Painterly Image Harmonization via Adversarial Residual Learning

Xudong Wang, Li Niu*, Junyan Cao, Yan Hong, Liqing Zhang*
Department of Computer Science and Engineering, MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
{wangxudong1998,ustcnewly,Joy_C1,hy2628982280,zhang-lq}@sjtu.edu.cn

Abstract

Image compositing plays a vital role in photo editing. After inserting a foreground object into another background image, the composite image may look unnatural and inharmonious. When the foreground is photorealistic and the background is an artistic painting, painterly image harmonization aims to transfer the style of background painting to the foreground object, which is a challenging task due to the large domain gap between foreground and background. In this work, we employ adversarial learning to bridge the domain gap between foreground feature map and background feature map. Specifically, we design a dual-encoder generator, in which the residual encoder produces the residual features added to the foreground feature map from main encoder. Then, a pixel-wise discriminator plays against the generator, encouraging the refined foreground feature map to be indistinguishable from background feature map. Extensive experiments demonstrate that our method could achieve more harmonious and visually appealing results than previous methods.

1. Introduction

In many photo editing applications, it is often necessary to cut a foreground object from one image and overlay it on another background image, which is referred to as image composition [37]. However, when combining the foreground and background from different image sources to produce a composite image, the styles of foreground and background may be inconsistent, which would severely harm the quality of composite image.

When the foreground and background are both photographic images, the style mainly refers to illumination statistics, *e.g.*, the foreground is captured in the daytime while the background is captured at night. To address the style inconsistency between foreground and background, image harmonization [9,10,32,50] aims to adjust the illumination statistics of foreground to be compatible with background, leading to a harmonious image. When the foreground is from a photographic image and the background is an artistic painting, the background style has the same meaning as in artistic style transfer [13, 19, 39], which includes color, texture, pattern, strokes, and so on. To address the style inconsistency between foreground and background, painterly image harmonization [35] aims to migrate the background style to the foreground, so that the stylized foreground is compatible with the background and naturally blended into the background.

Figure 1. Example of painterly image harmonization. From left to right are foreground object, background image, composite image, and harmonized image.

To the best of our knowledge, there are only few works on painterly image harmonization. To name a few, Luan et al. [35] proposed to update the composite foreground through iterative optimization process that minimizes the designed loss functions. However, the method [35] relies on slow iterative optimization process, which imposes restrictions on real-time application. Inspired by [19], Peng et al. [40] introduced AdaIN [19] to align the styles between foreground and background, which is trained with content loss and style loss. The method [40] runs much faster than [35], but performs poorly when transferring the color and brush texture of artistic paintings. Zhang et al. [56] jointly optimized the proposed Poisson blending loss with the style and content loss, and reconstructed the blending region by iteratively updating the pixels. Analogous to [35], the method [56] is also very time-consuming. In summary, the existing painterly image harmonization methods are either time-consuming or weak in style transfer. Additionally,

*Corresponding author.

the image harmonization methods [9, 10, 32, 50] for photographic images are not suitable for our task (see Section 2.1) and the artistic style transfer methods [13, 19, 39] have several limitations when applied to our task (see Section 2.3).

One critical issue that hinders the performance of painterly image harmonization is the large domain gap between photographic foreground and painterly background. Considering that adversarial learning has been widely used to close the gap between different domains [2, 51], we attempt to employ adversarial learning in the painterly image harmonization task. Actually, pixel-wise adversarial learning has been used in previous works [18, 23] from related fields (*e.g.*, video harmonization, photo retouching). They use a discriminator to distinguish foreground pixels from background pixels in the output image, which can help strengthen the generator in an adversarial manner.

In this work, we apply similar idea to the feature maps in the generator, that is, employing adversarial learning to bridge the gap between foreground feature map and background feature map. Specifically, we propose a novel painterly image harmonization network that contains a dual-encoder generator (main encoder and residual encoder) and pixel-wise feature discriminators. In the main encoder, we use pretrained VGG [43] encoder to extract multiple layers of feature maps from composite image and background image. Then, we apply AdaIN [19] to align the statistics between the foreground region in composite feature maps and the whole background feature maps, leading to stylized composite feature maps. To further reduce the domain gap between foreground and background, we also propose an extra residual encoder to learn residual features for each encoder layer. The learnt residual features are added to the foreground regions of stylized composite feature maps, leading to refined composite feature maps. Afterwards, for each encoder layer, our pixel-wise feature discriminator takes in the refined composite feature map and plays against our dual-encoder generator by telling disharmonious pixels from harmonious ones, which encourages the refined composite feature maps to be harmonious. Finally, the refined composite feature maps are delivered to the decoder to produce the harmonized image. We name our method as **P**ainterly **H**armonization via **A**dversarial **R**esidual **Net**work (PHARNet).

Following previous works [35, 40], we conduct experiments on COCO [31] and WikiArt [36], comparing with painterly image harmonization methods and artistic style transfer methods. Our major contributions can be summarized as follows. 1) We are the first to introduce pixel-wise adversarial learning to harmonize feature maps. 2) We propose PHARNet equipped with novel dual-encoder generator and pixel-wise feature discriminator. 3) Extensive experiments on benchmark datasets prove the effectiveness of our network design.

2. Related Work

2.1. Image Harmonization

The goal of image harmonization is to harmonize a composite image by adjusting the illumination information of foreground to match that of background. Early traditional image harmonization methods [26, 45, 46, 55] tended to match low-level color or brightness information between foreground and background. After that, unsupervised deep learning methods [58] were proposed to enhance the realism of harmonized image using adversarial learning. With the constructed large-scale dataset [9] containing paired training data, myriads of supervised deep learning approaches [8,38,44,48,50] have been developed to advance the harmonization performance. To name a few, [10, 16] designed various attention modules which are embedded in the network. [7, 9] treated foreground and background as different domains, thus converting image harmonization task to domain translation task. [14, 15] introduced intrinsic decomposition to image harmonization task. More recently, [8, 22, 29, 54] integrated color transformation with deep learning network to achieve better performance. However, the well-behaved supervised image harmonization methods require pairs of training data, which are almost impossible to acquire in painterly image harmonization task.

2.2. Painterly Image Harmonization

When overlaying a photographic foreground onto a painterly background, the task is called painterly image harmonization. This task targeted at migrating the background style to the foreground and preserving the foreground content. As far as we are concerned, there only exist few works concentrating on painterly image harmonization task. The existing approaches [35, 40, 56] can be divided into optimization-based approaches [35, 56] and feed-forward approaches [40]. The optimization-based approaches [35, 56] iteratively optimize over the foreground region of input composite image to minimize the designed loss functions (*e.g.*, content loss, style loss, Poisson loss), which is very inefficient. The feed-forward approaches [40] pass the composite image through the network once and output the harmonized image, which is much more efficient than optimization-based methods. PHDNet [4] performed image harmonization in both frequency domain and spatial domain. PHDiffusion [34] introduced diffusion model to painterly image harmonization.

Our proposed method belongs to feed-forward approaches. Although adversarial learning has been used in [40], they perform image-level and region-level adversarial learning, which is quite different from our pixel-wise adversarial learning. Moreover, [40] tends to make the output images indistinguishable from artistic ones, but lacks the ability to match foreground style with background style.

2.3. Artistic Style Transfer

Artistic style transfer [1, 5, 6, 13, 19, 21, 28, 30, 33, 47, 52, 53, 57] renders a photo with a specific visual style by transferring style patterns from a given style image to a content image. Similar to painterly image harmonization, artistic style transfer methods can also be divided into optimization-based methods [12, 13, 24, 28] and feed-forward methods [11, 19, 20, 27, 33, 39]. Artistic style transfer methods can be applied to painterly image harmonization task by transferring the style from background image to the whole content image and pasting the cropped stylized foreground on the background image. However, the foreground region is prone to be insufficiently stylized. Moreover, the pasted foreground may not be naturally blended into the background without considering the locality of compositing task.

3. Our Method

3.1. Overview

By pasting the foreground object from a photographic image on a painterly background image I_s, we can obtain the composite image I_c with foreground mask M. The goal of painterly image harmonization is transferring the style of background image I_s to the foreground object in the composite image I_c while preserving the foreground content.

The overview of our network is shown in Figure 2 , which contains a dual-encoder generator G, pixel-wise feature discriminators D_f^l, and a pixel-wise image discriminator D_m. The dual-encoder generator G consists of a main encoder E_m and a residual encoder E_r. The generator G takes in the background image I_s, the composite image I_c, and the foreground mask M, and generates a harmonized output image $\tilde{I}_o$. In addition, we employ L pixel-wise feature discriminators $\{D_f^l|_{l=1}^L\}$ and a pixel-wise image discriminator D_m to play against G by telling disharmonious pixels from harmonious ones. The pixel-wise feature discriminators D_f^l are attached to multiple layers of feature maps in the generator, while the pixel-wise image discriminator D_m is attached to the output image $\tilde{I}_o$. Next, we will introduce each component in our network.

3.2. Dual-encoder Generator

Our generator is composed of a main encoder E_m, a residual encoder E_r, and a decoder. The main encoder E_m contains the first few layers (up to $ReLU4\_1$) of a pre-trained VGG-19 [43] and the decoder structure is symmetrical to the main encoder. We fix the main encoder E_m when training our network. Following [42], we add skip connections on $ReLU1\_1$, $ReLU2\_1$, and $ReLU3\_1$ to preserve the content details in the low-level feature maps.

At first, E_m extracts $L = 4$ layers of feature maps from the background image I_s and the composite image I_c, leading to $\{F_s^l|_{l=1}^4\}$ and $\{F_c^l|_{l=1}^4\}$ from four encoder layers $ReLU1\_1$, $ReLU2\_1$, $ReLU3\_1$, and $ReLU4\_1$. For the l-th layer, we feed both feature maps F_s^l and F_c^l with the resized foreground mask M^l to the AdaIN layer [19] that aligns the statistics of the foreground region in F_c^l with those of F_s^l, producing the stylized feature maps F_a^l:

$$F_a^l = \Big(\sigma(F_s^l)\frac{F_c^l - \mu(F_c^l \circ M^l)}{\sigma(F_c^l \circ M^l)} + \mu(F_s^l)\Big) \circ M^l + F_c^l \circ (1 - M^l), \quad (1)$$

where $\circ$ is Hadamard product, $\mu(\cdot)$ and $\sigma(\cdot)$ are the channel-wise mean and standard deviation of a certain region of a feature map.

Although the AdaIN operation in Eqn. 1 roughly aligns the composite foreground with the background image, the domain gap between foreground and background in F_a^l may still exist. Therefore, we attempt to refine the foreground details in the stylized feature maps to further reduce the domain gap. To this end, we design a residual encoder E_r to learn multiple layers of residual features that are added to the foreground regions of stylized feature maps $\{F_a^l|_{l=1}^4\}$.

Our residual encoder E_r takes the concatenation of the composite image I_c and the foreground mask M as input. We employ four residual blocks to learn four layers of residual features. All residual blocks share the identical structure, that is, two convolutional filters followed by batch-normalization layer and ReLU activation. For the l-th layer, the learned residual features F_r^l, *i.e.*, the output from the l-th residual block, are added to the foreground region in the stylized feature map F_a^l, leading to refined feature map $\tilde{F}_a^l$:

$$\tilde{F}_a^l = F_a^l + F_r^l \circ M^l. \quad (2)$$

Then, multiple layers of refined feature maps are delivered to the decoder through bottleneck or skip connection to generate the output image I_o. Afterwards, inspired by [44], we adopt a blending layer to blend I_o with the background image I_s. In particular, we feed the concatenation of the final decoder feature map and the foreground mask M to the blending layer [44], generating a soft mask $\tilde{M}$. At last, we blend the output image I_o with the background image I_s using $\tilde{M}$ to obtain the final harmonized image $\tilde{I}_o$:

$$\tilde{I}_o = I_o \circ \tilde{M} + I_s \circ (1 - \tilde{M}). \quad (3)$$

3.3. Pixel-wise Feature Discriminator

To supervise the learned residual features and mitigate the foreground-background domain gap in the refined feature maps, we employ pixel-wise adversarial learning to encourage the foreground pixels to be indistinguishable from the background pixels in the refined feature maps.

We attach a pixel-wise feature discriminator D_f^l to the l-th layer of refined feature map $\tilde{F}_a^l$. D_f^l aims to distinguish

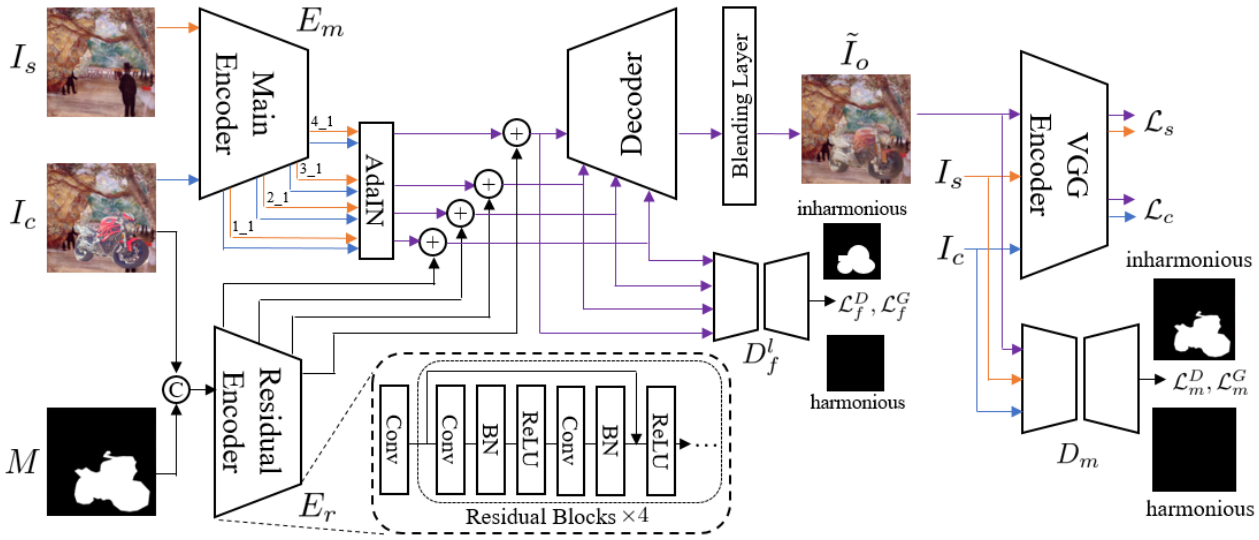

Figure 2. An overview of our painterly image harmonization network PHARNet. The network contains a dual-encoder generator G, pixel-wise feature discriminators D_f^l, and a pixel-wise image discriminator D_m.

inharmonious pixels from harmonious pixels and assign a class label to each pixel in the feature map. Considering the output format, we adopt encoder-decoder architecture for D_f^l, which produces a mask. We use $D_f^l(\tilde{F}_a^l)$ to denote the discriminator output for $\tilde{F}_a^l$. $D_f^l(\tilde{F}_a^l)$ should be close to M^l so that the discriminator is guided to distinguish the foreground pixels from background pixels, in which the foreground pixels are labeled as 1 and the background pixels are labeled as 0. We also feed the feature map F_s^l of background image into the discriminator. Since there are no inharmonious pixels in the background image, all pixels should be labeled as 0. Therefore, the loss function to train the discriminator D_f^l can be written as

$$L_f^D = \sum_{l=1}^{4} \|D_f^l(\tilde{F}_a^l) - M^l\|_2^2 + \sum_{l=1}^{4} \|D_f^l(F_s^l)\|_2^2. \quad (4)$$

When training the generator G, we expect that the foreground pixels are indistinguishable from the background pixels in the refined feature maps, that is, all pixels should be labeled as 0. Thus, the loss function for D_f^l can be written as

$$L_f^G = \sum_{l=1}^{4} \|D_f^l(\tilde{F}_a^l)\|_2^2. \quad (5)$$

Note that, unlike the commonly used global discriminator which classifies an image or a feature map to be real or fake as a whole, our pixel-wise feature discriminator learns to classify each pixel-wise feature vector separately.

3.4. Other Losses

In this section, we introduce the remaining losses imposed on the final harmonized image $\tilde{I}_o$.

We employ the style loss [19] to ensure that the style of foreground object is close to that of background image:

$$L_s = \sum_{l=1}^{4} \|\mu(\Psi^l(\tilde{I}_o) \circ M^l) - \mu(\Psi^l(I_s))\|_2^2 + \sum_{l=1}^{4} \|\sigma(\Psi^l(\tilde{I}_o) \circ M^l) - \sigma(\Psi^l(I_s))\|_2^2, \quad (6)$$

in which Ψ^l denotes the l-th *ReLU-l_1* layer in the pre-trained VGG-19 encoder.

We also employ the content loss [13] to enforce the harmonized image to retain the content of the foreground object:

$$L_c = \|\Psi^4(\tilde{I}_o) - \Psi^4(I_c)\|_2^2. \quad (7)$$

Inspired by [18], we also apply pixel-wise adversarial learning to the harmonized image $\tilde{I}_o$. Specifically, we train a pixel-wise image discriminator D_m to distinguish inharmonious pixels from harmonious pixels by minimizing the loss L_m^D, while the generator strives to make the foreground pixels indistinguishable from background pixels by minimizing the loss L_m^G. The definitions of L_m^D and L_m^G are similar to L_f^D in Eqn. 4 and L_f^G in Eqn. 5 except the input, so we omit the details here.

In summary, the total loss function for training the generator G is

$$L_G = L_c + L_s + L_f^G + L_m^G. \quad (8)$$

The total loss function for training the discriminators $\{D_f^l|_{l=1}^4\}$ and D_m is

$$L_D = L_f^D + L_m^D. \tag{9}$$

Under the adversarial learning framework, we update the generator and the discriminators alternatingly.

4. Experiments

4.1. Datasets

Following previous works on painterly image harmonization [35,40,56], we conduct experiments on COCO [31] and WikiArt [36] datasets. COCO is a large-scale dataset of 123,287 images, which have instance segmentation annotations for the objects from 80 categories. Wikiart is a large-scale digital art dataset which contains 81444 images from 27 different styles. In this work, we use the images from WikiaArt dataset as painterly background images and extract photographic foreground objects from COCO dataset using the provided instance segmentation masks. We randomly choose a segmented object whose area ratio in the original image is in the range of [0.05, 0.3], and paste it onto a randomly selected painting background, producing an inharmonious composite image. We follow the training and test split of COCO and WikiArt as [49], based on which we obtain 57,025 (*resp.*, 24,421) background images and 82783 (*resp.*, 40504) foreground objects for training (*resp.*, testing).

4.2. Implementation Details

The overall architecture of our network has been described in Section 3. For the residual encoder E_r, we use four residual blocks [17] to learn the residual features. All residual blocks share the identical structure, that is, two convolutional filters followed by batch-normalization layer and ReLU activation. The pixel-wise feature discriminators D_f^l are small-scale auto-encoders consisting of downsample (DS) and upsample (US) blocks. For $l \in \{1, 2\}$, D_f^l contains three DS blocks and three US blocks. For $l \in \{3, 4\}$, D_f^l contains two DS blocks and two US blocks. Each DS block contains a convolutional layer with kernel size being 4 and stride being 2, a batch normalization layer, and a LeakyReLU activation sequentially. Each US block contains an upsampling layer with scale factor being 2, a reflection padding layer, a convolutional layer with kernel size being 3 and stride being 1, a batch normalization layer, and a ReLU layer. The pixel-wise image discriminator D_m is also built upon DS blocks and US blocks as used in D_f^l. For D_m, we employ seven DS blocks and seven US blocks.

Our network is implemented with Pytorch 1.10.2 and trained using Adam optimizer with learning rate of $2e-4$ on ubuntu 18.04 LTS operating system, which has 32GB of memory, Intel Core i7-9700K CPU, and two GeForce GTX 2080 Ti GPUs. We resize the input images to 256×256 in the training stage. However, our network can be applied to the test images of arbitrary size due to the fully convolutional network structure.

4.3. Baselines

There are two groups of methods which can be applied to our task: painterly image harmonization [35, 40, 56] and artistic style transfer [19, 33].

For the first group of methods, we compare with Deep Image Blending [56] ("DIB" for short), Deep Painterly Harmonization [35] ("DPH" for short), and E2STN [40]. We also include traditional image blending method Poisson Image Editing [41] ("Poisson" for short) for comparison.

For the second group of methods, they were originally proposed to migrate the style of an artistic image to a complete photographic image, so some modifications are required to adapt them to our task. In particular, we first migrate the style of background image to the photographic image containing the foreground object, using the artistic style transfer methods. Then, we segment the stylized foreground object and overlay it on the background image to obtain a harmonized image. Since there are myriads of artistic style transfer methods, we choose several iconic or recent works for comparison: WCT [27], AdaIN [19], SANet [39], AdaAttN [33], and StyTr2 [11].

4.4. Qualitative Analysis

We show the comparison with the first group of baselines in Figure 3 and the comparison with the second group of baselines in Figure 4. More visualization results could be found in Supplementary.

As shown in Figure 3, Poisson [41] can smoothen the boundary between foreground and background, but the foreground content is severely distorted (*e.g.*, row 2, 5). DIB [56] and E2STN [40] preserve the foreground content well, but the foreground style is not very close to background style (*e.g.*, E2STN in row 4, DIB in row 2) and the harmonized foreground may be corrupted (*e.g.*, DIB in row 5). DPH [35] is a competitive baseline, which can achieve good harmonized results in some cases. However, the content structure and foreground boundary might be damaged or blurred (*e.g.*, row 2, 5). In comparison, our method can preserve the content structure, sharp boundaries, and rich details (*e.g.*, human face/clothes in row 1, 3 and the patterns on the giraffe body in row 5). In the meanwhile, the foreground is sufficiently stylized and harmonious with the background. Interestingly, without suppressing the stylization effect, our method can also maintain the color distribution of foreground (*e.g.*, white-and-red car in row 4), while other methods either understylize the foreground or lose partial color distribution information.

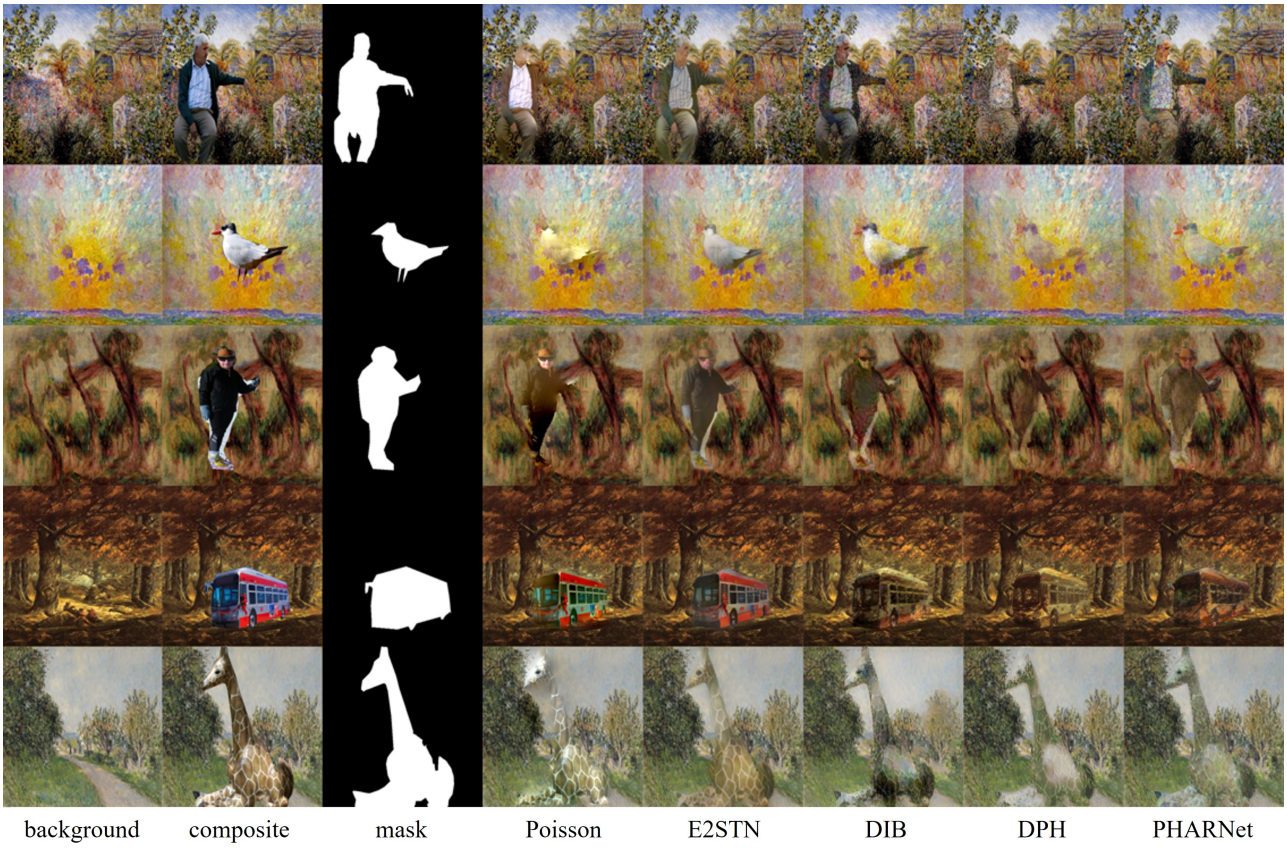

Figure 3. From left to right, we show the background image, composite image, foreground mask, the harmonized results of Poisson [41], E2STN [40], DIB [56], DPH [35], and our PHARNet.

Method	SANet [39]	AdaAttN [33]	StyTr2 [11]	E2STN [40]	DPH [35]	PHARNet
BT-Score	-1.8757	-1.0406	-0.3891	0.4677	0.7814	2.0562
Time(s)	0.0097	0.0115	0.0504	0.0078	270.96	0.0223

Table 1. The BT-score and inference time of different methods.

As shown in Figure 4, since the style transfer methods do not focus on stylizing the foreground region, the foreground may not be adequately stylized (*e.g.*, AdaIN and StyTr2 in row 2) and the content structure of foreground may be destroyed (*e.g.*, WCT in row 4, 5). Besides, since style transfer methods do not consider the location of foreground in the composite image, the stylized foreground may be incompatible with the surrounding background. In contrast, our method is able to transfer the style and retain the content structure, leading to more visually appealing results. The stylized foregrounds are harmonious with backgrounds, as if they originally exist in the paintings.

4.5. User Study

We randomly select 100 foreground objects and 100 background images to generate 100 composite images for user study. We compare with 5 representative baselines SANet [39], AdaAttN [33], StyTr2 [11], E2STN [40], DPH [35]. Specifically, for each composite image, we can obtain 6 harmonized images produced by 6 methods, based on which 2 images are selected to construct an image pair. Provided with 100 composite images, we can construct in total 1500 image pairs. Then, we ask 50 annotators to observe one image pair at a time and pick the better one. At last, we gather $30,000$ pairwise results and calculate the overall ranking of all methods using Bradley-Terry (B-T) model [3, 25]. As shown in Table 1, our method achieves the highest B-T score.

4.6. Efficiency Comparison

We compare the inference time between our method and baseline methods in Table 1. We test the inference speed of

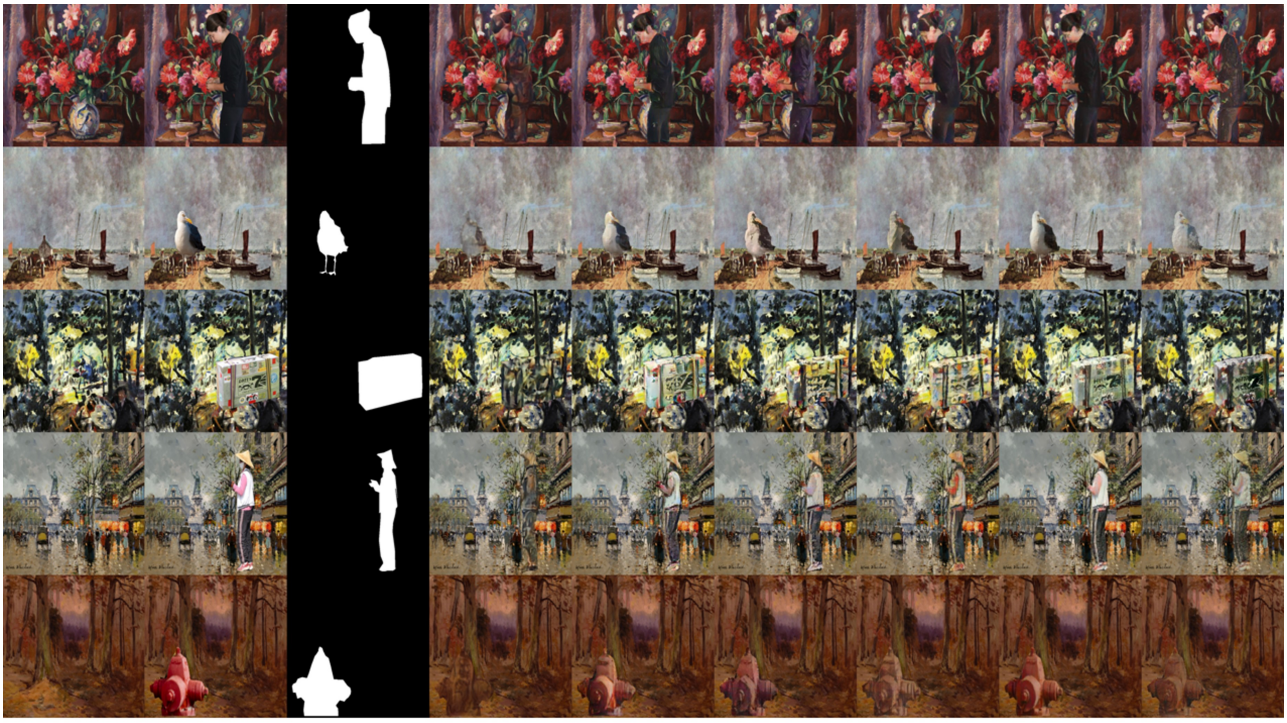

Figure 4. From left to right, we show the background image, composite image, foreground mask, the harmonized results of WCT [27], AdaIN [19], SANet [39], AdaAttN [33], StyTr2 [11], and our PHARNet.

Method	E_r	D_f^l	D_m	B-T score
V1				-4.4103
V2			✓	-0.6537
V3	✓		✓	1.4343
V4	✓	✓	✓	3.6297

Table 2. The B-T score of our different network structure. E_r refers to the residual encoder. D_f^l refers to the pixel-wise feature discriminators. D_m refers to the pixel-wise image discriminator.

all methods on one GeForce GTX 2080 Ti GPU, with input image size 256×256, and average the results over 100 test images. We observe that DPH is the slowest method because DPH is an optimization-based method which requires iterative optimization process. StyTr2 [11] is also very slow due to the Transformer network structure. Our method is relatively efficient and the inference speed is acceptable for real-time applications.

4.7. Ablation Studies

In this section, we investigate the effectiveness of each component in our method. We first remove all discriminators and the residual encoder, and obtain a basic network with multi-scale AdaIN, which is referred to as V1 in Table 2. Then, we add pixel-wise image discriminator D_m, which is referred to as V2. Furthermore, we add the residual encoder E_r to form the dual-encoder generator, which is referred to as V3. Finally, we apply pixel-wise feature discriminators and reach our full-fledged method, which is referred to as V4.

We show the harmonized results of ablated versions in Figure 5. It can be seen that the harmonized results of V1 have many strip artifacts, which significantly harms the quality of harmonized results. After using the pixel-wise image discriminator D_m in V2, the strip artifacts can be removed. Nevertheless, the harmonized results of V2 may still have some other types of artifacts (*e.g.*, row 4, row 5) and unsatisfactory details. After adding the residual features without the guidance of pixel-wise feature discriminator, the harmonized foregrounds of V3 may have distorted content (*e.g.*, row 5) and look incompatible with the background (*e.g.*, row 4). After applying pixel-wise feature discriminators to the refined feature map in V4, the learnt residual features become more reasonable and the harmonized results become more visually appealing. Compared with the ablated versions, the results of V4 have fully-transferred style, well-preserved content structure, and meaningful details (*e.g.*, dog eye in row 1, suspender in row 3).

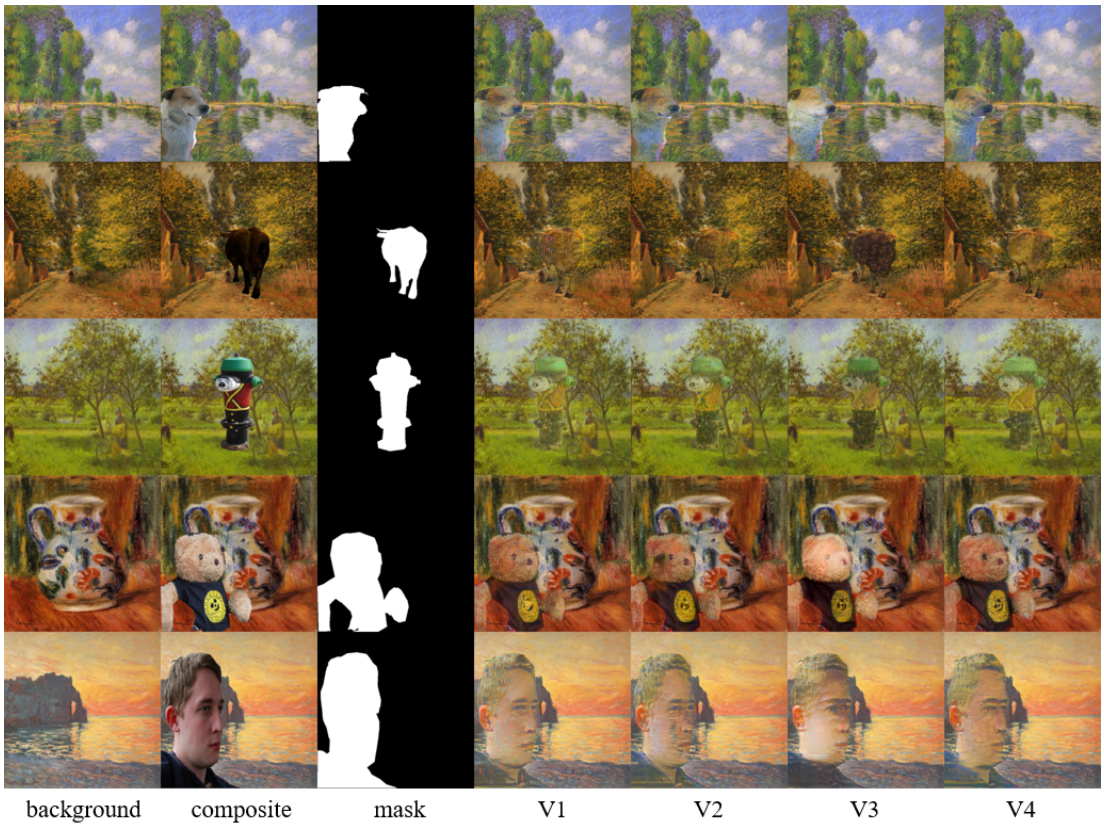

Figure 5. From left to right, we show the background image, composite image, foreground mask, the harmonized results of V1, V2, V3, V4 (full method).

The clear advantage of V4 could be attributed to the residual features and pixel-wise adversarial learning. The residual features, which are added to the foreground region of stylized feature map, could repair the content structure and enhance the style representations, leading to the refined feature map with improved quality. Moreover, the pixel-wise feature discriminator plays against the dual-encoder generator by telling disharmonious pixels from harmonious ones. Such pixel-wise adversarial learning encourages the refined foreground feature map to be indistinguishable from background feature map, so that the foreground is more harmonious with the surrounding background.

Similar to Table 1, we also conduct user study to compare different ablated versions. The results are summarized in Table 2, which again demonstrate the superiority of our full method.

5. Discussion on Limitation

Although our method can generally produce visually appealing results, there still exist some challenging cases in which our method may fail to produce satisfactory results. For example, as shown in Figure 6, when the foreground objects are very small, our method may fail in retaining the foreground content information and produce poor harmonized results.

Figure 6. Example failure cases of our method.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 62076162), the Shanghai Municipal Science and Technology Major/Key Project, China (Grant No. 2021SHZDZX0102, Grant No. 20511100300).

References

[1] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *CVPR*, 2021. 3

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017. 2

[3] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 6

[4] Junyan Cao, Yan Hong, and Li Niu. Painterly image harmonization in dual domains. In *AAAI*, 2023. 2

[5] Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. *NeurIPS*, 2021. 3

[6] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Dualast: Dual style-learning networks for artistic style transfer. In *CVPR*, 2021. 3

[7] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Bargainnet: Background-guided domain translation for image harmonization. In *ICME*, 2021. 2

[8] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *CVPR*, 2022. 2

[9] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020. 1, 2

[10] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020. 1, 2

[11] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *CVPR*, 2022. 3, 5, 6, 7

[12] Len Du. How much deep learning does neural style transfer really need? an ablation study. In *WACV*, 2020. 3

[13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 1, 2, 3, 4

[14] Zonghui Guo, Dongsheng Guo, Haiyong Zheng, Zhaorui Gu, Bing Zheng, and Junyu Dong. Image harmonization with transformer. In *ICCV*, 2021. 2

[15] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *CVPR*, 2021. 2

[16] Guoqing Hao, Satoshi Iizuka, and Kazuhiro Fukui. Image harmonization with attention-based deep feature modulation. In *BMVC*, 2020. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[18] Hao-Zhi Huang, Sen-Zhe Xu, Jun-Xiong Cai, Wei Liu, and Shi-Min Hu. Temporally coherent video harmonization using adversarial networks. *IEEE Transactions on Image Processing*, 29:214–224, 2019. 2, 4

[19] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1, 2, 3, 4, 5, 7

[20] Jing Huo, Shiyin Jin, Wenbin Li, Jing Wu, Yu-Kun Lai, Yinghuan Shi, and Yang Gao. Manifold alignment for semantically aligned style transfer. In *ICCV*, 2021. 3

[21] Yongcheng Jing, Yining Mao, Yiding Yang, Yibing Zhan, Mingli Song, Xinchao Wang, and Dacheng Tao. Learning graph neural networks for image style transfer. *ECCV*, 2022. 3

[22] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *ECCV*, 2022. 2

[23] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *NeurIPS*, 2019. 2

[24] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, 2019. 3

[25] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *CVPR*, 2016. 6

[26] Jean-Francois Lalonde and Alexei A Efros. Using color compatibility for assessing image realism. In *ICCV*, 2007. 2

[27] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *NeurIPS*, 2017. 3, 5, 7

[28] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 3

[29] Jingtang Liang, Xiaodong Cun, and Chi-Man Pun. Spatial-separated curve rendering network for efficient and high-resolution image harmonization. In *ECCV*, 2022. 2

[30] Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *CVPR*, 2021. 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

[32] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *CVPR*, 2021. 1, 2

[33] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *ICCV*, 2021. 3, 5, 6, 7

[34] Lingxiao Lu, Jiangtong Li, Junyan Cao, Li Niu, and Liqing Zhang. Painterly image harmonization using diffusion model. In *ACM MM*, 2023. 2

[35] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep painterly harmonization. In *CGF*, 2018. 1, 2, 5, 6

[36] Kiri Nichol. Painter by numbers. https://www.kaggle.com/competitions/painter-by-numbers/data, 2016. 2, 5

[37] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021. 1

[38] Li Niu, Linfeng Tan, Xinhao Tao, Junyan Cao, Fengjun Guo, Teng Long, and Liqing Zhang. Deep image harmonization with globally guided feature transformation and relation distillation. In *ICCV*, 2023. 2

[39] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7

[40] Hwai-Jin Peng, Chia-Ming Wang, and Yu-Chiang Frank Wang. Element-embedded style transfer networks for style harmonization. In *BMVC*, 2019. 1, 2, 5, 6

[41] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*. 2003. 5, 6

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 2, 3

[44] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. In *WACV*, 2021. 2, 3

[45] Shuangbing Song, Fan Zhong, Xueying Qin, and Changhe Tu. Illumination harmonization with gray mean scale. In *CGI*, 2020. 2

[46] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010. 2

[47] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *CVPR*, 2020. 3

[48] Linfeng Tan, Jiangtong Li, Li Niu, and Liqing Zhang. Deep image harmonization in dual color spaces. In *ACM MM*, 2023. 2

[49] Wei Ren Tan, Chee Seng Chan, Hernán E Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorical gans. In *ICIP*, 2017. 5

[50] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017. 1, 2

[51] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 2

[52] Zhijie Wu, Chunjin Song, Yang Zhou, Minglun Gong, and Hui Huang. Efanet: Exchangeable feature alignment network for arbitrary style transfer. In *AAAI*, 2020. 3

[53] Wenju Xu, Chengjiang Long, Ruisheng Wang, and Guanghui Wang. Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer. In *ICCV*, 2021. 3

[54] Ben Xue, Shenghui Ran, Quan Chen, Rongfei Jia, Binqiang Zhao, and Xing Tang. Dccf: Deep comprehensible color filter learning framework for high-resolution image harmonization. In *ECCV*, 2022. 2

[55] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and improving the realism of image composites. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 2

[56] Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. Deep image blending. In *WACV*, 2020. 1, 2, 5, 6

[57] Yuxin Zhang, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, Tong-Yee Lee, and Changsheng Xu. Domain enhanced arbitrary image style transfer via contrastive learning. *SIGGRAPH*, 2022. 3

[58] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *ICCV*, 2015. 2