Deep Image Harmonization with Learnable Augmentation

Li Niu[*], Junyan Cao, Wenyan Cong, Liqing Zhang
Department of Computer Science and Engineering, MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
{ustcnewly,Joy_C1}@sjtu.edu.cn, wycong@utexas.edu, zhang-lq@cs.sjtu.edu.cn

Abstract

The goal of image harmonization is adjusting the foreground appearance in a composite image to make the whole image harmonious. To construct paired training images, existing datasets adopt different ways to adjust the illumination statistics of foregrounds of real images to produce synthetic composite images. However, different datasets have considerable domain gap and the performances on small-scale datasets are limited by insufficient training data. In this work, we explore learnable augmentation to enrich the illumination diversity of small-scale datasets for better harmonization performance. In particular, our designed SYthetic COmposite Network (SycoNet) takes in a real image with foreground mask and a random vector to learn suitable color transformation, which is applied to the foreground of this real image to produce a synthetic composite image. Comprehensive experiments demonstrate the effectiveness of our proposed learnable augmentation for image harmonization. The code of SycoNet is released at https://github.com/bcmi/SycoNet-Adaptive-Image-Harmonization.

1. Introduction

As a prevalent image editing operation, image composition [28] aims to cut the foreground from one image and paste it on another background image, making a realistic-looking composite image. Image composition plays a critical role in artistic creation, augmented reality, automatic advertising, and so on [37, 44]. However, the illumination statistics of foreground and background could be inconsistent due to distinct capture conditions or capture devices (*e.g.*, season, weather, time of the day, camera setting), which makes the resultant composite image unrealistic. To address this issue, image harmonization [5] targets at adjusting the illumination statistics of foreground to make it compatible with the background, leading to a harmonious composite image.

Training data-hungry deep learning models for image harmonization relies on abundant pairs of composite images and harmonious images. Nevertheless, it is extremely difficult and expensive to manually adjust the foreground of composite image to produce harmonious image. Therefore, recent works turn to an inverse manner, that is, adjusting the foreground of real image to produce a synthetic composite image, resulting in pairs of synthetic composite images and ground-truth harmonious real images. In this inverse manner, the work in [5] constructed four datasets (HCOCO, HFlickr, HAdobe5k, and Hday2night), which are collectively called iHarmony4. Despite similar construction pipeline, four datasets adopt different foreground adjustment approaches. In particular, HCOCO and HFlickr adopt traditional color transfer methods [31, 38, 11, 30] to adjust the foreground, while HAdobe5k (*resp.*, Hday2night) replaces the foreground with the counterpart retouched by different experts (*resp.*, captured at different times). Previous works usually train a deep image harmonization model based on the union of training sets from four datasets. However, the data distributions of different datasets are considerably different, which is caused by many factors (*e.g.*, image source, capture device, scene type, foreground adjustment approach). Following the terminology of domain adaptation [35, 29], **we treat each dataset as one domain and four datasets have large domain gap**. We observe that finetuning on different datasets can bring notable performance gain (see Section 4.2), but the performance is still limited by insufficient training data on small-scale datasets (*e.g.*, HFlickr, Hday2night).

In this work, we attempt to augment small-scale datasets with more synthetic composite images to enrich the illumination diversity, which may not be easily achieved by using the original foreground adjustment approach. Specifically, for HFlickr, we need to manually filter unqualified synthetic composite images [5], otherwise the performance would be significantly compromised (see Section 4.2). For Hday2night, it is almost impossible to recapture the same scene at different times. Therefore, we design an aug-

[*] Corresponding author.

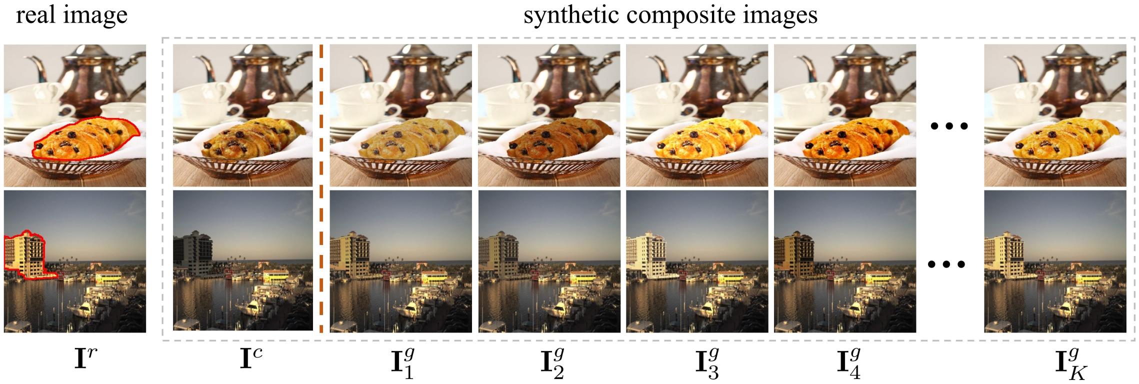


Figure 1: In the left two columns, we show a pair of real image $\mathbf{I}^r$ and original synthetic composite image $\mathbf{I}^c$ from HFlickr (*resp.*, Hday2night) dataset in the top (*resp.*, bottom) row, with the real foreground outlined in red. In the other columns, we show K augmented synthetic composite images $\{\mathbf{I}^g_k|_{k=1}^K\}$ generated by our SycoNet.

mentation network named SYthetic COmposite Network (SycoNet) to produce synthetic composite images automatically, which simulates the original foreground adjustment approach.

Our SycoNet takes in a real image with foreground mask and a random vector, generating a synthetic composite image with adjusted foreground. By sampling multiple random vectors, we can generate multiple synthetic composite images. As shown in Figure 2(a), we adopt a CNN encoder to extract feature from real image and foreground mask, which is concatenated with a random vector to produce suitable color transformation for the foreground. The color transformation is realized by a linear combination of basis LUTs [4]. The combined LUT is applied to the foreground to produce a synthetic composite image. Moreover, we establish a bijection between random vectors and original synthetic composite images in the dataset to ensure the quality and diversity of generated synthetic composite images. Concretely, we employ another CNN encoder to produce a latent code, which is expected to encode the necessary information required to transfer from real foreground to original composite foreground. Thus, when using this latent code to predict color transformation, we hope that the generated synthetic composite image can reconstruct the original synthetic composite image in the dataset.

After training SycoNet, we freeze the model parameters and integrate it with an existing image harmonization network, as shown in Figure 2(b). When training the image harmonization network, besides the original training pairs of synthetic composite images and real images, SycoNet produces extra synthetic composite images as augmented data. Both original and augmented synthetic composite images (see Figure 1) should be harmonized to approach ground-truth real images.

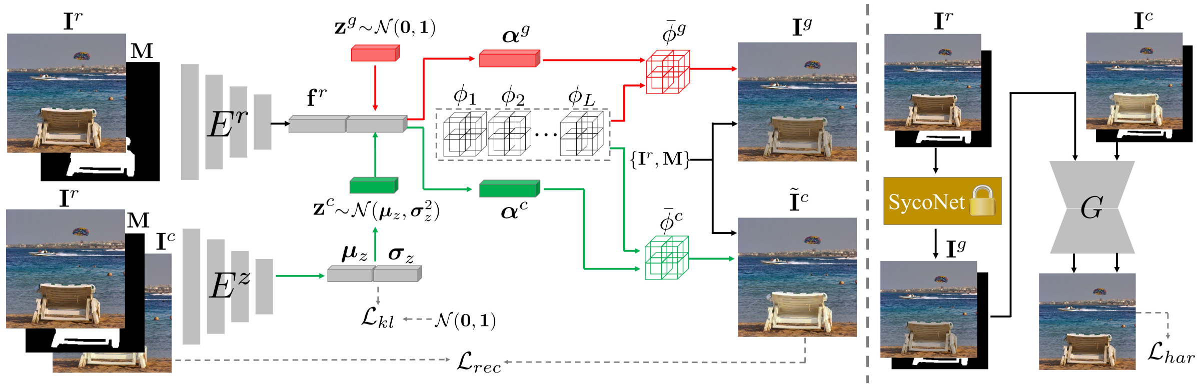
Our proposed learnable augmentation is helpful for adapting a pretrained image harmonization model to a new domain with limited data. Given a test image which we do not know which domain it belongs to, we can apply our learnable augmentation in the following two ways. 1) We use learnable augmentation to enhance the harmonization model of each domain. Given a test image, we can first predict its domain label using a domain classifier and apply the model of the corresponding domain (see Section 4.6). 2) We use learnable augmentation to enhance one unified harmonization model for all domains (see Section 4.7). The second option is more compact, at the cost of performance degradation.

We conduct experiments on iHarmony4, which demonstrates that our proposed learnable augmentation can significantly boost the harmonization performance. Our major contributions are summarized as follows: 1) We propose learnable augmentation to enrich the illumination diversity for image harmonization; 2) We design a novel augmentation network named SycoNet, which can automatically generate synthetic composite images by simulating the foreground adjustment in the original dataset; 3) Extensive experiments prove the effectiveness of our proposed learnable augmentation.

2. Related Work

2.1. Image Harmonization

Traditional image harmonization methods [2, 23, 40, 33] mainly leveraged traditional color transfer methods (*e.g.*, shifting and scaling, histogram matching) to align the color information between foreground and background. In [46], they designed a model to learn the color shifting matrix by using a discriminator to push the transformed image to be realistic. Recently, lots of deep image harmonization methods [5, 32, 16, 26, 19, 13, 1] have emerged, which usually



(a) Synthetic Composite Network (SycoNet)

(b) Image Harmonization

Figure 2: (a) Illustration of our augmentation network SycoNet. In the generation (*resp.*, reconstruction) branch indicated by red (*resp.*, green) arrows, we predict the color transformation LUT $\bar{\phi}^g$ (*resp.*, $\bar{\phi}^c$) to adjust the real foreground in $\mathbf{I}^r$, resulting in the synthetic composite image $\mathbf{I}^g$ (*resp.*, $\tilde{\mathbf{I}}^c$). (b) Image harmonization with augmented synthetic composite images $\mathbf{I}^g$ generated by fixed SycoNet (only generation branch). Both original $\mathbf{I}^c$ and augmented $\mathbf{I}^g$ are harmonized by the harmonization network G.

learn a mapping network from composite image to harmonized image. To name a few, [36] supplemented basic image harmonization network with extra semantic information. By treating different capture conditions as different domains, [5] proposed to pull close foreground domain and background domain via domain verification, while [3] used background domain code to guide the translation of foreground. [7] developed various types of attention blocks embedded in the image harmonization network. [26] borrowed the idea of adaptive instance normalization from style transfer to image harmonization, which is further extended in [15] by considering local style transfer. [14] decomposed a composite image to reflectance map and illumination map, followed by modulating the foreground illumination map. [19] constructed training pairs based on two crops from the same image and designed an image harmonization model requiring complete background image. [4, 20, 24, 39] merged color transformation into deep learning networks for efficient image harmonization.

Different from the above works, we explore learnable augmentation for image harmonization, which has never been studied before. Our designed augmentation network can be integrated with any image harmonization network.

2.2. Data Augmentation

Data augmentation targets at augmenting training set with new samples. Traditional data augmentation techniques (*e.g.*, crop, flip, affine transformation, cutout [8]) have been widely used in myriads of computer vision tasks. There are also some more advanced augmentation techniques [43, 41] which mix up training images and their labels. However, all the above augmentation techniques are non-learnable augmentation without learnable parameters. Our proposed learnable augmentation shares some similar thoughts with a research line called auto-augmentation. Specifically, AutoAugment [6] proposed to search the best data augmentation policy for a target dataset via reinforcement learning, which is accelerated by subsequent works [25, 34, 27] using different techniques (*e.g.*, efficient density matching, weight sharing strategy, bilevel optimization). PBA [18] proposed to learn augmentation policy schedule rather than a fixed policy. Instead of searching the optimal augmentation policy, we predict proper color transformation to generate augmented data.

This work makes the first attempt at learnable augmentation for the image harmonization task, in which we generate more synthetic composite images for a real image to construct more training pairs. **Our augmentation strategy focuses on enriching the illumination diversity of training set, which is complementary with other data augmentation techniques (*e.g.*, crop, flip, more foregrounds).** We show that enriching the illumination diversity of training set can greatly enhance the image harmonization performance.

3. Our Method

We suppose that the training set of an image harmonization dataset contains N pairs of synthetic composite images and real images, *i.e.*, $\mathcal{S} = \{(\mathbf{I}^c_n, \mathbf{I}^r_n)|_{n=1}^N\}$, in which $\mathbf{I}^c_n$ (*resp.*, $\mathbf{I}^r_n$) is the n-th synthetic composite image (*resp.*, real image). In the first stage, we train an augmentation network on the training set, which can generate multiple synthetic composite images $\{\mathbf{I}^g_{n,k}|_k\}$ for a real image $\mathbf{I}^r_n$. In the second stage, we integrate the augmentation network into an existing image harmonization network G and fix the model parameters of augmentation network. When training the image harmonization network, the augmentation network dynamically generates augmented synthetic composite images $\{\mathbf{I}^g_{n,k}|_{n=1,k}^{N}\}$ to supplement the original training set $\mathcal{S}$, aiming to train a better image harmonization model. Next, we will introduce the first stage in Section 3.1 and the second stage in Section 3.2.

3.1. Synthetic Composite Network

We refer to our augmentation network as SYnthetic COmposite Network (SycoNet), which can generate multiple synthetic composite images given a real image. Our SycoNet consists of a generation branch and a reconstruction branch, which will be detailed separately. In this section, we omit the subscript n for brevity. Besides, the foreground of $\mathbf{I}^r$ (*resp.*, $\mathbf{I}^c$, $\mathbf{I}^g$) is denoted as $\mathbf{F}^r$ (*resp.*, $\mathbf{F}^c$, $\mathbf{F}^g$).

3.1.1 Generation Branch

In the generation branch, we produce suitable color transformation function $h(\cdot)$ for the foreground $\mathbf{F}^r$ of real image $\mathbf{I}^r$, in which $h(\cdot)$ could convert the source color value $\mathbf{v}_{src}$ in $\mathbf{F}^r$ to a target color value $\mathbf{v}_{tgt} = h(\mathbf{v}_{src})$. The color transformation function $h(\cdot)$ can be realized in various forms. In this work, we opt for look-up table (LUT), which has been widely used in a variety of computer vision tasks [9, 12, 10, 42, 4]. Briefly speaking, a look-up table (LUT) is a 3D lattice in the RGB color space with each dimension corresponding to one color channel (*e.g.*, red). An LUT has $(B + 1)^3$ entries by uniformly slicing the color space into B bins in each dimension, where B is set as 16 following [42, 4]. Each entry in the LUT has an indexing color and its output color. Given a source color value $\mathbf{v}_{src}$, its target color value $\mathbf{v}_{tgt}$ could be obtained by looking up its eight nearest indexing colors in the LUT and performing trilinear interpolation based on their output colors. More technical details of LUT can be found in [42, 4].

Following [42, 4], we learn a group of L basis LUTs $\{\phi_l|_{l=1}^L\}$ shared among all images and predict image-specific combination coefficients of basis LUTs for each image. In this way, shared basis LUTs are combined adaptively to form image-specific LUT. In [42, 4], L basis LUTs are initialized as one identity map and $L-1$ zero maps. Besides, there is no constraint for the combination coefficients, that is, the coefficient can be either positive or negative. The $L-1$ LUTs initialized with zero maps actually function as residual LUTs [4]. However, during our experiments, we observe that the learnt residual LUTs are prone to be similar with each other, which severely limits the representation ability of combined LUT. Therefore, we modify the LUT initialization and combination strategy. Precisely, we initialize L basis LUTs with one identity LUT and $L-1$ representative LUTs (we collect 100 LUTs from Internet and perform clustering to get $L-1$ cluster centers). All L basis LUTs are updated during training. Moreover, the combination coefficients are softmax normalized, so that all coefficients are positive and sum up to one. The comparison results demonstrate the advantage of our LUT initialization and combination strategy (see Table 2).

After introducing the color transformation function $h(\cdot)$ realized in the form of LUT, we describe the network architecture of generation branch. Given a real image $\mathbf{I}^r$ and its foreground mask $\mathbf{M}$, we concatenate them and feed into an encoder E^r (*e.g.*, ResNet18 [17]) to produce a feature vector $\mathbf{f}^r$. For each real image, we hope to generate multiple synthetic composite images instead of a single deterministic one. As a common approach to support stochastic sampling, we sample a random vector $\mathbf{z}^g$ from unit Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and concatenate it with $\mathbf{f}^r$. Then, the concatenation $[\mathbf{f}^r, \mathbf{z}^g]$ passes through one fully-connected (FC) layer followed by softmax normalization to produce the combination coefficients $\boldsymbol{\alpha}^g$. Given the learnt basis LUTs $\{\phi_l|_{l=1}^L\}$, the combined LUT is $\bar{\phi}^g = \sum_{l=1}^L \alpha_l^g \phi_l$, in which α_l^g is the l-th element in $\boldsymbol{\alpha}^g$. Finally, we apply $\bar{\phi}^g$ to $\mathbf{F}^r$ and get the transformed foreground $\mathbf{F}^g$, which is combined with original background to compose a synthetic composite image $\mathbf{I}^g$.

3.1.2 Reconstruction Branch

Note that the generation branch ignores the original synthetic composite images $\mathbf{I}^c$ in the training set $\mathcal{S}$ and lacks supervision for the generated synthetic composite images $\mathbf{I}^g$. Hence, there is no guarantee for the plausibility of generated synthetic composite images. In the reconstruction branch, we aim to ensure the quality and diversity of generated synthetic composite images, by reconstructing original synthetic composite image $\mathbf{I}^c$ from real image $\mathbf{I}^r$.

Given the original synthetic composite image $\mathbf{I}^c$ for real image $\mathbf{I}^r$, we deliver the concatenation of $\mathbf{I}^c$, $\mathbf{I}^r$, and $\mathbf{M}$ to an encoder E^z (*e.g.*, ResNet18 [17]). E^z produces $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$, based on which the latent code $\mathbf{z}^c$ could be sampled from Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2)$. By using reparameterization trick [21], we obtain $\mathbf{z}^c = \boldsymbol{\mu}_z + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_z$, where $\boldsymbol{\epsilon}$ is a random vector sampled from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and $\odot$ means element-wise product. $\mathbf{z}^c$ is expected to encode the requisite information to generate $\mathbf{I}^c$ conditioned on $\mathbf{I}^r$. Analogous to the generation branch, we concatenate $\mathbf{z}^c$ with $\mathbf{f}^r$, and use the same FC layer to produce the combination coefficients $\boldsymbol{\alpha}^c$ of basis LUTs. Then, we apply the combined LUT $\bar{\phi}^c = \sum_{l=1}^L \alpha_l^c \phi_l$ to the real foreground $\mathbf{F}^r$ and get a synthetic composite image $\tilde{\mathbf{I}}^c$, which is pushed towards $\mathbf{I}^c$ using the reconstruction loss $\mathcal{L}_{rec} = \|\tilde{\mathbf{I}}^c - \mathbf{I}^c\|_1$. In the meanwhile, we use KL divergence loss [22] $\mathcal{L}_{kl} = KL[\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2)||\mathcal{N}(\mathbf{0}, \mathbf{1})]$ to enforce

$\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2)$ to approach $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

The above reconstruction process establishes a bijection between latent code and synthetic composite image conditioned on real image, which enables generating qualified and diverse synthetic composite images by sampling $\mathbf{z}^g \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. The reasons are explained as follows. In terms of quality, $\mathbf{I}^r$ and $\mathbf{z}^c$ can produce one plausible color transformation (*i.e.*, from $\mathbf{I}^r$ to $\mathbf{I}^c$) due to the bijection. We assume that such knowledge can be transferred across the joint space of real image and latent code, so that $\mathbf{I}^r$ and other sampled $\mathbf{z}^g$ can produce other plausible color transformations. More specifically, we denote that $\mathbf{I}^r_i$ and $\mathbf{z}^c_i$ can reconstruct plausible composite image $\mathbf{I}^c_i$ for $i = 1, \ldots, N$. Assuming that $\{\mathbf{z}^c_i|_{i=1}^N\}$ are transferrable across real images $\{\mathbf{I}^r_i|_{i=1}^N\}$, $\mathbf{I}^r_i$ and $\mathbf{z}^c_j$ for $j \neq i$ could also produce plausible composite images. Besides, $\{\mathbf{z}^c_i|_{i=1}^N\}$ are sampled from Gaussian distributions close to $\mathcal{N}(\mathbf{0}, \mathbf{1})$, so $\mathbf{I}^r_i$ and $\mathbf{z}^g \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ could also produce plausible composite images. In terms of diversity, the bijection can prevent two latent codes from producing the same synthetic composite image. If two latent codes produce the same synthetic composite image, one synthetic composite image cannot be mapped back to two different latent codes, which would violate the bijection. The experiments in Table 2 verify that the reconstruction branch can effectively promote the quality and diversity of generated synthetic composite images.

So far, the total loss function can be summarized as

$$\mathcal{L}_{syco} = \mathcal{L}_{kl} + \mathcal{L}_{rec}. \tag{1}$$

After training the augmentation network on the training set $\mathcal{S}$ using Eqn. (1), the obtained augmentation network could generate more synthetic composite images to augment the original training set. Formally, given the n-th real training image $\mathbf{I}^r_n$, we can produce K synthetic composite images $\{\mathbf{I}^g_{n,k}|_{k=1}^K\}$ by sampling K random vectors $\{\mathbf{z}^g_k|_{k=1}^K\}$. Recall that different datasets can be treated as different domains with different data distributions, it would be beneficial to train the augmentation network on each dataset separately (see Section 4.2). When training on a specific dataset, the reconstruction branch could simulate the foreground adjustment process of this dataset, and the generation branch could produce synthetic composite images with close data distribution to the original ones in this dataset.

3.2. Dynamic Augmentation

With the trained SycoNet in Section 3.1, we can integrate its generation branch with any existing image harmonization network G for dynamic augmentation, as illustrated in Figure 2. Normally, G is trained on the training set $\mathcal{S} = \{(\mathbf{I}^c_n, \mathbf{I}^r_n)|_{n=1}^N\}$, by harmonizing $\mathbf{I}^c_n$ to be close to $\mathbf{I}^r_n$. In our training process, for the n-th real training image $\mathbf{I}^r_n$ in the t-th training iteration, we use fixed SycoNet to produce a synthetic composite image $\mathbf{I}^g_{n,t}$ with sampled $\mathbf{z}^g_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. We denote the harmonization results of $\mathbf{I}^c_n$ and $\mathbf{I}^g_{n,t}$ as $\tilde{\mathbf{I}}^c_n$ and $\tilde{\mathbf{I}}^g_{n,t}$ respectively, both of which should be close to $\mathbf{I}^r_n$. The loss function used in original image harmonization network G is represented by $\mathcal{L}_{har}$ and may be different in various image harmonization methods. So we omit the details of $\mathcal{L}_{har}$ here. Then, the training loss can be written as

$$\mathcal{L}_{har} = \sum_{n=1}^{N} \sum_{t=1}^{T} \mathcal{L}_{har}(\tilde{\mathbf{I}}^c_n, \mathbf{I}^r_n) + \mathcal{L}_{har}(\tilde{\mathbf{I}}^g_{n,t}, \mathbf{I}^r_n), \tag{2}$$

in which T is the number of training iterations. Since the augmented composite images dynamically vary during the training procedure, we refer to this augmentation strategy as dynamic augmentation. We also compare this augmentation strategy with static augmentation, that is, generating adequate augmented composite images beforehand and merging them into the original training set. We find that dynamic augmentation is more elegant and effective than static augmentation (see Section 4.5).

Note that SycoNet is only used in the training stage. During inference, we only use the image harmonization network G, without introducing extra computational cost.

4. Experiments

4.1. Datasets and Implementation Details

We conduct experiments on iHarmony4 [5] which consists of four datasets: HCOCO, HFlickr, HAdobe5k, and Hday2night. HCOCO (*resp.*, HAdobe5k, HFlickr, and Hday2night) has 38545 (*resp.*, 19437, 7449, and 311) training images and 4283 (*resp.*, 2160, 828, and 133) test images. We mainly use HFlickr and Hday2night, which are two relatively small datasets, because the advantage of data augmentation could be exhibited more clearly on small-scale datasets. We also conduct similar experiments on the other two datasets (HCOCO and HAdobe5k) and observe that the performance gain decreases as the scale of training data increases, which is left to Supplementary.

In the augmentation network, we adopt ResNet18 [17] as the encoders E^r and E^z. We set the dimension of latent code $\mathbf{z}$ as $d_z = 32$ and the number of basis LUTs as $L = 20$. For the harmonization network G, since our augmentation network can cooperate with any existing image harmonization network, we take RainNet [26] and iS^{2}AM [32] as two examples to show the effectiveness of learnable augmentation in Section 4.2. In the other sections, we adopt iS^{2}AM as G by default considering its simplicity and effectiveness. Our network is implemented using Pytorch 1.7.0 and trained using Adam optimizer with learning rate of $1e{-}4$ on ubuntu 18.04 LTS operation system, with 32GB memory, Intel Core i7-9700K CPU, and two GeForce GTX 2080 Ti GPUs. We adopt MSE, foreground MSE (fMSE),

	Train	Aug	Hday2night			HFlickr		
			MSE↓	fMSE↓	fSSIM ↑	MSE↓	fMSE↓	fSSIM ↑
1	-	-	40.59	591.07	0.7726	69.68	443.63	0.9108
2	ft(o)	-	38.08	567.21	0.7763	64.07	411.12	0.9151
3	ft(oa)	CT	39.62	575.35	0.7672	73.00	461.90	0.9072
4	ft(oa)	LUT	42.97	630.55	0.7661	86.06	559.67	0.8948
5	ft(oa)	Ours	36.49	547.63	0.7766	63.53	407.53	0.9164
6	ft(a)	Ours(ft)	42.79	572.53	0.7757	68.05	431.20	0.9149
7	ft(oa)	Ours(ft)	**34.44**	**517.00**	**0.7844**	**58.46**	**385.43**	**0.9176**

(a) The results using iS^{2}AM [32] as the harmonization network.

	Train	Aug	Hday2night			HFlickr		
			MSE↓	fMSE↓	fSSIM ↑	MSE↓	fMSE↓	fSSIM ↑
1	-	-	47.24	852.12	0.7444	117.59	751.13	0.8573
2	ft(o)	-	44.10	785.79	0.7471	107.83	722.97	0.8629
3	ft(oa)	CT	51.10	950.66	0.7177	105.69	708.03	0.8685
4	ft(oa)	LUT	50.44	937.03	0.7349	117.72	788.21	0.8533
5	ft(oa)	Ours	39.86	717.05	0.7531	102.57	678.18	0.8731
6	ft(a)	Ours(ft)	44.47	792.99	0.6758	109.94	733.49	0.8614
7	ft(oa)	Ours(ft)	**37.28**	**643.78**	**0.7662**	**96.01**	**634.55**	**0.8780**

(b) The results using RainNet [26] as the harmonization network.

Table 1: In "Train" column, "ft" means finetuning the harmonization model on Hday2night/HFlickr, and "o" (*resp.*, "a") represents original (*resp.*, augmented) training images. In "Aug" column, "LUT" (*resp.*, "CT") means using random LUT (*resp.*, traditional color transfer methods) for data augmentation. "Ours" means using our SycoNet for data augmentation, and "Ours(ft)" means finetuning SycoNet on Hday2night/HFlickr. The best results are highlighted in boldface.

foreground SSIM (fSSIM) as evaluation metrics for harmonization performance, in which MSE is calculated based on the whole image while fMSE and fSSIM are calculated within the foreground region.

4.2. Effectiveness of Our Learnable Augmentation

In Table 1a, we first report the results of iS^{2}AM model trained on the whole iHarmony4 training set (row 1). Then, we finetune the harmonization model on the training set of Hday2night or HFlickr ("ft(o)"). The attained results in row 2 are significantly better than those in row 1, which verifies the domain gap between different datasets. Next, the following results are all obtained by finetuning the harmonization model on Hday2night or HFlickr, with both original training data and augmented training data ("ft(oa)") or only augmented training data ("ft(a)").

We first try two simple augmentation methods: traditional color transfer methods [31, 38, 11, 30] ("CT" in row 3) and random LUT ("LUT" in row 4). For "CT", following [5], for each real image in each training iteration, we randomly choose one reference image from ADE20k dataset [45] and one of the color transfer methods [31, 38, 11, 30] to perform color transfer on its foreground. The attained results in row 3 are worse than those in row 2. As claimed in [5], the obtained synthetic composite images without deliberate filtering may have noticeable artifacts or unreasonable albedo change, which would adversely affect the effectiveness of augmentation. For "LUT", we gather 100 LUTs from Internet and randomly choose one LUT for each real image in each training iteration. The obtained results in row 4 become even worse.

Then, we train our augmentation network SycoNet on the whole iHarmony4 training set and perform dynamic augmentation when finetuning the harmonization model. The obtained results in row 5 outperform two simple augmentation methods (row 3 and row 4), and generally outperform those without augmentation (row 2), which shows the superiority of our augmentation network. Next, we finetune SycoNet on Hday2night and HFlickr respectively, and use finetuned SycoNet for dynamic augmentation. We report the results in the last row (row 7), based on which the finetuned SycoNet performs more favorably (row 7 *v.s.* row 5). Due to the large domain gap between different datasets, the finetuned SycoNet can better approximate the foreground adjustment process of each dataset, and thus generate more suitable synthetic composite images for each dataset. Finally, we report the results (row 6) obtained by only using augmented data and discarding original training data. The

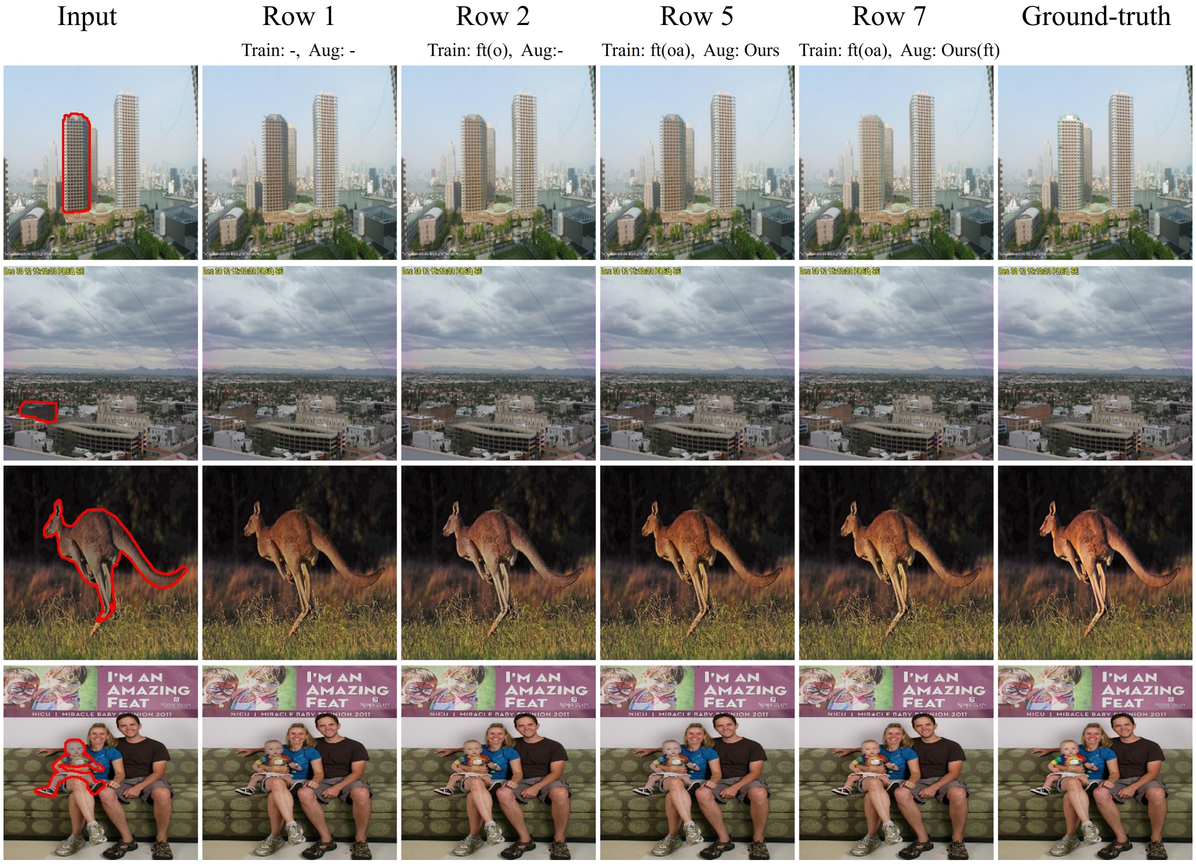


Figure 3: The leftmost (*resp.*, rightmost) column is the input composite image (*resp.*, ground-truth real image), in which the foreground in the input image is outlined in red. The rest columns show the harmonized results of row 1, 2, 5, 7 in Table 1a. *Train* and *Aug* below the row index means the corresponding training setting and augmentation setting as in Table 1. The first (*resp.*, last) two rows are from Hday2night (*resp.*, HFlickr) dataset.

comparison between row 6 and row 2 implies that the generated synthetic composite images still have certain gap with the original synthetic composite images. However, jointly using them can achieve significant improvement (row 7 *v.s.* row 2). Another observation is that the improvement of our learnable augmentation (row 7 *v.s.* row 2) on Hday2night dataset is more significant than that on HFlickr dataset, which confirms our conjecture that data augmentation is more effective on small-scale datasets.

In Table 1b, we report the results based on RainNet and have similar observations on the relation between different rows. For example, simple color augmentation (row 3, row 4) generally brings no performance gain, except when the performance without augmentation is very poor (*e.g.*, "CT" slightly improves RainNet on HFlickr). A common SycoNet can improve the results (row 5 *v.s.* row 2) and the finetuned SycoNet can achieve further improvement (row 7 *v.s.* row 5). The improvement on Hday2night is more notable than that on HFlickr.

4.3. Qualitative Results

For qualitative comparison, we show the visualization results of row 1, 2, 5, 7 in Table 1a on two datasets in Figure 3. From row 1 to row 7, the results are overall getting better. The results obtained by using our finetuned augmentation network (row 7) are more visually appealing and closer to the ground-truth real images, which proves that it is useful to finetune the harmonization model with the aid of finetuned augmentation network. More visualization results can be found in Supplementary.

4.4. Ablation Studies on Augmentation Network

By taking Hday2night as an example, we conduct ablation studies on our augmentation network SycoNet in Table 2. Besides the harmonization metrics used in Table 1, we also evaluate the diversity of generated composite images, which is referred to as "Div" in Table 2. In particular, for each real training image, we randomly sample $\mathbf{z}^g$ for 10 times to produce 10 synthetic composite images. Then, we calculate fMSE between each pair of composite images and compute the average over all pairs, and then compute the average over all real training images. For MSE, fMSE, fSSIM, the experimental setting is the same as row 7 in Table 1a. In detail, we finetune the harmonization model on each dataset using data augmentation, with the augmented images generated by different variants of our augmentation network. We include the results obtained by using our full-fledged augmentation network (row 7 in Table 1a) as "Ours" for comparison.

First, we delete the reconstruction branch by removing E^z and the associated inputs/outputs. After removal, there is no supervision for the generated synthetic composite image $\mathbf{I}^g$. Therefore, we add an adversarial loss to make $\mathbf{I}^g$ indistinguishable from original synthetic composite images $\mathbf{I}^c$ in the training set. We observe that the produced combination coefficients collapse to an one-hot vector and the generated synthetic composite images lack diversity, which is known as mode collapse issue [47]. As reported in row 1, "Div" is close to zero and the effect of augmentation is negligible compared with row 2 in Table 1a. We also try removing $\mathbf{I}^r$ from the input of E^z, because only using $\{\mathbf{I}^c, \mathbf{M}\}$ could also provide the target illumination information of composite foreground $\mathbf{F}^c$. The results become worse than "Ours" (row 2 *v.s.* row 6), which shows that it would be better to use both $\mathbf{I}^r$ and $\mathbf{I}^c$ to encode the information required to transfer from $\mathbf{F}^r$ to $\mathbf{F}^c$. Considering that the random vector $\mathbf{z}$ could be appended to different locations in E^z, we try an alternative location, *i.e.*, spatially replicating $\mathbf{z}$ and appending it to the input of E^z. The attained results become slightly worse than "Ours" (row 3 *v.s.* row 6). Recall that we make some modifications about the LUT design compared with [42, 4]. Here, we change the LUT design to be the same as in [42, 4] and report the results in row 4. As mentioned in Section 3.1.1, the learnt residual LUTs are prone to be similar with each other, so the diversity of generated composite images is degraded. The harmonization performance is also worse than "Ours", indicating that our LUT design is more effective. We also try sampling $\boldsymbol{z}^c$ to generate composite images and report the results in row 6. Sampling $\boldsymbol{z}^c$ can only reconstruct original synthetic composite images (see Section 3.1.2) and cannot achieve the goal of augmentation, so the obtained results are merely comparable with those without using augmentation.

4.5. Comparing Dynamic and Static Augmentation

As described in Section 3.2, we dynamically generate augmented composite images during the training procedure, which is dubbed as dynamic augmentation. There exists another straightforward augmentation strategy: generating augmented composite images beforehand and merging them into the original training set, which is dubbed as static augmentation. We compare dynamic augmentation with static augmentation in Supplementary.

	Method	MSE↓	fMSE↓	fSSIM ↑	Div↑
1	w/o rec	40.72	569.49	0.7754	0.01
2	w/o $\mathbf{I}^r$	37.64	554.48	0.7760	1094.57
3	move $\mathbf{z}$	34.83	525.54	0.7834	987.47
4	LUTv2	37.10	541.56	0.7810	847.01
5	$\mathbf{z}^c$	37.59	556.44	0.7758	8.37
6	Ours	34.44	517.00	0.7844	1132.71

Table 2: Ablation studies of our data augmentation network SycoNet on Hday2night. "Div" measures the diversity of augmented images. "w/o rec" means removing the reconstruction branch. "w/o $\mathbf{I}^r$" means removing $\mathbf{I}^r$ from E^z input. "move $\mathbf{z}$" means moving $\mathbf{z}$ to E^z input. "LUTv2" means using the LUT design in [42, 4]. $\mathbf{z}^c$ means using $\mathbf{z}^c \sim \mathcal{N}(\mu_z, \sigma_z^2)$ instead of $\mathbf{z}^g \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ to generate $\mathbf{I}^g$.

4.6. Evaluation on Real Composite Images

Following previous image harmonization works [36, 5, 26, 32], we also evaluate our method on 199 real composite images from [36, 4] (99 images from [36] and 100 images from [4]). As there are no ground-truth harmonious images for real composite images, we conduct user study for comparison. The detailed user study results and visualization results are left to Supplementary.

4.7. One Unified Model to Rule All Domains

We use SycoNet pretrained on iHarmony4 to generate augmented images for the whole iHarmony4 training set. Then, we train one unified model on the augmented training set, which is applied to all four domains. The detailed results and analyses are left to Supplementary.

5. Conclusion

In this work, we have proposed learnable augmentation to enrich the illumination diversity of image harmonization datasets. Specifically, we design a novel augmentation network named SycoNet, which can produce more synthetic composite images for a real image. Our SycoNet can be integrated into any existing image harmonization network for dynamic augmentation. Comprehensive experiments show that our proposed learnable augmentation can significantly boost the harmonization performance.

Acknowledgement

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

References

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

[2] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. *ACM Transactions on Graphics*, 25(3):624–630, 2006. 2

[3] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. BargainNet: Background-guided domain translation for image harmonization. In *ICME*, 2021. 3

[4] 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. 2, 3, 4, 8

[5] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020. 1, 2, 3, 5, 6, 8

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay K Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019. 3

[7] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Trans. Image Process.*, 29:4759–4771, 2020. 3

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[9] M. Elad, B. Matalon, and M. Zibulevsky. Image denoising with shrinkage and redundant representations. In *CVPR*, 2006. 4

[10] Bo Fan, Fugen Zhou, and Hongbin Han. Medical image enhancement based on modified lut-mapping derivative and multi-scale layer contrast modification. In *International Congress on Image and Signal Processing*, 2011. 4

[11] Ulrich Fecker, Marcus Barkowsky, and André Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1258–1267, 2008. 1, 6

[12] S. Fujita, N. Fukushima, M. Kimura, and Y. Ishibashi. Randomized redundant dct: efficient denoising by using random subsampling of dct patches. In *Siggraph Asia Technical Briefs*, 2015. 4

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

[14] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *CVPR*, 2021. 3

[15] Yucheng Hang, Bin Xia, Wenming Yang, and Qingmin Liao. Scs-co: Self-consistent style contrastive learning for image harmonization. In *CVPR*, 2022. 3

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

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[18] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *ICML*, 2019. 3

[19] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. Ssh: A self-supervised framework for image harmonization. In *ICCV*, 2021. 2, 3

[20] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *ECCV*, 2022. 3

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 4

[22] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

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

[24] Jingtang Liang, Xiaodong Cun, and Chi-Man Pun. Spatial-separated curve rendering network for efficient and high-resolution image harmonization. In *ECCV*, 2022. 3

[25] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, 2019. 3

[26] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *CVPR*, 2021. 2, 3, 5, 6, 8

[27] Saypraseuth Mounsaveng, Issam Laradji, Ismail Ben Ayed, David Vazquez, and Marco Pedersoli. Learning data augmentation with online bilevel optimization for image classification. In *WACV*, 2021. 3

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

[29] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 1

[30] François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007. 1, 6

[31] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 1, 6

[32] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. In *WACV*, 2021. 2, 5, 6, 8

[33] Kalyan Sunkavalli, Micah K. Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics*, 29(4):125:1–125:10, 2010. 2

[34] Keyu Tian, Chen Lin, Ming Sun, Luping Zhou, Junjie Yan, and Wanli Ouyang. Improving auto-augment via augmentation-wise weight sharing. *NeurIPS*, 2020. 3

[35] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1

[36] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017. 3, 8

[37] Shuchen Weng, Wenbo Li, Dawei Li, Hongxia Jin, and Boxin Shi. MISC: Multi-condition injection and spatially-adaptive compositing for conditional person image synthesis. In *CVPR*, 2020. 1

[38] Xuezhong Xiao and Lizhuang Ma. Color transfer in correlated color space. In *VRCAI*, 2006. 1, 6

[39] 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. 3

[40] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly E. Rushmeier. Understanding and improving the realism of image composites. *ACM Transactions on Graphics*, 31(4):84:1–84:10, 2012. 2

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3

[42] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning Image-adaptive 3D Lookup Tables for High Performance Photo Enhancement in Real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):1–1, 2020. 4, 8

[43] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[44] Song-Hai Zhang, Zhengping Zhou, Bin Liu, Xi Dong, and Peter Hall. What and where: A context-based recommendation system for object insertion. *Computational Visual Media*, 6(1):79–93, 2020. 1

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 6

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

[47] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *NIPS*, 30, 2017. 8