Deep Image Harmonization in Dual Color Spaces

Linfeng Tan
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
China
tanlinfeng@sjtu.edu.cn

Jiangtong Li
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
China
keep_moving-Lee@sjtu.edu.cn

Li Niu*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
China
ustcnewly@sjtu.edu.cn

Liqing Zhang*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
China
zhang-lq@cs.sjtu.edu.cn

ABSTRACT

Image harmonization is an essential step in image composition that adjusts the appearance of composite foreground to address the inconsistency between foreground and background. Existing methods primarily operate in correlated *RGB* color space, leading to entangled features and limited representation ability. In contrast, decorrelated color space (*e.g.*, *Lab*) has decorrelated channels that provide disentangled color and illumination statistics. In this paper, we explore image harmonization in dual color spaces, which supplements entangled *RGB* features with disentangled *L*, *a*, *b* features to alleviate the workload in harmonization process. The network comprises a *RGB* harmonization backbone, an *Lab* encoding module, and an *Lab* control module. The backbone is a U-Net network translating composite image to harmonized image. Three encoders in *Lab* encoding module extract three control codes independently from *L*, *a*, *b* channels, which are used to manipulate the decoder features in harmonization backbone via *Lab* control module. Our code and model are available at https://github.com/bcmi/DucoNet-Image-Harmonization.

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; **Computer vision**.

KEYWORDS

image harmonization,decorelated color space,image composition

ACM Reference Format:
Linfeng Tan, Jiangtong Li, Li Niu, and Liqing Zhang. 2023. Deep Image Harmonization in Dual Color Spaces. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612404

*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '23, October 29–November 3, 2023, Ottawa, ON, Canada
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612404

1 INTRODUCTION

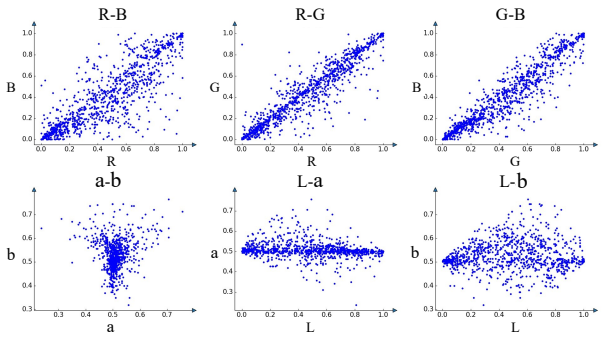

Figure 1: We randomly sample 1000 pixels from 100 real images in iHarmony4 [9] and plot the correlation between every two channels in *RGB* (*resp.*, *Lab*) color space in the top (*resp.*, bottom) row. It can be seen that *RGB* channels have strong positive correlations, while *Lab* channels are decorrelated.

Image composition [29] targets at generating a composite image by merging foreground and background. Nevertheless, the foreground and background in the obtained composite image might have appearance discrepancy, which is caused by different lighting, climate, and capture devices between foreground and background. To tackle this challenge, image harmonization [8, 9, 35, 38, 44] modifies the foreground appearance to ensure its compatibility with the background. Early traditional image harmonization methods [22, 36, 38, 44] are often designed based on low-level color and illumination statistics. However, with the rapid advance of deep learning techniques, deep image harmonization methods [8, 9, 20, 35] have become dominant and achieved impressive results.

Existing deep image harmonization methods have been developed from different aspects (*e.g.*, attention mechanism, domain/style transfer, Retinex theory, color transfer) to address the appearance mismatch between foreground and background. In detail, some

works [10, 16] explored attention mechanism to adjust the foreground features more effectively. Besides, some works [7, 9] approached image harmonization as the translation from foreground domain to background domain with additional loss to guide the domain transfer. Moreover, some works [13, 14] introduced Retinex [23] theory to image harmonization tasks by decoupling an image into reflectance and illumination. Recently, some works [8, 20] considered the balance between effectiveness and efficiency, and solved image harmonization in the form of color transfer. Despite the success achieved by existing methods, they mainly operate in *RGB* color space to extract and adjust features. However, *RGB* color space is a correlated color space and the entangled *RGB* features may increase the workload of existing harmonization methods.

As known to all, an image can be represented in various color spaces, such as *RGB*, *XYZ*, or *Lab*. These color spaces can be categorized into two groups: correlated color spaces and decorrelated color spaces. In correlated color spaces (*e.g.*, *RGB*, *XYZ*), different channels are strongly correlated and tend to change simultaneously. In contrast, in decorrelated color spaces (*e.g.*, *YUV*, *Lab*), different channels are decorrelated. By taking *Lab* as an example decorrelated color space, *L* represents lightness, *a* represents the spectrum from green to red, and *b* represents the spectrum from blue to yellow. In Figure 1, we plot the correlation between every two channels in *RGB* (*resp.*, *Lab*) color spaces in the top (*resp.*, bottom) row. It can be observed that *RG*, *RB*, and *GB* in *RGB* color space exhibit strong positive correlations, while *La*, *Lb*, and *ab* in *Lab* color space are decorrelated. Considering the correlation within the *RGB* color space, the extracted *RGB* features may not effectively disentangle the independent factors of color and illumination statistics, which potentially complicates the harmonization process [9, 10, 27, 35]. However, the decorrelated *Lab* color space contains decorrelated factors (*i.e.*, lightness, orthogonal colors) in three channels, serving as a valuable complement to the entangled features extracted from *RGB* color space. Moreover, recent studies [26, 41] on inharmonious region localization have revealed that the decorrelated color space can help identify the inharmonious region, which also motivates us to explore image harmonization in the decorrelated color space.

Our primary insight for image harmonization is to alleviate the workload of harmonization process by supplementing the entangled *RGB* features with the disentangled *L*, *a*, *b* features. To this end, we propose a novel image harmonization network in **Du**al **Co**lor Spaces (**DucoNet**). Our DucoNet comprises a *RGB* harmonization backbone, an *Lab* encoding module, and an *Lab* control module. The harmonization backbone is a U-Net network responsible for harmonizing the input composite image in the *RGB* color space. In detail, the backbone takes in the *RGB* channels and the foreground mask, producing the *RGB* channels of the harmonized image. The *Lab* encoding module consists of three encoders to extract the *L*, *a*, *b* control codes from *L*, *a*, *b* channels of the composite image independently. The *Lab* control module interacts with the harmonization backbone to adjust the decoder features with *L*, *a*, *b* control codes. Each control code adjusts the decoder features in multiple decoder layers of the harmonization backbone. Specifically, each control code is used to generate dynamic convolution kernels [19], which are applied to the foreground region in the decoder feature maps. The decoder feature maps manipulated using three control codes are fused to produce the harmonized image. Considering that *L*, *a*, *b* channels may contribute differently to various images or even various pixels, we tend to learn pixel-wise weights for three channels when fusing the decoder feature maps manipulated using three control codes, which could also provide hints for the contributions of *L*, *a*, *b* channels when harmonizing a specific image.

The effectiveness of our DucoNet is verified through extensive experiments of low/high-resolution harmonization on the benchmark dataset iHarmony4 [9] and real composite images. Our contribution can be summarized as follows: 1) To the best of our knowledge, we are the first to investigate image harmonization in both correlated and decorrelated color spaces. 2) We propose a novel image harmonization network in Dual Color Spaces (DucoNet) with *Lab* encoding module and control module, which supplements entangled *RGB* features with disentangled *L*, *a*, *b* features. 3) Extensive experiments on the benchmark dataset demonstrate that our DucoNet outperforms the state-of-the-art approaches by a large margin.

2 RELATED WORK

2.1 Image Harmonization

As a subtask in image composition [29], image harmonization aims to create a harmonious composite image by ensuring that the appearances of foreground and background are consistent. In the early stage, traditional image harmonization methods [22] focused on adjusting the low-level illumination and color statistics of foreground to match the background.

In recent years, deep learning based harmonization methods have brought significant advance to this research field. Unsupervised image harmonization methods [47] were initially explored using adversarial learning. With the introduction of the first large-scale image harmonization dataset iHarmony4 [9], supervised image harmonization methods [1–5, 15, 18, 25, 30, 33, 42, 48] have received increasing attention. Among them, some works [10, 16, 35] designed attention modules to extract background features and adjust the foreground features through channel-wise adjustment [10], semantic representation [35, 39], and modulation-demodulation [16]. Additionally, some works [7, 9, 27] formulated image harmonization as domain/style translation, and employed adversarial learning [9], region-aware AdaIn [27], and contrastive loss [7] to transfer the foreground into the background domain/style. Moreover, some works [12–14] introduced Retinex [23] theory to image harmonization by decomposing the harmonization task into reflectance maintenance and illumination adjustment. Recently, some works [8, 20] treated image harmonization as color-to-color transformation [8] or image-level regression [20], striking a good balance between effectiveness and efficiency in high-resolution image harmonization.

Existing methods mainly rely on the correlated *RGB* space to extract the background features and adjust the foreground features. However, the entangled *RGB* features may increase the workload of harmonization network and impede the harmonization performance. Our work focuses on dual color spaces (*i.e.*, *RGB* and *Lab*), by using the decorrelated *Lab* color space to generate *L*, *a*, and *b* control codes for feature manipulation in harmonization backbone.

2.2 Color Spaces

There are multiple color spaces to represent images, such as *RGB*, *Lab*, *XYZ*, which can be divided into correlated and decorrelated

color spaces based on whether each color channel correlates with each other. The correlated color space can be directly shown in different monitors and reflect the basic physics rules, for example, *RGB* represents three primary colors of light. However, the correlations among different color channels may prevent the critical factors to be encoded independently and complicate the color transformation [32]. On the contrary, the decorrelated color space usually disentangles some critical factors (*i.e.*, lightness), which may help extract the corresponding features independently. Most works in computer vision field predominantly use *RGB* color space. Nevertheless, some works also utilize multiple color spaces [24, 31] to achieve the desired effect.

For example, in underwater image enhancement [24, 28, 31, 46], it is important to incorporate multiple color spaces to enhance model capabilities. Among them, Peng *et al.*[31] integrated *RGB*, *Lab*, and *LCH* color spaces into a loss function to improve the contrast and saturation of the enhanced image. Li *et al.* [24] proposed a multi-color encoder to enrich the diversity of feature representations by incorporating the characteristics of *RGB*, *HSV*, and *Lab* color spaces into a unified structure. Zhang *et al.* [46] studied the near-independent properties of *Lab* color space, and proposed an adaptive method to enhance the contrast and saturation in *RGB* color. In grayscale image coloring, Wan *et al.* [40] utilized the *RGB* color space to colorize the initialized super-pixel, and then employed the *YUV* color space for color propagation to achieve a balance between efficiency and effectiveness. In video tracking, Lai *et al.* [21] investigated loss designation in terms of different color spaces (*e.g.*, *RGB*, *Lab*, and *HSV*), revealing that the decorrelated color space could force models to learn more robust features.

Our work is the first deep image harmonization method using multiple color spaces. Specifically, we extract disentangled *L*, *a*, *b* features from decorrelated *Lab* color space, to supplement the entangled *RGB* features extracted from correlated *RGB* color space.

2.3 Dynamic Neural Network

Dynamic neural networks aim to dynamically adjust the model parameters or structures to cope with different conditions, which can improve the generalization and representation ability of models.

For dynamic neural networks with dynamic parameters, Chen *et al.* [6] were the first to propose dynamic convolution, which aggregates multiple convolution kernels based on attention weight. CondConv [45] introduced the idea of learning sample-dependent convolution kernels to replace original convolution layers, resulting in improved model performance for classification and detection tasks. PAC [37] proposed the pixel-adaptive convolution operation by combining learnable local pixel features with the filter weights to change the standard convolution operation. In terms of dynamic neural networks with dynamic structures, MSDNet [17] proposed a multi-scale DenseNet with an early-exit strategy that decides when to exit the network for different samples. ATC [11] developed an algorithm that enables recurrent neural networks to learn the number of computational steps between receiving an input and emitting an output, making previously inaccessible problems manageable.

In our *Lab* control module, inspired by StyleGANv2 [19], we use *L*, *a*, and *b* control codes to generate dynamic convolution kernels for feature manipulation, which falls within the scope of dynamic parameters. This approach enables us to adjust the decoder features in the harmonization backbone using the *L*, *a*, and *b* control codes.

3 METHOD

In this section, we will set forth to our DucoNet. In detail, we will first briefly introduce our overall framework in Section 3.1, and our used harmonization backbone in Section 3.2. In Section 3.3, we will detail the process to extract the *L*, *a*, *b* control codes. In Section 3.4, we will describe how our *Lab* control module (*Lab*-CM) exploits the *L*, *a*, *b* control codes to adjust the decoder features in the harmonization backbone.

3.1 Overview

Given a composite image $\boldsymbol{I}_c$ and its foreground mask $\boldsymbol{M}$, the goal of image harmonization is adjusting the foreground of $\boldsymbol{I}_c$ and producing the harmonized image $\boldsymbol{I}_h$ as output. Prior works [8, 9, 13, 20, 35] only use the composite image in the *RGB* color space as input. However, *RGB* color space is a correlated color space, which may increase the workload of existing methods to disentangle independent factors (*e.g.*, lightness, orthogonal colors), potentially complicating the harmonization process. Considering that the decorrelated *Lab* color space contains disentangled color and illumination statistics, we additionally use the composite image with *Lab* channels as input to help improve the harmonization performance.

As shown in Figure 2, the overall framework consists of three parts: the harmonization backbone, the *Lab* encoding module, and the *Lab* control module. Following previous works [10, 35], the harmonization backbone uses the composite image with *RGB* channels $\boldsymbol{I}_{c,RGB} \in \mathbb{R}^{H\times W\times 3}$ concatenated with the foreground mask $\boldsymbol{M} \in \mathbb{R}^{H\times W\times 1}$ as input. We have also tried using *Lab* color space in harmonization backbone, but the results are compromised (see Section 4.4). Therefore, we still use *RGB* color space in harmonization backbone. Considering the effectiveness and efficiency, we adopt iSSAM [35] as our harmonization backbone, which can also be easily replaced by other harmonization backbones. For the *Lab* encoding module, we use the composite image with *Lab* channels $\boldsymbol{I}_{c,Lab} \in \mathbb{R}^{H\times W\times 3}$ concatenated with the foreground mask $\boldsymbol{M}$ as input. Considering that the *L*, *a*, and *b* channels are near-independent, we process different channels $\boldsymbol{I}_{c,L}, \boldsymbol{I}_{c,a}, \boldsymbol{I}_{c,b} \in \mathbb{R}^{H\times W\times 1}$ using three encoders E_L, E_a, E_b separately to obtain the corresponding *L*, *a*, and *b* control codes $\boldsymbol{s}_L, \boldsymbol{s}_a, \boldsymbol{s}_b \in \mathbb{R}^{d_s}$, $d_s = 256$. *Lab* control module uses *L*, *a*, and *b* control codes to adjust the decoder feature maps in the harmonization backbone. Finally, the decoder of harmonization backbone outputs the harmonized image $\boldsymbol{I}_h$, which is supervised by the ground-truth image $\boldsymbol{I}_g$ using L_1 loss $\mathcal{L} = ||\boldsymbol{I}_h - \boldsymbol{I}_g||_1$.

3.2 Harmonization Backbone

The choice of harmonization backbones should balance effectiveness and efficiency simultaneously. Therefore, we opt for iSSAM [35] as our harmonization backbone, which is framed as a U-Net [34] with four encoder layers and three decoder layers. The first three encoder layers output features, which are connected with the corresponding decoder layers via skip connections to preserve the encoded information. To tailor for image harmonization, an Spatial-Separated Attention Module [10] and a blending layer [35] are

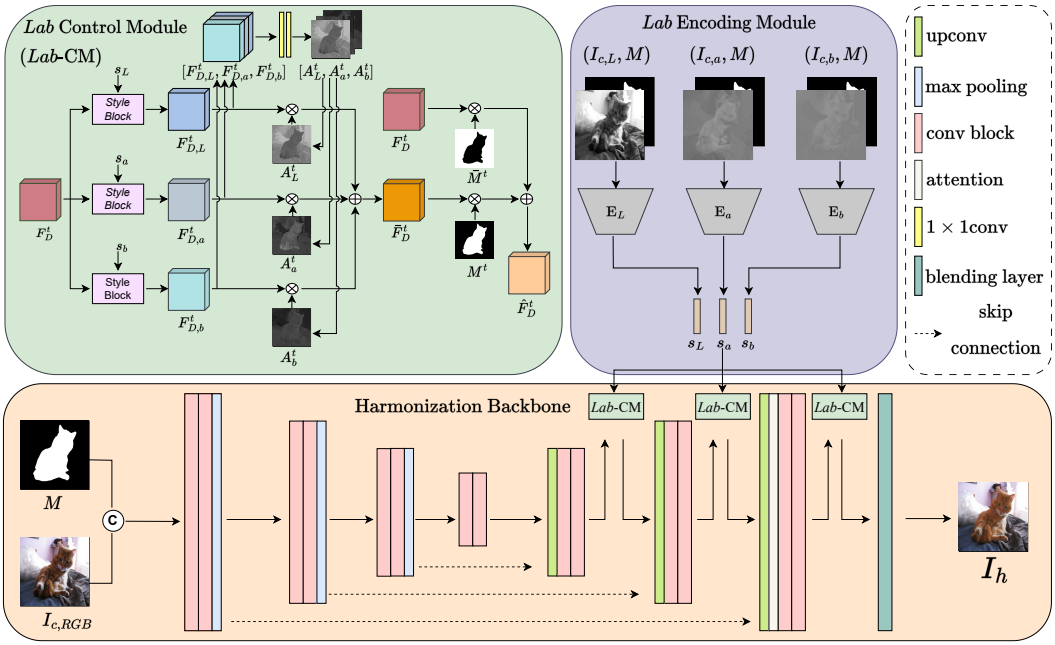

Figure 2: The illustration of our harmonization network with Dual Color Spaces (DucoNet). Given a composite image I_c and its foreground mask M, the harmonization backbone [35] takes *RGB* channels of composite image ($I_{c,RGB}$) concatenated with M as input, and generates the harmonized image I_h. In *Lab* encoding module, three encoders extract control codes s_L, s_a, and s_b from L, a, and b channels of composite image $I_{c,L}$, $I_{c,a}$, $I_{c,b}$, respectively, which are used to manipulate the decoder feature maps in the harmonization backbone. We insert *Lab* control module (*Lab*-CM) into each decoder layer. For the t-th decode feature map F_D^t output from the t-th decoder layer, we use s_L, s_a, and s_b to manipulate F_D^t independently through style blocks [19]. Then, three manipulated decoder feature maps are fused as $\bar{F}_D^t$ with learnt pixel-wise weights. Finally, the foreground of $\bar{F}_D^t$ and the background of F_D^t are combined as $\hat{F}_D^t$ and sent back to the decoder to produce the harmonized image I_h.

inserted to the last decoder layer. For more details, please refer to iSSAM [35].

As mentioned earlier, the harmonization backbone still uses the composite image with *RGB* channels $\boldsymbol{I}_{c,RGB} \in \mathbb{R}^{H\times W\times 3}$ concatenated with the foreground mask $\boldsymbol{M}$ as input, and outputs the harmonized result $\boldsymbol{I}_h$. To adjust the decoder feature maps with the L, a, and b control codes, each decoder feature map is sent into our *Lab*-CM along with the L, a, and b control codes, which allows disentangled L, a, b features to help produce more harmonious images. The details of *Lab*-CM will be introduced in Section 3.4.

3.3 *Lab* Encoding Module

The *RGB* color space has been well explored in image harmonization tasks [7–10, 13, 14, 16, 20, 27, 35]. Due to the correlation among *RGB* channels, the extracted RGB features may not disentangle the independent factors (*e.g.*, lightness, orthogonal colors) effectively. Thus, we additionally use the decorrelated *Lab* color space to supplement *RGB* color space. As introduced in Section 1, L, a, and b channels in *Lab* color space represent lightness, the spectrum from green to red, and the spectrum from blue to yellow, respectively.

In the *Lab* color space, we attempt to obtain the control code of each channel using the respective control encoder. Each encoder E_L, E_a, and E_b) in the *Lab* encoding module has the same structure as the encoder of the harmonization backbone, followed by a pooling layer and a fully-connected layer. Each encoder extracts the independent feature from one channel, which serves as the control code to manipulate the decoder feature maps through our *Lab* control module (*Lab*-CM). In detail, we first convert the composite image from *RGB* color space $\boldsymbol{I}_{c,RGB} \in \mathbb{R}^{H\times W\times 3}$ to *Lab* color space $\boldsymbol{I}_{c,Lab} \in \mathbb{R}^{H\times W\times 3}$, and obtain three separate channels $\boldsymbol{I}_{c,L}$, $\boldsymbol{I}_{c,a}$, and $\boldsymbol{I}_{c,b} \in \mathbb{R}^{H\times W\times 1}$. These three single-channel composite images are concatenated with the $\boldsymbol{M}$ and delivered to the corresponding control encoders to yield the corresponding control code.

By taking the L channel $\boldsymbol{I}_{c,L}$ as an example, the L control code $\boldsymbol{s}_L$ is generated through the following steps. We first scale the range of $\boldsymbol{I}_{c,L}$ to $[0, 1]$, and then concatenate it with $\boldsymbol{M}$ as input. The concatenation is sent into E_L to produce the feature map $\boldsymbol{F}_L$, which is then transformed into the L control code $\boldsymbol{s}_L$ through one pooling layer AvgPool and one fully connected layers FC_L. The whole process for generating L, a, and b control codes can be formulated as

$$\begin{aligned} \boldsymbol{F}_L &= \mathrm{E}_L(\boldsymbol{I}_{c,L}, \boldsymbol{M}), \quad \boldsymbol{s}_L = \mathrm{FC}_L(\mathrm{AvgPool}(\boldsymbol{F}_L)), \\ \boldsymbol{F}_a &= \mathrm{E}_a(\boldsymbol{I}_{c,a}, \boldsymbol{M}), \quad \boldsymbol{s}_a = \mathrm{FC}_a(\mathrm{AvgPool}(\boldsymbol{F}_a)), \\ \boldsymbol{F}_b &= \mathrm{E}_b(\boldsymbol{I}_{c,b}, \boldsymbol{M}), \quad \boldsymbol{s}_b = \mathrm{FC}_b(\mathrm{AvgPool}(\boldsymbol{F}_b)). \end{aligned} \tag{1}$$

With three control encoders, we get three control codes $\boldsymbol{s}_L$, $\boldsymbol{s}_a$, and $\boldsymbol{s}_b$ corresponding to three channels. They encode the independent factors of color and illumination statistics from the composite

image in *Lab* color space, which can further provide guidance for decoder feature manipulation in our *Lab* control module.

3.4 *Lab* Control Module

Our *Lab* Control Module (*Lab*-CM) aims to migrate useful information from the decorrelated *Lab* color space to the *RGB* color space, by using three control codes to manipulate the decoder feature maps in the harmonization backbone. Recall that our harmonization backbone has three decoder layers and the output feature map from the t-th decoder layer is denoted as $\boldsymbol{F}_D^t$. We insert *Lab*-CM after each decoder layer. For the t-th decoder layer, *Lab*-CM takes $\boldsymbol{F}_D^t$ along with L, a, b control codes as input, producing the *Lab*-enhanced decoder feature map $\hat{\boldsymbol{F}}_D^t$. Precisely, we first use three control codes to get three manipulated feature maps independently, and then fuse them using learnt pixel weights.

Feature Map Manipulation: By taking the decoder feature map $\boldsymbol{F}_D^1$ from the first decoder layer as an example, we attempt to use three control codes $\boldsymbol{s}_L, \boldsymbol{s}_a$, and $\boldsymbol{s}_b$ to manipulate $\boldsymbol{F}_D^1$ independently and obtain three manipulated decoder feature maps. In this work, we adopt the style block proposed in StyleGANv2 [19], which is essentially dynamic convolution. The style block produces dynamic convolution kernel using the control code and apply it to the decoder feature map.

Specifically, for each color channel c from {*L, a, b*}, we have one 3×3 base convolution kernel $\boldsymbol{W}_c$, and use control code $\boldsymbol{s}_c$ to dynamically scale the input channels of $\boldsymbol{W}_c$. We first project $\boldsymbol{s}_c$ to a scale vector $\boldsymbol{u}_c$ using two fully-connected layers, in which $\boldsymbol{u}_c$ contains the scales for each input channel. The scaling process is represented by

$$\hat{w}_c^{i,j,k} = u_c^i \cdot w_c^{i,j,k}, \tag{2}$$

in which $w_c^{i,j,k}$ is the (i, j, k)-th entry in $\boldsymbol{W}_c$ with i, j, k enumerating the input channel, output channel, and the spatial location respectively. u_c^i is the i-th entry in $\boldsymbol{u}_c$, representing the scale for the i-th input channel. Then, we normalize $\hat{w}_c^{i,j,k}$ as

$$\bar{w}_c^{i,j,k} = \hat{w}_c^{i,j,k} \Big/ \sqrt{(\sum_{i,k} \hat{w}_c^{i,j,k})^2 + \epsilon}, \tag{3}$$

where ϵ is a small constant to prevent numerical errors. $\bar{w}_c^{i,j,k}$ form the dynamic convolution kernel $\bar{\boldsymbol{W}}_c$, which acts upon the decoder feature map $\boldsymbol{F}_D^1$ to produce the manipulated feature map $\boldsymbol{F}_{D,L}^1$. For more details of the style block, please refer to StyleGANv2 [19].

With three control codes, we can get three manipulated feature maps $\boldsymbol{F}_{D,L}^1, \boldsymbol{F}_{D,a}^1, \boldsymbol{F}_{D,b}^1$. By using P_c to denote the style block for the color channel c, the feature map manipulation can be formulated as

$$\boldsymbol{F}_{D,L}^1 = \mathrm{P}_L(\boldsymbol{F}_D^1, \boldsymbol{s}_L), \quad \boldsymbol{F}_{D,a}^1 = \mathrm{P}_a(\boldsymbol{F}_D^1, \boldsymbol{s}_a), \quad \boldsymbol{F}_{D,b}^1 = \mathrm{P}_b(\boldsymbol{F}_D^1, \boldsymbol{s}_b). \tag{4}$$

Feature Map Fusion: Considering that L, a, b channels may contribute differently to various images or even various pixels, we learn pixel-wise weights $\{\boldsymbol{A}_L^1, \boldsymbol{A}_a^1, \boldsymbol{A}_b^1\}$ for three channels when fusing three manipulated feature maps $\{\boldsymbol{F}_{D,L}^1, \boldsymbol{F}_{D,a}^1, \boldsymbol{F}_{D,b}^1\}$. Specifically, we concatenate three manipulated feature maps and send them to G^1:

$$[\boldsymbol{A}_L^1, \boldsymbol{A}_a^1, \boldsymbol{A}_b^1] = \mathrm{G}^1\left([\boldsymbol{F}_{D,L}^1, \boldsymbol{F}_{D,a}^1, \boldsymbol{F}_{D,b}^1]\right), \tag{5}$$

where G^1 is constructed by a 1×1 convolution layer and a softmax layer, $\{\boldsymbol{A}_L^1, \boldsymbol{A}_a^1, \boldsymbol{A}_b^1\}$ are single-channel weight maps. After that, we fuse three manipulated feature maps $(\boldsymbol{F}_{D,L}^1, \boldsymbol{F}_{D,a}^1, \boldsymbol{F}_{D,b}^1)$ using the predicted pixel-wise weights. Note that we only manipulate the foreground feature map, aiming to make it compatible with the background. Thus, the original background feature map in $\boldsymbol{F}_D^1$ is preserved. The above process is represented by

$$\begin{aligned} \bar{\boldsymbol{F}}_D^1 &= \boldsymbol{F}_{D,L}^1 \circ \boldsymbol{A}_L^1 + \boldsymbol{F}_{D,a}^1 \circ \boldsymbol{A}_a^1 + \boldsymbol{F}_{D,b}^1 \circ \boldsymbol{A}_b^1, \\ \hat{\boldsymbol{F}}_D^1 &= \bar{\boldsymbol{F}}_D^1 \circ \boldsymbol{M}^1 + (1 - \boldsymbol{M}^1) \circ \boldsymbol{F}_D^1, \end{aligned} \tag{6}$$

where $\circ$ means element-wise product and $\hat{\boldsymbol{F}}_D^1$ is the final *Lab*-enhanced feature map.

Similar steps can be applied to decoder feature maps $\boldsymbol{F}_D^2$ and $\boldsymbol{F}_D^3$ to get the *Lab*-enhanced feature maps $\hat{\boldsymbol{F}}_D^2$ and $\hat{\boldsymbol{F}}_D^3$. The *Lab*-enhanced feature maps are sent back to the decoder of the harmonization backbone to generate the final harmonized image $\boldsymbol{I}_h$.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

4.1.1 Dataset. Following previous image harmonization works, we conduct experiments on the benchmark dataset iHarmony4 [9] to evaluate the effectiveness of our DucoNet, where the iHarmony4 [9] has been widely used in supervised image harmonization. In detail, iHarmony4 [9] consists of four sub-datasets, including HFlickr, Hday2night, HCOCO, and HAdobe5K, with 73,146 samples in total. For each sample in iHarmony4 [9], it includes a composite image, its foreground mask, and the corresponding ground-truth image.

We perform both low-resolution and high-resolution image harmonization based on iHarmony4. For low-resolution harmonization, we conduct experiments with image size 256×256 following previous works [35]. For high-resolution harmonization, we follow the experimental setting in CDTNet [8]. Specifically, we perform training and testing based on the HAdobe5k dataset with image size 1024×1024. Moreover, we also evaluate our trained model on 100 high-resolution real composite images collected in CDTNet [8]. Since real composite images have no ground-truth image for evaluation, we present the user study results.

4.1.2 Evaluation Metrics. We adopt the evaluation metrics which are commonly used in previous image harmonization works [8, 9, 13, 20, 35, 43], including MSE (Mean-Square-Error), fMSE (foreground Mean-Square-Error), and PSNR (Peak Signal to Noise Ratio).

4.2 Implementation Details

Our network is implemented with PyTorch 1.10.1, optimized by Adam optimizer with initial learning rate as 1×10^{-3}. The batch size is set as 64 and we train our DucoNet for 120 epochs in total. The learning rate decay starts at epoch 105 and epoch 115 with a decay factor of 10. The hardware devices used for training are Intel(R) Xeon(R) Silver 4116 CPU, with 128GB memory and two NVIDIA GeForce RTX 3090 GPUs. More details about the implementation can be found in Supplementary.

Figure 3: From left to right, we show the composite image (foreground outlined in green), the harmonized results of iSSAM [35], CDTNet [8], Harmonizer [20], DCCF [43], our DucoNet, and the ground-truth in iHarmony4 [9] dataset. Best viewed in color and zoom in.

4.3 Comparison with Start-of-the-Art Methods

Low-resolution Harmonization: We compare our method with the existing methods. In the low-resolution setting with image size 256×256, we compare our method with DoveNet [9], RainNet [27], Instrinsic [14], IHT (Image Harmonization with Transformer) [13], iSSAM [35], CDTNet [8], Harmonizer [20], and DCCF [43]. The experiment results are copied from original papers or reproduced with the released models.

In Table 1, we report the results on four sub test sets and the whole test set in the low-resolution setting. For the results on the whole test set, our DucoNet outperforms the SOTA method by a large margin. Specifically, our DucoNet achieves 15.68% relative

Method	All			HCOCO			HFlickr			HAdobe5k			Hday2night		
	MSE ↓	fMSE ↓	PSNR ↑	MSE ↓	fMSE ↓	PSNR ↑	MSE ↓	fMSE ↓	PSNR ↑	MSE ↓	fMSE ↓	PSNR ↑	MSE ↓	fMSE ↓	PSNR ↑
Composite images	172.47	1387.30	31.63	69.37	1013.27	33.94	264.35	1612.59	28.32	345.54	2137.07	28.16	109.65	1443.05	34.01
DoveNet [9]	52.36	549.96	34.75	36.72	554.55	35.83	133.14	823.64	30.21	52.32	383.91	34.34	54.05	1075.42	35.18
RainNet [27]	40.29	469.60	36.12	31.12	535.40	37.08	117.59	751.12	31.64	42.85	320.43	36.22	47.24	852.12	34.83
Instrinsic [14]	38.71	400.29	35.90	24.92	416.38	37.16	105.13	716.60	31.34	43.02	284.21	35.20	55.53	797.04	35.96
IHT [13]	27.89	295.56	37.94	14.98	274.67	39.22	67.88	471.04	33.55	36.83	242.57	37.17	49.67	736.55	36.38
iSSAM [35]	24.64	262.67	37.95	16.48	266.14	39.16	69.68	443.63	33.56	22.59	166.19	37.24	40.59	591.07	37.72
CDTNet [8]	23.75	252.05	38.23	16.25	261.29	39.15	68.61	423.03	33.55	20.62	149.88	38.24	36.72	549.47	37.95
Harmonizer [20]	24.26	280.51	37.84	17.34	298.42	38.77	64.81	434.06	33.63	21.89	170.05	37.64	**33.14**	542.07	37.56
DCCF [43]	22.05	266.49	38.50	14.87	272.09	39.52	60.41	411.53	33.94	19.90	175.82	38.27	49.32	655.43	37.88
DucoNet	**18.47**	**212.53**	**39.17**	**12.12**	**211.25**	**40.23**	**51.71**	**353.81**	**34.65**	**17.06**	**141.55**	**38.87**	38.70	**527.07**	**38.11**

Table 1: Comparison of different methods with image size 256×256 on iHarmony4. ↓ (*resp.*, ↑) indicates that lower (*resp.*, higher) values are better. The best results are highlighted in bold face.

Method	MSE ↓	fMSE ↓	PSNR ↑
Composite images	352.05	2122.37	28.10
iSSAM [35]	25.03	168.85	38.29
CDTNet-256(sim) [8]	31.15	195.93	37.65
CDTNet-256 [8]	21.24	152.13	38.77
Harmonizer [20]	20.12	150.99	38.45
DCCF [43]	21.12	171.17	38.38
DucoNet	**10.94**	**80.69**	**41.37**

Table 2: Comparison of different methods with image size 1024×1024 on HAdobe5k. ↓ (*resp.*, ↑) indicates that lower (*resp.*, higher) values are better. The best results are denoted in bold face.

improvement over CDTNet [8] in terms of fMSE and 16.23% relative improvement over DCCF [43] in terms of MSE. Considering each sub test set, our DucoNet achieves the best results on HCOCO, HFlickr, and HAdobe5k, which indicates the generation ability our method. On Hday2night, our method achieves the best results in terms of fMSE and PSNR, and the third best result for MSE, probably due to the small-scale training set and test set (only 311 images for training and 133 image for test).

We further visualize the harmonized results of different methods in Figure 3. It can be seen that our method can produce more visually appealing and harmonious results, that are closer to the ground-truth real images. These visualisation results again demonstrate the effectiveness of our proposed method.

High-resolution Harmonization: Recently, there are also a few works that focus on high-resolution image harmonization. In the high-resolution setting with image size 1024×1024, we compare our DucoNet with iSSAM [35], CDTNet [8], Harmonizer [20], DCCF [43] in HAdobe5k subset with image size 1024×1024. CDTNet-256 is the CDTNet [8] model with the input size of harmonization backbone being 256×256, and CDTNet-256(sim) is a simplified version of CDTNet-256. The experimental results for DCCF, CDTNet-256 and CDTNet-256(sim) are copied from the corresponding paper. Harmonizer did not report their results in the same high-resolution setting as CDTNet [8], so we train the corresponding models on HAdobe5k training set with image size 1024×1024 for fair comparison.

In Table 2, we report the results on HAdobe5k in the high-resolution image harmonization setting. Our DucoNet outperforms all the baselines by a large margin in terms of all evaluation metrics in high-resolution image harmonization. Specifically, our DucoNet achieves 45.63% relative improvement over Harmonizer [20] in terms of MSE and achieves 46.56% relative improvement over Harmonizer [20] in terms of fMSE.

4.4 Ablation Study

As described in Section 3, our DucoNet consists of the harmonization backbone, the *Lab* encoding module, and the *Lab* control module (*Lab*-CM). In this section, we demonstrate the effectiveness of each component and each color space by ablating each component or comparing with alternatives.

The results of our ablation studies are presented in Table 3. Firstly, when only using the harmonization backbone, we compare using the input composite image with *RGB* channels (row 1) and using the input composite image with *Lab* channels (row 2). By comparing row 1 and row 2, we see that *RGB* channels outperforms *Lab* channels, revealing that *RGB* channels are still more suitable as the input for the U-Net structure. Note that although the inputs to the network are different, the loss and evaluation metrics are all calculated based on *RGB* channels for fair comparison. In detail, when using the input composite image with *Lab*, we first generate the harmonized image with *Lab* channels and then convert it into *RGB* color space for loss calculation and evaluation.

To evaluate the effectiveness of *Lab* color space for feature manipulation, we treat the *Lab* (*resp.*, *RGB*) channels as a whole input in the encoding module and use a single control code in the control module, leading to the results in row 3 (*resp.*, row 4). Comparing row 3 and row 4, we can find that the *Lab* channels are more helpful for feature manipulation, because *Lab* color space could supplement *RGB* color space with extra useful guidance.

In row 5, we study a simple way to fuse *RGB* and *Lab* features. In particular, we treat the *Lab* channels as a whole input in encoding module and send multi-scale encoder features to the decoder via skip-connection, in the same way as the backbone encoder. The

	RGB	*Lab*	**Fusion**	**MSE ↓**	**fMSE ↓**	**PSNR ↑**
1	iSSAM	-	-	24.64	262.67	37.95
2	-	iSSAM	-	28.13	296.59	37.20
3	iSSAM	E(*Lab*)	CM	19.30	222.22	38.93
4	iSSAM	E(*RGB*)	CM	22.66	245.38	38.62
5	iSSAM	E(*Lab*)	SC	21.76	243.06	38.47
6	iSSAM	E(*L*)	CM	21.43	234.70	38.72
7	iSSAM	E(*a*)	CM	23.32	256.34	38.39
8	iSSAM	E(*b*)	CM	23.46	255.29	38.36
9	iSSAM	E(*L*,*a*,*b*)	CM-avg	20.45	227.71	38.88
10	iSSAM	E(*L*,*a*,*b*)	CM-pix	18.47	212.53	39.17

Table 3: The ablation study of our DucoNet. "iSSAM" indicates using the harmonization backbone [35] in the corresponding color space. "E(*Lab*)", "E(*RGB*)", "E(*L*)", "E(*a*)", "E(*a*)", and "E(*L*,*a*,*b*)" indicate that we treat *Lab* as a whole, *RGB* as a whole, only *L*, only *a*, only *b*, and *L*,*a*,*b* separately as input in the *Lab* encoding module. "SC" is short for skip-connection. "CM" is short for *Lab*-CM. "CM-avg" indicates average fusion. "CM-pix" indicates weighted fusion with pixel-wise weights.

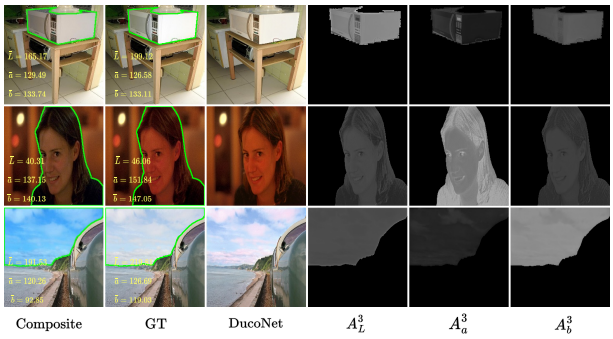

Figure 4: From left to right, we show the composite image (foreground outlined in green), the ground-truth, the harmonized results of our method, visualization of $\{A_L^3, A_a^3, A_b^3\}$ in *Lab*-CM. In composite image and ground-truth image, we also show the average value in *L*, *a*, *b* channels within the foreground region. Best viewed in color and zoom in.

obtained performance is worse than row 3, which demonstrates the effectiveness of feature manipulation in our *Lab*-CM.

Furthermore, we conduct experiments by treating each individual *L*, *a*, *b* channel as the input in the encoding module and use the single control code in the control module (row 6 *v.s.* row 7 *v.s.* row 8). Experimental results shows the *L* channel is the most effective one among all three channels. To provide some insights for the importance of *L* channel, we calculate the amount of change between the foreground area of composite image and the ground-truth image for each channel (*L*, *a*, and *b*), the average amount of change in three channels are 25.90, 3.88, and 6.65 respectively over the entire test set. The average amount of change in *L* channel is significantly higher than the other two channels, which corroborates that merely using *L* channel could achieve compelling results (row 6).

Finally, we conduct experiments to verify the effectiveness of the pixel-wise weighting strategy. Comparing row 9 with row 10, we can find that simply averaging the manipulated feature maps $\{\boldsymbol{F}_{D,L}^t, \boldsymbol{F}_{D,a}^t, \boldsymbol{F}_{D,b}^t\}$ undermines the representation ability of *Lab*-CM, since *L*, *a*, *b* channels contribute differently to the harmonization results.

4.5 Visualization of Weight Map

To show the effectiveness of our proposed *Lab*-CM, we visualize the weight maps $\{\boldsymbol{A}_L^3, \boldsymbol{A}_a^3, \boldsymbol{A}_b^3\}$ from the third decoder layer in Figure 4. Recall that we only manipulate the foreground region of decoder feature map and the background pixel weights do not contribute to the final output. Thus, we mask out the background pixels and only show the pixel weights in the foreground region in Figure 4, in which brighter pixel indicates higher weight. For composite image and ground-truth, we also show the average value in *L*, *a*, *b* channels within the foreground region, which reflects the amount of change in each channel.

Based on Figure 4, we observe that the learnt weight map is closely related to the amount of change in each channel. Recall that *L*, *a*, and *b* channels in *Lab* color space represent lightness, the spectrum from green to red, and the spectrum from blue to yellow, respectively. When the lightness between foreground and background in the composite image is contrastively different (row 1), the value of *L* channel would change greatly after harmonization, in which case the weight map $\boldsymbol{A}_L$ corresponding to the *L* channel has the largest values. When the foreground object has dominant color (row 3) or the lighting has color cast (row 2), the value of the corresponding color channel (*e.g.*, red, blue) would vary greatly after harmonization, in which the corresponding weight map has the largest values (Figure 4).

4.6 Real Composite Images

Following previous works, we also evaluate different methods on 100 real composite images in CDTNet [8]. The visualization results of different baseline methods are provided in the Supplementary. Since these real composite images do not have ground-truth image, we conduct user study to compare different methods, which is also left to the Supplementary.

5 CONCLUSION

In this paper, we have explored image harmonization in dual color spaces, where we additionally use the decorrelated color space *Lab* to relieve the burden of the harmonization process when compared with using *RGB* color space alone. We have proposed a novel network DucoNet, which manipulates the foreground of the decoder feature maps from the harmonization backbone using the control codes from *Lab* color space. Experiments conducted on the benchmark dataset have shown that our approach significantly outperforms the state-of-the-art methods.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Zhongyun Bao, Chengjiang Long, Gang Fu, Daquan Liu, Yuanzhen Li, Jiaming Wu, and Chunxia Xiao. 2022. Deep Image-based Illumination Harmonization. In *CVPR*.

[2] Junyan Cao, Wenyan Cong, Li Niu, Jianfu Zhang, and Liqing Zhang. 2022. Deep Image Harmonization by Bridging the Reality Gap. In *BMVC*.

[3] Junyan Cao, Yan Hong, and Li Niu. 2023. Painterly Image Harmonization in Dual Domains. In *AAAI*.

[4] Haoxing Chen, Zhangxuan Gu, Yaohui Li, Jun Lan, Changhua Meng, Weiqiang Wang, and Huaxiong Li. 2022. Hierarchical Dynamic Image Harmonization. *arXiv preprint arXiv:2211.08639* (2022).

[5] Jianqi Chen, Yilan Zhang, Zhengxia Zou, Keyan Chen, and Zhenwei Shi. 2023. Dense Pixel-to-Pixel Harmonization via Continuous Image Representation. *arXiv preprint arXiv:2303.01681* (2023).

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. 2020. Dynamic convolution: Attention over convolution kernels. In *CVPR*.

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

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

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

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

[11] Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983* (2016).

[12] Zonghui Guo, Zhaorui Gu, Bing Zheng, Junyu Dong, and Haiyong Zheng. 2022. Transformer for Image Harmonization and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022).

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

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

[15] Yucheng Hang, Bin Xia, Wenming Yang, and Qingmin Liao. 2022. Scs-co: Self-consistent style contrastive learning for image harmonization. In *CVPR*.

[16] Guoqing Hao, Satoshi Iizuka, and Kazuhiro Fukui. 2020. Image Harmonization with Attention-based Deep Feature Modulation.. In *BMVC*.

[17] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844* (2017).

[18] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. 2021. SSH: a self-supervised framework for image harmonization. In *ICCV*.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *CVPR*.

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

[21] Zihang Lai, Erika Lu, and Weidi Xie. 2020. Mast: A memory-augmented self-supervised tracker. In *CVPR*.

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

[23] Edwin H Land and John J McCann. 1971. Lightness and retinex theory. *Josa* 61, 1 (1971), 1–11.

[24] Chongyi Li, Saeed Anwar, Junhui Hou, Runmin Cong, Chunle Guo, and Wenqi Ren. 2021. Underwater image enhancement via medium transmission-guided multi-color space embedding. *IEEE Transactions on Image Processing* 30 (2021), 4985–5000.

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

[26] Jing Liang, Li Niu, Penghao Wu, Fengjun Guo, and Teng Long. 2022. Inharmonious region localization by magnifying domain discrepancy. In *AAAI*.

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

[28] Ziyin Ma and Changjae Oh. 2022. A wavelet-based dual-stream network for underwater image enhancement. In *ICASSP*.

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

[30] Jinlong Peng, Zekun Luo, Liang Liu, Boshen Zhang, Tao Wang, Yabiao Wang, Ying Tai, Chengjie Wang, and Weiyao Lin. 2022. FRIH: Fine-grained Region-aware Image Harmonization. *arXiv preprint arXiv:2205.06448* (2022).

[31] Lintao Peng, Chunli Zhu, and Liheng Bian. 2023. U-shape transformer for underwater image enhancement. *IEEE Transactions on Image Processing* (2023).

[32] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.

[33] Xuqian Ren and Yifan Liu. 2022. Semantic-guided multi-mask image harmonization. In *ECCV*.

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

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

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

[37] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. 2019. Pixel-adaptive convolutional neural networks. In *CVPR*.

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

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

[40] Shaohua Wan, Yu Xia, Lianyong Qi, Yee-Hong Yang, and Mohammed Atiquzzaman. 2020. Automated colorization of a grayscale image with seed points propagation. *IEEE Transactions on Multimedia* 22, 7 (2020), 1756–1768.

[41] Penghao Wu, Li Niu, and Liqing Zhang. 2022. Inharmonious Region Localization with Auxiliary Style Feature. *BMVC* (2022).

[42] Yazhou Xing, Yu Li, Xintao Wang, Ye Zhu, and Qifeng Chen. 2022. Composite photograph harmonization with complete background cues. In *ACM MM*.

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

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

[45] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. 2019. Condconv: Conditionally parameterized convolutions for efficient inference. *NeurIPS*.

[46] Weidong Zhang, Peixian Zhuang, Hai-Han Sun, Guohou Li, Sam Kwong, and Chongyi Li. 2022. Underwater image enhancement via minimal color loss and locally adaptive contrast enhancement. *IEEE Transactions on Image Processing* 31 (2022), 3997–4010.

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

[48] Ziyue Zhu, Zhao Zhang, Zheng Lin, Ruiqi Wu, Zhi Chai, and Chun-Le Guo. 2022. Image Harmonization by Matching Regional References. *arXiv preprint arXiv:2204.04715* (2022).