

Semantic-guided Multi-Mask Image Harmonization

Xuqian Ren¹[0000-0002-3811-0235] and Yifan Liu²[0000-0002-2746-8186]

¹ Beijing Institute of Technology

² University of Adelaide

Abstract. Previous harmonization methods focus on adjusting one in-harmonious region in an image based on an input mask. They may face problems when dealing with different perturbations on different semantic regions without available input masks. To deal with the problem that one image has been pasted with several foregrounds coming from different images and needs to harmonize them towards different domain directions without any mask as input, we propose a new semantic-guided multi-mask image harmonization task. Different from the previous single-mask image harmonization task, each inharmonious image is perturbed with different methods according to the semantic segmentation masks. Two challenging benchmarks, HScene and HLIP, are constructed based on 150 and 19 semantic classes, respectively. Furthermore, previous baselines focus on regressing the exact value for each pixel of the harmonized images. The generated results are in the ‘black box’ and cannot be edited. In this work, we propose a novel way to edit the inharmonious images by predicting a series of operator masks. The masks indicate the level and the position to apply a certain image editing operation, which could be the brightness, the saturation, and the color in a specific dimension. The operator masks provide more flexibility for users to edit the image further. Extensive experiments verify that the operator mask-based network can further improve those state-of-the-art methods which directly regress RGB images when the perturbations are structural. Experiments have been conducted on our constructed benchmarks to verify that our proposed operator mask-based framework can locate and modify the inharmonious regions in more complex scenes. Our code and models are available at <https://github.com/XuqianRen/Semantic-guided-Multi-mask-Image-Harmonization.git>.

Keywords: Multi-Mask Image Harmonization, GAN, Image Editing

1 Introduction

Image editing techniques play a more crucial role in our daily life. They have been widely used in advertisement propaganda, social media sharing, and digital entertainment. Furthermore, with the rapid expansion of electronic devices and image processing applications, such as PhotoShop and MeituPic, image composition has become a more accessible technique. However, without professional

photo-editing experience, a fake composited image will have lower evaluation credibility due to its inharmonious color, texture, or illumination. Thus, image harmonization is an imperative process to improve composite image quality.

Previous image harmonization aims to improve the quality of image composition by matching the appearance between a pre-defined foreground region and background to make the whole composited image more realistic. The adjusted appearance can be color, brightness, contrast, etc. Deep-learning based methods [3, 2] usually handle this task as an image-to-image translation task [8, 14]. These methods have two limitations. First, these networks focus on the problem that one composite image only has one inharmonious region to change and do not consider the situation that the composite image may have multi-inharmonious regions with different kinds of perturbations but without masks to indicate. In a real application, the user may cut out the portraits from different photos and paste them on the same background. Second, the generated result is not explainable and editable, which is not flexible for users to further edit if they do not like the results generated by the automation output.

To solve these drawbacks, we propose a new setting named Semantic-guided Multi-Mask Image Harmonization(Sg-MMH), which aims to adjust the inharmonious regions according to their semantics. We construct two new benchmarks, named HScene and HLIP, which originated from public semantic segmentation datasets ADE20K [29] and LIP [5], separately. For each image, we randomly select several masks with different semantics to apply the perturbations. For the HScene, we focus on harmonizing various things and stuff in natural scenes, such as houses, lawns, and animals. And for the HLIP, we pay attention to adjusting the body parts and clothes of a person.

To realize the perturbation, we use Instagram filters, LAB perturb methods, blur perturbations to enrich disturbance scenarios, and apply different perturb functions to different regions randomly. That means the framework needs to harmonize the foreground regions towards different domains, which is a more challenging task.

Furthermore, we propose a new framework that can generate Operator Masks. Different from previous works that directly regress RGB images, we choose to generate operator masks for some pre-defined operators. These operator masks can simulate the manipulation functions in the image processing software. In this work, we define six feasible operator masks, and we multiply and add every two masks to each L, A, and B channel of inharmonious input images. Hence, each mask can be seen as an operation that can change the illumination, contrast, or color. The mask location indicates where to apply the operation, and the value of each pixel can be seen as the level each function is acted. The final generated mask is also editable and explainable, and it can locate the inharmonious region automatically. Our framework can be applied to any off-the-shelf backbones to adapt them to our task and make their output editable.

We summarize our contributions in three-fold:

- We propose a new setting that aims to simulate the situation that the foregrounds come from different images and need to be harmonized towards

different domains. We propose two benchmarks, HScene and HLIP, to represent the scenery and person harmonization task separately. To enrich the training set, we perturb several foreground images with multiple semantic-guided masks on the fly.

- We propose a new framework to generate operator masks to simulate image editing operations in image processing software to harmonize images instead of generating each pixel separately in the training process. The masks indicate the level and the position to apply a certain operation, which could be the brightness, the saturation, and the color in a specific dimension.
- Quantitative and qualitative comparisons demonstrate that the proposed operator mask-based network can have a reasonable result on our multi-mask image harmonization benchmarks. It can also provide more explainable and editable outputs for further change.

2 Related Works

Image harmonization: Traditional image harmonization methods concentrated on learning and adjusting statistical appearance between hand-crafted heuristics appearance features, such as [18, 20, 1, 10, 17, 22, 21]. Some of them have been adapted to image editing software. However, these methods are not very reliable when dealing with more complex situations. Recently several CNN networks started to use learning methods to make the harmonized image more reasonable. Zhu *et al.* [30] trained a discriminative model to rank the composite images according to their realism level and optimized the model by maximizing the output visual realism score. DIH [23] proposed the first end-to-end CNN network to produce harmonized images with auxiliary semantics prediction.

Since the remarkable effect of GAN in image-to-image translation [8, 15, 31], some GAN frameworks have also been proposed to facilitate image harmonization. *S²AM* [4] considered the harmonization task as just changing the foreground appearance by keeping semantic information. So it used channel-wise and spatial-wise attention mechanisms to focus on the appearance changes in the foreground mask region. DoveNet [3] further conceived the image harmonization task as a domain transfer problem and hypothesized that the foreground and background of an inharmonious image are captured in different conditions. So, they chose to select the foreground objects in the datasets and perturbed their color or illumination to simulate an inharmonious composite, and trained the network to regress close to the original input image in a GAN framework with a domain verification discriminator. This is a good start to deeply researching harmonization tasks with a large dataset. Other methods also re-defined the image harmonization task to make this task more practical. BarginNet [2] formulated image harmonization as background-guided domain translation. They used a domain code extractor to extract foreground and background domain code and use background domain code to guide the foreground domain during translation. RainNet [13] treated image harmonization as a style transfer problem and extracted background style information, and applied it to the foreground. D-HT [6]

used the strong power of Transformer to model long-range context information to perceive information for better harmonization. These methods all consider that only one foreground inharmonious region needs to be edited and directly generate and regress each pixel towards the target, which lacks interpretability. Instead, in our work, we resolve the image harmonization task using preset image editing operations to edit the inharmonious images. Specifically, we predict a series of operator masks to be applied to the inharmonious input. To tackle our proposed setting, we construct two benchmarks with semantic-guided perturbed regions that need to be adjusted in one composite image.

Photo retouch: Photo Retouch aims to learn an automatic network to correct a raw image to a destinate expert-retouched one to enhance image quality. UEGAN [16] learned in an unsupervised fashion to realize image-to-image mapping. Deep Preset [7] realized color transfer by predicting various presets in a labeled reference photo and applied them to another image to blend and retouch a photo. Their filters are supervised and preset, however, in our framework, our filters are unsupervised and we only supervise the final harmonization output, so the filter style is not static and can be defined by users. Inspired by Deep Preset, we predict operator masks for pre-defined operations implicitly. The network can predict the position and level that each operation is applied to the picture and generate operator masks to simulate the manipulations.

3 Methods

3.1 Overview

Most image harmonization tasks try to solve the problem that a composite image comes from two pictures captured in different situations, which is a simple scene in image composition. In a more general case in our daily life, users may paste several foregrounds to one background image to make the picture meet their expectations. For example, users may cut the picture of a skirt from one picture, cut the picture of a hat from another picture, and then stick them to their portrait image. In this situation, the composite image needs to be changed in different directions to adjust the whole foreground harmonious with the background portrait. So in order to solve the problem, we propose a new setting to simulate this situation, named Semantic-guided Multi-Mask Image Harmonization(Sg-MMH) task, and construct two benchmarks, HScene and HLIP, for this task.

As shown in Figure 1, our framework includes one generator, which contains an encoder-decoder structure and a retoucher, and a patch discriminator. First, the inharmonious image I_c is the input of our operator mask generator. I_c can be generated by perturbing selected regions according to the semantics of a natural or person image I as described in Section 3.2. The encoder in the generator can locate the inharmonious region automatically. The decoder aims to predict the operator masks according to the gap between the inharmonious region and natural image distributions. These two structures can be implemented by any off-the-shelf backbone used in image harmonization tasks in previous research, such as [3, 2, 13, 6]. The output operator masks from the decoder will pass a

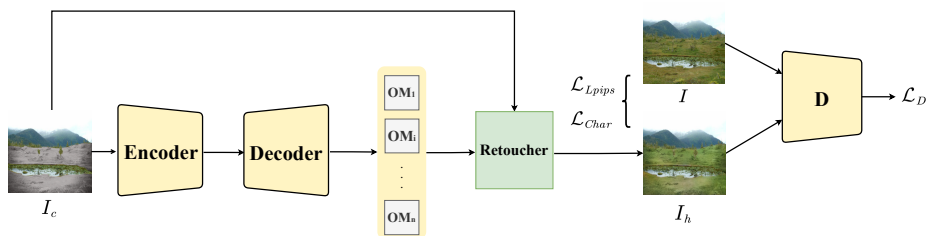


Fig. 1: This image shows the training pipeline of our framework. We first generate a composited image I_c guided by multiple semantic masks M from the original real image I , serving as the input image. The encode-decode structure generates operator masks. Then the operator masks OM will be used to retouch the composited image I_c and formulate I_h . The discriminator will compare I_h and I to make the output I_h harmonious.

retouch module, in which the predefined operators are applied to I_c according to the predicted locations and levels to harmonize the composited image I_c and generate a harmonious image I_h . We supervised the network by minimizing the difference between I_h and the ground truth harmonious image I . A binary regression loss (\mathcal{L}_{Char}) and a perceptual loss (\mathcal{L}_{Lpips}) will be applied to train the model and will be further introduced in Section 3.4. The discriminator we use is a patch discriminator. In this way, we force the discriminator to focus on some tiny objects and details.

In the following sections, we will first introduce how we construct our benchmarks and gradually illustrate the detailed structure of our framework and loss functions.

3.2 Multi-Mask Image Harmonization

We define a new task called Semantic-guided Multi-mask Image Harmonization (Sg-MMH). The input is a composited image in which foregrounds are pasted from different pictures, and the background is another real image. The output is a harmonious image in which the foregrounds have been adjusted to the background.

We build two benchmarks, **HScene** and **HLIP**. And their foregrounds have been perturbed according to their semantics. The details are as follows:

HScene: The harmonized images and the semantic masks of HScene are from ADE20K Dataset. ADE20K Dataset [28, 29] consists of natural images with high accuracy pixel level human annotations. There are 150 semantic classes, such as animals, bulidings, sky, trees, etc. There are 20k/2k/30k images in the training/validation/ testing set. Since the ground truth semantic label of the testing set is not released, we use the total images in the train set to construct our train set and use the validation set as our test set, and neglect some images with no foreground to construct HScene.

HLIP: HLIP constructed on the basis of The Look into Person(LIP) dataset, which is a large-scale dataset that focuses on the semantic understanding of a person. It contains 50462 images with elaborated pixel-wise annotations of 19 semantic human part labels, such as dress, coat, hair and face. The original training set contains 30462 images, and the validation images contain 10k pictures. Since the images contain poses, views, occlusions, appearances, and resolution variations, they can simulate real-world portrait scenarios. To construct HLIP, we also perturb the original train set as our training set, perturb the validation set as our test set, and neglect some images without proper foreground.

The statistics of training and testing images in the original datasets and our two constructed benchmarks are shown in Table 1.

Table 1: The statistics of the original datasets and our two constructed benchmarks.

Benchmark name	ADE20K	LIP	HScene	HLIP
Training set	20210	30462	20196	30385
Testing set	2000	10000	2000	9972

Previous works often use fixed fake foregrounds during training. So in this work, we randomly select $0.2 \times \text{len}(\text{classes}) + 1$ perturb regions as the foreground and disturb them with several structural perturbations. The *classes* number depends on the total semantic regions in each image. First, we mainly use Instagram filters[19], which is a set of camera filters usually used in image-editing software. We implement them by a python API called pilgram. Here we use the total of 23 Instagram filters and randomly select one to perturb one region. Meanwhile, we superimpose a filter from pilgram.css to each region with a probability of 0.5. So in each benchmark, we apply different disturbances to multiple regions randomly. These Instagram filters can represent most of the common filters in image processing applications, and most of them can change the color, illumination, and saturation of the foreground structurally. Second, to enrich the perturbation methods, we also add some other perturb methods to add more scenarios. One is to simulate the situation that a high-resolution foreground is pasted on a low-resolution background. The blurring process is conducted by using several blur or noise methods, which are usually used in image super-resolution tasks to blur the real image I and the background of I_c , and try to regress the network to blur the foreground and make the resolution of the whole image consistent. We use Gaussian noise, Laplace noise, Poisson noise, motion blur, and jpeg compression function and randomly select one blur method if blur perturbation is chosen. The other additional perturb method is to randomly multiply a number l, a, b to each channel of the LAB version of I_c as the following equation to add more structural perturbations.

$$I_{c_i} = I_{c_i} \times i, i = (l, a, b) \quad (1)$$

Some samples of our benchmarks have been shown in Figure 2. In this way, we want to simulate some structured perturb situation in which one region can be learned linearly in the same direction and at the same level. The reason we want to focus on this situation is that many changes in nature can be achieved by overall linear migration on a single meaningful channel, such as illumination change. So this situation is more suitable to operator masks format for they can change the appearance of one region in one direction centralized. Therefore, we have different foregrounds and perturb functions every iteration, and the network needs to regress the multi-mask region in a different direction, which adds variability and complexity to each training pair.

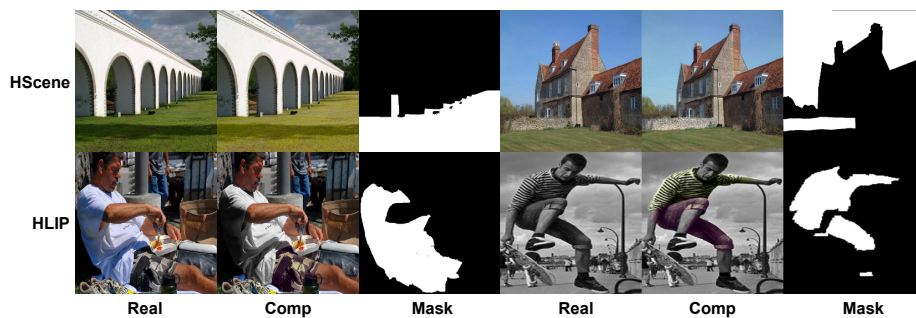


Fig. 2: Here we show some perturb visualization of our benchmarks. The first line shows samples from HScene, and the second line shows examples from HLIP. We can see that the foreground color, illumination, and contrast can be changed.

3.3 Framework Structure

We first use the U-Net structure from DoveNet [3] in there to illustrate our framework. It should be noted that the backbone can be replaced with any other more complex structures to boost the performance further. With our framework, the results generated in the form of operator masks are more editable and explainable. To make the fair comparison with DoveNet [3], we replace the input of the generator as I_c by removing the foreground masks, and the output of the decoder is operator masks, which its location can indicate where needs to change, and the value can represent what level needs to be changed. These operator masks could be pre-defined as certain functions and will be used to edit the inharmonious image in a Retoucher module.

Retoucher: The operator masks will be passed into a retouch module to harmonize I_c . The retouch module has some preset operations which users can define. Here we employ several simple but effective preset methods. Here we show an example. We first split I_c to L, A, and B channels. Then, we generate six operator masks: $OM_{i_{mul}}, OM_{i_{add}}, i = (l, a, b)$, and each will be multiplied or

added to each L, A, B channel of the image I_c , to generate I_h , as described in the following equation,

$$I_{h_i} = I_{c_i} \odot OM_{i_{mut}} \oplus OM_{i_{add}}, i = (l, a, b) \quad (2)$$

where " \odot " represents the Hadamard product operation and " \oplus " represents the Hadamard add operation. It should be noted that other forms of retouch can also be used, and the type of the pre-defined operator can be set according to different applications or dataset volume.

Patch discriminator: In our task, not only do the global features need to be discriminated, but accurate gradient feedback for local details also needs to be produced. Therefore, we employ a patch discriminator [8, 9] in our framework, which has a fixed receptive field, and each output value of the discriminator is only related to the local fixed area. Therefore, we want the discriminator to focus on some tiny objects and local details in the image. The final loss is the average of all local losses to guarantee global consistency [9].

3.4 Loss functions

We use four type losses in our framework:

- We apply a binary Charbonnier loss [12] to directly minimize the distance between our final output I_h and the ground truth I as:

$$\mathcal{L}_{Char} = \frac{1}{N} \sum_{i=1}^N \sqrt{(I_i - I_{h_i})^2 + \epsilon^2}, \quad (3)$$

where N is the batch size, ϵ is a very small constant in order to stabilize the value.

- In order to enhance the contextual details, we employ perceptual loss LPIPS[11]:

$$\mathcal{L}_{Lpips} = \text{LPIPS}(\hat{I}, I_h) \quad (4)$$

- Different from standard discriminator loss, we use a relativistic discriminator loss and try to maximize the probability that a real image is relatively more realistic than a fake one [25]. The discriminator loss is defined as:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_I [\log(D(I, I_h))] \\ &\quad - \mathbb{E}_{I_h} [\log(1 - D(I_h, I))] \\ D(I, I_h) &= \sigma(C(I) - \mathbb{E}[C(I_h)]) \rightarrow 1 \\ D(I_h, I) &= \sigma(C(I_h) - \mathbb{E}[C(I)]) \rightarrow 0 \end{aligned} \quad (5)$$

σ is the sigmoid activation layer, and $C(x)$ is the output of the discriminator [25]. $\mathbb{E}_{I_h}[\cdot]$ represents the final loss is averaged on the batch-level [25].

- The adversarial loss for the generator is as following [25]:

$$\mathcal{L}_G = -\mathbb{E}_I [\log(1 - D(I, I_h))] - \mathbb{E}_{I_h} [\log(D(I_h, I))] \quad (6)$$

In this way, the generator benefits from the gradients from both outputs harmonized image and real image in adversarial training.

Finally, our total loss function used for the generator is :

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{Char} + \beta\mathcal{L}_{Lpips} + \gamma\mathcal{L}_G, \quad (7)$$

where α, β, γ are set as 1, 1, 0.005 empirically.

4 Experiments

In this section, we introduce implementation details. Then we evaluate our proposed framework on our constructed datasets.

4.1 Implementation

As for the structure, in our framework, we first apply the backbone of an attention U-Net from DoveNet [3]. The pre-defined operator is set to the form of formula 2. So the input channel of the backbone is 3, the output channel is 6, and the number of filters in the last conv layer is 64. The number of downsamplings in U-Net is set to 8. The normalization type is instance normalization.

As for the optimization, we first trained the generator from scratch with the learning rate of $2e-4$. The milestone is set as 20k, 30k, and the total iteration is 40k for the HScene benchmark. And the milestone is set as 55k, 75k, and the total iteration is 85k for the HLIP benchmark. When training the whole framework with discriminator, we loaded the pre-trained generator model and trained with another 10k for HScene and HLIP. We used the ADAM optimizer, and the batch size in each GPU is set to 64. The whole training process is on Nvidia 3090 GPUs. We implemented it based on BasicSR [24] framework. We used 256×256 resolution for both training and testing evaluation.

The metric we use is Mean square Error(MSE), Peak Signal-to-Noise Ration (PSNR), Structural SIMilarity (SSIM) [26] and Perceptual metrics (LPIPS) [27]. PSNR and SSIM pay more attention to the fidelity of the image rather than visual quality [9]. The larger the two metrics, the less likely the image is distorted. LPIPS mainly focuses on comparing the distance of visual features between the generated image and the real one. In this work, we use pre-trained VGG to extract image features, and the smaller the LPIPS is, the generated image is closer to the real one visually.

4.2 Comparison to recent works

In this section, we compare our framework with the original output, which directly regresses the RGB value of each image in both quantitative and qualitative methods. We also conduct a user study test to demonstrate the subjective effect of these algorithms.

Quantitative Comparison: We evaluate the performance of our framework on our conducted two benchmarks, HScene and HLIP. The baseline network is adjusted from Dovenet[3]. We re-implemented it on the proposed multi-mask

Table 2: The comparison of the composite, RGB format output, and our framework output. Our framework can have a reasonable improvement on both the two benchmarks.

Method	HScene				HLIP			
Metric	MSE↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	PSNR↑	SSIM↑	LPIPS↓
Composite	271.89	27.83	0.96	0.027	111.95	30.99	0.96	0.021
Sg-MHH(RGB)	141.75	28.73	0.95	0.026	75.45	31.34	0.95	0.025
Sg-MHH(OM)	99.84	30.59	0.96	0.020	48.02	33.11	0.96	0.021
Sg-MHH(OMGAN)	94.00	30.88	0.96	0.020	42.62	33.92	0.97	0.018

dataset by removing the input mask. The quantitative results are shown in Table 2. The first line shows the metrics of the composited images on test datasets. When directly outputting the RGB version of the harmonious image, the result cannot be interpretable and modifiable, and the results are also sub-optimal in our setting. Our framework makes an improvement in terms of all evaluation metrics on two benchmarks.

To make a fair comparison with other SOTA methods, we also conduct experiments on previous popular single-mask benchmarks, such as iHarmony4 [3]. We train and test on iHarmony4 [3] datasets and show the results in the supplementary materials. The experiment results demonstrate that our methods can also have a reasonable effect on the single-mask image harmonization task.

Qualitative Comparison: We show visual comparison results on our proposed multi-mask harmonization datasets in Figure 3 and show more visualization on supplementary. The first column shows the real images, and the second column shows the perturbed images. In the third to fifth columns, we show the results output in the previous version and the results output from our framework without or with a discriminator. As the perturb methods on different segmentation masks are different, our framework can deal with the situation with different adjustment directions properly. As demonstrated in the red dotted box, the methods that directly output the RGB format only adjust one inharmonious region (the results in the last line) or adjust two areas but are not in place (the results in the first line). In this way, we demonstrate that our framework has a more flexible and realistic effect and can handle more complicated scenes.

User Study: We conduct user study to compare the results of our framework which the backbone is from DoveNet with the original DoveNet[3], BarginNet[2] and RainNet[13] methods and show the results in Table 3. We test all methods on real composited images.

We consider three realistic situations to construct real composited images. 1) Real composite images that need structured adjustment, such as only changing the illumination. 2) Real composite images that foregrounds are cut from pictures that have already been processed by Instagram filters and want to be re-processed. 3) Real composite images that the foregrounds come from two real pictures but already look realistic. Both of the composited images contain multiple regions that need to be changed. We construct 11 image pairs showing one

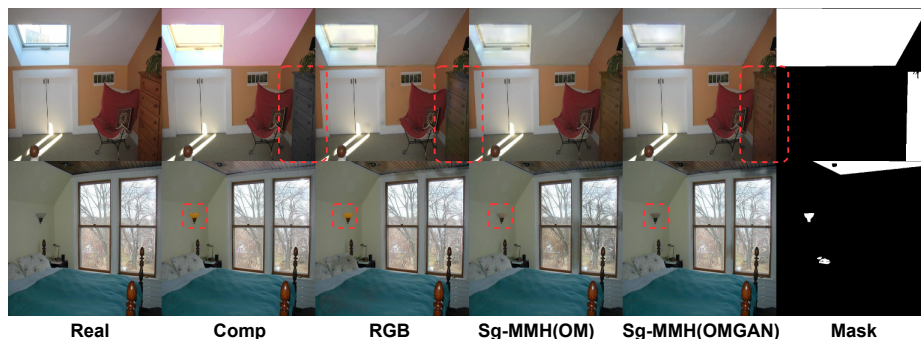


Fig. 3: **Qualitative harmonization results.** The first column is the real image. The second column is the composited image. The third to the fifth column is the harmonization results of a baseline RGB format and our framework with or without GAN structure. OM means *only use generator in our framework*. OMGAN represents *generate operator masks with GAN framework*.

Table 3: **User study results on real composited images.** We show four images to users and ask them to choose the most realistic one. The number means what percentage of subjects think the images produced by one method are more realistic.

Methods	User Study Score(%) \uparrow
DoveNet[3]	20.6
BarginNet[2]	24.2
RainNet[13]	18.5
Sg-MMH(OMGAN)	36.7

picture of each setting in Figure 4. Since there is no real image as reference, we cannot use evaluation like proposed in 4.1 to evaluate image quality. We asked 60 subjects, and each subject needed to independently select one image they think is the most realistic of the four images. The images from the four methods are presented randomly. From the qualitative results and user study, we can see that, when dealing with the illumination change task, our method only change the brightness level but retain other color prosperity. This is meaningful because, in most cases, we just want to change the illumination of the foreground but keep the color of the foreground itself. When dealing with pictures such as downloading from websites which already processed by camera filters, our method can alleviate these filters and change them to a more natural appearance. This situation is also worth paying attention to because, in many cases, the landscape images posted to the Internet have already been processed with camera filters, but traditional color transfer methods cannot mimic the realistic filters. When dealing with pictures in which the foregrounds are cut from real pictures, our model mostly keeps remaining the foreground color because although the fore-

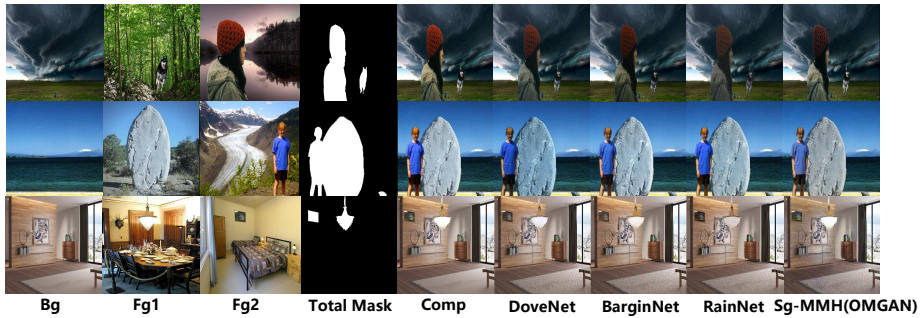


Fig. 4: **Qualitative results on real composited images.** We compare our method with recent state-of-the-art harmonization methods, including DoveNet, BarginNet, and RainNet. Note that they all have input masks to indicate the inharmonious regions while our method adjusts the input automatically. Our proposed framework can apply different operations for different semantic regions appropriately.

grounds and background are not from the same picture, the original foreground is also reasonable in the new environment. In many cases, when the foreground and background colors are not very consistent, the composite image also looks actual and sensible and does not necessarily need to be adjusted. Only when the color difference is very large and unreasonable the harmonization task needs to adjust the composite image appropriately.

4.3 Discussion

Explainable property of operator masks: Our operator masks can sense the changes and respond appropriately to the corresponding operator masks. Here in Figure 5 we visualized our operator masks effect on our datasets. For example, in the first line, the sky is perturbed to gray, and to recover it into blue, the operator mask $OM_{b_{add}}$ has a largely negative response since when the picture is in LAB format, the smaller the value of channel B, the bluer the color. In the second line, the foregrounds are disturbed in two directions, the hat is perturbed greener, and the shoes become redder, so $OM_{a_{add}}$ and $OM_{b_{add}}$ has a positive response in the hat region and has a negative response in the region of the shoe. These pictures show that our algorithm can explain the changes the network makes for the inharmonious input.

To illustrate that our method can also explain when dealing with real images, we cut a foreground and paste it into a real background picture. Here in Figure 6(a) show that our operator mask $OM_{l_{add}}$ has a strong negative activation, indicating that the brightness of the inharmonious region is reduced.

Editable property of operator masks: Previous work often generates harmonized images directly, and the results cannot be edited if users do not like the automatic outputs. However, in our work, we locate and generate operator

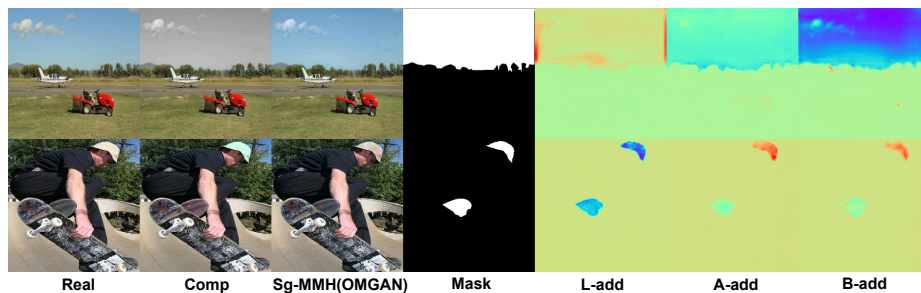


Fig. 5: **Visualization of the operator masks in our datasets.** The first line is the result from the HScene dataset, and the last line is the result from the HLIP datasets. The last three columns visualize the operator masks of ‘add’ of L, A, and B channels.

masks, which can be edited if users do not like the results. Here Figure 6(b) shows the edible effect of our algorithm. Users can add or multiply a number to deepen the color or change the illumination of the foreground. This figure shows that our method can edit the harmonization results based on the automated generation.

Limitation and future work: Our method attempt to first realize the harmonization framework with operator masks. Although we have explored the effectiveness of operator masks and show their editable and explainable features, there are still some limitations that need further exploration. First, the training of the harmonization network may heavily depend on the perturbations. The particularity of the filters we use makes our model have many performance limitations. The performance and generalization ability of the network can be further improved if more perturbations are included in the data pre-processing. Also, without special restrictions, some filters applied in our benchmarks make some rare composited images occur. In future research, filters should be applied according to the semantics of objects, such as a green filter should not be applied to a face. Second, when dealing with the foreground color, current image harmonization methods usually choose to transform the color of the foreground to be consistent with the background, but this action is often unnecessary. Sometimes the foreground can keep its own color, and the whole composite image also looks reasonable and realistic. Future work could go further step to explore in which region the foreground color can be consistent with the background and give a color range, rather than just outputting one result. Third, we propose six alternative operator masks to build our retoucher. However, more complicated operators require further exploration, for example, the shadow effects, the grainy effect, or de-noise. Forth, as our operator masks are trained in an unsupervised manner, there may be coupling between different masks. The controllability can be further improved if the operator masks can be decoupled or with specific supervision.

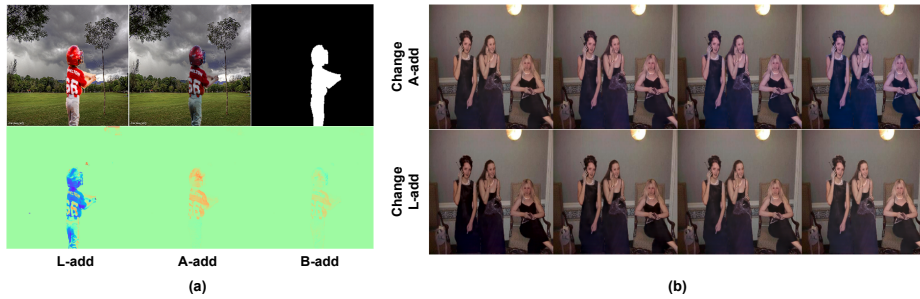


Fig. 6: Here we show the explainable and editable attributes of our operator masks. Figure 5(a) shows the **Visualization of the operator masks in real data**. First row: the composite, the harmonious result, and the ground truth foreground mask. Second row: the operator masks of ‘add’ operator masks in L, A, and B channels. The L channel has a strong negative activation, indicating the brightness of the inharmonious region has been reduced. Figure 5(b) shows the **Editing attribute of the operator masks**. First row: gradually enlarge the value of the ‘add’ operator mask in the A channel. Second row: gradually enlarge the value level of the ‘add’ operator mask in the L channel, which is the brightness. This figure shows that our method can edit the harmonization results based on the generated results.

5 Conclusion

In order to tackle the problem that one composite image has multiple foregrounds pasted from other images, we set up a new setting Sg-MMH and propose two benchmarks, HScene and HLIP, to simulate the situation. To solve the benchmarks, we propose a new framework that uses operator masks to solve the multi-mask image harmonization task. The proposed methods can simulate preset operations in the PhotoShop to edit each image channel and can be explained and edited by users. Also, the operator masks can locate the inharmonious region automatically even when the indicator masks are not given. Experiments show that our framework can handle the more complicated scenes and achieve good results. There are also some limitations for the current work, such as the design of the operator masks is easy and can not decoupled well. In the future, more parametric operator masks can be applied to this framework based on expert knowledge, which will add more control and interpretability to the image harmonization task.

References

1. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.Q.: Color harmonization. In: ACM SIGGRAPH 2006 Papers, pp. 624–630 (2006)
2. Cong, W., Niu, L., Zhang, J., Liang, J., Zhang, L.: Bargainnet: Background-guided domain translation for image harmonization. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2021)
3. Cong, W., Zhang, J., Niu, L., Liu, L., Ling, Z., Li, W., Zhang, L.: Dovenet: Deep image harmonization via domain verification. In: CVPR. pp. 8394–8403 (2020)
4. Cun, X., Pun, C.M.: Improving the harmony of the composite image by spatial-separated attention module. *IEEE TIP* **29**, 4759–4771 (2020)
5. Gong, K., Liang, X., Zhang, D., Shen, X., Lin, L.: Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 932–940 (2017)
6. Guo, Z., Guo, D., Zheng, H., Gu, Z., Zheng, B., Dong, J.: Image harmonization with transformer. In: ICCV. pp. 14870–14879 (2021)
7. Ho, M.M., Zhou, J.: Deep preset: Blending and retouching photos with color style transfer. In: ICCV. pp. 2113–2121 (2021)
8. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. pp. 1125–1134 (2017)
9. Ji, X., Cao, Y., Tai, Y., Wang, C., Li, J., Huang, F.: Real-world super-resolution via kernel estimation and noise injection. In: CVPR. pp. 466–467 (2020)
10. Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. *ACM TOG* **25**(3), 631–637 (2006)
11. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
12. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE TPAMI* **41**(11), 2599–2613 (2018)
13. Ling, J., Xue, H., Song, L., Xie, R., Gu, X.: Region-aware adaptive instance normalization for image harmonization. In: CVPR. pp. 9361–9370 (2021)
14. Liu, Y., Qin, Z., Wan, T., Luo, Z.: Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks. *Neurocomputing* **311**, 78–87 (2018)
15. Mirza, M., Osindero, S.: Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014)
16. Ni, Z., Yang, W., Wang, S., Ma, L., Kwong, S.: Towards unsupervised deep image enhancement with generative adversarial network. *IEEE TIP* **29**, 9140–9151 (2020)
17. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM SIGGRAPH 2003 Papers, pp. 313–318 (2003)
18. Pitie, F., Kokaram, A.C., Dahyot, R.: N-dimensional probability density function transfer and its application to color transfer. In: ICCV. vol. 2, pp. 1434–1439. IEEE (2005)
19. PyPI: pilgram. <https://pypi.org/project/pilgram/>
20. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Computer graphics and applications* **21**(5), 34–41 (2001)
21. Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale image harmonization. *ACM TOG* **29**(4), 1–10 (2010)

22. Tao, M.W., Johnson, M.K., Paris, S.: Error-tolerant image compositing. In: ECCV. pp. 31–44. Springer (2010)
23. Tsai, Y.H., Shen, X., Lin, e.: Deep image harmonization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3789–3797 (2017)
24. Wang, X., Yu, K., Chan, K.C., Dong, C., Loy, C.C.: BasicSR: Open source image and video restoration toolbox. <https://github.com/xinntao/BasicSR> (2020)
25. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCV. pp. 0–0 (2018)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE TIP* **13**(4), 600–612 (2004)
27. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
28. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. pp. 633–641 (2017)
29. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset **127**(3), 302–321 (2019)
30. Zhu, J.Y., Krahenbuhl, P., Shechtman, E., Efros, A.A.: Learning a discriminative model for the perception of realism in composite images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3943–3951 (2015)
31. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. pp. 2223–2232 (2017)