Region-aware Adaptive Instance Normalization for Image Harmonization

Jun Ling[1], Han Xue[1], Li Song[1,2]✉, Rong Xie[1], Xiao Gu[1]
[1]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, China
[2]MOE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
{lingjun, xue_han, song_li, xierong, gugu97}@sjtu.edu.cn

Abstract

Image composition plays a common but important role in photo editing. To acquire photo-realistic composite images, one must adjust the appearance and visual style of the foreground to be compatible with the background. Existing deep learning methods for harmonizing composite images directly learn an image mapping network from the composite to real one, without explicit exploration on visual style consistency between the background and the foreground images. To ensure the visual style consistency between the foreground and the background, in this paper, we treat image harmonization as a ***style*** *transfer problem. In particular, we propose a simple yet effective Region-aware Adaptive Instance Normalization (RAIN) module, which explicitly formulates the visual* ***style*** *from the background and adaptively applies them to the foreground. With our settings, our RAIN module can be used as a drop-in module for existing image harmonization networks and is able to bring significant improvements. Extensive experiments on the existing image harmonization benchmark datasets shows the superior capability of the proposed method. Code is available at https://github.com/junleen/RainNet.*

1. Introduction

Image composition is one of the most common operations in image editing [39, 3] and data augmentation [6, 42], *etc*. However, generating a realistic composite image by taking an object from one image and combining it with a new background image usually requires professional compositors to adjust the appearance of the foreground objects by photo editing software like Adobe Photoshop, and ensure the realism of the generated image. To alleviate this burden, image harmonization is introduced for adjusting the foreground and making it seamlessly integrated into the new image with less human involvement, especially for non-expert users.

However, what makes a composite image appear more realistic? In this paper, we present a new perspective for image harmonization. Let us take Fig. 1 for example. Fig. 1 shows three different real photos (small-sized images with red border) that hold different visual properties. When an unbefitting foreground object with special visual properties is pasted into a new image with incompatible visual features, we can easily distinguish it from real photos. This is an unsolved problem and has emerged for years, which we call visual *style* discrepancy. Specifically, in this paper, we define the visual *style* of an image as visual properties including illumination, color temperature, saturation, hue, texture *etc*., which varies from image to image. To make a composite image look more realistic, we must ensure a more consistent visual *style* between the foreground and the background.

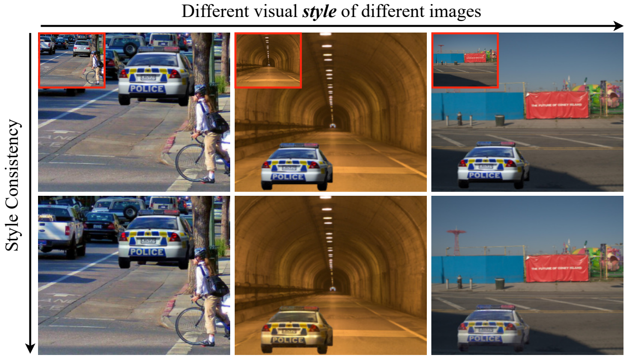

Figure 1. Illustration of our motivation. If we want to put a police car into these images with different visual *style* , we must ensure that the car is compatible with the background images (small-sized images with red boundaries in the *top row*). Simple cut-and-paste operations introduce unrealistic results (*top row*). Our method aims to adaptively learn high-level visual *style* from different backgrounds and produce harmonious composite images (*bottom row*).

Abundant image harmonization approaches have been proposed for improving the realism of composite images. Traditional methods address the harmonization problem by transferring statistics of hand-crafted features between foreground and background regions, such as color [26, 27, 39, 30]. However, these methods only work in simple cases where the foreground image is already consistent with the background image. Recently, more deep learning-based methods [2, 3, 32, 43] have been proposed for generating harmonious images in an end-to-end manner. Zhu *et al*. [43] propose to adopt a discriminative model to predict the realism of a compsite image and assist optimization of color adjustment. Tsai *et al*. [32] propose an end-to-end learning approach for image harmonization while only constraining semantic information learning in the encoder. Cun *et al*. [3] adopt a spatial-separated attention module to enforce the network to learn the foreground and background features separately, failing to ensure the *style* consistency between these two parts. To sum up, none of these methods really consider the realism from the perspective of visual *style* consistency. Cong *et al*. [2] propose to use a domain verification discriminator and adversarial loss [10] to improve domain-consistency between foreground and background regions but neglect to explicitly transform the foreground features in the generator. However, performance improvement brought by such an auxiliary discriminator is limited (*i.e*., 0.27dB for PSNR, which is revealed in [2]).

To address these issues, in this work, we reframe image harmonization as a background-to-foreground *style* transfer problem, where we render the foreground image to hold similar visual *style* of the background image. Taking *style* guidance from background information is of great importance because the foreground image should be converted to own different appearances when pasted into different background images (as illustrated in Fig. 1). To generate style-consistent and realistic-looking composite images, we expect a unified transferring operation to adaptively adjust the *style* of the foreground objects to be in perfect harmony with new background images even collected in different environments. Therefore, in this work, we propose a learnable layer, named Region-aware Adaptive Instance Normalization (RAIN) layer, to learn the style from background images and apply it to the foreground objects. By taking convolutional features and the foreground mask as input, the RAIN layer aligns the channel-wise mean and variance of the foreground activation to match those learned from the background. The details of the proposed RAIN module are presented in Fig. 3. It is worth mentioning that our RAIN layer can be easily applied to existing image harmonization networks and encourage performance improvements.

The contributions of this work are as follows. 1) To the best of our knowledge, we are the first to introduce the *style* concept of background images and regard the image harmonization task as a *style* transferring problem. 2) We propose a novel Region-aware Adaptive Instance Normalization (RAIN) method, which captures the *style* information only from the background features and applies it to the foreground for image harmonization tasks. Our RAIN module is simple yet effective and can be used as a *plug-and-play* module for existing image harmonization networks to enhance their performance. 3) Extensive experiments demonstrate that our method surpasses the state-of-the-art methods by a large margin.

2. Related work

Image harmonization aims to adjust a foreground image to seamlessly match a background image. Traditional methods mainly focus on matching the appearance of the foreground with background regions based on handful of handcrafted heuristics, such as color statistics [27, 26, 39], gradient information [15, 25, 31], multi-scale statistical features [30], semantic information [32, 33]. These methods directly match appearance to harmonize a composite image while paying less attention to visual realism. Johnson *et al*. [17] introduce a data-driven approach to improve the realism of computer-generated images by retrieving a small number of real images from an image dataset and transfer the features of color, tone, texture, *etc*. Lalonde *et al*. [19] predict the realism of images by learning global and local statistics from natural images. With the advances of deep learning, more deep learning-based methods [3, 2, 32, 43] draw much attention due to their impressive results. Different from these works, we start from the perspective of background-to-foreground *style* transfer, and push the limit of image harmonization performance by introducing a novel RAIN module, which separates our approach from previous methods.

Neural style transfer is designed to render a photo with special visual style captured from artistic creations while retaining the content information from the original image. Earlier style transfer methods concentrate on texture synthesis or transfer [7, 8, 20, 34]. Gatys *et al*. [9] first introduce a method to match feature statistics in pre-trained convolutional networks and demonstrate impressive artistic style transfer. To achieve the goal of real-time style transfer, Johnson *et al*. [16] propose a novel feed-forward perceptual loss with a pre-trained VGG network [29]. Later, Huang *et al*. [11] propose Adaptive Instance Normalization (AdaIN) to achieve arbitrary style transfer from the perspective of feature normalization. Besides AdaIN, other normalization methods [5, 35] were also proposed for fast stylization and later adopted in various vision tasks [12, 21, 22, 41, 38].

Normalization layers include unconditional normalization (Batch Normalization (BN) [13], Instance Normalization (IN) [35], Layer Normalization (LN) [1], Group Normalization (GN) [37], *etc*.) and conditional normalization (Conditional Batch Normalization (CBN) [4], Conditional Instance Normalization (CIN) [5], SPADE [23], Region Normalization (RN) [40], and AdaIN [11], *etc*.). Note that unconditional normalization aligns the mean and variance of feaures without guidance from external data. On the con-

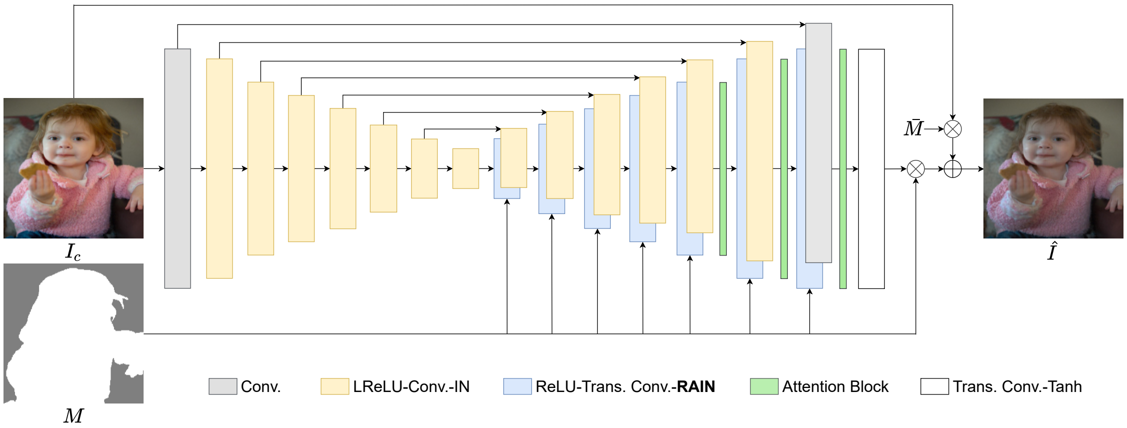

Figure 2. Overview of the proposed generator. We provide a detailed structure of our RainNet to ensure better understanding and reproducibility. The bottom legend: Conv.= Convolution, Trans. = Transposed.

trary, conditional normalization [4, 5, 11, 23] requires external data to provide affine parameters, which embed new information from the external data. SPADE [23] applies spatially-varying transformations from semantic masks for image synthesis, which cannot be used in our image harmonization task due to the irregular shapes of foreground objects. RN [40] is designed for image inpainting which aims to alleviate the mean and variance shift problem but it does not consider the semantic connection between the background and the foreground. AdaIN [11] is proposed for real-time image stylization which uses a pre-trained VGG network to extract style code. However, it is not practical for our task because the *style* defined in this work is considered to be consistent with image realism instead of texture. Besides, the background image with one region removed cannot be extracted by a pre-trained network, which will introduce new problems of mean and variance shift. In this paper, we seek ways to establish a connection between the background and the foreground. Therefore, we regard image harmonization as a new *style* transfer task in which we transfer *style* from the background to the foreground instance.

3. Our approach

Our goal is to learn a mapping network for the foreground image and ensure that the foreground image is compatible with the background. To achieve this goal, we introduce our Region-aware Adaptive Instance Normalization (RAIN) for improving the performance of basic networks.

3.1. Problem formulation

We consider a foreground image and a background image as I_f and I_b respectively. The foreground mask is denoted by M, which indicates the region to be harmonized in the composite image I_c. Accordingly, the background mask is $\bar{M} = 1 - M$. The object composition process is formulated as $I_c = M \circ I_f + (1 - M) \circ I_b$, where $\circ$ is the Hadamard product. In this paper, we define the harmonization model as generator G, and the harmonized image as $\hat{I} = G(I_c, M)$, where G is a learnable model that we expect to optimize for making $\hat{I}$ close to the ground truth image I by $\|G(I_c, M) - I\|_1$.

3.2. Region-aware Adaptive Instance Normalization (RAIN)

The input of our normalization module consists of two parts, *i.e.*, the foreground mask, and the convolutional features (see in Fig. 3). Without loss of generality, we take the RAIN module in the i-th layer of G for example. Let $F^i \in \mathbb{R}^{H^i \times W^i \times C^i}$ be the activations and $M^i \in \mathbb{R}^{H^i \times W^i}$ be the resized foreground mask in the i-th layer, where H^i, W^i, C^i denote the height, width, and number of channels of feature F^i, respectively. We propose a simple yet effective normalizing method called Region-aware Adaptive Instance Normalization (RAIN).

As depicted in Fig. 3, we first multiply the input features F^i by the foreground mask and its corresponding background mask. Then we normalize the foreground features by IN [35], and then affine the normalized features with learned scale and bias from the background features. The new activation value $\bar{F^i}$ at site (h, w, c) in the foreground region is computed by:

$$\bar{F^i}_{h,w,c} = \gamma^i_c \frac{F^i_{h,w,c} - \mu^i_c}{\sigma^i_c} + \beta^i_c, \qquad (1)$$

where μ^i_c and σ^i_c are the channel-wise mean and variance of the foreground feature in i-th layer:

$$\mu^i_c = \frac{1}{\#\{M^i = 1\}} \sum_{h,w} F^i_{h,w,c} \circ M^i_{h,w}, \qquad (2)$$

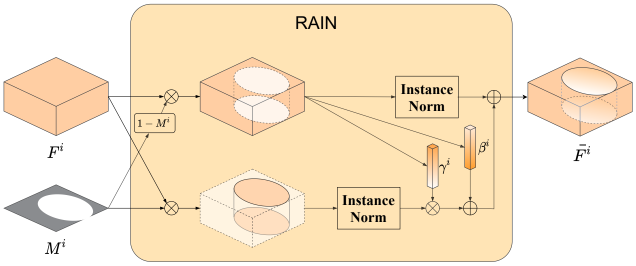

Figure 3. Our RAIN module takes the input feature F^i and resized mask M^i as input. Then we obtain the statistical style parameters γ^i and β^i from only background features. The produced γ^i and β^i are multiplied and added to the normalized foreground features in a channel-wise manner.

$$\sigma_c^i = \sqrt{\frac{1}{\#\{M^i = 1\}} \sum_{h,w} (F^i_{h,w,c} \circ M^i_{h,w} - \mu_c^i)^2 + \epsilon}. \quad (3)$$

The expression $\#\{x = k\}$ means the number of pixels which equal to value k in x. The γ_c^i and β_c^i are the mean and standard deviation of the activations of the background in channel c of layer i:

$$\gamma_c^i = \frac{1}{\#\{\bar{M}^i = 1\}} \sum_{h,w} F^i_{h,w,c} \circ \bar{M}^i_{h,w} \quad (4)$$

$$\beta_c^i = \sqrt{\frac{1}{\#\{\bar{M}^i = 1\}} \sum_{h,w} (F^i_{h,w,c} \circ \bar{M}^i_{h,w} - \gamma_c^i)^2 + \epsilon} \quad (5)$$

where $\bar{M}^i$ is the background mask in i-th layer.

Our method is different from AdaIN in two aspects. First, our method focuses on transferring the visual *style* from background to foreground only within the same image while AdaIN considers the style of features from another whole external image. Second, AdaIN uses a pre-trained VGG network to extract and calculate the statistics of the features, which cannot be directly employed in our task. Contrarily, our RAIN is designed and trained for image harmonization, such that the style parameters are better fitted for the foreground adjustment operations. Moreover, comprehensive experimental results demonstrate the efficacy of the proposed method.

RainNet. We take a simple U-Net [28, 14] alike network without any feature normalization layers as our basic network architecture. Following [2, 3], in this work, we add three attention blocks in the decoder part for our **Baseline** network. Theoretically, our RAIN module can be applied in any layers of the basic network. In this work, we train our baseline with different normalization methods and exploit the design strategy of implementing our RAIN module to obtain the best model, denoted as RainNet. The structure of our RainNet is depicted in Fig. 2.

Why is RAIN effective? Briefly, RAIN helps the model to capture the visual *style* information from the background image and inject it into the foreground, so that the generated foreground objects are more compatible with the new background.

Consider a simple case with Region Normalization (RN) [40] that performs feature normalization for the foreground features and the background features separately. In each normalization layer, the background features will not provide any guidance for the model to transform the foreground features. Consequently, the model can only transform the foreground image to hold the average back-ground visual statistics in the training data, leading to unsatisfactory harmonizing results. However, when performing normalization with BN or IN, the foreground features will be normalized with the same mean and variance as the background features, where the mean and variance are statistically measured from the whole global feature map. Unfortunately, the styles of background features will be shifted by those statistics from the foreground and limit the style consistency learning in subsequent layers.

In contrast with other normalization methods, our RAIN module only transfers the statistics from the background features to the normalized foreground features, without the influences from inconsistent foreground objects. As plotted in Fig. 6, IN and BN outperform RN, while our RAIN outperforms IN and BN by a large margin, demonstrating the reasonableness of our aforementioned analysis.

4. Implementation

Datasets. To demonstrate the efficacy of our approach, we analyze the performance of our model against previous methods on the benchmark dataset iHarmony4 [2]. According to [2], iHarmony4 consists of 4 sub-datasets (*i.e.*, HCOCO, HAdobe5K, HFlicker and Hday2night), and 73147 pairs of synthesized composite images and corresponding ground truth images are provided. In our experiments, we follow the train-test split as [2] suggested.

Training. We trained the model by Adam [18] optimizer with a learning rate of 0.0002, and optimized our model with the same objective that DoveNet [2] uses. Our model was optimized for 100 epochs on an Nvidia GTX 2080Ti GPU, with input images resized to 256×256 and batch size set to 12. Detailed training objectives of our model are presented in the supplementary materials.

5. Experimental Results

In this section, we conduct extensive experiments to demonstrate the efficacy of our method. We first compare our best model (RainNet) to current state-of-the-art methods both qualitatively and quantitatively in Sec. 5.1. Then, we investigate the design choice of RAIN for our generator in Sec. 5.2. Subjective evaluations and further discussions are presented in Sec. 5.3 and Sec. 5.4, respectively.

Method	Venue	HCOCO	HAdobe5k	HFlickr	Hday2night	Average
Input composite	-	33.94	28.16	28.32	34.01	31.63
Lalonde and Efros [19]	ICCV'07	31.14	29.66	26.43	29.80	30.16
Xue *et al*. [39]	TOG'12	33.32	28.79	28.32	31.24	31.40
Zhu *et al*. [43]	ICCV'15	33.04	27.26	27.52	32.32	30.72
DIH [32]	CVPR'17	34.69	32.28	29.55	34.62	33.41
S^{2}AM [3]	TIP'20	35.47	33.77	30.03	34.50	34.35
DoveNet [2]	CVPR'20	35.83	34.34	30.21	**35.18**	34.75
Baseline	This work	35.03	33.35	29.50	35.02	33.92
RainNet	Ours	**37.08**	**36.22**	**31.64**	34.83	**36.12**

Table 1. Quantitative performance comparisons of PSNR metric on the four sub-datasets of iHarmoni4 [2]. The numbers in **red** and blue represent the best and second best performance. As can be found from the results, our approach performs favorably against other methods.

Method	Venue	0% ∼5%		5% ∼15%		15% ∼100%		Average	
		MSE	fMSE	MSE	fMSE	MSE	fMSE	MSE	fMSE
Lalonde and Efros [19]	ICCV'07	41.52	1481.59	120.62	1309.79	444.65	1467.98	150.53	1433.21
Xue *et al*. [39]	TOG'12	31.24	1325.96	132.12	1459.28	479.53	1555.69	155.87	1141.40
Zhu *et al*. [43]	ICCV'15	33.30	1297.65	145.14	1577.70	682.69	2251.76	204.77	1580.17
DIH [32]	CVPR'17	18.92	799.17	64.23	725.86	228.86	768.89	76.77	773.18
S^{2}AM [3]	TIP'20	15.09	623.11	48.33	540.54	177.62	592.83	59.67	594.67
DoveNet [2]	CVPR'20	14.03	591.88	44.90	504.42	152.07	505.82	52.36	549.96
Baseline	This work	19.21	841.61	64.54	749.36	241.15	803.05	79.97	808.68
RainNet	Ours	**11.66**	**550.38**	**32.05**	**378.69**	**117.41**	**389.80**	**40.29**	**469.60**

Table 2. We measure the error of different methods in foreground ratio range based on the whole test set. fMSE indicates the mean square error of the foreground region. The numbers in **red** and blue indicate the best and second-best results.

Method	0% ∼5%			5% ∼15%			15% ∼30%			30% ∼100%			Average		
	fL1	PSNR	SSIM	fL1	PSNR	SSIM	fL1	PSNR	SSIM	fL1	PSNR	SSIM	fL1	PSNR	SSIM
Baseline	21.76	37.99	0.9951	20.55	32.05	0.9838	20.97	27.85	0.9631	21.49	24.39	0.9285	21.31	33.92	0.9824
+ IN [35]	18.61	39.08	0.9959	16.53	33.75	0.9870	16.34	29.77	0.9711	17.97	25.97	0.9384	17.69	35.32	0.9855
+ BN [13]	17.81	39.48	0.9962	16.79	33.60	0.9876	17.76	29.15	0.9704	19.32.	25.10	0.9395	17.65	35.34	0.9859
+ RN [40]	18.85	38.74	0.9959	17.54	32.85	0.9864	18.77	28.42	0.9673	20.55	24.37	0.9326	18.62	34.57	0.9842
+ RAIN-1	**17.10**	**39.67**	**0.9963**	14.70	34.69	0.9882	14.20	31.02	0.9742	14.92	27.36	0.9478	**15.88**	36.06	0.9873
+ RAIN-2	17.71	39.39	0.9961	14.88	34.52	0.9882	13.89	31.19	0.9737	14.39	27.72	0.9491	16.16	36.01	0.9871
+ RAIN-3	17.97	39.28	0.9960	15.00	34.54	0.9881	13.82	31.19	0.9743	**14.21**	**27.75**	0.9493	16.30	35.95	0.9872
+ RAIN-4	17.95	39.27	0.9959	14.95	34.51	0.9878	**13.75**	**31.23**	0.9735	14.75	27.51	0.9469	16.31	35.96	0.9868
+ RAIN-Encoder	19.29	38.81	0.9957	16.64	33.79	0.9869	15.96	30.15	0.9719	16.40	26.72	0.9449	17.89	35.31	0.9861
+ RAIN-Decoder	17.41	39.50	0.9962	**14.32**	**34.89**	**0.9889**	14.18	31.01	**0.9746**	14.75	27.60	**0.9507**	15.92	**36.12**	**0.9877**

Table 3. Ablation studies. The numbers in **red** and blue represent the best and second-best performance.

5.1. Comparison with existing methods

Performance on different sub-datasets. To quantitatively validate our approach, we adopt the evaluation protocols from previous work [2, 32, 3]. We first train our model on the whole training set. Then we evaluate the trained model on given testing images by measuring mean square error (MSE) and PSNR score for the synthesized images. The results of all previous methods as well as our RainNet are given in Table 1. It can be observed that the baseline model attains comparable performance of DIH [32]. Benefiting from the proposed RAIN module, our RainNet improves the baseline by a reduction of 39.68 in MSE metric, and a performance gain of 2.2 in PSNR for all datasets. Although DoveNet [2] is slightly favorable to our approach in Hday2night dataset, our model achieves the best results on HCOCO, HAdobe5k, and HFlickr and outperforms [2] by a large margin in average performance.

Influence of foreground ratios. We next examine the influence of different foreground ratios on the harmonization models. Following [2], we split the images into three

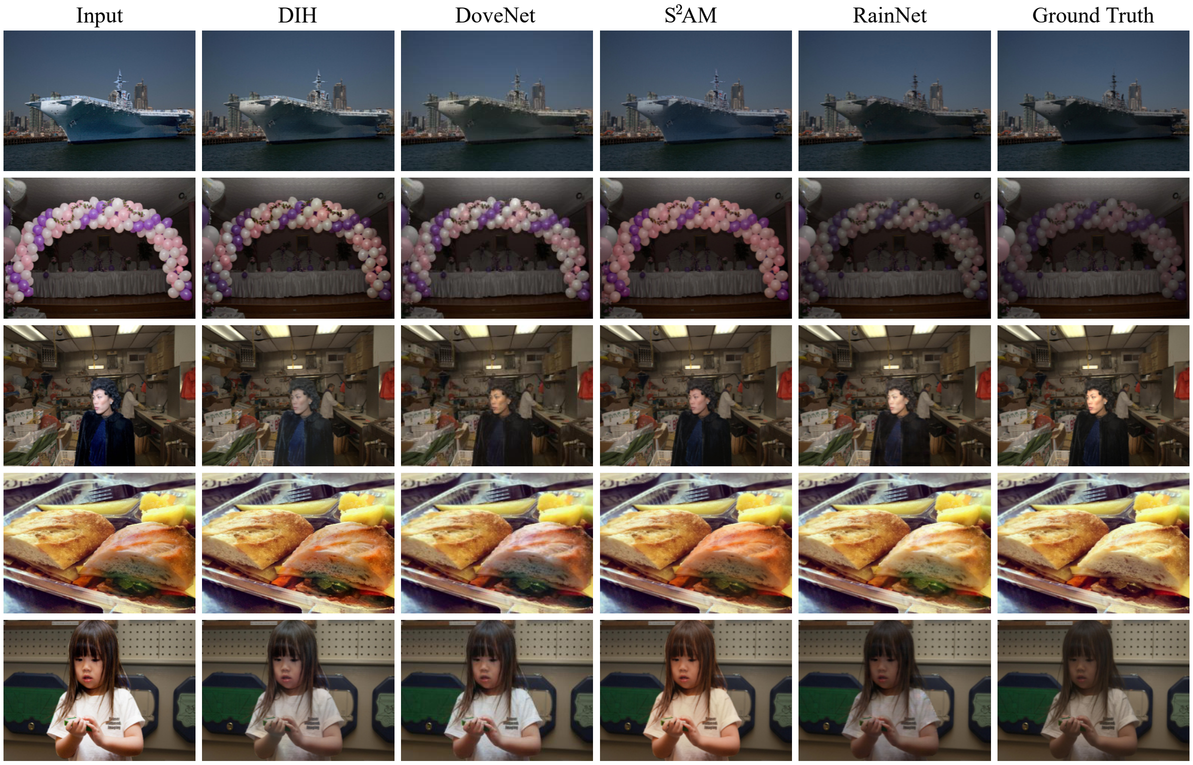

Figure 4. **Qualitative comparison**. We present example results of our RainNet against three state-of-the-art methods. The samples are taken from the testing dataset of iHarmony4 [2].

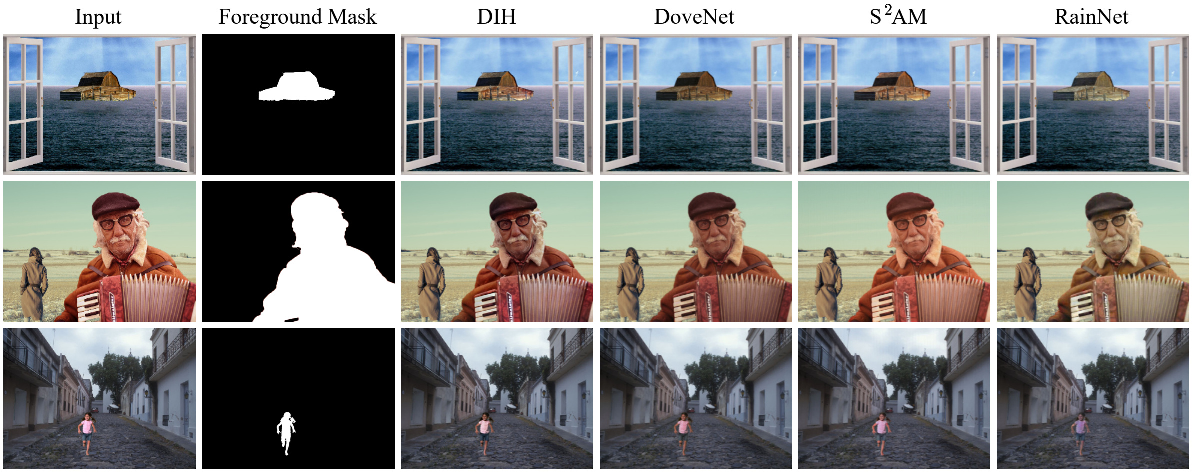

Figure 5. **Example results on real composite images.**. We present real composite images, foreground mask, the results of three state-of-the-art methods, and the proposed model. The samples are taken from the testing dataset of [32]. Our method achieves better harmonized visual results than competing methods.

groups according to different foreground ratio ranges, *i.e.*, 0% ∼5%, 5% ∼15%, and 15% ∼100%. We compare the performance by metrics of MSE and fMSE. For fMSE, we only calculate the MSE of the foreground regions. The comparison results are presented in Table 2. As can be found, on one hand, the model performance in terms of MSE downgrades as the foreground ratios increases while fMSE is less likely to be influenced by foreground ratios. On the other hand, our model outperforms [2] by 80.36 in the fMSE metric and improves the performance of the baseline model by

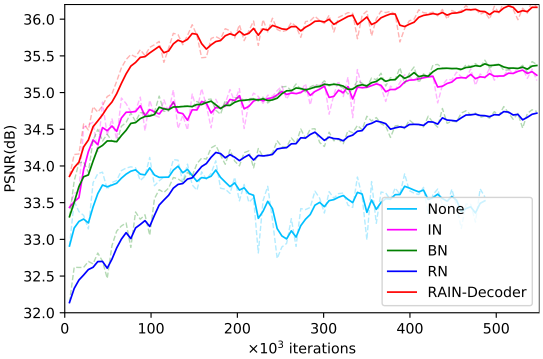

Figure 6. Comparisons of different normalization methods on PSNR metric. Without normalization (labeled by *None*), the model performance heavily deteriorates.

39.68, 339.08 in MSE, fMSE, respectively.

Qualitative comparisons. We proceed to take a closer look at model performance and provide qualitative comparisons with the previous competing methods. From the sample results in Fig. 4, it can be easily observed that our method better integrates the foreground objects into the background image, achieving much better visual consistency compared to other methods. For instance, in the second row of Fig. 4, the background image is underexposed, while the foreground objects (balloons) are much brighter, leading to unrealistic visual results. Both DIH and DoveNet cannot adjust the foreground to be compatible with the dim backgrounds, while S^2AM generates the least realistic result. Our RainNet achieves more photorealistic results with context consistency by adaptively learning the *style* features from the background and applying to the foreground objects. Fig. 5 gives another three typical samples picked from 99 real composited images evaluated in [32]. Although there is no ground truth image as a reference, we can still observe significant improvements of visual *style* consistency achieved by our approach.

5.2. Ablation study

In this section, we conduct comprehensive ablation studies to demonstrate the effectiveness of our RAIN module. Different from Sec. 5.1, we resort to three alternative measures (*i.e*., foreground L1 norm (fL1), PSNR, and SSIM [36]) for quantitative evaluation.

Efficacy of RAIN. We first investigate the performance gain brought by our RAIN module compared to other normalization methods, *i.e*., RN, IN, and BN. To begin with, we apply RN to the baseline model and observe stable model training curves and better performance than that without noralization layers (See in Table 3 and Fig. 6). Note that RN only performs batch normalization for the background (foreground) features within all background (foreground) regions, respectively. This operation splits the background and foreground features and prevents the network from propagating information from the background to the foreground, thus cannot generalize well in image harmonization tasks.

We proceed to add IN and BN to the baseline. As can be found in Table 3 and Fig. 6 (the purple and green curves), the baseline+IN/BN outperforms the baseline method and baseline+RN by a large margin. Potential explanations can be analyzed from two aspects. On one hand, feature normalizing operations can help to stabilize and benefit the training process of deep neural networks, yielding better convergence. On the other hand, performing feature normalization with IN or BN enables the foreground features to be modified by the mean and variance statistically measured from both the foreground features and the background features. Therefore, the model can learn to adjust the visual properties of the foreground objects somehow.

Furthermore, we replace the normalization layer in the decoder network with RAIN while setting the normalization layer to IN in the encoder, then train the network under the same settings. The results are plotted in Fig. 6 (red curve). Obviously, thanks to our novel RAIN module, the model with RAIN-Decoder outperforms other normalization methods and achieves the best performance on average.

Which layer to add RAIN? In order to exploit the best implementation strategy for RAIN, we conduct experiments by gradually adding and removing the RAIN layers in the RainNet network. Here we compare several variants that are boosted by RAIN module in different convolutional stages (more variants and comparisons are presented in the supplementary materials). Note that in the middle layers of the generator, the spatial size of convolutional features decreases significantly. For instance, when we resize the foreground mask to 4×4, the valid pixels of the foreground mask are rather rare. Under these circumstances, our RAIN downgrades to Instance Normalization. So we gradually remove RAIN layers from the 4 outermost layers in the encoder and decoder. **(a) Baseline+RAIN-Decoder**: we add RAIN layer to the decoder and IN to the encoder. **(b) Baseline+RAIN-Encoder**: in contrast to **(a)**, we use RAIN a layer only for the encoder and use IN for the decoder. **(c) Baseline+RAIN-k**: we add k (k=1,2,3,...) RAIN layers to the outermost four layers of the encoder and decoder, and IN to the remaining layers. The quantitative comparison results are provided in Table 3 and Fig. 7. Our observations can be summarized as follows:

1) Baseline+RAIN-Encoder achieves comparable performance of that with IN, while Baseline+RAIN-Decoder outperforms RAIN-Encoder by a large margin. The differences indicate the better choices of RAIN for the decoder and IN for the encoder.

2) Starting from Baseline+RAIN-Decoder, we decrease the number of RAIN layers in the decoder, while adding as

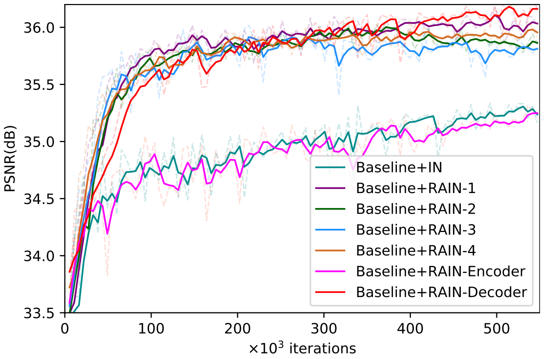

Figure 7. Comparisons of different implementation strategies of RAIN on PSNR metric.

Method	Input	DIH [32]	S^2AM [3]	DoveNet [2]	RainNet
Total votes	113	203	193	226	**354**
Preference	10.4%	18.6%	17.7%	20.8%	**32.5%**

Table 4. Comparisons between our method and other competing methods under user study.

many RAIN layers to the outermost parts of the encoder, *i.e*., Baseline+RAIN-4. The model attains dropped performance but still better than Baseline+IN.
3) Baseline+RAIN-1 slightly outperforms Baseline+RAIN-2, Baseline+RAIN-3, and Baseline+RAIN-4 by minor improvements. However, when compared to IN, BN, and RN, the improvements brought by our RAIN are significant.

From the experimental results, we conclude that adopting RAIN in the decoder and IN in the encoder or using the similar structure as Baseline+RAIN-k are better choices. One probable reason is that some visual-consistency related features (*e.g*., color tone, illumination *etc*.) are likely to be related to the low-level features extracted in the shallow layers of convolutional neural networks, so the layers that are closest to the network's input and output impose greater impacts on estimation error. Another reason is that the deployment of the RAIN in the symmetrical layers of the encoder and decoder helps the concatenated features have the same mean-variance in the background and foreground regions, which is helpful for the filters to stabilize the training and converge to better performance.

Adding RAIN to previous work. To apply RAIN in existing methods, we conduct experiments with DIH [32]. We first implement DIH (with segmentation branch) in Pytorch [24] and then train the basic network. In order to add RAIN to DIH, we replace BN with IN in the encoder, and RAIN with BN in the harmonization decoder. The performance of DIH model reaches to 33.36dB of PSNR while the new model with RAIN achieves 33.84dB (+0.48dB). Detailed illustrations can be found in the supplementary materials.

5.3. User study

Table 4 shows the user evaluation results on real-world composited images collected by DIH [32]. Specifically, we invited 11 volunteers to rate and choose the most realistic harmonized images from 5 given images. Those 5 images include the original composite image and its corresponding 4 harmonized versions created by DIH, S^2AM, DoveNet, and Ours. We randomly shuffle the displaying order of 5 images to ensure that the users do not know which model each image belongs to. Each user is asked to evaluate for the whole set (99 images). As shown in the Table 4, RainNet attains more votes than the rest, which demonstrates the effectiveness of the proposed approach.

5.4. Discussions and limitations

Discussions. Obviously, benefiting from RAIN module, RainNet achieves a higher PSNR score and lower estimation error than previous DoveNet [2] by 1.37dB and 12.07, respectively. Although we found that parts of these improvements are attributed to our generator settings, in which we only learn to modify the foreground image and copy the background pixels from the input, thus reducing the error of the background, we attain lower foregroud estimation errors (fMSE). fMSE is fair for all methods. Furthermore, comparing to IN, RainNet remarkably improves the performance of a baseline model and achieves the best scores on average, which demonstrates the superiority of the proposed RAIN module.
Limitations. Despite the improvements, our proposed approach still faces with two major confusions. First, it is not very clear why applying RAIN only in the encoder brings little improvement. Second, our model will soften the sharp foreground object and reduce the visual style discrepancy in the samples with dark background and sharp foreground objects. Future investigation in these issues should be required.

6. Conclusion

In this paper, we propose to solve the visual *style* inconsistency problem in image harmonization and present a simple yet effective Region-aware Adaptive Instance Normalization (RAIN) module, which outperforms previous normalization methods by a large margin. We have also exploited the best implementation choice of RAIN for the baseline network. Moreover, we demonstrate the efficacy of RAIN by applying RAIN into existing networks, *e.g*., DIH, and observe performance gains over these models.

Acknowledgements. This work was supported in part by MoE-China Mobile Research Fund Project under Grant MCM20180702, National Key R&D Project of China under Grant 2019YFB1802701, and Shanghai Key Laboratory of Digital Media Processing and Transmissions.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8394–8403, 2020.

[3] 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.

[4] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Adv. Neural Inform. Process. Syst.*, pages 6594–6604, 2017.

[5] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *Int. Conf. Learn. Represent.*, 2017.

[6] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Int. Conf. Comput. Vis.*, pages 1301–1310, 2017.

[7] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001.

[8] Michael Elad and Peyman Milanfar. Style transfer via texture synthesis. *IEEE Trans. Image Process.*, 26(5):2338–2351, 2017.

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2414–2423, 2016.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, pages 2672–2680, 2014.

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, pages 1501–1510, 2017.

[12] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Eur. Conf. Comput. Vis.*, pages 172–189, 2018.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1125–1134, 2017.

[15] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. *ACM Trans. Graph.*, 25(3):631–637, 2006.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Eur. Conf. Comput. Vis.*, pages 694–711. Springer, 2016.

[17] Micah K Johnson, Kevin Dale, Shai Avidan, Hanspeter Pfister, William T Freeman, and Wojciech Matusik. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Trans. Vis. Comput. Graph.*, 17(9):1273–1285, 2010.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Jean-Francois Lalonde and Alexei A Efros. Using color compatibility for assessing image realism. In *Int. Conf. Comput. Vis.*, 2007.

[20] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Eur. Conf. Comput. Vis.*, pages 702–716. Springer, 2016.

[21] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5074–5083, 2020.

[22] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Int. Conf. Comput. Vis.*, pages 10551–10560, 2019.

[23] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2337–2346, 2019.

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch, 2017.

[25] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH*, pages 313–318. ACM New York, NY, USA, 2003.

[26] François Pitié and Anil Kokaram. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. *4th European Conference on Visual Media Production*, 2007.

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

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015.

[30] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Trans. Graph.*, 29(4):1–10, 2010.

[31] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. *Int. J. Comput. Vis.*, 103(2):178–189, 2013.

[32] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3789–3797, 2017.

[33] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, and Ming-Hsuan Yang. Sky is not the limit: semantic-aware sky replacement. *ACM Trans. Graph.*, 35(4):149–1, 2016.

[34] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Int. Conf. Mach. Learn.*, volume 1, page 4, 2016.

[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[37] Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018.

[38] Han Xue, Jun Ling, Li Song, Rong Xie, and Wenjun Zhang. Realistic talking face synthesis with geometry-aware feature transformation. In *IEEE Int. Conf. Image Process.*, pages 1581–1585. IEEE, 2020.

[39] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and improving the realism of image composites. *ACM Trans. Graph.*, 31(4):1–10, 2012.

[40] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. Region normalization for image inpainting. In *AAAI*, pages 12733–12740, 2020.

[41] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Int. Conf. Comput. Vis.*, pages 9459–9468, 2019.

[42] Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. In *Eur. Conf. Comput. Vis.*, 2020.

[43] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *Int. Conf. Comput. Vis.*, pages 3943–3951, 2015.