Region-aware Adaptive Instance Normalization for Image Harmonization Supplementary Material

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

A. Overview

In this supplementary, we provide implementation details in Sec. B, including the detailed architecture of attention block, and model training objective. We also conduct more ablation studies to exploit better applying strategy of the proposed RAIN method (Sec. C). More comparison results on real composite images are presented in Sec. D. Finally, we discuss the failure case in Sec. E

B. Implementation Details

B.1. Attention Block

Attention block has been proven to bring noticeable improvements to the simple U-Net architecutre [1, 2]. Following the prior work, we add three attention blocks in the decoder part for baseline network (the structure of generator is presented in Section 3 of the main paper). The detailed structure of attention block is presented in Fig. 1.

Specifically, in each attention block, we take the concatenation of the encoder feature and the decoder feature $F_{in} \in \mathbb{R}^{C\times H\times W}$ as the input of the block. To fuse the concatenated features, we use an 1×1 convolutional layer and a Sigmoid activation function σ to acquire coefficients map, which is denoted as $W \in \mathbb{R}^{C\times H\times W}$. Then we acquired the modulated feature F_{out} by multiplying the concatenated features by the map in element-wise manner:

$$F_{out} = W \circ F_{in}, \tag{1}$$

where $\circ$ denotes the element-wise multiplication.

B.2. Improving Image Composites

In this paper, we define the composite image as I_c, the foreground mask as M. The harmonization model is denoted by G, and the harmonized image by $\hat{I} = G(I_c, M)$. Our aim is to optimize the model G to make $\hat{I}$ close to the ground truth image I by a reconstruction loss:

$$\mathcal{L}_{rec}(G, I, I_c, M) = \|G(I_c, M) - I\|_1. \tag{2}$$

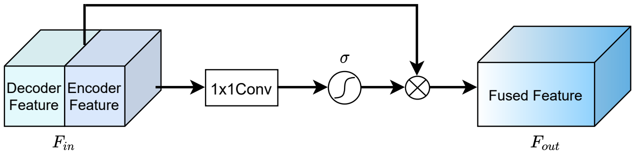


Figure 1. Illustration of the adopted attention block.

Due to the widespread applications of adversarial training in many computer vision tasks, we also adopt adversarial training method and follow the training strategy in [1, 2]. The adversarial loss can be written as follows:

$$\begin{aligned}\mathcal{L}_{adv}(D, I, \hat{I}) = \mathbb{E}_I[\max(0, 1 - D(I))] \\ + \mathbb{E}_{\hat{I}}[\max(0, 1 + D(\hat{I}))],\end{aligned} \tag{3}$$

and

$$\mathcal{L}_{adv}(G, I_c, M) = -\mathbb{E}_{I_c}[D(G(I_c, M))], \tag{4}$$

where D tries to distinguish between natural-realistic images I and harmonized samples $\hat{I}$, while G aims to generate samples that look similar to the real observations. Introducing adversarial loss can, in theory, learn the model G that generate images as realistic as the real [3, 4].

Besides the global discriminator, we also adopt the setting of domain verification loss [1], which has been proved to bring modest improvements for image harmonization. Specifically, we construct real and fake samples by grouping image pairs of $(I \circ M, I \circ (1 - M))$ and $(\hat{I} \circ M, \hat{I} \circ (1 - M))$, respectively. To perform domain-oriented optimization, we first utilize a domain encoder E_D to obtain the feature representations of the foreground image and the background image. We denote the feature representations as l_f and l_b, respectively. Equally, $\hat{l}_f$ and $\hat{l}_b$ are extracted from harmonized image $\hat{I}$ by the same domain encoder. To acquire domain verification loss, following [1], we use one more domain discriminator D_v which incorporate the domain encoder E_D and measure the similarity of l_f and l_b

Type	Method	Index of feature normalization layer													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
I	Baseline + RAIN-Decoder-1	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**
	Baseline + RAIN-Decoder-2	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**	**R**
	Baseline + RAIN-Decoder-3	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**	**R**	**R**
	Baseline + RAIN-Decoder-4	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**	**R**	**R**	**R**
	Baseline + RAIN-Decoder	IN	IN	IN	IN	IN	IN	IN	**R**	**R**	**R**	**R**	**R**	**R**	**R**
	Baseline + RAIN-Encoder	**R**	**R**	**R**	**R**	**R**	**R**	**R**	IN	IN	IN	IN	IN	IN	IN
II	Baseline + RAIN-1	**R**	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**
	Baseline + RAIN-2	**R**	**R**	IN	IN	IN	IN	IN	IN	IN	IN	IN	IN	**R**	**R**
	Baseline + RAIN-3	**R**	**R**	**R**	IN	IN	IN	IN	IN	IN	IN	IN	**R**	**R**	**R**
	Baseline + RAIN-4	**R**	**R**	**R**	**R**	IN	IN	IN	IN	IN	IN	**R**	**R**	**R**	**R**
	Baseline + RAIN-5	**R**	**R**	**R**	**R**	**R**	IN	IN	IN	IN	**R**	**R**	**R**	**R**	**R**
	Baseline + RAIN-6	**R**	**R**	**R**	**R**	**R**	**R**	IN	IN	**R**	**R**	**R**	**R**	**R**	**R**
III	Baseline + RAIN-Inner-3	IN	IN	IN	IN	**R**	**R**	**R**	**R**	**R**	**R**	IN	IN	IN	IN
	Baseline + RAIN-Inner-4	IN	IN	IN	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	IN	IN	IN
	Baseline + RAIN-Inner-5	IN	IN	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	IN	IN

Table 1. Designing choices of RAIN. IN: Instance Normalization, **R**: RAIN.

by:

$$D_v(I, M) = l_f \cdot l_b, \tag{5}$$

where · means the inner product of two vectors.

Afterward, we measure the domain verification loss as follows:

$$\begin{aligned}\mathcal{L}_v(D_v, I, \hat{I}, M) = \mathbb{E}_I[\max(0, 1 - D_v(I, M))] \\ + \mathbb{E}_{\hat{I}}[\max(0, 1 + D_v(\hat{I}, M))],\end{aligned} \tag{6}$$

$$\mathcal{L}_v(G, I_c, M) = -\mathbb{E}_{I_c}[D_v(G(I_c, M), M)]. \tag{7}$$

By using domain verification loss, the discriminator is encouraged to distinguish similar domain features for positive foreground-background pairs from negative foreground-background pairs.

In our experiments, D and D_v share the same structure as [1], and we apply the well-know spectral normalization [6] for two discriminators to stabilize training procedure. The domain encoder utilizes Partial Convolutions [5] to extract domain code for regions with irregular shape, avoiding information leakage from unmasked regions.

Our full objective is:

$$\mathcal{L}(D, D_v, I, \hat{I}, M) = \lambda_1 \mathcal{L}_{adv}(D, I, \hat{I}) + \lambda_2 \mathcal{L}_v(D_v, I, \hat{I}, M), \tag{8}$$

$$\begin{aligned}\mathcal{L}(G, I, I_c, M) = \lambda_1 \mathcal{L}_{adv}(G, I_c, M) + \lambda_2 \mathcal{L}_v(G, I_c, M) \\ + \lambda_3 \mathcal{L}_{rec}(G, I, I_c, M),\end{aligned} \tag{9}$$

where $\lambda_1 = \lambda_2 = 1$, and $\lambda_3 = 100$.

C. More ablation studies

In this section, we conduct more experiments to validate the efficacy of our method. Theoretically, our RAIN module can be applied in any layers of the basic network. In this section, we train our baseline with different designing strategies of applying our RAIN module.

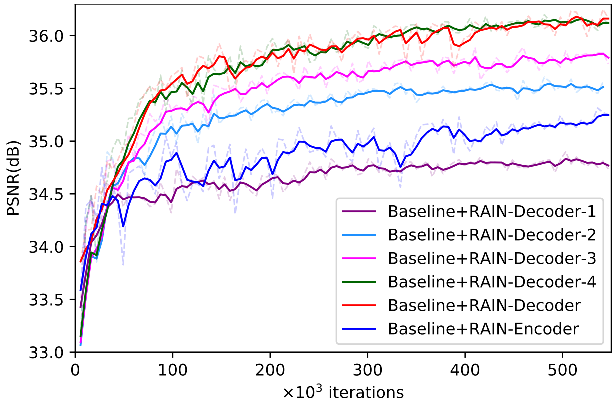
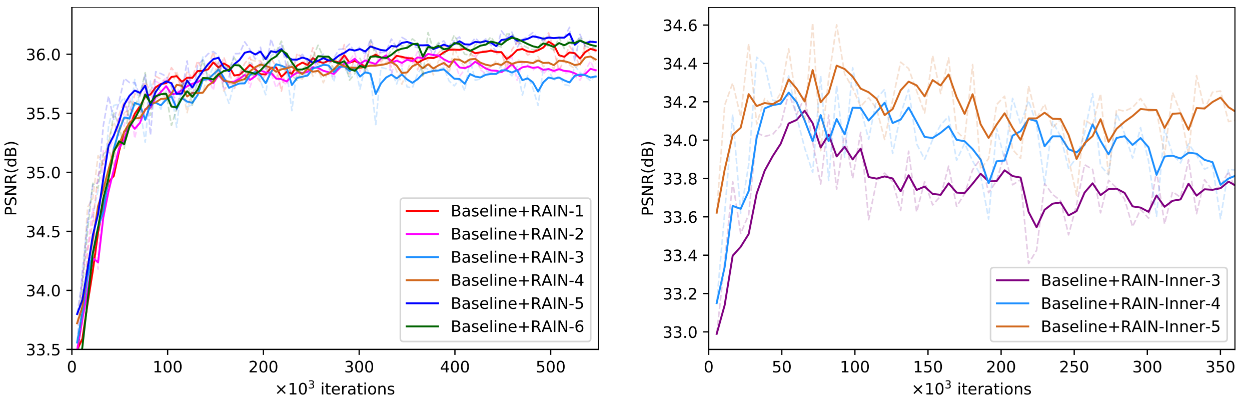


Figure 2. Convergence curves of methods in Type I. We only add RAIN layers to the decoder or encoder. Better viewed in color with zoom in.

As presented in Table 1, we exploit better implementations of RAIN module by designing three main types of structures of the basic network: **I**) we gradually replace IN with RAIN in the decoder or encoder; **II**) we add RAIN modules to the outermost layers of the network; **III**) we add RAIN modules to the innermost layers of the network. We conduct these experiments with fixed random seed for better reproduction. The convergence results are presented in Fig. 2 and 3.

From Fig. 2, it is obvious that more RAIN layers in the decoder brings more stable training process and better convergence performance. When we only add one RAIN layer at the last normalization layer of the network, *i.e.*, Baseline+RAIN-Decoder-1, we attain the least PSNR re-



(a)

(b)

Figure 3. Convergence curves on PSNR metric. (a) Type II: we add RAIN modules to the outermost layers of the network; (b) Type III: we add RAIN modules to the innermost layers of the network. Better viewed in color with zoom in.

sults (purple curve). As we add more RAIN layers to the decoder, we obtain noticeable improvements. Another interesting conclusion is that adding more RAIN layers to the decoder brings no benifits when we have already added four RAIN layers in the decoder(green curve and red curve). This may be ascribed to the reasons that when the feature size is small enough, *e.g.*, 4×4 or 8×8, our RAIN module will equal to IN. Therefore, equal performances of Baseline+RAIN-Decoder and Baseline+RAIN-Decoder-4 are observed in our experiments.

In Fig. 3, we visualize the convergence curves of methods within type II and III. It is clear that using symmetric normalization method for the network benefits the model optimizing process and leads to better convergent performance. Specifically, from subfigure 3(a), Baseline+RAIN-5 and Baseline+RAIN-6 outperform other methods, while Baseline+RAIN-1 performs slightly better than the Baseline+RAIN-2/3/4.

In the right figure of Fig. 3, we visualize the convergent curves of those methods with RAIN modules inserted in the middle part of the network. It can be observed that Baseline+RAIN-Inner-5 is much better than Baseline+RAIN-Inner-3 but much worse than Baseline+RAIN-5. To analyze the observation, note that the visual style defined in this work is close to image visual properties, including illumination, color temperature, saturation, hue, and texture, *etc*. In other words, visual properties in image harmonization task are more related to low-level feature representations learnd by convolutional network in the first few layers of the encoder and the last few layers in the decoder. Therefore, adding the same amount of RAIN layers to the middle layers of the baseline network is less competitive than that to the outermost layers.

Adding RAIN to previous work. We first re-implemented DIH in PyTorch and pretrain the whole model for the first 400 epochs. Then we freeze the segmentation branch and optimize the encoder and harmonization branch for another 600 epochs. To add RAIN to DIH, we replace BN with IN in the encoder, and BN with RAIN in the harmonization decoder. Note that we only predict the foreground objects like RainNet does. In Fig. 4, we present the performance curve of DIH and its variant. It can be easily conclude that RAIN module stabilizes the optimizing process and brings significant improvements to existing network.

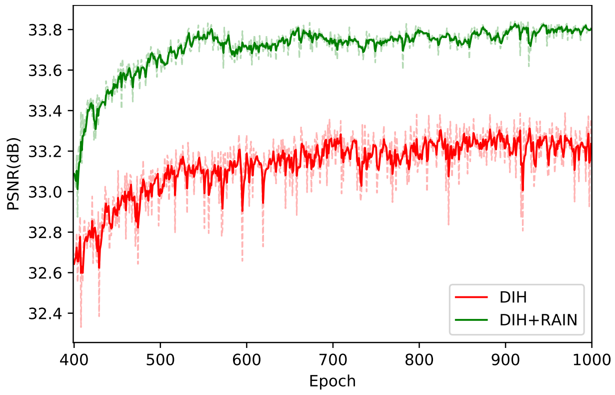


Figure 4. Convergence curves of DIH [7] and DIH+RAIN on PSNR metric.

D. Results on real composite images

In this section, we present the sample results of real composite image used in [7] and [1] and compare our method to other competing methods in Fig. 5, 6 and 7. As can be found, our method chieves better visual consistency between the foreground and the background images and outperforms other methods in most cases.

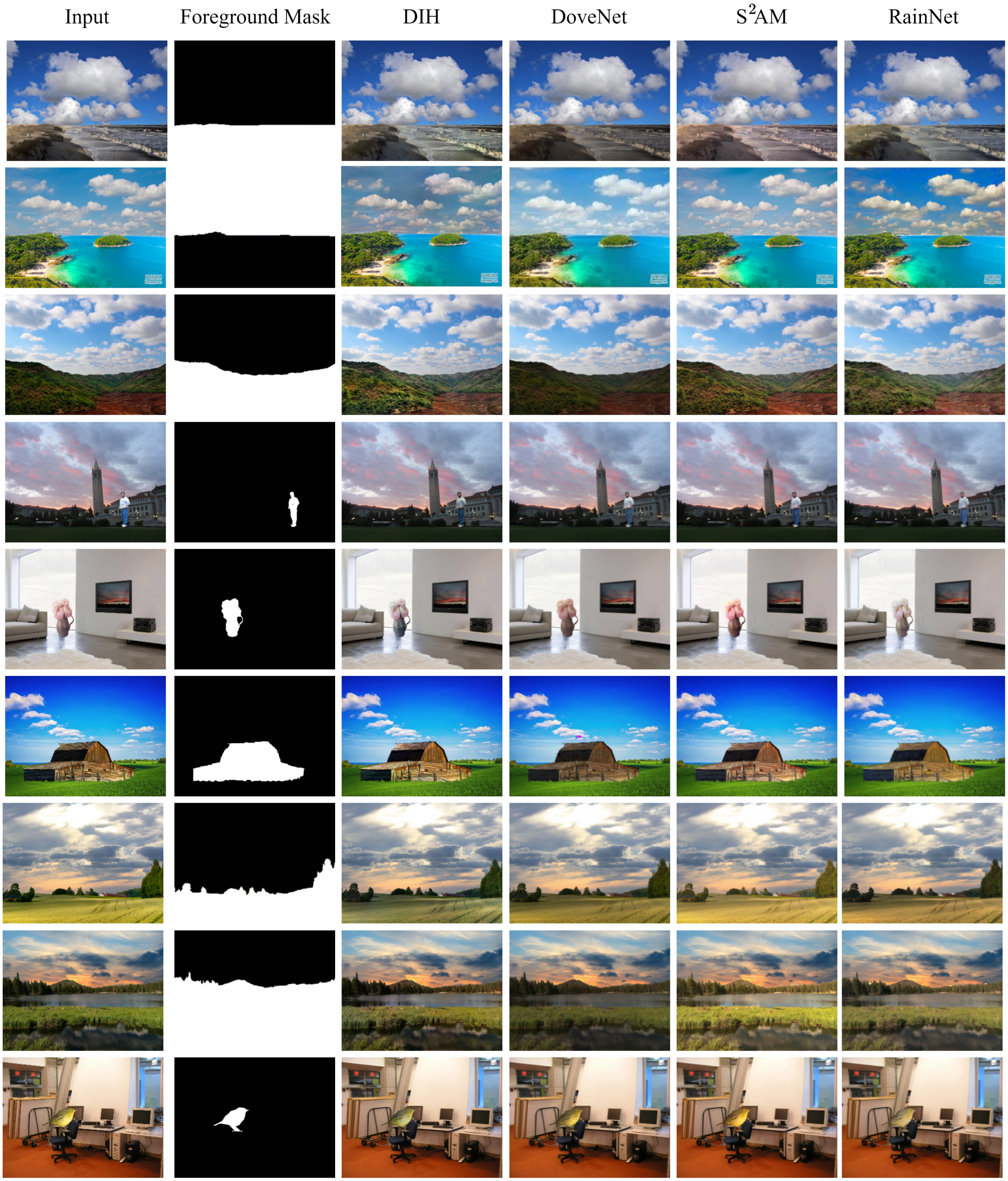


Figure 5. **Example results on real composite images.**. We present real composite images, foreground mask, three state-of-the-art methods, and the proposed model. The samples are taken from the testing dataset of [7]. Our method achieves better harmonized visual results than competing methods.

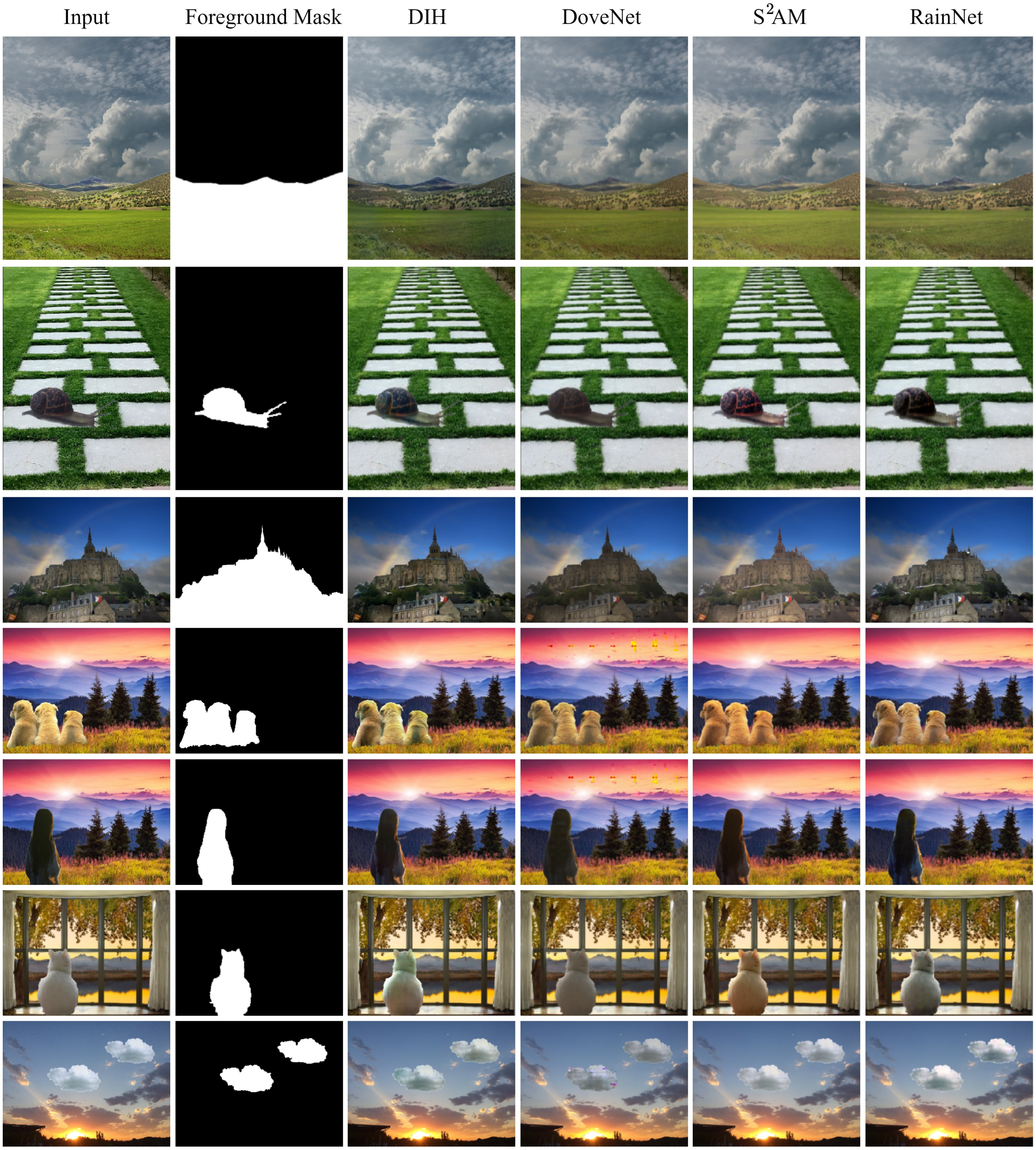


Figure 6. **Example results on real composite images.**. We present real composite images, foreground mask, three state-of-the-art methods, and the proposed model. The samples are taken from the testing dataset of [7]. Our method achieves better harmonized visual results than competing methods.

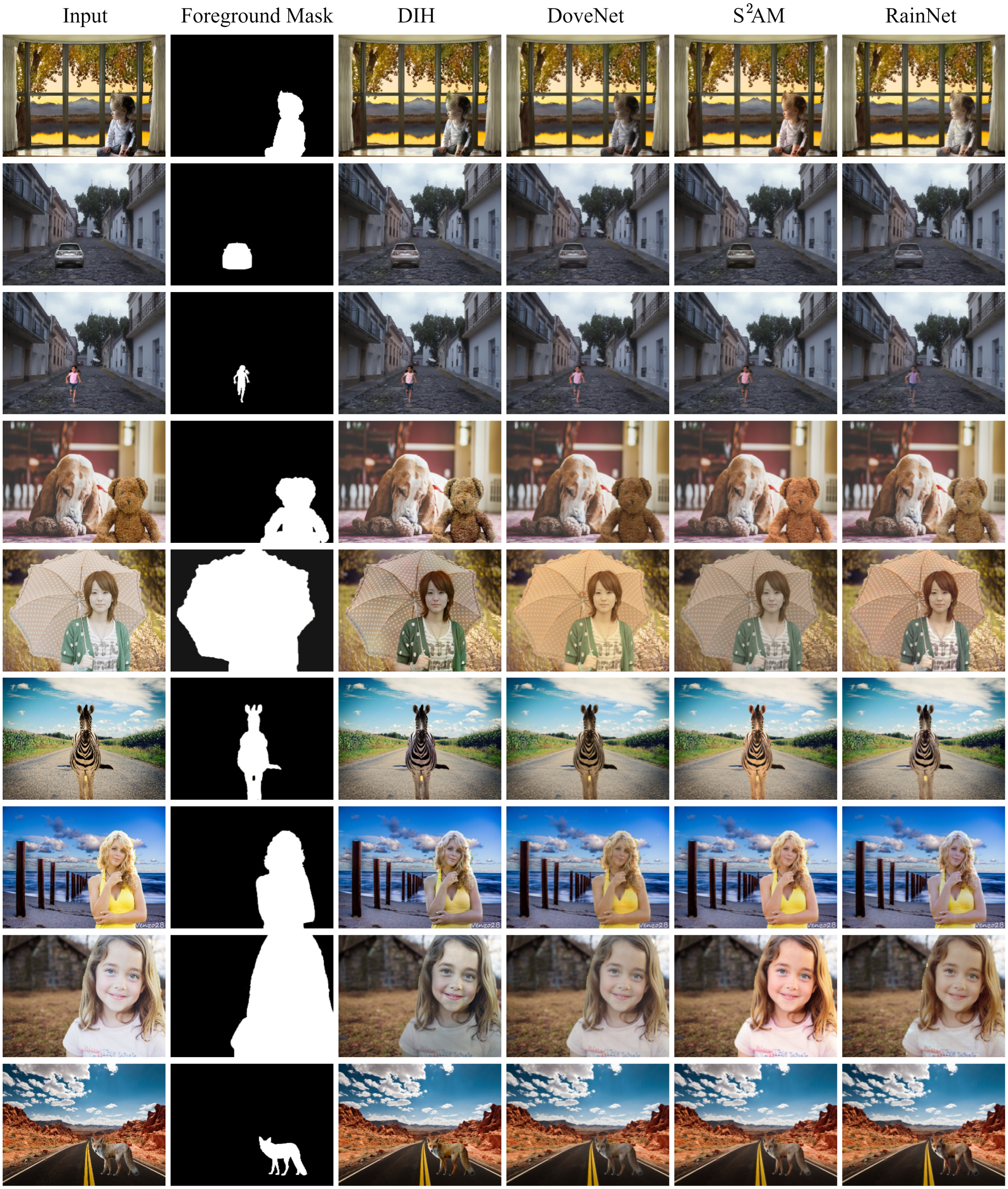


Figure 7. **Example results on real composite images.**. We present real composite images, foreground mask, three state-of-the-art methods, and the proposed model. The samples are taken from the testing dataset of [7]. Our method achieves better harmonized visual results than competing methods.

E. Failure case

As has been refered in the main submission, the proposed RainNet fails to deal with the case of images with a blurred background with a sharp foreground object. Figure 8 shows an example. As can be found in Fig. 8, S^{2}AM performs better than the proposed RainNet and other methods. However, these methods also fail to produce consistent boundary, introducing observable visual artifacts and deteriorating the visual quality. Our future work should focus on this issue.

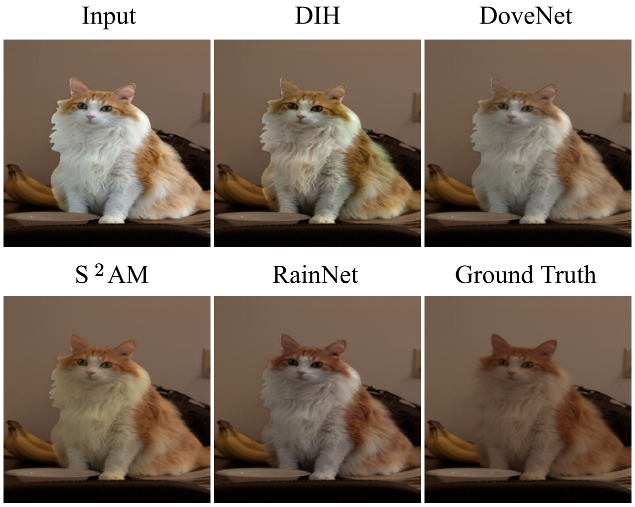


Figure 8. **Failure case.**. The proposed RainNet fails to harmonize the composite image with sharp foreground object and dim or blurry background image.

References

[1] 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.

[2] 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] 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.

[4] 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.

[5] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Eur. Conf. Comput. Vis.*, pages 85–100, 2018.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2018.

[7] 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.