Progressive Painterly Image Harmonization from Low-Level Styles to High-Level Styles

Li Niu[*], Yan Hong, Junyan Cao, Liqing Zhang

MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
{ustcnewly, hy2628982280, joy_c1, lqzhang}@sjtu.edu.cn

Abstract

Painterly image harmonization aims to harmonize a photographic foreground object on the painterly background. Different from previous auto-encoder based harmonization networks, we develop a progressive multi-stage harmonization network, which harmonizes the composite foreground from low-level styles (*e.g.*, color, simple texture) to high-level styles (*e.g.*, complex texture). Our network has better interpretability and harmonization performance. Moreover, we design an early-exit strategy to automatically decide the proper stage to exit, which can skip the unnecessary and even harmful late stages. Extensive experiments on the benchmark dataset demonstrate the effectiveness of our progressive harmonization network.

1 Introduction

As a common photo editing technique, image composition refers to splicing a foreground object from one image and overlaying it on another background image. However, the foreground and background in the obtained composite image may have incompatible styles, which severely harms the quality of composite image. Given a composite image, if the foreground is from a photographic image and the background is an artistic painting (see Figure 1), the background style could be defined similarly as in artistic style transfer (Gatys, Ecker, and Bethge 2016; Huang and Belongie 2017; Park and Lee 2019), which includes color, texture, and so on. The task to mitigate the style mismatch between foreground and background in such composite images is called painterly image harmonization (Luan et al. 2018), which adjusts the style of foreground to make it compatible with the background and naturally embedded into the background.

As far as we are concerned, there exist only few works on painterly image harmonization, which can be divided into optimization-based methods (Luan et al. 2018; Zhang, Wen, and Shi 2020) and feed-forward methods (Cao, Hong, and Niu 2023; Yan et al. 2022). Optimization-based methods (Luan et al. 2018; Zhang, Wen, and Shi 2020) update the composite foreground iteratively to minimize the designed loss functions, which is too slow for real-time application. In contrast, feed-forward methods only pass the composite image through the network once to produce a harmonized image, which is much more efficient than optimization-based methods. Among the feed-forward methods, Peng, Wang, and Wang (2019) introduced AdaIN (Huang and Belongie 2017) into painterly image harmonization. Yan et al. (2022) proposed to combine the advantages of transformer and CNN. Cao, Hong, and Niu (2023) explored harmonizing the composite image in both spatial domain and frequency domain. Some diffusion model based methods (Meng et al. 2021; Hachnochi et al. 2023; Lu, Liu, and Kong 2023) for cross-domain image composition can also be applied to this task. However, the above methods are still struggling to preserve the content details and transfer the styles sufficiently. Moreover, the above methods directly output the final harmonized image, which function like a black box.

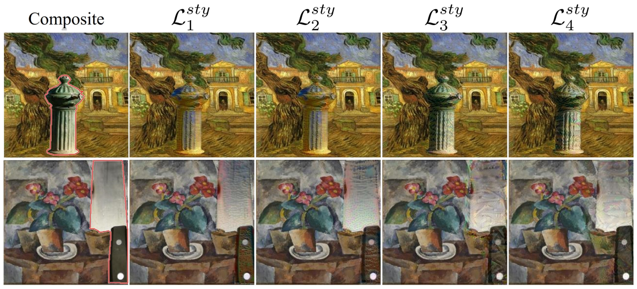

Figure 1: The leftmost column shows the composite images with the foregrounds outlined in red. The rest columns show the harmonized images obtained by minimizing the style loss $\mathcal{L}^{sty}$ in different VGG-19 encoder stages, *e.g.*, $\mathcal{L}_1^{sty}$ is the style loss in the first VGG-19 encoder stage.

In this work, we propose an interpretable network structure which harmonizes the composite image progressively. In previous works (Peng, Wang, and Wang 2019; Cao, Hong, and Niu 2023), style is usually represented by the feature statistics (*e.g.*, mean, variance) of pretrained VGG-19 (Simonyan and Zisserman 2015) encoder. We observe that the feature statistics in shallow and deep encoder stages represent low-level and high-level styles respectively, which has also been discussed in previous literature (Yao et al. 2019; Cho et al. 2019; Liu et al. 2021). As shown in Figure 1, we optimize the composite foreground by minimizing the style

[*]Corresponding author.

loss in different encoder stages and the content loss (see Section 3.1 for the details of style and content loss). From left to right, the adjusted style of composite foreground transits from low-level style (*e.g.*, color, simple texture) to high-level style (*e.g.*, complex texture). In the first row, $\mathcal{L}_1^{sty}$ changes the foreground color from green to yellow, while $\mathcal{L}_4^{sty}$ adds wave texture to the foreground. In the second row, $\mathcal{L}_1^{sty}$ changes the small-scale texture of foreground, while $\mathcal{L}_4^{sty}$ changes the large-scale texture, in which the large-scale texture means that the texture is perceivable with large receptive field.

Motivated by these observations, we design a dual-branch network which harmonizes the composite image from low-level styles to high-level styles progressively. The top branch is a pretrained VGG-19 (Simonyan and Zisserman 2015) encoder with four encoder stages, in which we apply AdaIN (Huang and Belongie 2017) to harmonize the output feature map in each encoder stage. The bottom branch has four stages corresponding to four encoder stages in the top branch, and maintain the full resolution through four stages. In the k-th stage, the bottom branch fuses the harmonized feature maps from the first to the k-th encoder stage in the top branch, and the output image is harmonized up to the k-th style level. *The progressive network has several advantages over previous networks that entangle all levels of styles.* Firstly, our network owns better interpretability, reflecting how the composite image is harmonized from low-level styles to high-level styles. Secondly, from low-level style transfer to high-level style transfer, the multi-stage network gradually raises the difficulty level of harmonization task, which might be easier than handling all style levels simultaneously.

In practice, we observe that for some composite images, the early stages could produce comparable or even better results than the late stages. One reason is that in some cases, it is sufficient to only transfer low-level styles. Another reason is that when struggling to transfer high-level styles, the content information may be sacrificed. Therefore, we further propose an early-exit strategy, which uses sequential model to forecast the proper stage to exit the network. *The early-exit strategy can benefit the harmonization quality by discarding unnecessary and even harmful late stages.* We name our method as ProPIH (**Pro**gressive **P**ainterly **I**mage **H**armonization).

Our contributions can be summarized as follows: 1) We design a progressive painterly harmonization network ProPIH, which can harmonize the composite image from low-level styles to high-level styles. 2) We propose an early-exit strategy, which can automatically determine the proper stage to exit the network. 3) Extensive experiments on COCO (Lin et al. 2014) and WikiArt (Nichol 2016) demonstrate the effectiveness of our proposed method.

2 Related Work

2.1 Image Harmonization

Image harmonization aims to harmonize a composite image by adjusting foreground illumination to match background illumination. In recent years, abundant deep image harmonization methods (Tsai et al. 2017; Jiang et al. 2021; Xing et al. 2022; Peng et al. 2022; Zhu et al. 2022; Valanarasu et al. 2023; Liu et al. 2023) have been developed. For example, (Cun and Pun 2020; Hao, Iizuka, and Fukui 2020; Sofiiuk, Popenova, and Konushin 2021) proposed diverse attention modules to treat the foreground and background separately, or establish the relation between foreground and background. (Cong et al. 2020; Ling et al. 2021; Hang et al. 2022) directed image harmonization to domain translation or style transfer by treating different illumination conditions as different domains or styles. (Guo et al. 2021a,b, 2022) decomposed an image into reflectance map and illumination map. More recently, (Cong et al. 2022; Ke et al. 2022; Liang, Cun, and Pun 2022; Xue et al. 2022; Guerreiro, Nakazawa, and Stenger 2023; Wang et al. 2023) used deep network to predict color transformation, striking a good balance between efficiency and effectiveness. However, most image harmonization methods only adjust illumination and require ground-truth supervision, which is unsuitable for painterly image harmonization.

2.2 Painterly Image Harmonization

Different from Section 2.1, in painterly image harmonization, the foreground is a photographic object while the background is an artistic painting. The goal of painterly image harmonization is adjusting the foreground style to match background style and preserving the foreground content. The existing methods can be roughly categorized into optimization-based methods and feed-forward methods. The optimization-based methods (Luan et al. 2018; Zhang, Wen, and Shi 2020) iteratively optimize over the composite foreground to minimize the designed loss functions. The feed-forward methods (Peng, Wang, and Wang 2019; Yan et al. 2022; Cao, Hong, and Niu 2023) send the composite image through the harmonization network once and generate the harmonized image. Several diffusion model based methods (Meng et al. 2021; Hachnochi et al. 2023) for cross-domain image composition can also be used to harmonize the composite image.

Different from previous network structures, we design a novel progressive harmonization network which can harmonize a composite image from low-level to high-level styles.

2.3 Artistic Style Transfer

Artistic style transfer (Kolkin, Salavon, and Shakhnarovich 2019; Jing et al. 2020; Chen et al. 2021, 2017; Sanakoyeu et al. 2018; Wang et al. 2020; Li et al. 2018; Chen and Schmidt 2016; Sheng et al. 2018; Gu et al. 2018; Zhang et al. 2019; Chen et al. 2022; Huo et al. 2021) targets at recomposing a content image in the style of a style image. Amounts of works (Huang and Belongie 2017; Li et al. 2017, 2019b) focus on global style transfer. Recently, some works (Park and Lee 2019; Liu et al. 2021; Deng et al. 2022) turn to fine-grained local style transfer by establishing local correspondences between content image and style image. More recently, diffusion model has been explored for artistic style transfer (Zhang et al. 2023). Artistic style transfer stylizes an

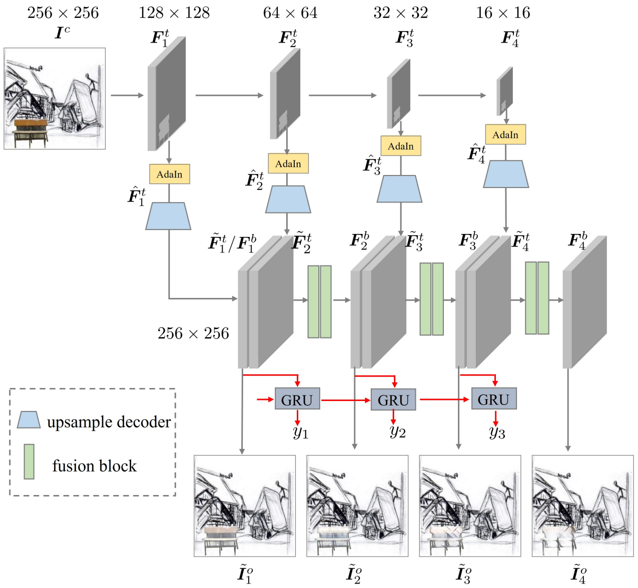
256×256
$\boldsymbol{I}^c$
128×128
$\boldsymbol{F}_1^t$
64×64
$\boldsymbol{F}_2^t$
32×32
$\boldsymbol{F}_3^t$
16×16
$\boldsymbol{F}_4^t$
AdaIn
$\hat{\boldsymbol{F}}_1^t$
$\hat{\boldsymbol{F}}_2^t$
$\hat{\boldsymbol{F}}_3^t$
$\hat{\boldsymbol{F}}_4^t$
$\tilde{\boldsymbol{F}}_1^t/\boldsymbol{F}_1^b$
$\tilde{\boldsymbol{F}}_2^t$
$\boldsymbol{F}_2^b$
$\tilde{\boldsymbol{F}}_3^t$
$\boldsymbol{F}_3^b$
$\tilde{\boldsymbol{F}}_4^t$
$\boldsymbol{F}_4^b$
256×256
GRU
y_1
y_2
y_3
upsample decoder
fusion block
$\tilde{\boldsymbol{I}}_1^o$
$\tilde{\boldsymbol{I}}_2^o$
$\tilde{\boldsymbol{I}}_3^o$
$\tilde{\boldsymbol{I}}_4^o$

Figure 2: Our ProPIH consists of the top branch (pretrained VGG-19 encoder) and the bottom branch. The k-stage in the bottom branch fuses the encoder features up to the k-th stage, and produces the image harmonized up to the k-th style level. GRU predicts the exit label indicating whether to exit the network in the current stage.

entire content image, whereas painterly image harmonization stylizes the foreground object in the composite image.

A few style transfer methods (Wang et al. 2017; Luo, Han, and Yang 2022) have coarse-to-fine network structure, but *"coarse-to-fine" means moving from coarse-grained structure to fine-grained details, which is essentially different from our transition from low-level styles to high-level styles.* Some style transfer works (Yao et al. 2019; Cho et al. 2019; Liu et al. 2021) have discussed low-level and high-level styles, but they still transfer multi-level styles simultaneously, instead of from low-level styles to high-level styles progressively.

3 Our Method

We suppose that a composite image $\boldsymbol{I}^c$ is obtained by pasting a photographic foreground on the painterly background $\boldsymbol{I}^b$. We use foreground mask $\boldsymbol{M}^f$ to indicate the foreground region and $\boldsymbol{M}^b$ is the complementary background mask. Painterly image harmonization aims to transfer the style from background region to foreground region while preserving the content of foreground, giving rise to a harmonized image $\tilde{\boldsymbol{I}}^o$.

As shown in Figure 2, our network has two branches. The top branch is the pretrained VGG-19 network (Simonyan and Zisserman 2015). The bottom branch progressively fuses the upsampled encoder feature maps from the top branch to produce the multi-stage outputs which are harmonized up to different style levels. Specifically, the output from early (*resp.*, late) stage is harmonized up to low (*resp.*, high) style level. We also attach GRU to the bottom branch to predict the exit stage, so that we can exit the network at the earliest proper stage. Next, we will introduce our network structure in Section 3.1 and early-exit strategy in Section 3.2.

3.1 Progressive Harmonization Network

As illustrated in Figure 2, the top branch is the pretrained VGG-19 encoder (Simonyan and Zisserman 2015), where we only use the first few layers up to *ReLU-4_1*. We divide the encoder to four stages, and the last layer of four stages are *ReLU-1_1*, *ReLU-2_1*, *ReLU-3_1*, and *ReLU-4_1* respectively. The output feature map from the k-th stage in the top branch is denoted as $\boldsymbol{F}_k^t$, which is comprised of the foreground feature map $\boldsymbol{F}_k^{t,f}$ and background feature map $\boldsymbol{F}_k^{t,b}$. To transfer the background style to the foreground, we match the channel-wise statistics (mean, standard deviation) of $\boldsymbol{F}_k^{t,f}$ with those of $\boldsymbol{F}_k^{t,b}$ using Adaptive Instance Normalization (AdaIN) (Huang and Belongie 2017), which can be formulated as follows,

$$\hat{\boldsymbol{F}}_k^{t,f} = \sigma(\boldsymbol{F}_k^{t,b}) \frac{\boldsymbol{F}_k^{t,f} - \mu(\boldsymbol{F}_k^{t,f})}{\sigma(\boldsymbol{F}_k^{t,f})} + \mu(\boldsymbol{F}_k^{t,b}), \tag{1}$$

in which $\mu(\cdot)$ (*resp.*, $\sigma(\cdot)$) means calculating the mean (*resp.*, standard deviation) of the specified feature map. The stylized foreground $\hat{\boldsymbol{F}}_k^{t,f}$ is combined with the unchanged background $\boldsymbol{F}_k^{t,b}$ to form the harmonized feature map $\hat{\boldsymbol{F}}_k^t$.

For each stage in the top branch, we attach a lightweight decoder which upsamples the harmonized feature map to the full resolution (*e.g.*, 256×256). The upsample decoder for the k-th stage contains one conv block and k upsample blocks. We denote the output feature map from the k-th upsample decoder as $\tilde{\boldsymbol{F}}_k^t$.

In the bottom branch, for the first stage, we directly use the output feature map $\tilde{\boldsymbol{F}}_1^t$ to produce the harmonized image $\tilde{\boldsymbol{I}}_1^o$ through a 3×3 conv layer. We use $\boldsymbol{F}_k^b$ to represent the final feature map of the k-th stage in the bottom branch, so $\boldsymbol{F}_1^b = \tilde{\boldsymbol{F}}_1^t$ in the first stage. For the rest of stages, we adopt two fusion blocks to fuse the final feature map in previous stage and the output from the upsample decoder. Specifically, for the k-th stage ($k > 1$), we concatenate $\boldsymbol{F}_{k-1}^b$ with $\tilde{\boldsymbol{F}}_k^t$, which passes through two fusion blocks to arrive at $\boldsymbol{F}_k^b$. $\boldsymbol{F}_k^b$ is used to produce the harmonized image $\tilde{\boldsymbol{I}}_k^o$ through a 3×3 conv layer. The details of conv/upsample block in the upsample decoder and fusion block in the bottom branch can be found in Section 4.1.

The k-th stage in the bottom branch integrates the information of the harmonized feature maps up to the k-th stage in the top branch, that is, $\{\hat{\boldsymbol{F}}_1^t, \ldots, \hat{\boldsymbol{F}}_k^t\}$. Accordingly, we enforce the output image $\tilde{\boldsymbol{I}}_k^o$ in the k-th stage to be harmonized up to the k-th style level, so the style loss (Huang and Belongie 2017) for $\tilde{\boldsymbol{I}}_k^o$ can be written as

$$\mathcal{L}_k^{sty} = \sum_{k'=1}^{k} \|\mu\left(\phi_{k'}(\tilde{\boldsymbol{I}}_k^o) \circ \boldsymbol{M}^f\right) - \mu(\hat{\boldsymbol{F}}_{k'}^{t,f})\|^2$$

$$+\sum_{k'=1}^{k} \|\sigma\left(\phi_{k'}(\tilde{\boldsymbol{I}}_k^o)\circ \boldsymbol{M}^f\right) - \sigma(\hat{\boldsymbol{F}}_{k'}^{t,f})\|^2, \tag{2}$$

where each $\phi_{k'}(\cdot), k' \in \{1,2,3,4\}$ denotes the output feature map from the k'-th VGG-19 encoder stage, and $\circ$ means element-wise product. $\mu(\cdot)$ and $\sigma(\cdot)$ are the same as defined in Eqn. 1.

Additionally, for the harmonized images of all stages, we utilize the same content loss (Gatys, Ecker, and Bethge 2016) to ensure that the foreground content is maintained:

$$\mathcal{L}_k^{con} = \left\|\phi_4(\tilde{\boldsymbol{I}}_k^o) - \phi_4(\boldsymbol{I}^c)\right\|^2, \tag{3}$$

in which $\phi_4(\cdot)$ is the same as defined in Eqn. 2. For the k-th stage, the total loss is given by $\mathcal{L}_k^{total} = \mathcal{L}_k^{con} + \mathcal{L}_k^{sty}$.

From the first stage to the last stage, the bottom branch gradually harmonizes the composite foreground from low-level styles to high-level styles. Specifically, we first adjust the low-level styles (*e.g.*, color, simple pattern) of foreground and then adjust the high-level styles (*e.g.*, complex pattern) of foreground. As low-level styles could be easily transferred by only using shallow features, low-level style transfer is a simpler task than high-level style transfer. Therefore, the bottom branch transits from easy task to hard task, which might be easier than handling all tasks simultaneously, as discussed in the realm of curriculum learning (Bengio et al. 2009). Moreover, during the transition from easy task to hard task, we can terminate at the proper stage to prevent the adverse effect brought by challenging the hard task, which will be introduced in Section 3.2.

3.2 Early-exit Strategy

Based on the harmonized results from different stages, we observe that in some cases, the harmonized results of early stages are comparable or even better than those of late stages. One reason is that it is sufficient to only transfer low-level styles for certain composite images. Another reason is that the network is sometimes struggling to transfer high-level styles at the cost of sacrificing the content information, so that the content structure of harmonized image is severely distorted. Motivated by the above observations, we design an early-exit strategy, which enables the network to terminate at the proper stage. Some previous works (Huang et al. 2018; Teerapittayanon, McDanel, and Kung 2016; Bolukbasi et al. 2017; McGill and Perona 2017; Jie et al. 2019; Li et al. 2019a; Figurnov et al. 2017; Leroux et al. 2018) on dynamic routing explored exiting the network early. We apply similar idea to our progressive harmonization network, but the motivation and technical approach are considerably different from theirs.

To provide effective supervision for when to exit, we manually annotate the exit stages for 5000 randomly selected composite images in the training set (see details in the supplementary), in which the exit stage means the earliest stage that could produce satisfactory result. Given a composite image with annotated exit stage $\hat{k}$, the exit labels of stages prior to the exit stage are 0, that is, $y_{k'} = 0$ for $k' < \hat{k}$, otherwise $y_{k'} = 1$.

We adopt sequential model to predict the exit label for each stage sequentially. For simplicity, we choose GRU (Cho et al. 2014) as the sequential model. At the k-th stage in the bottom branch, we perform global average pooling over the final feature map $\boldsymbol{F}_k^b$ and produce $\boldsymbol{f}_k^b$. Then, we send $\boldsymbol{f}_k^b$ and the hidden state $\boldsymbol{h}_{k-1}$ of GRU at the $(k-1)$-th stage to the GRU cell. Formally, the function of GRU cell at the k-th stage can be represented by

$$[\tilde{y}_k, \boldsymbol{h}_k] = GRU(\boldsymbol{h}_{k-1}, \boldsymbol{f}_k^b). \tag{4}$$

In practice, we only add GRU cells to the first three stages and employ the binary cross-entropy loss $\mathcal{L}_k^{bce}(y_k, \tilde{y}_k)$ for the k-th stage. Thus, the total loss function can be written as

$$\mathcal{L}^{all} = \sum_{k=1}^{4} \mathcal{L}_k^{total} + \sum_{k=1}^{3} \mathcal{L}_k^{bce}. \tag{5}$$

During inference, we choose the earliest stage with predicted exit score larger than the threshold 0.5 as the exit stage. If the predicted exit scores of the first three stages are all smaller than the threshold, we proceed till the last stage.

4 Experiments

4.1 Datasets and Implementation Details

Following previous works (Peng, Wang, and Wang 2019; Cao, Hong, and Niu 2023), we conduct experiments on COCO (Lin et al. 2014) and WikiArt (Nichol 2016). COCO (Lin et al. 2014) contains instance segmentation annotations for 80 object categories, while WikiArt (Nichol 2016) contains digital artistic paintings from different styles. We create composite images based on these two datasets, with the photographic objects from COCO and the painterly backgrounds from WikiArt. In particular, we randomly select a photographic object from COCO with the foreground ratio in $[0.05, 0.3]$. Then, we paste this photographic object on a randomly selected painterly background from WikiArt, resulting in an inharmonious composite image.

Our model is implemented by PyTorch 1.10.0, which is distributed on ubuntu 20.04 LTS operation system, with 128GB memory, Intel(R) Xeon(R) Silver 4116 CPU, and one GeForce RTX 3090 GPU. We adopt Adam (Kingma and Ba 2015) with learning rate of 0.0001 as the optimization solver. We resize the input images as 256×256 and set the batch size as 4 for model training. Each upsample decoder consists of one conv block and several upsample blocks. The conv block has a 3×3 conv followed by ReLU. Each upsample block has an upsampling layer and a 3×3 conv followed by ReLU. The conv block and upsample blocks all reduce the channel dimension by half. In the bottom branch, each fusion block has a sequence of 3×3 conv, ReLU, 3×3 conv, ReLU. The first 3×3 conv in the first fusion block in each stage reduces the channel dimension by half.

4.2 Multiple Stages and Exit Stages

We annotate the exit stages for 5000 randomly selected composite images in the training set. For each composite image, we produce the results of four stages and ask human annotators to annotate its exit stage (see details in the Supplementary). The percentages that four stages are annotated as

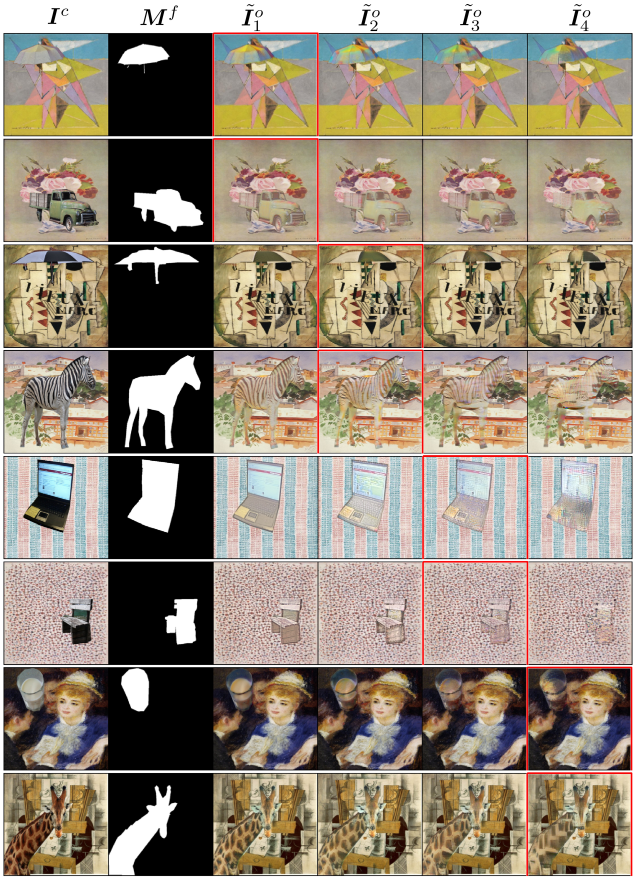

Figure 3: From left to right, we show the composite image, composite mask, the harmonized results from four stages in the test set. The predicted exit stages are marked with red bounding boxes.

the exit stage are 2.1%, 22.3%, 47.4%, and 28.2% respectively. Overall, the outputs from late stages (*e.g.*, 3, 4) are better than those from early stages (*e.g.*, 1, 2), because the former ones are prone to have incompatible foreground and background *w.r.t.* high-level styles. Some annotation examples are provided in the Supplementary.

We train GRU based on the annotated training images, which is then applied to the test set. Given a composite image in the test set, its exit stage is the earliest stage with the predicted exit score larger than the threshold 0.5. If the predicted exit scores in the first three stages are all smaller than 0.5, the last stage is deemed as the exit stage.

We show the harmonized results from four stages and the predicted exit stages in Figure 3. It can be seen that the outputs from early stages are harmonized up to low style level (*e.g.*, color, simple pattern) while the outputs from late stages are harmonized up to high style level (*e.g.*, complex pattern). For example, in row 7-8, the outputs from the last stage have better visualization effect with more compatible foreground and background, *e.g.*, fluffy glass in row 7 and the giraffe with square spots in row 8.

However, in some cases, the early stages are predicted as the exit stages. As explained in Section 3.2, one reason is that it is sufficient to only transfer low-level styles for certain composite images. For example, in row 1-2, the outputs from the first stage have been sufficiently harmonized, probably because the background style is simple or the foreground naturally matches the background. Hence, there is no need to go through the rest of stages. Another reason is that the network is sometimes struggling to transfer high-level styles at the cost of sacrificing the content information, so that the content structure of harmonized image is distorted. For example, in row 3-6, the results from the last stage have undesired artifacts (the uneven black spots on the umbrella in row 3) or distorted content structure (the corrupted zebra strips in row 4, the distorted laptop/chair structure in row 5, 6), so we need to exit the network before the last stage, to balance between stylization and content preservation.

Method	B-T score	Time(s)	FLOPs(G)
SANet	-0.429	0.0097	43.32
AdaAttN	-0.626	0.0115	49.64
StyTr2	0.225	0.0504	39.74
QuantArt	0.336	0.1031	133.34
Inst	-0.762	2.2996	3378.43
SDEdit	-0.654	2.1321	3164.52
CDC	0.204	2.3427	3299.81
E2STN	-0.188	0.0079	29.28
DPH	0.311	55.24	-
PHDNet	0.474	0.0321	158.41
ProPIH	0.917	0.0067	23.45

Table 1: The comparison between different methods.

4.3 Comparison with Baselines

We compare with two groups of baselines: artistic style transfer methods and painterly image harmonization methods. Since the standard image harmonization methods introduced in Section 2.1 only adjust the illumination statistics and require ground-truth images as supervision, they are not suitable for painterly image harmonization.

For the first group, we first use artistic style transfer methods to stylize the entire photographic image according to the painterly background, and then paste the stylized photographic object on the painterly background. We compare with typical or recent style transfer methods: SANet (Park and Lee 2019), AdaAttN (Liu et al. 2021), StyTr2 (Deng et al. 2022), QuantArt (Huang et al. 2023), INST (Zhang et al. 2023). For the second group, we compare with SDEdit (Meng et al. 2021), CDC (Hachnochi et al. 2023), E2STN (Peng, Wang, and Wang 2019), DPH (Luan et al. 2018), PHDNet (Cao, Hong, and Niu 2023).

For our method, we use the harmonized images from the predicted exit stage for comparison.

Visualization Results: The comparison with the first group of baselines is illustrated in Figure 4. It can be seen that the baselines often fail to harmonize the composite foreground sufficiently, especially when the background has complicated texture (*e.g.*, row 2). In contrast, our method

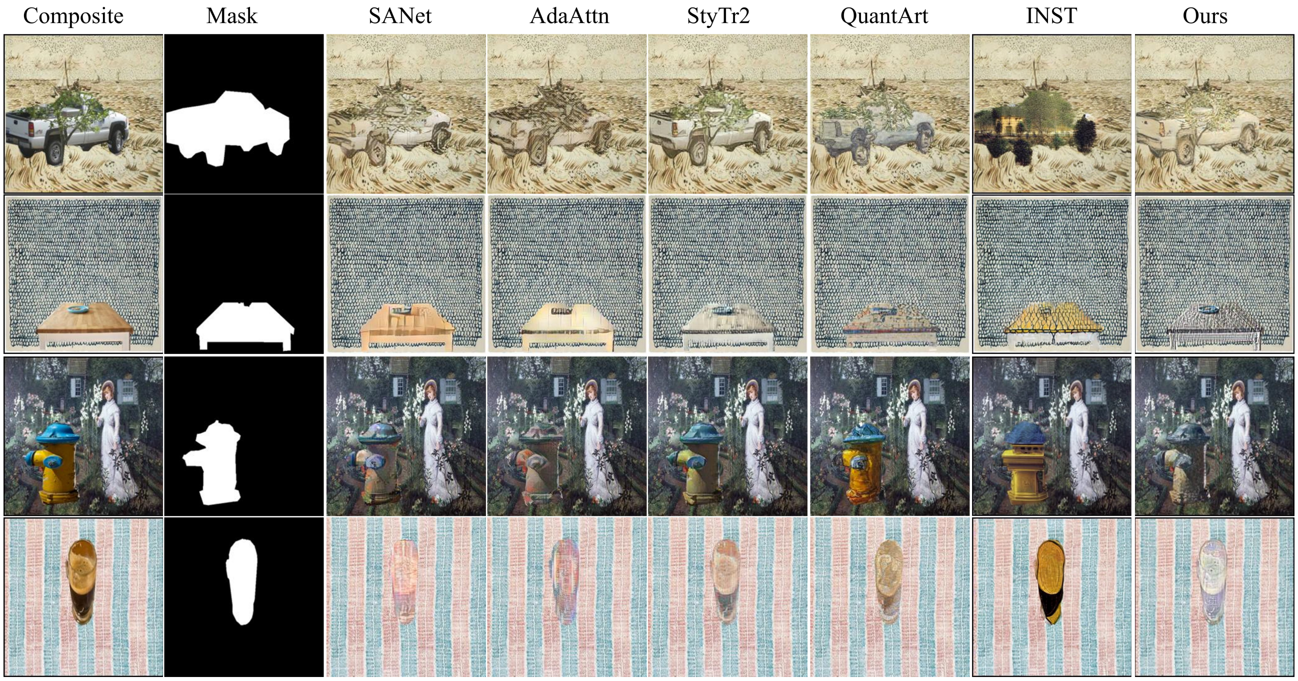

Figure 4: From left to right, we show the composite image, composite foreground mask, the harmonized results of SANet (Park and Lee 2019), AdaAttN (Liu et al. 2021), StyTr2 (Deng et al. 2022), QuantArt (Huang et al. 2023), INST (Zhang et al. 2023), and our ProPIH.

could handle diversified background styles and produce visually appealing harmonized images.

The comparison with the second group of baselines is illustrated in Figure 5. Again, our method can produce the harmonized foreground, which has more compatible color and texture with the background. In row 1, the harmonized foregrounds of baselines have inharmonious color or distorted content structure. In row 2, the harmonized foregrounds of baselines have severe artifacts, while our method can roughly maintain the zebra strips after harmonization. In row 3 and row 4, our method can achieve a good balance between content preservation and style migration, while baselines either fail in style migration or severely distort the content structure.

User Study: Following (Cao, Hong, and Niu 2023), we also conduct user study to compare different methods. We randomly select 100 content images from COCO and 100 style images from WikiArt to generate 100 composite images for user study. We compare the harmonized results generated by 10 baselines and our method.

Specifically, for each composite image, we obtain 11 harmonized outputs and use every two images from these 11 images to construct image pairs. With 100 composite images, we construct 5500 image pairs. Then, we invite 100 participants to watch one image pair each time and choose the better one. Finally, we collect $550,000$ pairwise results and calculate the overall ranking of all methods using Bradley-Terry (B-T) model (Bradley and Terry 1952; Lai et al. 2016). The results are summarized in Table 1, in which our method achieves the highest B-T score and once again outperforms other baselines.

Efficiency Analyses: For efficiency comparison, we report the inference time and FLOPs. We evaluate with the image size 256×256 and the inference time is averaged over 100 test images on a single RTX 3090 GPU. Note that DPH is optimization-based method, so we omit its FLOPs. For our method, different test images may exit the network from different stages, we count the time cost and computational cost up to the predicted exit stage for each test image.

The results are reported in Table 1. The optimization-based method DPH is time-consuming, due to iteratively updating the input composite image. Diffusion-based methods are much slower than the other feed-forward methods, which limits their real-world applicability. Our method is relatively efficient, because of our lightweight network structure and early-exit strategy.

4.4 Ablation Studies

Firstly, we try using the full style loss (*i.e.*, $\mathcal{L}_4^{sty}$ in Eqn. 2) for all four stages and the harmonized results from four stages are shown in the top row in Figure 6. We also show the harmonized results of our default method for comparison in the bottom row. In the top row, even using the full style loss, the outputs ($\tilde{\boldsymbol{I}}_1^o$, $\tilde{\boldsymbol{I}}_2^o$) from early stages can still only be harmonized up to the low style level. The content structures of outputs ($\tilde{\boldsymbol{I}}_3^o$) may even be destroyed when using low-level features for high-level harmonization. The results demonstrate that low-level harmonization is simpler than high-level harmonization and high-level features are manda-

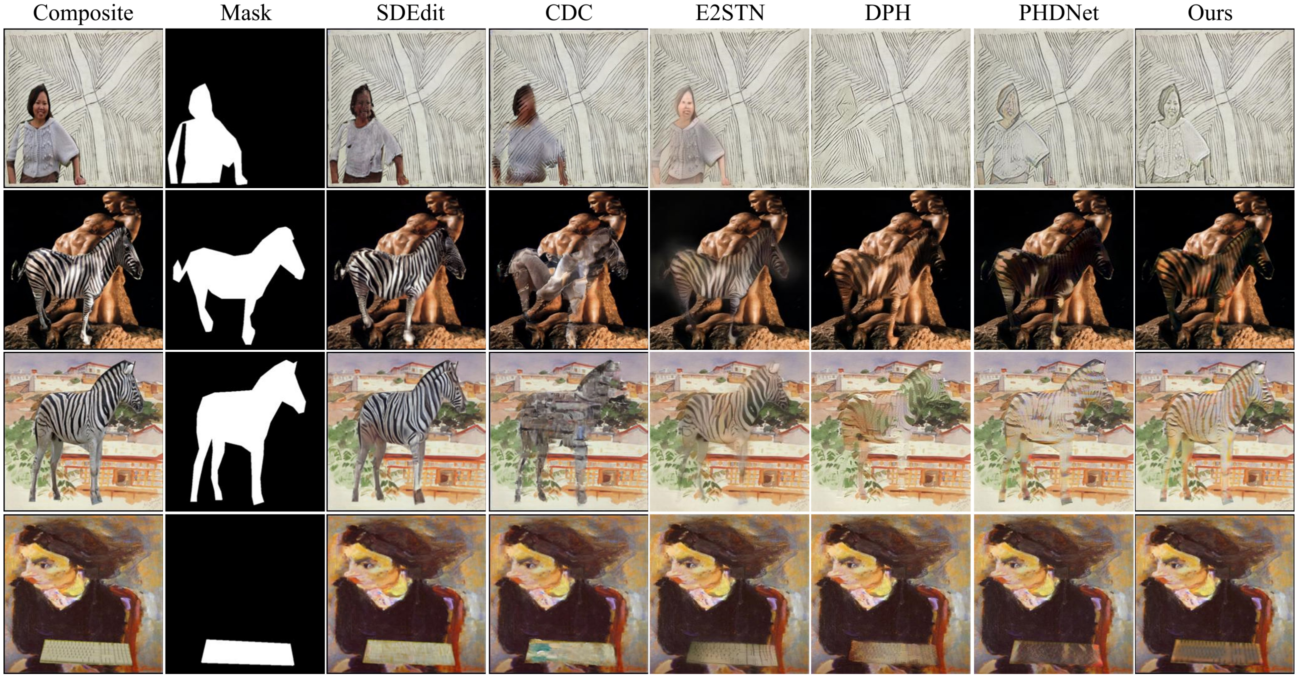

Figure 5: From left to right, we show the composite image, composite foreground mask, the harmonized results of SDEdit (Meng et al. 2021), CDC (Hachnochi et al. 2023), E2STN (Peng, Wang, and Wang 2019), DPH (Luan et al. 2018), PHDNet (Cao, Hong, and Niu 2023), and our ProPIH.

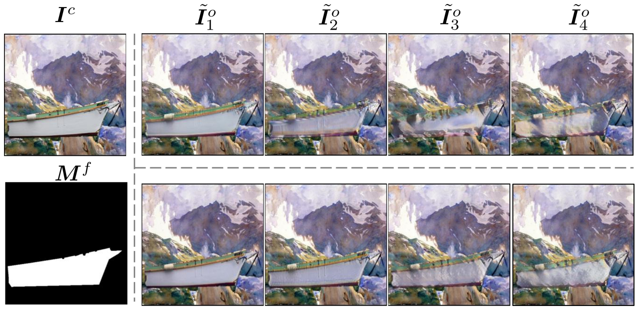
$\boldsymbol{I}^c$
$\tilde{\boldsymbol{I}}^o_1$
$\tilde{\boldsymbol{I}}^o_2$
$\tilde{\boldsymbol{I}}^o_3$
$\tilde{\boldsymbol{I}}^o_4$
$\boldsymbol{M}^f$

Figure 6: The top row shows the harmonized results from four stages when using the full style loss for all stages. The bottom row shows the results of our default method for comparison.

tory for high-level harmonization.

Secondly, we remove the losses for the first three stages and only use the loss for the final stage. As there is no supervision for the first three stages, we adopt the output from the last stage, which is referred to as V_1. We also try some variants of our network structure. We replace the upsample decoder with bilinear upsampling, which is referred to as V_2. We also reduce the number of fusion blocks from two to one in each stage, which is referred to as V_3. For V_2 and V_3, we adopt the output from the predicted exit stage. The visualization results of V_1, V_2, and V_3 are shown in Figure 7. The harmonized results of V_1 become worse than our full method, because the losses in the first three stages can provide useful intermediate supervision and help learn more informative intermediate features. The harmonized results of V_2 and V_3 are also worse than our method, because the simplified network structure reduces the model capacity and degrades the harmonization ability. The above results verify the necessity of each component in our network.

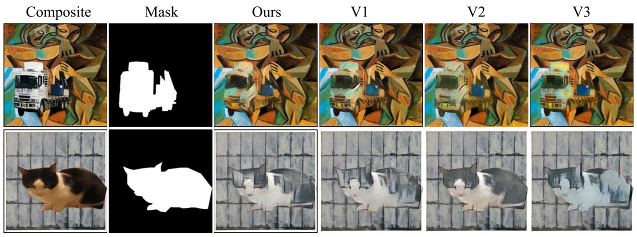

Figure 7: From left to right, we show the composite image, the composite mask, the harmonized results of our method and three ablated versions.

5 Conclusion

In this work, we have developed a progressive harmonization network which can harmonize the composite foreground from low-level styles to high-level styles progressively. We have also designed early-exit strategy, which enables the network to exit at the proper stage. Extensive experiments on the benchmark dataset have demonstrated the superiority of our progressive harmonization network.

Acknowledgments

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

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *ICML*.

Bolukbasi, T.; Wang, J.; Dekel, O.; and Saligrama, V. 2017. Adaptive neural networks for efficient inference. In *ICML*.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Cao, J.; Hong, Y.; and Niu, L. 2023. Painterly Image Harmonization in Dual Domains. *AAAI*.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *CVPR*.

Chen, H.; Zhao, L.; Wang, Z.; Zhang, H.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021. Dualast: Dual style-learning networks for artistic style transfer. In *CVPR*.

Chen, T. Q.; and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*.

Chen, Z.; Wang, W.; Xie, E.; Lu, T.; and Luo, P. 2022. Towards Ultra-Resolution Neural Style Transfer via Thumbnail Instance Normalization. In *AAAI*.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, W.; Choi, S.; Park, D. K.; Shin, I.; and Choo, J. 2019. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *CVPR*.

Cong, W.; Tao, X.; Niu, L.; Liang, J.; Gao, X.; Sun, Q.; and Zhang, L. 2022. High-Resolution Image Harmonization via Collaborative Dual Transformations. In *CVPR*.

Cong, W.; Zhang, J.; Niu, L.; Liu, L.; Ling, Z.; Li, W.; and Zhang, L. 2020. Dovenet: Deep image harmonization via domain verification. In *CVPR*.

Cun, X.; and Pun, C. 2020. Improving the Harmony of the Composite Image by Spatial-Separated Attention Module. *IEEE Trans. Image Process.*, 29: 4759–4771.

Deng, Y.; Tang, F.; Dong, W.; Ma, C.; Pan, X.; Wang, L.; and Xu, C. 2022. StyTr2: Image Style Transfer with Transformers. In *CVPR*.

Figurnov, M.; Collins, M. D.; Zhu, Y.; Zhang, L.; Huang, J.; Vetrov, D.; and Salakhutdinov, R. 2017. Spatially adaptive computation time for residual networks. In *CVPR*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *CVPR*.

Gu, S.; Chen, C.; Liao, J.; and Yuan, L. 2018. Arbitrary style transfer with deep feature reshuffle. In *CVPR*.

Guerreiro, J. J. A.; Nakazawa, M.; and Stenger, B. 2023. PCT-Net: Full Resolution Image Harmonization Using Pixel-Wise Color Transformations. In *CVPR*.

Guo, Z.; Gu, Z.; Zheng, B.; Dong, J.; and Zheng, H. 2022. Transformer for Image Harmonization and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Guo, Z.; Guo, D.; Zheng, H.; Gu, Z.; Zheng, B.; and Dong, J. 2021a. Image harmonization with transformer. In *ICCV*.

Guo, Z.; Zheng, H.; Jiang, Y.; Gu, Z.; and Zheng, B. 2021b. Intrinsic image harmonization. In *CVPR*.

Hachnochi, R.; Zhao, M.; Orzech, N.; Gal, R.; Mahdavi-Amiri, A.; Cohen-Or, D.; and Bermano, A. H. 2023. Cross-domain Compositing with Pretrained Diffusion Models. *arXiv preprint arXiv:2302.10167*.

Hang, Y.; Xia, B.; Yang, W.; and Liao, Q. 2022. SCS-Co: Self-Consistent Style Contrastive Learning for Image Harmonization. In *CVPR*.

Hao, G.; Iizuka, S.; and Fukui, K. 2020. Image Harmonization with Attention-based Deep Feature Modulation. In *BMVC*.

Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; and Weinberger, K. 2018. Multi-Scale Dense Networks for Resource Efficient Image Classification. In *ICLR*.

Huang, S.; An, J.; Wei, D.; Luo, J.; and Pfister, H. 2023. QuantArt: Quantizing Image Style Transfer Towards High Visual Fidelity. In *CVPR*.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.

Huo, J.; Jin, S.; Li, W.; Wu, J.; Lai, Y.-K.; Shi, Y.; and Gao, Y. 2021. Manifold alignment for semantically aligned style transfer. In *ICCV*.

Jiang, Y.; Zhang, H.; Zhang, J.; Wang, Y.; Lin, Z.; Sunkavalli, K.; Chen, S.; Amirghodsi, S.; Kong, S.; and Wang, Z. 2021. SSH: A Self-Supervised Framework for Image Harmonization. In *ICCV*.

Jie, Z.; Sun, P.; Li, X.; Feng, J.; and Liu, W. 2019. Anytime recognition with routing convolutional networks. *PAMI*, 43(6): 1875–1886.

Jing, Y.; Liu, X.; Ding, Y.; Wang, X.; Ding, E.; Song, M.; and Wen, S. 2020. Dynamic instance normalization for arbitrary style transfer. In *AAAI*.

Ke, Z.; Sun, C.; Zhu, L.; Xu, K.; and Lau, R. W. 2022. Harmonizer: Learning to Perform White-Box Image and Video Harmonization. In *ECCV*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kolkin, N.; Salavon, J.; and Shakhnarovich, G. 2019. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*.

Lai, W.-S.; Huang, J.-B.; Hu, Z.; Ahuja, N.; and Yang, M.-H. 2016. A comparative study for single image blind deblurring. In *CVPR*.

Leroux, S.; Simoens, P.; Dhoedt, B.; Molchanov, P.; Breuel, T.; and Kautz, J. 2018. IamNN: iterative and adaptive mobile neural network for efficient image classification. In *ICLR*.

Li, H.; Zhang, H.; Qi, X.; Yang, R.; and Huang, G. 2019a. Improved techniques for training adaptive deep networks. In *ICCV*.

Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2018. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv:1808.04537*.

Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2019b. Learning linear transformations for fast image and video style transfer. In *CVPR*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. *NeurIPS*.

Liang, J.; Cun, X.; and Pun, C.-M. 2022. Spatial-Separated Curve Rendering Network for Efficient and High-Resolution Image Harmonization. In *ECCV*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Ling, J.; Xue, H.; Song, L.; Xie, R.; and Gu, X. 2021. Region-aware adaptive instance normalization for image harmonization. In *CVPR*.

Liu, S.; Huynh, C. P.; Chen, C.; Arap, M.; and Hamid, R. 2023. LEMaRT: Label-Efficient Masked Region Transform for Image Harmonization. In *CVPR*.

Liu, S.; Lin, T.; He, D.; Li, F.; Wang, M.; Li, X.; Sun, Z.; Li, Q.; and Ding, E. 2021. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *ICCV*.

Lu, S.; Liu, Y.; and Kong, A. W. 2023. TF-ICON: Diffusion-Based Training-Free Cross-Domain Image Composition. In *ICCV*.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2018. Deep painterly harmonization. In *CGF*.

Luo, X.; Han, Z.; and Yang, L. 2022. Progressive Attentional Manifold Alignment for Arbitrary Style Transfer. In *ACCV*.

McGill, M.; and Perona, P. 2017. Deciding how to decide: Dynamic routing in artificial neural networks. In *ICML*.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*.

Nichol, K. 2016. Painter by numbers. https://www.kaggle.com/competitions/painter-by-numbers/data.

Park, D. Y.; and Lee, K. H. 2019. Arbitrary style transfer with style-attentional networks. In *CVPR*.

Peng, H.-J.; Wang, C.-M.; and Wang, Y.-C. F. 2019. Element-Embedded Style Transfer Networks for Style Harmonization. In *BMVC*.

Peng, J.; Luo, Z.; Liu, L.; Zhang, B.; Wang, T.; Wang, Y.; Tai, Y.; Wang, C.; and Lin, W. 2022. FRIH: Fine-grained Region-aware Image Harmonization. *arXiv preprint arXiv:2205.06448*.

Sanakoyeu, A.; Kotovenko, D.; Lang, S.; and Ommer, B. 2018. A style-aware content loss for real-time hd style transfer. In *ECCV*.

Sheng, L.; Lin, Z.; Shao, J.; and Wang, X. 2018. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

Sofiiuk, K.; Popenova, P.; and Konushin, A. 2021. Foreground-aware semantic representations for image harmonization. In *WACV*.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *ICPR*.

Tsai, Y.-H.; Shen, X.; Lin, Z.; Sunkavalli, K.; Lu, X.; and Yang, M.-H. 2017. Deep image harmonization. In *CVPR*.

Valanarasu, J. M. J.; Zhang, H.; Zhang, J.; Wang, Y.; Lin, Z.; Echevarria, J.; Ma, Y.; Wei, Z.; Sunkavalli, K.; and Patel, V. M. 2023. Interactive portrait harmonization. *ICLR*.

Wang, H.; Li, Y.; Wang, Y.; Hu, H.; and Yang, M.-H. 2020. Collaborative distillation for ultra-resolution universal style transfer. In *CVPR*.

Wang, K.; Gharbi, M.; Zhang, H.; Xia, Z.; and Shechtman, E. 2023. Semi-supervised Parametric Real-world Image Harmonization. In *CVPR*.

Wang, X.; Oxholm, G.; Zhang, D.; and Wang, Y.-F. 2017. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *CVPR*.

Xing, Y.; Li, Y.; Wang, X.; Zhu, Y.; and Chen, Q. 2022. Composite photograph harmonization with complete background cues. In *ACM MM*.

Xue, B.; Ran, S.; Chen, Q.; Jia, R.; Zhao, B.; and Tang, X. 2022. DCCF: Deep Comprehensible Color Filter Learning Framework for High-Resolution Image Harmonization. In *ECCV*.

Yan, X.; Lu, Y.; Shuai, J.; and Zhang, S. 2022. Style Image Harmonization via Global-Local Style Mutual Guided. In *ACCV*.

Yao, Y.; Ren, J.; Xie, X.; Liu, W.; Liu, Y.-J.; and Wang, J. 2019. Attention-aware multi-stroke style transfer. In *CVPR*.

Zhang, L.; Wen, T.; and Shi, J. 2020. Deep image blending. In *WACV*.

Zhang, Y.; Fang, C.; Wang, Y.; Wang, Z.; Lin, Z.; Fu, Y.; and Yang, J. 2019. Multimodal style transfer via graph cuts. In *ICCV*.

Zhang, Y.; Huang, N.; Tang, F.; Huang, H.; Ma, C.; Dong, W.; and Xu, C. 2023. Inversion-Based Creativity Transfer with Diffusion Models. In *CVPR*.

Zhu, Z.; Zhang, Z.; Lin, Z.; Wu, R.; Chai, Z.; and Guo, C.-L. 2022. Image Harmonization by Matching Regional References. *arXiv preprint arXiv:2204.04715*.