decoder feature $\Phi_{n-l}(y)$. $\Psi_l(M)$ is computed by an extra network that has the same architecture as the U-net encoder but with a network width of 1.

The pix2pix framework is designed to generate low-resolution images if applied directly to 512 × 512 resolution image synthesis. We find that the training is unstable and the

generated results are unsatisfactory. Since stacked networks can be competent for high-resolution image synthesis because of their progressive refinement capability [26, 28, 29], we introduce this concept to our compositing task. Our network consists of two generators in which the second one is stacked upon the first. We call the first generator G_1 and the second generator G_2. Given a cut-and-paste image y, generator G_1 is trained to produce a first feature map $G_1(y)$. Then, $G_1(y)$ is concatenated with the original image y and serves as the input for the second generator G_2. The generator is given by the tuple $G = \{G_1, G_2\}$, as showed in Fig. 2. The detailed architecture of the stacked generators is listed in Table 1.

3.2 Attention guidance compositing

As a state-of-art appearance compositing method, the deep image harmonization method [3] adjusts the masked parts conditioned on their surroundings. However, we have found that this method can produce a distorted appearance or structural inconsistency between the foreground and background when the appearance of particular scenes is improperly remembered due to the limitation of training samples. In contrast, as a traditional compositing method, the image melding method [8] uses exemplar-based synthesis to smoothly transform from the source region to the target region, which avoids obviously inconsistent appearance. This suggests that matching by patches might lead to a more harmonious result. Motivated by these observations, our network takes into account the advantages of learning-based and exemplar-based methods for image compositing. We introduce the shift-connection attention layer [12] in our generators, which can guide the generator to obtain global semantic and structural information, improving the structure and texture consistency of the result.

Table 1 The architecture of G_1/G_2 network. "IN" represents InstanceNorm, "LReLU" represents Leaky ReLU activation, "Conv."/"DeConv." denotes convolutional/transposed convolutional layer with kernel size of 4, "st" means stride, "Concat" explains the skip connections, "Guidance" means guidance loss operation, and "Shift" means shift-connection operation. The different layers of G_1 and G_2 are listed separately

The generative model G_1/G_2
Input G_1: Image (512 × 512 × 3)/ G_2: Feature (512 × 512 × 6)
Layer 1 G_1: Conv. (3, 64), st=2;/ G_2: Conv. (6, 64), st=2;
Layer 2 LReLU; Conv.(64, 128), st=2; IN;
Layer 3 LReLU; Conv.(128, 256), st=2; IN;
Layer 4 LReLU; Conv.(256, 512), st=2;IN;
Layer 5 LReLU; Conv.(512, 512), st=2; IN;
Layer 6 LReLU; Conv.(512, 512), st=2; IN;
Layer 7 LReLU; Conv.(512, 512), st=2; IN;
Layer 8 LReLU; Conv.(512, 512), st=2;
Layer 9 ReLU; DeConv.(512, 512), st=2; IN; Concat.(9, 7);
Layer 10 ReLU; DeConv.(1024, 512), st=2; IN; Concat.(10, 6);
Layer 11 ReLU; DeConv.(1024, 512), st=2; IN; Concat.(11, 5);
Layer 12 ReLU; DeConv.(1024, 512), st=2; IN; Concat.(12, 4);
Layer 13 ReLU; DeConv.(1024, 256), st=2; IN; Concat.(13, 3);
Layer 14 ReLU; Guidance; Shift; DeConv.(768, 128), st=2; IN; Concat.(14, 2);
Layer 15 ReLU; DeConv.(256, 64), st=2; IN; Concat.(15, 1);
Layer 16 ReLU; DeConv.(128, 3), st=2; Tanh;
Output G_1: Feature (512 × 512 × 3)/ G_2: Image (512 × 512 × 3)

Formally, let Ω be the foreground region and $\overline{\Omega}$ be the background region. For each $(\Phi_{n-l}(y))_p$ with location $p \in \Omega$, its nearest neighbor searching in $(\Phi_l(y))_q$ (location $q \in \overline{\Omega}$) can be independently defined as [12]:

$$q^*(p) = \arg\max_{q \in \overline{\Omega}} \frac{\left\langle (\Phi_{n-l}(y))_p, (\Phi_l(y))_q \right\rangle}{\left\| (\Phi_{n-l}(y))_p \right\|_2 \left\| (\Phi_l(y))_q \right\|_2} \tag{2}$$

and the shift vector is obtained by [12]:

$$u_p = q^*(p) - p \tag{3}$$

Then, we spatially rearrange the encoder feature $(\Phi_l(y))_q$ according to the shift vector to obtain a new estimate [12]:

$$(\Phi_{n-l}^{\text{shift}}(y))_p = (\Phi_l(y))_{p+u_p} \tag{4}$$

The shift-connection layer takes $\Phi_l(y)$, $\Phi_{n-l}(y)$, and $\Psi_l(M)$ as inputs and outputs a new shift-connection feature $\Phi_{n-l}^{\text{shift}}(y)$. The layer is embedded in the decoders of both G_1 and G_2 to guide generation. On the one hand, the layer can thus use the information from the background region of the feature to generate new appearances in the foreground region. On the other hand, the layer also helps to model global dependencies across generated regions, ensuring that the details at each location are carefully coordinated with the details at a distance.

3.3 Training losses

The choice of GAN discriminator is especially important for learning-based high-resolution image editing tasks. To obtain realistic-looking generated results, multiple discriminators at different image scales [24] or different image patches [30] have been proposed. Considering that the shape and size of the foreground region in the cut-and-paste image are arbitrary and the resolution of the generating task is high, our compositing network constructs three diverse PatchGAN discriminators [22, 23]. The discriminators receive the generated composite or the ground truth at different scales and attempt to classify the content as either "real" or "fake." We denote the discriminators as D_1, D_2, and D_3. The discriminator is given by the tuple $D = \{D1, D2, D3\}$, as shown in Fig. 2. Specifically, the generated and real high-resolution images are downsampled by a factor of 2 to obtain image pyramids of 2 scales. Then, D_1 is trained to differentiate real and generated images at the finest scale, and D_2 and D_3 are both trained to differentiate images at the coarsest scale. The detailed architecture of the discriminators is presented in Table 2. The discriminators D_1 and D_2 have identical network architectures, while D_3 differs from them. With the discriminators, our adversarial loss is defined as:

$$\mathcal{L}_{\text{adv}} = \min_G \max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) \tag{5}$$

where k is the number of discriminators. The objective function $\mathcal{L}_{GAN}(G, D_k)$ is given by:

$$\mathcal{L}_{GAN}(G, D_k) = E_{x_k \sim p_{\text{data}}(x_k)}[\, log D_k(x_k)] + E_{y_k \sim p_{\text{data}}(y_k)}[\, log(1 - D_k(G(y_k)))] \tag{6}$$

where y_k is a cut-and-paste image and x_k is the corresponding ground truth image. Specifically, y_1 and x_1 correspond to the finest scale, and y_2, y_3 and x_2, x_3 correspond to the coarsest scale. $E_{x_k \sim p_{\text{data}}(x_k)}$ represents the mathematical expectation of $log D_k(x_k)$, where x_k follows the probability distribution $p_{\text{data}}(x_k)$. $E_{y_k \sim p_{\text{data}}(y_k)}$ represents the mathematical expectation of $log(1 - D_k(G(y_k)))$, where y_k follows the probability distribution $p_{\text{data}}(y_k)$.

Table 2 The architecture of $D_1/D_2/D_3$ network. Annotations are the same as Table 1. The different layers of D_1, D_2, and D_3 are listed separately

The discriminative model $D_1/D_2/D_3$	
Input	D_1: Image (512 × 512 × 3)/ D_2, D_3: Image (256 × 256 × 3)
Layer 1	D_1, D_2: Conv. (3, 64), st=2; LRelu;/ D_3: Conv. (3, 32), st=2; LRelu;
Layer 2	D_1, D_2: Conv. (64, 128), st=2; IN; LRelu;/ D_3: Conv. (32, 64), st=2; IN; LRelu;
Layer 3	D_1, D_2: Conv. (128, 256), st=2; IN; LRelu;/ D_3: Conv. (64, 128), st=2; IN; LRelu;
Layer 4	D_1, D_2: Conv. (256, 512), st=1; IN; LRelu;/ D_3: Conv. (128, 256), st=1; IN; LRelu;
Layer 5	D_1, D_2: Conv. (512, 1), st=1; Sigmoid;/ D_3: Conv. (256, 1), st=1; Sigmoid;
Output	D_1: Real or Fake (62 × 62 × 1)/ D_2, D_3: Real or Fake (30 × 30 × 1)

Recent GAN methods [25, 27] have found it effective to combine the adversarial loss with other additional multiple loss terms. First, we choose to use the traditional $L2$ pixel loss to stabilize the training. It is defined as the mean squared error (MSE) between a generated image and its reference image:

$$\mathcal{L}_{L2} = \left\| G(y) - x \right\|_2^2 \tag{7}$$

where $G(y)$ is the output of a given cut-and-paste composite using a generator and x is the corresponding ground truth.

Next, we further include the perceptual loss term, which is used in various editing tasks, such as image inpainting [27] and image super-resolution [31]. Given a cut-and-paste input, we would like the composite result to look realistic and the foreground and background regions to be compatible. The features extracted from the middle layers of the pretrained very deep network represent high-level semantic perception. We defined the perceptual loss using the active layer of the pretrained VGG-19 [32] network on the ImageNet dataset [33]. The loss is defined as the MSE between the feature representations of a generated image and its ground truth:

$$\mathcal{L}_{\text{per}} = \left\| \phi(G(y)) - \phi(x) \right\|_2^2 \tag{8}$$

where $\phi(\cdot)$ is the activation map of the selected layer.

Our final loss term is used to encourage the compositing network to focus on the masked foreground region. We use the guidance loss on the decoder feature of U-net proposed by Yan et al. [12]. It is defined as the MSE between the masked feature representations:

$$\mathcal{L}_{\text{gui}} = \sum_{j=1,2} \left\| (\Psi_l(M) \odot \Phi^j_{n-l}(y)) - (\Psi_l(M) \odot \Phi^j_l(x)) \right\|_2^2 \tag{9}$$

where j is the generator number, $\Phi^j_{n-l}(y)$ is the decoder feature of cut-and-paste input on the $(n-l)$th layer for G_1 or G_2, and $\Phi^j_l(x)$ is the encoder feature of ground truth on the lth layer. Note that the guidance loss is only deployed to the decoder feature maps of the $(n-3)$th layer for G_1 and G_2 in our method.

Our combined loss is defined as the sum of all the above loss functions:

$$\mathcal{L}_{\text{total}} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{L2}\mathcal{L}_{L2} + w_{\text{per}}\mathcal{L}_{\text{per}} + w_{\text{gui}}\mathcal{L}_{\text{gui}} \tag{10}$$

where w_{adv}, w_{L2}, w_{per}, and w_{gui} are the weight parameters for the adversarial, $L2$, perceptual, and guidance losses, respectively.

3.4 Training details

During the training, three discriminators are trained to distinguish the generated results from the ground truth, while the stacked compositing networks are trained to fake the discriminators. Since the high-resolution image compositing task itself is very challenging, we need to train the network carefully to make it converge. The training procedure is divided into three phases. First, generator G_1 and discriminator D are trained for T_{G_1} epochs. Then, generator G_1 is fixed, and generator G_2 is trained from scratch jointly with discriminator D for T_{G_2} epochs. Finally, generator G_1, generator G_2, and discriminator D are trained jointly until the end of the training. An overview of the training procedure is shown in Algorithm 1.

In all experiments, we set the weight $w_{\text{adv}} = 0.002$, $w_{L2} = 1$, $w_{\text{per}} = 0.01$, and $w_{\text{gui}} = 1$. Our network is optimized using the Adam algorithm [34] with a learning rate of 0.0002. We train our models at an input resolution of 512×512, and the batch size is 1. Data augmentation, such as cropping, is also adopted during training.

Algorithm 1 Training of our proposed framework

while epochs $t_1 < T_{G_1}$ **do**
  Initialize generator G_1.
  Initialize discriminators $D = \{D_1, D_2, D_3\}$.
  Sample an image set $\{y, x, M\}$ from training data.
  Update D with adversarial loss.
  Update G_1 with combined losses.
end while
while epochs $t_2 < T_G$ **do**
  Sample an image set $\{y, x, M\}$ from training data.
  if $t_2 < T_{G_2}$ **then**
    Load model G_1 and D.
    Initialize generator G_2.
    Append G_2 to the end of G_1.
    Update D with adversarial loss.
    Fix G_1 and update G_2 with combined losses.
  else
    Update D with adversarial loss.
    Update G_1 and G_2 jointly with combined losses.
  end if
end while

3.5 Synthetic datasets

Data acquisition is the foundation of a successful training network. In our experiment, a masked image pair containing the cut-and-paste and composite result is required as the

input and ground truth for the network. However, there are currently no public datasets for our task. To solve this problem, we selected two public datasets (MIT-Adobe FiveK [18] and Archive of Many Outdoor Scenes (AMOS) [35]) to create our multi-scenario training dataset for compositing through appearance editing. Two different processes are described in Fig. 3.

MIT-Adobe FiveK consists of 5000 raw images, each of which is paired with five retouched images using Adobe Lightroom by 5 trained photographers, A/B/C/D/E. The 6 editions of the same image have different styles. We randomly select one of the 6 versions of the image as the target image and then randomly select one of the remaining 5 versions as the source image. Therefore, there are 30 sets of 5000 target-source paired images (i.e., 150,000 paired images). To create more foreground objects or scenes from the source image, we manually annotate multiple object-level masks for each image in the dataset using the LabelMe annotation tool [36]. When generating input data, we first randomly select a mask and manually segment a region from the source image. Then, we crop this segmented region and overlay on the target image (i.e., ground truth image) to generate the cut-and-paste composite. We reserve 109 images (i.e., 3270 masked paired images) for testing, and the model is trained on the remaining 4891 (i.e., 146,730 paired images) images.

To cover richer object categories and scene styles, we use images from outdoor webcams, which contain images that are captured at the same location but change dramatically with lighting, weather, and season. We construct the compositing dataset using sequences from 92 webcams (the webcam numbers are the same as the famous Transient Attributes Database [37]) selected from AMOS by color transfer. First, given a target image from the camera sequence, we pick 20–30 other images of the same camera taking

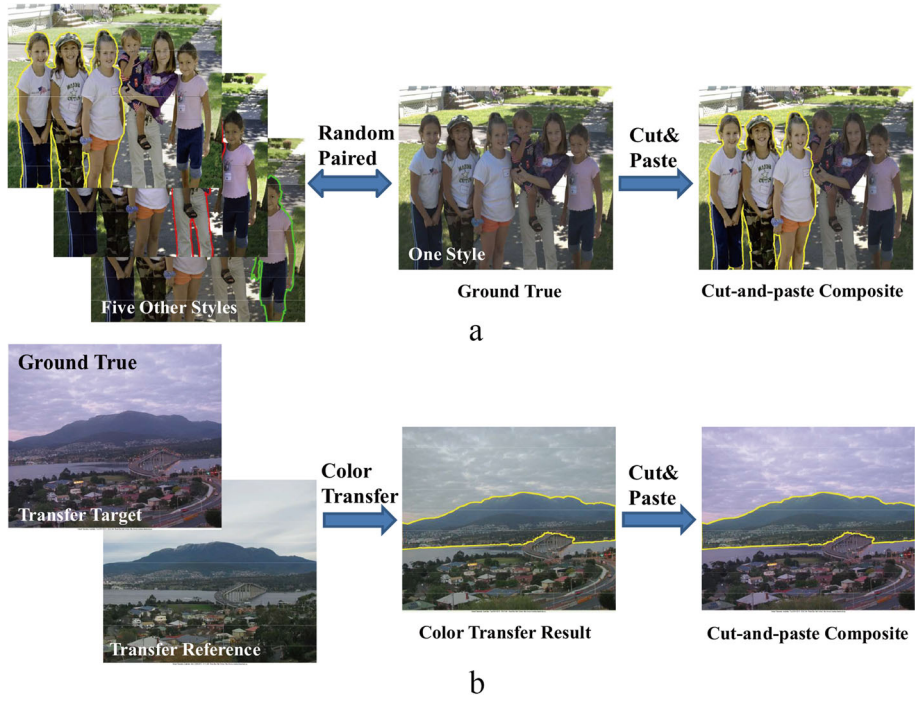

Fig. 3 Data acquisition methods for our multi-scenario dataset. **a** MIT-Adobe FiveK. **b** AMOS

pictures at other times as transfer reference images. Second, instead of using the simple color and illumination histogram statistics method in Tsai et al. [3], we use a patch-based matching method [38] to transfer the appearance between two images with similar content. In this way, we produce 20–30 images of different styles from the given target image while maintaining the same content and scene. Third, for each camera sequence, we repeat the above steps to select 3–10 target images and produce multiple images of different styles. Fourth, all original targets and color transfer results are manually reviewed to ensure that there will be no artifacts or noise. Fifth, we obtain multiple object-level masks for each target image using the LabelMe tool. We use the original target image as the ground truth and crop a segmented foreground from its corresponding produced image in a different style to overlay on the original image. We reserve 1365 masked paired images from 7 webcams for testing and train the model on the remaining 21,658 paired images from another 85 webcams. To distinguish them from the original datasets, we call our compositing datasets FiveK and AMOS in the following experimental discussion.

4 Results and discussion

In this section, we first describe the experimental setup. We then provide comparisons of synthesized images and real images with several metric methods, including user studies. Finally, we conduct five ablation studies on our network design.

4.1 Experimental setup

Our model is implemented on PyTorch v0.3.1, CUDNN v7.0.5, and CUDA v9.0 and run on hardware with an NVIDIA TITAN X GPU (12GB). We separately train and test on our two synthesized datasets. Since the GAN loss curve does not reveal much information in training image-to-image translation GANs [23], we check whether the training has converged by observing $L2$ and perceptual loss curves. On the one hand, the $L2$ term can reflect how close the results are to ground truth images at the pixel level. On the other hand, the perceptual term can reflect the perceptual similarity between generated images and ground truth images. Figures 4 and 5 show the $L2$ and perceptual loss convergence curves of different training phases on the two datasets, respectively. For FiveK, we set $T_{G_1} = 6$ (880,380 iterations), $T_{G_2} = 1$ (146,730 iterations), and $T_G = 3$ (440,190 iterations). For AMOS, we set $T_{G_1} = 16$ (346,528 iterations), $T_{G_2} = 10$ (216,580 iterations), and $T_G = 30$ (649,740 iterations). For each dataset, the training takes approximately 3 weeks. Compositing a single cut-and-paste image of 512×512 takes less than 0.7 s.

4.2 Comparison with existing methods

For synthesized images, we compare our results with MVCC [1], IM [8], DIH [3], and GP [11] at 512×512 resolution. For DIH [3] and GP [11], we use the pretrained models provided by the authors. Note that DIH [3] uses a combination of three public datasets, including MIT-Adobe FiveK, to train the model, and GP [11] uses the transient attributes database as the training dataset.

The images shown in Figs. 6 and 7 are taken from the FiveK and AMOS test datasets. Although the foreground appearances of the MVCC results are well blended using mean-value coordinates, some obvious artifacts can be found, as shown in Figs. 6c and 7c. IM results showed no significant improvement in visual appearance. DIH is effective in semantic compositing, and the visual appearance of the results shows better performance

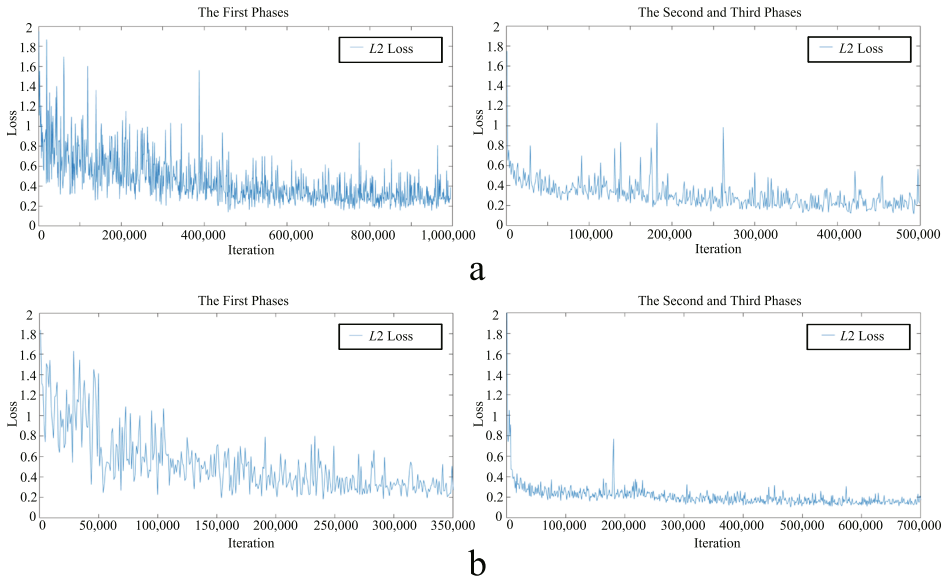

Fig. 4 Training convergence curves of *L*2 loss. **a** FiveK. **b** AMOS

than MVCC and IM. However, the boundary between the foreground and background of the DIH results is not seamless enough, and there are obvious jagged edges. In addition, DIH models the dependencies between the scene semantics and its surface appearance, but these kinds of semantic priors do not always work well; for example, the yellow flower foreground in the composite of the three rows of Fig. 6e is adjusted to green, and the result is far from the ground truth (GT). GP adopts a multistage scheme to combine a deep network and Poisson blending, while its GAN model generates poor results at low resolution and leads to incorrect enlargement in the subsequent high-resolution optimization step,

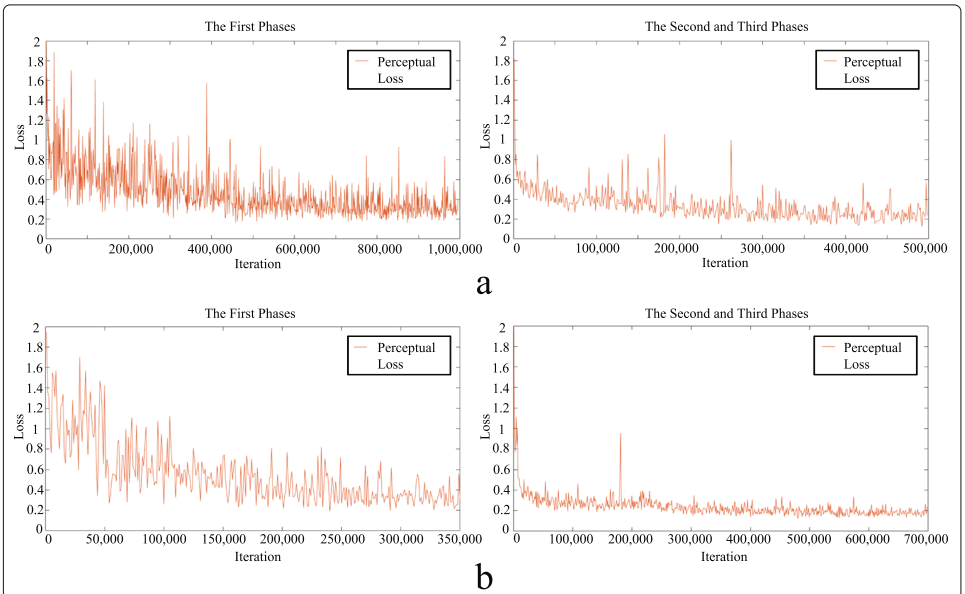

Fig. 5 Training convergence curves of perceptual loss. **a** FiveK. **b** AMOS

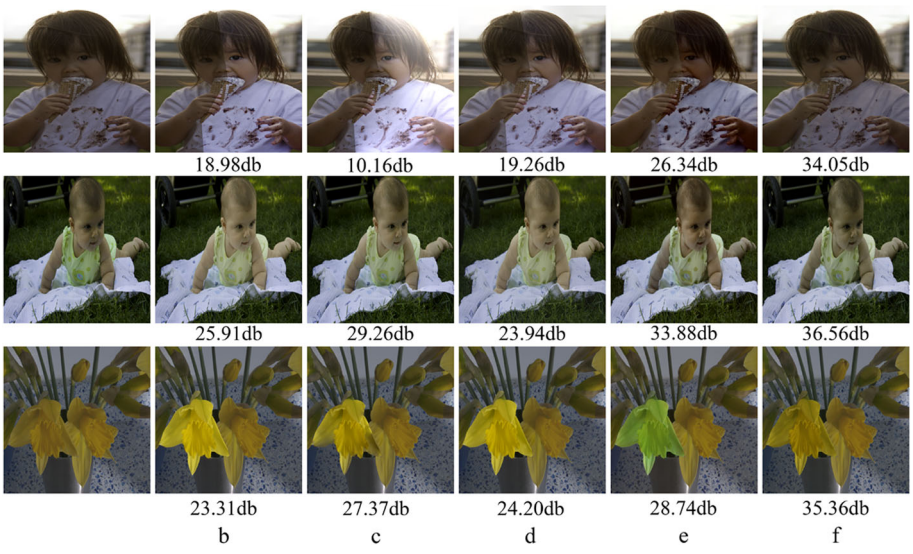

Fig. 6 Example results on synthesized FiveK dataset. **a** GT. **b** Cut-and-paste. **c** MVCC [1]. **d** IM [8]. **e** DIH [3]. **f** Our proposed network. Our composite results obtained the highest PSNR value scores

resulting in unrealistic images, as shown in Fig. 7e. Overall, the proposed method performs favorably in generating realistic, seamless, and harmonious images. The foreground appearance of our results is most consistent with the corresponding background.

In addition, we use three quantitative criteria to evaluate the proposed and other methods. First, the peak signal-to-noise ratio (PSNR), which is used by Tsai et al. [3], can reflect how close the result is to GT. Second, the structural similarity index (SSIM) attempts to quantify the visibility of structural differences between the result and GT. Third, the learned perceptual image patch similarity (LPIPS) [39], which agrees surprisingly well with human judgment, is used to assess the perceptual similarity between two images. Note that unlike PSNR and SSIM, smaller values mean greater perceptual similarity for LPIPS. Tables 3 and 4 show the quantitative scores between GT and composite results for FiveK and AMOS, respectively. The scores are calculated based on the mean values of a random subset of 300 images selected from each of the two test datasets. Our proposed

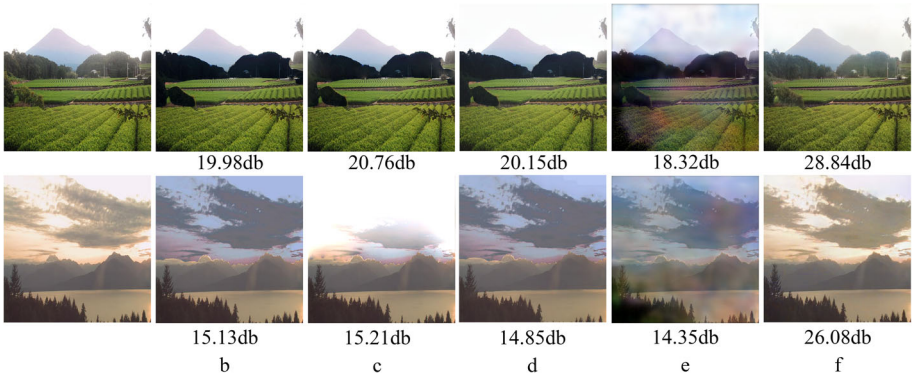

Fig. 7 Example results on synthesized AMOS dataset. **a** GT. **b** Cut-and-paste. **c** MVCC [1]. **d** IM [8]. **e** GP [11]. **f** Our proposed network. Our composite results obtained the highest PSNR value scores

Table 3 Comparisons of methods on the FiveK test dataset

	PSNR	SSIM	LPIPS
Cut-and-paste	24.40db	0.9547	0.0410
MVCC [1]	27.23db	0.9487	0.0363
IM [8]	22.95db	0.8349	0.0945
DIH [3]	29.96db	0.9368	0.0401
Ours (w/o G_2)	33.22db	0.9628	0.0298
Ours (w/ D_1 only)	33.66db	0.9517	0.0248
Ours (w/o Shift)	32.46db	0.9504	0.0252
Ours (w/o $\mathcal{L}_{\text{per}}$)	33.41db	0.9512	0.0422
Ours (w/o $\mathcal{L}_{\text{gui}}$)	34.42db	0.9580	0.0217
Ours	34.74db	0.9692	0.0209

image compositing network performs better than other methods in terms of PSNR, SSIM, and LPIPS metrics.

For real images, we compare our results with MVCC [1], IM [8], RC [2], GP [11], and DIH [3]. To demonstrate that the models trained on our multi-scenario dataset can be generalized to real cut-and-paste composite images, we created a test set of 30 high-resolution real composite images and combined 50 public high-quality images collected by Xue et al. [13] and Tsai et al. [3], resulting in a real cut-and-paste composite set that contains 80 images. Since Xue et al.'s statistical method has no public code, our results are not compared with it.

Figure 8 shows some experimental comparisons selected from the real composite set. The MVCC and IM can solve some of the inconsistencies between two parts of inputs, but the results are not satisfactory. RC's realism prediction model is effective in handling easily distinguishable cut-and-paste composite input; nevertheless, it generates unsatisfactory results, especially the transition region between foreground and background (e.g., there are distinct jagged outlines at the boundary of the foreground in the results of the second, third, and fifth rows). For GP, it generates visually poor results. For DIH, because the model utilizes semantic information to adjust cut-and-paste input, it is limited by the training dataset. If the scene semantics are incorrectly judged, this will lead to unrealistic outputs (e.g., the fourth, sixth, and eighth rows). Compared with others, the proposed model can better predict the global structure and thus maintain the consistency of the context, resulting in realistic-looking composites.

Figure 9 illustrates one example where the same foreground (i.e., the zebra) is copied to different backgrounds (i.e., a street in dim light and a zebra herd in the sun). For RC, the discriminative model cannot correctly predict the degree of perceived visual realism of the given inputs, so the appearance of the foreground is almost never adjusted. For DIH, regardless of the scene, the context-aware encoder-decoder recovers the fur color

Table 4 Comparisons of methods on the AMOS test dataset

	PSNR	SSIM	LPIPS
Cut-and-paste	17.66db	0.7616	0.1423
MVCC [1]	17.64db	0.8382	0.1617
IM [8]	17.49db	0.6885	0.1908
GP [11]	16.90db	0.5635	0.2697
Ours	29.15db	0.9048	0.0878

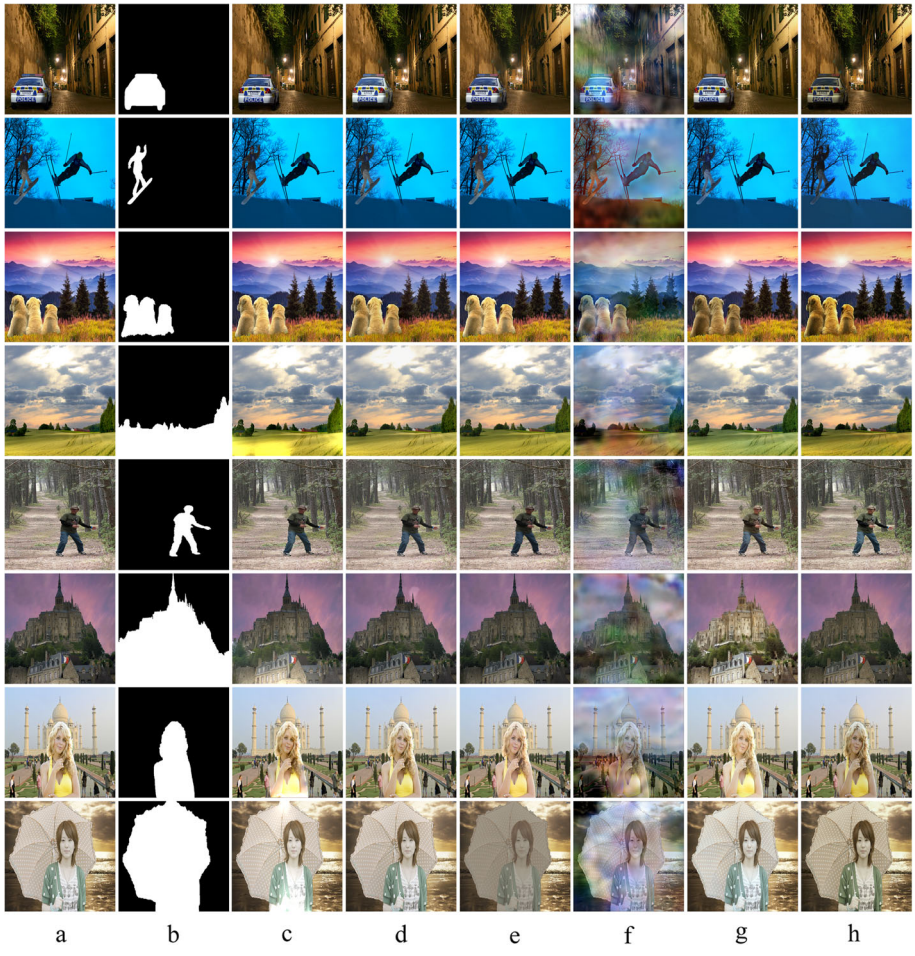

Fig. 8 Example results on real images. **a** Cut-and-paste. **b** Mask. **c** MVCC [1]. **d** IM [8]. **e** RC [2]. **f** GP [11]. **g** DIH [3]. **h** Our proposed network

constrained by the trained prior knowledge to almost invariable results. In contrast to the two methods mentioned above, with the proposed network, the foregrounds can be adjusted according to the surrounding scene and luminance.

To better understand the performance of our methods, we conducted quantitative assessment studies with users, similar to Tsai et al. [9]. Participants were shown an input

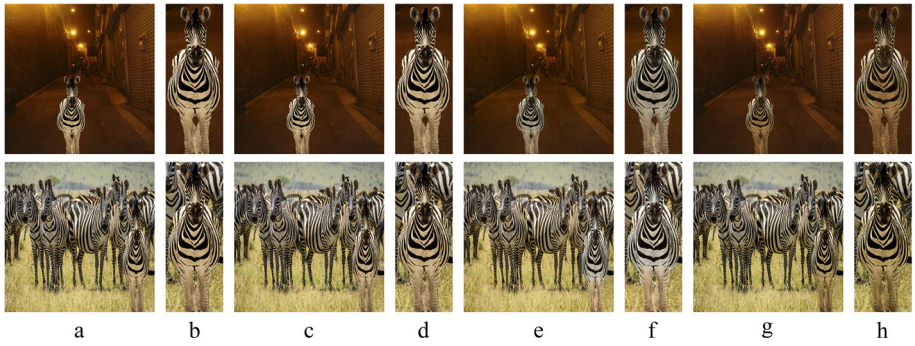

Fig. 9 Real example with same foreground and different backgrounds. **a** Cut-and-paste. **b** Partial enlarged details in **a**. **c** RC [2]. **d** Partial enlarged details in **c**. **e** DIH [3]. **f** Partial enlarged details in **e**. **g** Our proposed network. **h** Partial enlarged details in **g**

cut-and-paste composite and six results from MVCC, IM, RC, GP, DIH, and the proposed method. Each participant was asked to rate each group according to the realistic nature of the images using a 5-point Likert scale (1 for worst, 5 for best). We asked 20 users to provide feedback by giving users 30 tuples of images selected from our real cut-and-paste composite set. The average scores of individual images in the evaluation set are shown in Fig. 10. Most of our scores are above 3.0. Our scores outperform MVCC in 80%, IM in 80%, RC in 80%, GP in 100%, and DIH in 73%.

4.3 Ablation studies

The main differences between our compositing method and other methods are the stacked generative adversarial network architecture and the combined loss function. Thus, five groups of experiments in the FiveK dataset were conducted to analyze the effect of stacked generators, diverse discriminators, shift-connection operations, perceptual loss, and guidance loss on composite results. Table 3 shows that the proposed network achieved better scores in terms of PSNR, SSIM, and LPIPS metrics compared to the other five strategies.

To evaluate the effectiveness of stacked generators for high-resolution compositing, we trained our network without using generator G_2. The number of training epochs was constrained to be the same as the original model. As shown in Fig. 11, the results generated by a single (non-stacked) generator may not be satisfactory and have obvious artifacts. In addition, the consistent improvement in the quantitative assessment scores of our models clearly demonstrates the benefits of the cascaded refinement approach.

To evaluate the effectiveness of our specialized discriminator for high-resolution compositing, we trained our network only with discriminator D_1. Visually, as shown in Fig. 12, we observed that the model using the combination of three diverse PatchGAN discriminators could reduce artifacts and improve appearance in terms of realism.

We trained a model without using the shift-connection layer. As shown in Fig. 13, the operation helps to obtain representation for the foreground (i.e., the man or the flower) from the background region, resulting in composites with consistent regions.

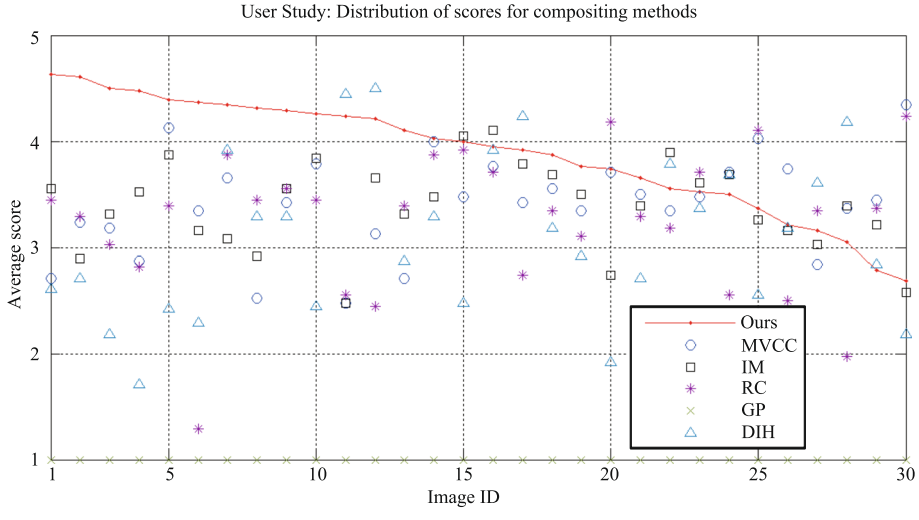

Fig. 10 Average evaluation scores for each image, sorted by our score. The proposed method performs better than others in most cases

Fig. 11 Effect of stacked generators. **a** GT. **b** Cut-and-paste [PSNR = 26.19 db, LPIPS = 0.0750]. **c** Ours (w/o G_2) [PSNR = 30.91 db, LPIPS = 0.0986]. **d** Ours [PSNR = 32.48 db, LPIPS = 0.0387]. **e** GT. **f** Cut-and-paste [PSNR = 19.00 db, LPIPS = 0.06326]. **g** Ours (w/o G_2) [PSNR = 33.30 db, LPIPS = 0.0428]. **h** Ours [PSNR = 35.00 db, LPIPS = 0.0274]

We trained a model without perceptual loss. As shown in Fig. 14, the composites generated by the model without $\mathcal{L}_{\text{per}}$ have ghosting. In addition, the significant advantage in LPIPS scores for the model with $\mathcal{L}_{\text{per}}$ shows that the perceptual loss can greatly improve visual perception.

We trained a model without guidance loss. As shown in Fig. 15, guidance loss is helpful in preserving better visual appearance. We observed that the color and luminance of foregrounds with $\mathcal{L}_{\text{gui}}$ were closer to GT.

4.4 Limitations

Our model trained on the proposed multi-scenario dataset can handle the composition of high-resolution real cut-and-paste images in most cases. However, if the input image is significantly different from the training data, it may still fail. Figure 16 shows two examples of our failure case, where the appearance of foregrounds and backgrounds is not sufficiently natural and harmonious.

5 Conclusion

In this paper, we proposed a stacked GAN method for high-resolution image compositing. Given a cut-and-paste composite, the proposed network can adjust the foreground appearance and output a harmonized image that looks realistic. We have shown that by using stacked generators, diverse discriminators, and multiple loss constraints, it is possible to train a good performance model. In addition, we demonstrated that our network can be implemented in three steps to achieve stable training and faster convergence. Our method utilizes a cascade attention guidance generation strategy and generates more harmonious and consistent results than state-of-the-art methods. Future studies will focus on improving the speed of high-resolution compositing of the proposed network and expanding the training dataset.

Fig. 12 Effect of diverse discriminators. **a** GT. **b** Cut-and-paste [PSNR = 23.06 db, LPIPS = 0.0669]. **c** Ours (w/ D_1 only) [PSNR = 24.75 db, LPIPS = 0.0518]. **d** Ours [PSNR = 28.15 db, LPIPS = 0.0444]. **e** GT. **f** Cut-and-paste [PSNR = 20.27 db, LPIPS = 0.0873]. **g** Ours (w/ D_1 only) [PSNR = 28.66 db, LPIPS = 0.0685]. **h** Ours [PSNR = 32.10 db, LPIPS = 0.0495]

Fig. 13 Effect of shift-connection layer. **a** GT. **b** Cut-and-paste [PSNR = 27.17 db, LPIPS = 0.0592]. **c** Ours (w/o Shift) [PSNR = 29.59 db, LPIPS = 0.0548]. **d** Ours [PSNR = 33.82 db, LPIPS = 0.03167]. **e** GT. **f** Cut-and-paste [PSNR = 28.40 db, LPIPS = 0.0422]. **g** Ours (w/o Shift) [PSNR = 29.90 db, LPIPS = 0.0402]. **h** Ours [PSNR = 32.71 db, LPIPS = 0.0257]

Fig. 14 Effect of perceptual loss. **a** GT. **b** Cut-and-paste [PSNR = 33.82 db, LPIPS = 0.0196]. **c** Ours (w/o $\mathcal{L}_{per}$) [PSNR = 31.64 db, LPIPS = 0.0580]. **d** Ours [PSNR = 35.83 db, LPIPS = 0.0167]. **e** GT. **f** Cut-and-paste [PSNR = 31.66 db, LPIPS = 0.0100]. **g** Ours (w/o $\mathcal{L}_{per}$) [PSNR = 34.33 db, LPIPS = 0.0464]. **h** Ours [PSNR = 36.37db, LPIPS = 0.009]

Fig. 15 Effect of guidance loss. **a** GT. **b** Cut-and-paste [PSNR = 28.65 db, LPIPS = 0.0257]. **c** Ours (w/o $\mathcal{L}_{gui}$) [PSNR = 27.98 db, LPIPS = 0.0281]. **d** Ours [PSNR = 29.86 db, LPIPS = 0.0248]. **e** GT. **f** Cut-and-paste [PSNR = 17.12 db, LPIPS = 0.0921]. **g** Ours (w/o $\mathcal{L}_{gui}$) [PSNR = 21.72 db, LPIPS = 0.0424]. **h** Ours [PSNR = 33.19 db, LPIPS = 0.0247]

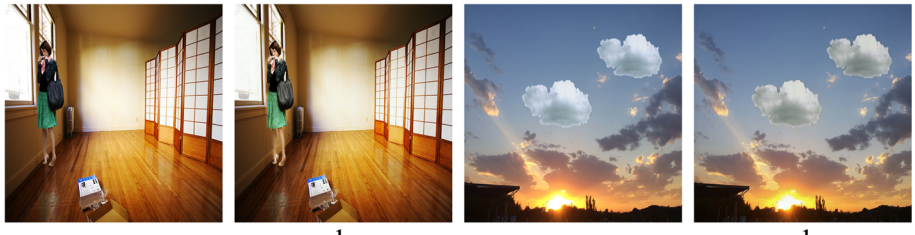

Fig. 16 Failure cases. **a** Cut-and-paste. **b** Ours. **c** Cut-and-paste. **d** Ours

Abbreviations
CNN: Convolutional neural network; GAN: generative adversarial network; AMOS: Archive of many outdoor scenes; PSNR: Peak signal to noise ratio; SSIM: Structural similarity index; LPIPS: Learned perceptual image patch similarity; MVCC: Mean-value coordinates cloning; RC: Realism CNN; DIH: Deep image harmonization; DPH: Deep painterly harmonization; IM: Image melding; GP: Gaussian-Poisson

Acknowledgements
The authors thank the editor and anonymous reviewers.

Authors' contributions
All authors take part in the discussion of the work described in this manuscript. YB wrote the first version of the manuscript. DY, XZ, and HD did part experiments of the paper. All authors read and approved the final manuscript.

Funding
This work was supported by the National Natural Science Foundation of China (61303093, 61402278) and the Shanghai Natural Science Foundation (19ZR1419100).

Authors' information
Bing Yu received his Ph.D. degree in digital media technology from Shanghai University, Shanghai, China, in 2020. He received his BS degrees in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2011, and his MS degree in computer science and technology from North Minzu University, Yinchuan, China, in 2015. He is now a lecturer with Shanghai University, Shanghai, China. His current research interests include deep learning and image processing.
Youdong Ding received his Ph.D. degree in mathematics from University of Science and Technology of China, Hefei, China, in 1997. He was a post-doctor at the Department of Mathematics of Fudan University, Shanghai, China, from 1997 to 1999. He is now a professor with Shanghai University, Shanghai, China. His research interests are computer graphics, image processing, and digital media technology.
Zhifeng Xie received his Ph.D. degree in computer application technology from Shanghai Jiao Tong University, Shanghai, China, in 2013. He was a research assistant at the Department of Computer Science, City University of Hong Kong, Hong Kong, China, in 2011. He is now an associate professor with Shanghai University, Shanghai, China. His research interests include image/video editing, computer graphics, and digital media technology.
Dongjin Huang received his Ph.D. degree in computer application technology from Shanghai University, Shanghai, China, in 2011. He was a post-doctor with University of Bradford, UK, from 2012 to 2013. He is now an assistant professor with Shanghai University, Shanghai, China. His research interests are augmented reality, computer vision, and computer graphics.

Availability of data and materials
The datasets for high-resolution image compositing generated during the current study are available in the Baidu Cloud repository, https://pan.baidu.com/s/1WmJ5P7ToSeA9FS4vgmMfaA (download password: 1111).

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 16 May 2019 Accepted: 1 March 2021
Published online: 29 March 2021

References
1. Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, D. Lischinski, Coordinates for instant image cloning. ACM Trans. Graph. **28**, 67 (2009)
2. J.-Y. Zhu, P. Krahenbuhl, E. Shechtman, A. A. Efros, in *2015 IEEE International Conference on Computer Vision (ICCV)*, Learning a discriminative model for the perception of realism in composite images (IEEE, Piscataway, 2015), pp. 3943–3951
3. Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, M.-H. Yang, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Deep image harmonization (IEEE, Piscataway, 2017), pp. 2799–2807
4. F. Luan, S. Paris, E. Shechtman, K. Bala, Deep painterly harmonization. Comput. Graph. Forum. **37**, 95–106 (2018)
5. P. Perez, M. Gangnet, A. Blake, Poisson image editing. ACM Trans. Graph. **22**, 313–318 (2003)
6. J. Wang, M. Agrawala, M. F. Cohen, Soft scissors: an interactive tool for realtime high quality matting. ACM Trans. Graph. **26**, 9 (2007)
7. K. Sunkavalli, M. K. Johnson, W. Matusik, H. Pfister, Multi-scale image harmonization. ACM Trans. Graph. **29**, 125 (2010)
8. S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, P. Sen, Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graph. **31**, 82 (2012)
9. Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, M.-H. Yang, Sky is not the limit: semantic-aware sky replacement. ACM Trans. Graph. **35**, 149 (2016)
10. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Context encoders: feature learning by inpainting (IEEE, Piscataway, 2016), pp. 2536–2544
11. H. Wu, S. Zheng, J. Zhang, K. Huang, in *2019 ACM International Conference on Multimedia*, Gp-gan: Towards realistic high-resolution image blending (ACM, New York, 2019), pp. 2487–2495

12. Z. Yan, X. Li, M. Li, W. Zuo, S. Shan, in *Computer Vision – ECCV 2018*, Shift-net: image inpainting via deep feature rearrangement (Springer, New York, 2018), pp. 3–19
13. S. Xue, A. Agarwala, J. Dorsey, H. Rushmeier, Understanding and improving the realism of image composites. ACM Trans. Graph. **31**, 84 (2012)
14. F. Tan, C. Bernier, B. Cohen, V. Ordonez, C. Barnes, in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Where and who automatic semantic-aware person composition (IEEE, Piscataway, 2018), pp. 1519–1528
15. R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, A. A. Efros, Real-time user-guided image colorization with learned deep priors. ACM Trans. Graph. **36**, 119 (2017)
16. X. Wang, K. Yu, C. Dong, C. C. Loy, in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Recovering realistic texture in image super-resolution by deep spatial feature transform (IEEE, Piscataway, 2018), pp. 606–615
17. J. Park, J.-Y. Lee, D. Yoo, I. S. Kweon, in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, Distort-and-recover: color enhancement using deep reinforcement learning (IEEE, Piscataway, 2018), pp. 5928–5936
18. V. Bychkovsky, S. Paris, E. Chan, F. Durand, in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Learning photographic global tonal adjustment with a database of input/output image pairs (IEEE, Piscataway, 2011), pp. 97–104
19. O. Ronneberger, P. Fischer, T. Brox, in *Lecture Notes in Computer Science*, U-net: convolutional networks for biomedical image segmentation (Springer, New York, 2015), pp. 234–241
20. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in *Advances in Neural Information Processing Systems*, Generative adversarial nets (MIT Press, Cambridge, 2014), pp. 2672–2680
21. M. Mirza, S. Osindero, Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014). https://arxiv.org/abs/1411.1784
22. J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, E. Shechtman, in *Advances in Neural Information Processing Systems*, Toward multimodal image-to-image translation (MIT Press, Cambridge, 2017), pp. 465–476
23. P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Image-to-image translation with conditional adversarial networks (IEEE, Piscataway, 2017), pp. 5967–5976
24. T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, High-resolution image synthesis and semantic manipulation with conditional gans (IEEE, Piscataway, 2018), pp. 8798–8807
25. W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, J. Hays, in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Texturegan: controlling deep image synthesis with texture patches (IEEE, Piscataway, 2018), pp. 8456–8465
26. T. Dekel, C. Gan, D. Krishnan, C. Liu, W. T. Freeman, in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, Sparse, smart contours to represent and edit images (IEEE, Piscataway, 2018), pp. 8456–8465
27. G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, in *Computer Vision – ECCV 2018*, Image inpainting for irregular holes using partial convolutions (Springer, New York, 2018), pp. 89–105
28. H. Zhang, T. Xu, H. Li, in *2017 IEEE International Conference on Computer Vision (ICCV)*, StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks (IEEE, Piscataway, 2017), pp. 5908–5916
29. Q. Chen, V. Koltun, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Photographic image synthesis with cascaded refinement networks (IEEE, Piscataway, 2017), pp. 1520–1529
30. S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion. ACM Trans. Graph. **36**, 107 (2017)
31. J. Johnson, A. Alahi, L. Fei-Fei, in *Computer Vision – ECCV 2016*, Perceptual losses for real-time style transfer and super-resolution (Springer, New York, 2016), pp. 694–711
32. K. Simonyan, Z. Andrew, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014). http://arxiv.org/abs/1409.1556
33. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Imagenet: a large-scale hierarchical image database (IEEE, Piscataway, 2009), pp. 248–255
34. D. P. Kingma, J. L. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014). https://arxiv.org/abs/1412.6980
35. N. Jacobs, N. Roman, R. Pless, in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Consistent temporal variations in many outdoor scenes (IEEE, Piscataway, 2007), pp. 1–6
36. B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**, 157–173 (2007)
37. P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, J. Hays, Transient attributes for high-level understanding and editing of outdoor scenes. ACM Trans. Graph. **33**, 149 (2014)
38. Y. HaCohen, E. Shechtman, D. B. Goldman, D. Lischinski, Non-rigid dense correspondence with applications for image enhancement. ACM Trans. Graph. **30**, 70 (2011)
39. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, The unreasonable effectiveness of deep features as a perceptual metric (IEEE, Piscataway, 2018), pp. 586–595

Publisher's Note