Deep Image Harmonization with Globally Guided Feature Transformation and Relation Distillation

Li Niu¹*, Linfeng Tan¹, Xin Hao Tao¹, Junyan Cao¹, Fengjun Guo², Teng Long², Li Qing Zhang¹
¹ Department of Computer Science and Engineering, MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
² INTSIG
{ustcnewly, tanlinfeng, taoxinhao, Joy_sl}@sjtu.edu.cn, {fengjun_guo, mike_long}@intsig.net, zhang-lq@cs.sjtu.edu.cn

Abstract

Given a composite image, image harmonization aims to adjust the foreground illumination to be consistent with background. Previous methods have explored transforming foreground features to achieve competitive performance. In this work, we show that using global information to guide foreground feature transformation could achieve significant improvement. Besides, we propose to transfer the foreground-background relation from real images to composite images, which can provide intermediate supervision for the transformed encoder features. Additionally, considering the drawbacks of existing harmonization datasets, we also contribute a ccHarmony dataset which simulates the natural illumination variation. Extensive experiments on iHarmony4 and our contributed dataset demonstrate the superiority of our method. Our ccHarmony dataset is released at https://github.com/bcmi/Image-Harmonization-Dataset-ccHarmony.

1. Introduction

Image composition [31, 6] is an essential editing operation to combine regions from different images to produce composite images, which has a variety of vision applications like advertisement propaganda and digital entertainment [34, 28]. Nevertheless, when pasting the foreground extracted from one image on another background image, the resultant composite image may have inconsistent illumination statistics. Image harmonization [10, 29] aims to adjust the foreground illumination for visual consistency within the composite image. In recent years, deep learning based image harmonization methods [10, 36, 18, 29, 20, 15] have sprouted out and achieved remarkable progress.

Existing methods [33, 49, 11, 34, 22] have developed myriads of techniques to address the mismatch between foreground and background illumination. Among them, some works [29, 11] revealed that specially transforming the foreground features can enhance the performance, since the main goal of image harmonization is adjusting the foreground. For example, [29, 17] transfer the feature statistics (e.g., mean, variance) from background to foreground. [11, 36] process foreground and background features separately using different channel attentions. These methods only leverage the information within the same feature map to transform the foreground features, lacking the guidance of global information. However, the global information of the whole composite image is very likely to be beneficial for foreground feature transformation.

In this work, we extract the bottleneck feature from encoder as the global feature, which is used to guide the transformation of foreground features in each feature map. Regarding the transformation manner, we opt for modulated convolution kernel proposed in [21], and other transformation manners are also applicable. In particular, we use global feature to obtain the modulated convolution kernel, which is applied to the foreground features. We name our method as GiftNet (globally guided feature transformation). It is worth noting that our method shares similar spirit with CDTNet [9]. CDTNet uses global feature to predict color transformation, while our GiftNet uses global feature to predict feature transformation.

In practice, we observe that globally guided feature transformation is effective for decoder features, but ineffective for encoder features. We conjecture that one reason is the lack of intermediate supervision for encoder features. Specifically, most of previous harmonization methods [33, 49, 11, 34, 22] only use the ground-truth image as the final supervision, without any intermediate supervision. In the auto-encoder based network, the encoder features may not be sufficiently harmonized, which could have negative impact on the harmonization performance. To address the above issue, we provide useful guidance for en-

*Corresponding author.
We name our dataset as ccHarmony, we can transfer each image across different illumination conditions, resulting in a group of images. Then, they swap foregrounds within the same group to construct pairs of composite images and ground-truth images. However, rendered images have huge domain gap with real images, which may not be helpful when we have adequate real training data [4]. In this work, we propose a novel way to approximate natural illumination variation, based on the images with recorded illumination information.

2.2. Image Harmonization Datasets

Previous works have explored various ways to construct image harmonization dataset. 1) [10] contributes iHarmony4 dataset by adjusting the foregrounds in real images to create synthetic composite images. However, such color adjustment may not faithfully reflect natural illumination variation. 2) The Hday2night subdataset in [10] captures a group of images for the same scene under different illumination conditions, which can reflect natural illumination variation. Nevertheless, such data collection process is extremely expensive. 3) [20] constructed RealHM by manually harmonizing the composite images, which is extremely expensive and not scalable. Moreover, the manually harmonized images are subjective and may not be reliable. 4) Other works [16, 4, 2] use 3D render to render the same 3D scene or the same 3D foreground object with different illumination conditions, resulting in a group of images. Then, they swap foregrounds within the same group to construct pairs of composite images and ground-truth images. However, rendered images have huge domain gap with real images, which may not be helpful when we have adequate real training data [4]. In this work, we propose a novel way to approximate natural illumination variation, based on the images with recorded illumination information.

2.3. Knowledge Distillation

Knowledge distillation [19] targets at transferring knowledge from teacher network to student network. Based on the type of knowledge, knowledge distillation can be divided into three groups [13]: response-based [19], feature-based [27], and relation-based [46]. Response-based knowledge distillation usually uses the neural response of the last output layer of the teacher model to supervise the student model [19, 47]. Feature-based knowledge distillation
targets at matching the feature activations between teacher network and student network. Relation-based knowledge distillation [46, 32] pays more attention to the relation between different model layers or data samples. Our work belongs to relation distillation. Precisely, we distill foreground-background relation from reconstruction network to harmonization network.

3. Our Method

We denote the composite image as \( I \), which is composed of foreground \( I^f \) and background \( I^b \). The ground-truth real image is \( \hat{I} \). The goal of image harmonization task is adjusting the composite foreground \( I^f \) and produce the harmonized image \( \hat{I} \), which should be close to \( I \). Our method is built upon the UNet-like network used in [36], which consists of an encoder \( E \) with four encoder blocks, a decoder \( G \) with three decoder blocks, and skip connections from the first three encoder blocks to the corresponding decoder blocks. We insert our designed globally guided feature transformation (GIFT) module into the network, which will be detailed in Section 3.1. Then, we introduce another reconstruction branch to distill the relation knowledge to the harmonization branch, which will be detailed in Section 3.2.

3.1. Globally Guided Feature Transformation

We design a globally guided feature transformation (GIFT) module, which transforms the foreground feature map using global guidance. We use \( F_{e,l} \) (resp., \( F_{d,l} \)) to denote the output feature map from the \( l \)-th encoder (resp., decoder) block. There are four encoder blocks and thus \( F_{e,4} \) is the bottleneck feature map. We perform global average pooling over \( F_{e,4} \) and get the global feature vector \( f_e \), which encodes the global information beneficial for image harmonization task. We use \( f_e \) to guide the feature transformation for encoder/decoder feature maps. We adopt the feature transformation technique proposed in [21], which applies modulated convolution weights to the given feature map. Other feature transformation techniques are also applicable. In our task, we use global feature \( f_e \) to guide the modulation of convolution weights, which provides global guidance for feature transformation. Besides, we only transform the foreground feature map, since image harmonization aims to adjust the foreground to be compatible with the background.

Without loss of generality, we take one encoder/decoder feature map as an example. We have learnable base convolution weights \( \tilde{W} \), in which each entry \( \tilde{W}(m,n,p) \) is the weight for the \( m \)-th input channel, \( n \)-th output channel, and the \( p \)-th spatial location. Then, we pass \( f_e \) through an Multi-Layer Perceptron (MLP) to predict the scale vector \( s \) corresponding to input channels. Our used MLP contains four fully-connected layers, in which the first three are shared by all feature maps and the last one is specific for each feature map. The predicted scale vector is used to modulate the base convolution weights:

\[
\tilde{W}'(m,n,p) = \tilde{W}(m,n,p) \cdot s(m),
\]

where \( s(m) \) is the \( m \)-th entry in \( s \) (the scale for the \( m \)-th input channel), and \( W'(m,n,p) \) is the modulated weight. Next, we normalize the modulated weights following [21]:

\[
\tilde{W}''(m,n,p) = \frac{\tilde{W}'(m,n,p)}{\sqrt{\sum_{m',n',p'}{\tilde{W}'(m',n',p')^2} + \epsilon}},
\]

in which \( \epsilon \) is a small constant to avoid numerical error. For more details of convolution weights modulation, please refer to [21]. Assuming that the feature map to be transformed is \( F_e \), which consists of the foreground feature map \( F_e^f \) and background feature map \( F_e^b \). As illustrated in Figure 1(b), we apply the modulated convolution weights \( \tilde{W}'' \) to the foreground feature map \( F_e^f \) and produce the transformed foreground feature map \( F_e^{f'} \). Then, we combine \( F_e^{f'} \) and \( F_e^{b'} \) to obtain the transformed feature map \( F_e' \).

We try applying our designed GIFT module to the output feature map from each encoder block or decoder block. We observe that the last two decoder blocks can obviously manifest the advantage of our designed GIFT, while the encoder blocks do not show clear advantage (see Section 5.3). We conjecture that the whole network is only supervised by the final ground-truth image and there is no intermediate supervision for the encoder feature maps. Lack of intermediate supervision might hinder the potential of our designed GIFT module. To provide intermediate supervision for the encoder features, we borrow the idea from knowledge distillation, which will be introduced next.

3.2. Relation Distillation

The difficulty of supervising encoder feature maps lies in the absence of ground-truth harmonized encoder feature maps. One intuitive thought is that the harmonized encoder feature maps should be close to the encoder feature maps of ground-truth real images. Therefore, besides the harmonization branch, we introduce another reconstruction branch which reconstructs the ground-truth real image \( \hat{I} \). The reconstruction branch shares similar UNet network structure (encoder \( E \) and decoder \( G \)) with the harmonization branch. We denote the \( l \)-th encoder feature map in the reconstruction branch as \( F_{e,l} \). Note that we perform distillation on the encoder feature maps \( F_{e,l} \) transformed by our GIFT module.

One naive approach is feature distillation, which enforces the encoder feature maps between two branches to be close, that is, \( \tilde{F}_{e,l} \) is close to \( F_{e,l}' \) for \( l = 1, \ldots, 4 \). However, the performance using feature distillation is unsatisfactory (see Section 5.3). One possible reason is that
the encoder feature maps contain lots of redundant information and distilling the entire feature maps increases the difficulty of network training. Therefore, we turn to only distill the key information: foreground-background relation. Distilling foreground-background relation can ensure that the foreground feature map is compatible with the background feature map in $F_e$, as in $\tilde{F}_e$.

There could be many possible ways to characterize foreground-background relation. To avoid the heavy computational cost of pixel-to-pixel relation, we first calculate the averaged foreground feature and then calculate the similarity between it and pixel-wise features. By taking $F'_{e,l}$ as an example, we calculate the averaged foreground feature $f'_{e,l}$ by performing average pooling within the foreground region. Then, we calculate the similarity between $f'_{e,l}$ and the $i$-th pixel-wise feature in $F'_{e,l}$ as follows,

$$R'_{f}[i] = \frac{\exp(-\gamma \| f'_{e,l} - F'_{e,l}[i] \|^2)}{\sum_j \exp(-\gamma \| f'_{e,l} - F'_{e,l}[j] \|^2)}$$

in which $\gamma$ is a hyper-parameter. The similarities $R'_{f}[i]$ of all pixels form a similarity map $R'_f$, in which the values in the background region represent the relation between foreground and background. The background regions with similar appearance to the foreground should have higher values. The similarity map $\hat{R}_f$ for $\hat{F}_e$ could be calculated in a similar way. Then, we distill foreground-background relation from $\tilde{F}_e$ to $F'_{e,l}$ using the distillation loss $L_{dis} = \| R'_f - \hat{R}_f \|^2$, which pushes the foreground-background relation in the harmonized feature map towards that in the real feature map, so that foreground is compatible with the background in the harmonized feature map.

During training, we first train and fix the reconstruction branch. Then, we train the harmonization branch. In addition to the distillation loss, we enforce the harmonized image $\hat{I}$ to be close to the ground-truth image $I$ by $L_{har} = \| \hat{I} - I \|_1$.

After summing up the distillation losses for four encoder blocks, the total loss for the harmonization branch is

$$L_{all} = L_{har} + \lambda \sum_{l=1}^{4} L_{dis}$$

in which $\lambda$ is a hyper-parameter.

4. Our ccHarmony Dataset

In this work, we explore a novel transitive way to construct image harmonization dataset, aiming to simulate the natural illumination variation. Specifically, based on the existing datasets with recorded illumination information, we first convert the foreground in a real image to the standard illumination condition, and then convert it to another illumination condition, which is combined with the original background to produce a synthetic composite image.

We build our dataset upon two publicly datasets with recorded illumination information: NUS dataset [7] and Gehler dataset [12], which were originally constructed for the research on color constancy. In these two datasets, each image is captured with a color checker placed in the scene that provides ground-truth reference for illumination estimation. Given a 24-patch Macbeth color checker, we have the original colors of 24 patches in standard illumination condition (see Figure 2(a)), which is referred to as standard patch color. Given an image with color checker, we can extract the colors of 24 patches, which is referred to as image patch color.

Inspired by previous works [48, 30, 1], we use polynomial matching to characterize the color transformation between standard patch colors and image patch colors. Formally, we denote the standard patch colors of 24 patches
Finally, we replace the foreground patches in image $I_a$ to $I_b$ in standard illumination condition, and then convert $I_b$ to other illumination conditions. Given a real image $I_a$, we convert its foreground $I_a'$ to $I_b'$ in standard illumination condition using polynomial matching matrix $T_a$. Then, we convert $I_b'$ to $I_b^T$ in the illumination condition of reference image $I_b$ using the inverse polynomial matching matrix $T_b^T$. Finally, $I_b^T$ is combined with the background of $I_a$ to produce a synthetic composite image $I_{a\rightarrow b}$.

as $C_o = \{c_o^1, c_o^2, \ldots, c_o^{24}\}$ and image patch colors of 24 patches in image $I_a$ as $C_a = \{c_a^1, c_a^2, \ldots, c_a^{24}\}$. We can learn a polynomial matching matrix $T_a$ which converts $C_o$ to $C_a$. $T_a$ could convert $I_a$ to $I_b$ in standard illumination condition. We can also learn an inverse polynomial matching matrix $T_a'$ which converts $C_o$ to $C_a$. Provided with inverse polynomial matching matrices from other images, we can convert $I_a$ to the illumination conditions of other images. As shown in Figure 2(b), with standard illumination condition being the transitional state, we can transfer across different illumination conditions to simulate natural illumination variation.

Next, we briefly describe our way of creating synthetic composite images. As depicted in Figure 2(c), given an image $I_a$ with foreground mask, we convert its foreground $I_a'$ to the one $I_b'$ in standard illumination condition using polynomial matching matrix $T_a$. Next, by using the inverse polynomial matching matrix $T_b'$ of another reference image $I_b$, we convert $I_b'$ to $I_b^T$ in the illumination condition of $I_b$. Finally, we replace the foreground $I_a'$ in $I_a$ with its counterpart $I_b'$, yielding a synthetic composite image $I_{a\rightarrow b}$. In this manner, we can acquire adequate pairs of synthetic composite images and real images like $\{I_{a\rightarrow b}, I_a\}$. Because our dataset is constructed based on images with color checker (cc), we name our dataset as ccHarmony. The details of our dataset construction are left to supplementary due to space limitation.

5. Experiments

5.1. Datasets and Implementation Details

We conduct experiments on the benchmark iHarmony4 [10] and our constructed ccHarmony dataset. iHarmony4 [10] is composed of four sub-datasets: HCOCO, HFlickr, HAdobe5K, Hday2night, which contains totally 65,742 (resp., 7, 404) pairs of composite images and ground-truth real images in the training (resp., test) set. Following previous works [10, 36], we merge the training sets of four subdatasets as the whole training set and evaluate on each subdataset. Our constructed ccHarmony dataset has 3,080 (resp., 1, 180) pairs of composite images and ground-truth real images in the training (resp., test) set. For the experiments on ccHarmony, we first pretrain the model on iHarmony4 [10] and then finetune the model on ccHarmony, in which the input image size is set as $256 \times 256$.

For evaluation metrics, following previous works [10, 36, 16, 9], we adopt PSNR, MSE, fMSE, fSSIM, in which fMSE (resp., fSSIM) means MSE (resp., SSIM) within the foreground region. Our model is implemented using Pytorch 1.10.1 and trained using Adam optimizer with learning rate being $1e-3$ on ubuntu 20.04 operation system, Intel(R) Xeon(R) Silver 4116 CPU, and RTX 3090 GPUs with 24GB memory. We empirically set $\gamma$ as 0.01 and $\lambda$ as 0.001.

5.2. Comparison with Baselines

On iHarmony4, we compare with following image harmonization methods: DoveNet [36], RainNet [29], IIH [16], IHT [15], iSSAM [36], CDTNet [9], Harmonizer [22], DCCF [44]. In Table 1, we report the results on four sub test sets and the whole test set, which are copied from original papers or reproduced with the released models. For the overall results on the whole test set, our method outperforms the SOTA method by a large margin, e.g., 10.61% relative improvement over CDTNet on fMSE and 11.75% relative improvement over DCCF on MSE. Our method achieves the best results on HCOCO, HFlickr, HAdobe5K. Our method does not achieve satisfactory results on Hday2night, probably due to the limited training
Table 1. Quantitative comparison on iHarmony4 [10] dataset. The best results are denoted in boldface.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE</th>
<th>fMSE</th>
<th>PSNR</th>
<th>MSE</th>
<th>fMSE</th>
<th>PSNR</th>
<th>MSE</th>
<th>fMSE</th>
<th>PSNR</th>
<th>MSE</th>
<th>fMSE</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoveNet [10]</td>
<td>52.36</td>
<td>549.96</td>
<td>34.75</td>
<td>36.72</td>
<td>554.55</td>
<td>35.83</td>
<td>133.14</td>
<td>832.64</td>
<td>30.21</td>
<td>52.32</td>
<td>383.91</td>
<td>34.34</td>
</tr>
<tr>
<td>RainNet [29]</td>
<td>40.29</td>
<td>469.60</td>
<td>36.12</td>
<td>31.12</td>
<td>535.40</td>
<td>37.08</td>
<td>117.59</td>
<td>751.12</td>
<td>31.64</td>
<td>42.85</td>
<td>320.43</td>
<td>36.22</td>
</tr>
<tr>
<td>IHH [16]</td>
<td>38.71</td>
<td>400.29</td>
<td>35.90</td>
<td>24.92</td>
<td>416.38</td>
<td>37.16</td>
<td>105.13</td>
<td>716.60</td>
<td>31.34</td>
<td>43.02</td>
<td>284.21</td>
<td>35.20</td>
</tr>
<tr>
<td>iSSAM [36]</td>
<td>24.64</td>
<td>262.67</td>
<td>37.95</td>
<td>16.48</td>
<td>266.14</td>
<td>39.16</td>
<td>69.68</td>
<td>443.63</td>
<td>33.56</td>
<td>22.59</td>
<td>166.19</td>
<td>37.24</td>
</tr>
<tr>
<td>CDTNet [9]</td>
<td>23.75</td>
<td>252.05</td>
<td>38.23</td>
<td>16.25</td>
<td>261.29</td>
<td>39.15</td>
<td>68.61</td>
<td>423.03</td>
<td>33.55</td>
<td>20.62</td>
<td>149.88</td>
<td>38.24</td>
</tr>
<tr>
<td>Harmonizer</td>
<td>24.64</td>
<td>280.51</td>
<td>37.84</td>
<td>17.34</td>
<td>298.42</td>
<td>38.77</td>
<td>64.81</td>
<td>443.06</td>
<td>33.63</td>
<td>21.89</td>
<td>170.05</td>
<td>37.64</td>
</tr>
<tr>
<td>DCCF [44]</td>
<td>22.05</td>
<td>266.49</td>
<td>38.50</td>
<td>14.87</td>
<td>272.09</td>
<td>39.52</td>
<td>60.41</td>
<td>411.53</td>
<td>33.94</td>
<td>19.90</td>
<td>175.82</td>
<td>38.27</td>
</tr>
</tbody>
</table>

| GiftNet      | 19.46 | 225.30 | 38.92 | 12.70 | 229.67 | 39.91 | 54.33 | 360.08 | 34.44 | 18.35 | 143.96 | 38.76 | 38.28 | 566.47 | 37.81 |

Table 2. Quantitative comparison on our ccHarmony dataset. The best results are denoted in boldface.

<table>
<thead>
<tr>
<th>Row</th>
<th>GIFT</th>
<th>Distill</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Layer</td>
<td>Op</td>
<td>Layer</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>D₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>D₂,₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>D₁,₂,₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>D₂,₃, E</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>D₃</td>
<td>w/o g</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>D₃</td>
<td></td>
<td>D₁,₃</td>
</tr>
<tr>
<td>8</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>E</td>
</tr>
<tr>
<td>9</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>D₂,₃, E</td>
</tr>
<tr>
<td>10</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Row</th>
<th>GIFT</th>
<th>Distill</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Layer</td>
<td>Op</td>
<td>Layer</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>D₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>D₂,₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>D₁,₂,₃</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>D₂,₃, E</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>D₃</td>
<td>w/o g</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>D₃</td>
<td></td>
<td>D₁,₃</td>
</tr>
<tr>
<td>8</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>E</td>
</tr>
<tr>
<td>9</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>D₂,₃, E</td>
</tr>
<tr>
<td>10</td>
<td>D₂,₃, E</td>
<td>*</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies on iHarmony4 [10] dataset. * means the default operation. Dₖ means the k-th decoder block. E means all encoder blocks. "FD" is short for feature distillation. "w/o g" means without guidance. "D₃ → g" means using D₃ feature map as guidance. The best results are denoted in boldface.

5.3. Ablation Studies

We conduct ablation studies and report the results in Table 3. After removing GIFT module and relation distillation, our network reduces to [36], so we include the results of [36] in row 1 as the performance of basic model.

GIFT Module: We first investigate the effectiveness of our GIFT module. Recall that we have four encoder blocks and three decoder blocks. We first append GIFT module to each encoder block or decoder block. We observe that appending GIFT module to the last decoder block achieves the largest improvement. For brevity, we only report the performance of appending to the last decoder block (D₃ in row 2) in Table 3. Based on row 2, we further append GIFT module to the penultimate decoder block (D₂,₃ in row 3) and the performance gain is relatively small. Then, we further append GIFT module to the first decoder block (D₁,₂,₃ in row 4) or the encoder blocks (E in row 5), but the performance gains are very marginal.

Based on row 2, we compare with other types of guidance information for modulating convolution weights. The first type is no guidance information, that is, we use the same learnable input channel scales for all images. The performance is only comparable with row 1, which shows that simply adding more convolution layers does not work. The second type is using the feature map to be transformed to predict the input channel scales. In the case of D₃, we perform average pooling over the last decoder feature map and pass the averaged feature vector through an MLP to predict the input channel scales. Again, there is no obvious improvement over row 1, demonstrating the necessity of using global information to guide feature transformation.

Relation Distillation: Based on row 5, we add relation distillation to encoder feature maps and observe slight performance improvement (row 8 v.s. row 5). We also try adding relation distillation to decoder feature maps, but the performance becomes worse (row 9 v.s. row 8). One possible reason is that the supervision from the final ground-truth image is sufficient to learn effective decoder features, whereas the relation distillation may disturb the learning of decoder fea-

set (only 311 images).

On ccHarmony, we also report the results of our method and baselines in Table 2. Again, our method outperforms the baselines and achieves significant improvement.

For the competitive baselines [36, 9, 22, 44] and our method, we show the harmonized results on iHarmony4 in Figure 3 and the results on ccHarmony in Figure 4. Our method can usually produce visually pleasant results which are closer to the ground-truth real images. More visualization results can be found in the supplementary.
Figure 3. From left to right, we show the composite image (foreground outlined in red), the harmonized results of iSSAM [36], CDTNet [9], Harmonizer [22], DCCF [44], our GiftNet, and the ground-truth on iHarmony4 [10] dataset.

Therefore, we only employ relation distillation on encoder feature maps. We also compare relation distillation with feature distillation. In particular, we replace relation distillation loss with feature distillation loss. The obtained performance in row 10 is not satisfactory, which demonstrates the superiority of relation distillation. As explained in Section 1, the whole feature maps contain much redundant and noisy information, which may enlarge the training difficulty of network. In contrast, relation distillation solely distills the key information for image harmonization task.

5.4. Real Composite Images

We additionally compare different methods on 100 real composite images provided by [9]. Since there are no ground-truth for these real composite images, we conduct user study to compare with competitive baselines [36, 9, 22, 44]. Following [9], given each composite image and its 5 harmonized results from different methods (4 baselines and our method), we can create image pairs by randomly choosing 2 images from 6 images (5 harmonized results and one composite image). Thus, we can construct 1500 image pairs based on 100 real composite images. We ask 50 users to watch an image pair each time and choose the more harmonious image, leading to 75,000 pairwise results. Then, we use the Bradley-Terry (B-T) model [3, 23] to calculate the global ranking of all methods, as reported in Table 4. Our method achieves the highest B-T score, which demonstrates the effectiveness of globally guided feature transformation.
and relation distillation. The visualization results on real composite images are left to the supplementary due to space limitation.

6. Conclusion

In this work, we have proposed globally guided feature transformation and relation distillation for image harmonization. We have also contributed a new dataset named ccHarmony, which provides a new perspective for harmonization dataset construction. Comprehensive experiments have demonstrated the superiority of our method.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 62076162), the Shanghai Municipal Science and Technology Major/Key Project, China (Grant No. 2021SHZDZX0102, Grant No. 2051100300).
References


[9] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. CVPR, 2022. 1, 2, 5, 6, 7, 8


[22] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rysson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. In ECCV, 2022. 1, 2, 5, 6, 7, 8


[27] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In CVPR, 2017. 2


[29] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In CVPR, 2021. 1, 2, 5, 6


[34] Xuqian Ren and Yifan Liu. Semantic-guided multi-mask image blending. In CVPR, 2022. 1, 2


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


[40] Jeya Maria Jose Valanarasu, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Jose Echevarria, Yinglan Ma, Zijun Wei, Kalyan Sunkavalli, and Vishal M Patel. Interactive portrait harmonization. ICLR, 2023. 2


[44] Ben Xue, Shenghui Ran, Quan Chen, Rongfei Jia, Binqiang Zhao, and Xing Tang. Dccf: Deep comprehensible color filter learning framework for high-resolution image harmonization. In ECCV, 2022. 2, 5, 6, 7, 8


