Supplementary Materials for "SIDNet: Learning Shading-aware Illumination Descriptor for Image Harmonization"

Zhongyun Hu, Ntumba Elie Nsampi, Xue Wang, and Qing Wang, *Senior Member, IEEE*

I. Implementation Details

In this section, we mainly introduce network architectures and training details.

A. Network Architecture

U-Net based network. We start with our U-Net based network depicted in Tab. I. This network is used for both the Shading Bases Module, the Albedo Estimation Module and the Rendering Module. Different from that in the main paper, the Albedo Estimation Module here includes h and t for simplicity. The U-Net network first processes its input using two convolutional layers. These two layers are used to extract general images features while preserving the image's original resolution. Following the two convolutional layers, we use a series of Residual Dense Blocks (RDBs). The dense connections in RDBs allow for information to be received from each previous layer. Consequently, RDBs are able to preserve the number of channels at a given layer, which facilitates collective features to be reused. We use a total of two RDBs per layer, for a total of three layers. After each pair of RDBs, we double the number of channels and down-sample the feature maps by a factor of two. Prior to the down-sampling operation, a 1x1 convolution is used to double the number of channels.

Our Global Attention Block (GAB), which we use around the network's bottleneck, is composed of 6 transformer layers. Note that the GAB is only used in the Shading Bases Module. The GAB takes features from the encoder, which are first collapsed across spatial dimensions. The resulting tensor has a size of $N \times S$, where $N = \left(\frac{H}{16} \times \frac{W}{16}\right)$, and S is the dimension of the embedding. For each transformer layer, we use 8 attention heads and an internal representation of size 512. The decoder part of our U-Net takes the features from the GAB as input. These features are first reshaped back to their pre-GAB shape and then fed to the decoder. The decoder is composed of up-sampling blocks followed by convolutional layers. Skip connections are used between the encoder and decoder.

The default U-Net based network architecture depicted in Tab. I is adopted as the Shading Bases Module. The Albedo Estimation Module and the Rendering Module are slightly different from the default U-Net based network architecture. Both the Shading Bases Module and the Albedo Estimation Module take an unharmonized input image of three channels. The Rendering Module accepts multiple inputs of twenty five channels (i.e., a shading image of three channels, albedo feature of sixteen channels, an input unharmonized image of three channels, and a background image of three channels). The output for the Shading Bases Module is a set of shading bases of K channels, where K is set to 32. The outputs of the Albedo Estimation Module and the Rendering Module are an albedo RGB image and a rendered RGB image, respectively. Moreover, the **chns** and **n-l** of both the Albedo Estimation Module and the Rendering Module within the U-Net based network are half those of the Shading Bases Module, with the aim of reducing parameters.

Illumination network. The illumination network is used for both the Illumination Encoder Module and the Background Illumination Estimation Module. The illumination network depicted in Tab. II shares the same structure as that of our U-Net based network's encoder. But the illumination network does not make use of the GAB. We also use an additional pair of RDBs. Following the RDBs, we use a 1x1 convolutional layer. We then use an adaptive average pooling layer to average each feature map separately. Finally, three 1x1 convolutional layers, which are equivalent to three fully-connected layers, are used to process the resulting feature maps and produce the illumination descriptor.

B. Training Details

Training SOTA models. All competing methods are re-trained from scratch on our dataset following instructions provided by the authors. Except for [1], the other methods [2]–[4] are trained with an image resolution of 512x512. Note that the model [1] only supports training on images of a specific resolution (i.e., 256x256), and the other models [2]–[4] only support training on images of particular resolutions (e.g., 256x256 or 512x512). Each model is trained on a single RTX TITAN. The training losses of these methods are shown in Fig. 1. We report their results on the test set when the training losses converge. During the testing phase, only [1] is tested with a resolution of 256x256, and the other models are tested with a resolution of 512x512.

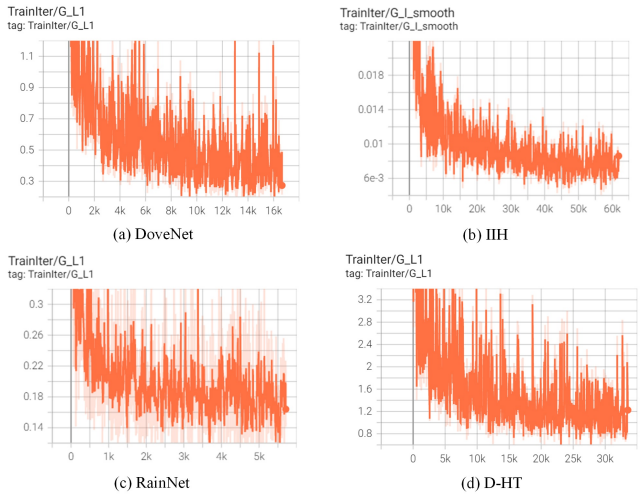
Training our model. Our model is trained in two main stages. During the first stage, we aim to learn our illumination descriptors. The second stage is trained with the objective to infer the illumination descriptor from the background images. Our model is trained with an original resolution of 480x640. For both stages, we use Adam [5] with a learning rate of 1e-4 and betas = $(0.9, 0.999)$ as our optimization algorithm. In

TABLE I
OUR DEFAULT U-NET BASED NETWORK ARCHITECTURE. WE USE THIS ARCHITECTURE FOR BOTH THE **SHADING MODULE** AND **ALBEDO ESTIMATION MODULE**. HERE, **K** IS THE KERNEL SIZE, **S** THE STRIDE, **D** THE KERNEL DILATION, **P** THE IMAGE PADDING. **CHNS** AND **INPUT** ARE THE NUMBER OF INPUT/OUTPUT CHANNELS AND THE INPUT TO THE LAYER. FOR RDBS AND THE GAB, **N-L** IS THE NUMBER OF LAYERS, **G-R** IS THE GROWTH RATE, **E-S** IS THE EMBEDDING SIZE, **N-H** IS THE NUMBER OF HEAD AND **I-R** IS THE INTERNAL REPRESENTATION SIZE. δ DENOTES RECTIFIED LINEAR ACTIVATION FUNCTION. NOTE THAT THE GAB IS ONLY USED IN THE **SHADING MODULE**.

Layers	Parameters						
	k	**s**	**d**	**p**	**chns**	**input**	
conv3x3$_1$	3	1	1	1	4/32	I_{in}	
conv3x3$_2$	3	1	1	1	32/32	δ(conv3x3$_1$)	
Pool$_1$	2	1	-	0	32/32	δ(conv3x3$_2$)	
	k	**s**	d	**chns**	**n-l**	**g-r**	**input**
RDB$_{1,1}$	3	1	1	32/32	8	16	Pool$_1$
RDB$_{1,2}$	3	1	1	32/32	8	16	RDB$_{1,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_1$	1	1	1	0	32/64	RDB$_{1,2}$	
Pool$_2$	2	1	-	0	64/64	conv1x1$_1$	
	k	**s**	**d**	**chns**	**n-l**	**g-r**	**input**
RDB$_{2,1}$	3	1	1	64/64	8	16	Pool$_2$
RDB$_{2,2}$	3	1	1	64/64	8	16	RDB$_{2,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_2$	1	1	-	0	64/128	RDB$_{2,2}$	
Pool$_3$	2	1	-	0	128/128	conv1x1$_2$	
	k	**s**	**d**	**chns**	**n-l**	**g-r**	**input**
RDB$_{3,1}$	3	1	1	128/128	8	16	Pool$_3$
RDB$_{3,2}$	3	1	1	128/128	8	16	RDB$_{3,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_3$	1	1	-	0	128/256	RDB$_{3,2}$	
Pool$_4$	2	1	-	0	256/256	conv1x1$_3$	
	e-s	**n-l**	**n-h**	**i-r**	**input**		
GAB	256	6	8	512	Pool$_4$		
	k	**s**	**d**	**p**	**chns**	**input**	
conv3x3$_3$	3	1	1	1	256/128	GAB	
Up$_1$	-	-	-	-	-	conv3x3$_3$	
conv1x1$_4$	1	1	-	0	512/256	Up$_1$ + RDB$_{3,2}$ + RDB$_{3,1}$	
conv3x3$_4$	3	1	1	1	256/128	δ(conv1x1$_4$)	
Up$_2$	-	-	-	-	-	δ(conv3x3$_4$)	
conv1x1$_5$	1	1	-	0	256/128	Up$_2$ + RDB$_{2,2}$ + RDB$_{2,1}$	
conv3x3$_5$	3	1	1	1	128/64	δ(conv1x1$_5$)	
Up$_3$	-	-	-	-	-	δ(conv3x3$_5$)	
conv1x1$_6$	1	1	-	0	128/64	Up$_3$ + RDB$_{1,2}$ + RDB$_{1,1}$	
conv3x3$_6$	3	1	1	1	64/32	δ(conv1x1$_6$)	
Up$_4$	-	-	-	-	-	δ(conv3x3$_6$)	
conv3x3$_7$	3	1	1	1	64/32	Up$_4$ + conv3x3$_2$	
conv3x3$_8$	3	1	1	1	32/32	δ(conv3x3$_7$)	

TABLE II
OUR ILLUMINATION NETWORK ARCHITECTURE. WE USE THIS ARCHITECTURE FOR BOTH THE **BACKGROUND ILLUMINATION ESTIMATION MODULE** AND **ILLUMINATION ENCODER MODULE**.

Layers	Parameters						
	k	**s**	**d**	**p**	**chns**	**input**	
conv3x3$_1$	3	1	1	1	3/32	I_{in}	
conv3x3$_2$	3	1	1	1	32/32	δ(conv3x3$_1$)	
Pool$_1$	2	1	-	0	32/32	δ(conv3x3$_2$)	
	k	**s**	d	**chns**	**n-l**	**g-r**	**input**
RDB$_{1,1}$	3	1	1	32/32	8	16	Pool$_1$
RDB$_{1,2}$	3	1	1	32/32	8	16	RDB$_{1,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_1$	1	1	1	0	32/64	RDB$_{1,2}$	
Pool$_2$	2	1	-	0	64/64	conv1x1$_1$	
	k	**s**	**d**	**chns**	**n-l**	**g-r**	**input**
RDB$_{2,1}$	3	1	1	64/64	8	16	Pool$_2$
RDB$_{2,2}$	3	1	1	64/64	8	16	RDB$_{2,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_2$	1	1	-	0	64/128	RDB$_{2,2}$	
Pool$_3$	2	1	-	0	128/128	conv1x1$_2$	
	k	**s**	**d**	**chns**	**n-l**	**g-r**	**input**
RDB$_{3,1}$	3	1	1	128/128	8	16	Pool$_3$
RDB$_{3,2}$	3	1	1	128/128	8	16	RDB$_{3,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_3$	1	1	-	0	128/256	RDB$_{3,2}$	
Pool$_4$	2	1	-	0	256/256	conv1x1$_3$	
	k	**s**	**d**	**chns**	**n-l**	**g-r**	**input**
RDB$_{4,1}$	3	1	1	128/128	8	16	Pool$_4$
RDB$_{4,2}$	3	1	1	128/128	8	16	RDB$_{4,1}$
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_4$	1	1	-	0	128/256	RDB$_{4,2}$	
AdaptiveAvgPool	-	-	-	-	256/256	conv1x1$_4$	
	k	**s**	**d**	**p**	**chns**	**input**	
conv1x1$_5$	1	1	-	1	256/128	AdaptiveAvgPool	
conv1x1$_6$	1	1	-	1	128/64	δ(conv1x1$_5$)	
conv1x1$_7$	1	1	-	1	64/96	δ(conv1x1$_6$)	



(a) DoveNet (b) IIH (c) RainNet (d) D-HT

Fig. 1. The training losses of SOTA models.

addition, to fairly compare our results against these competing methods, during the quantitative evaluation phase we resize all our images to a resolution of 512x512, so as to match them with those produced by competing methods.

II. MORE EXPERIMENTS AND RESULTS

In this section, we present more experiments and results to validate the effectiveness of our method.

A. More Qualitative Results

As shown in Fig. 2, Fig. 3, Fig. 4, and Fig. 5, we provide more qualitative results on four sub-datasets. We execute our method on representative images and compare our results against those produced by competing methods.

Our method produces results that are more realistic, where the foreground is more compatible with the background image. For example, in third column of Fig. 2, the foreground object appears to be taken from a sunny scene, where the main illumination is behind the lady. But the background image indicates that its primary illumination is located on the right side of the image, as is evident from shadows of the trees in the background. As shown in close-up details, our method produces results that are consistent with the background illumination. In contrast, the harmonized results produced by competing methods such as RainNet and DoveNet still retain the original illumination effects.

In the second column of Fig. 3, the foreground object of the input composite image appears to be illuminated from the left of the image, as is evident on the boy's face. Given that the background is cloudy, the resulting harmonized foreground should be under a smooth illumination. DoveNet, RainNet and Guo et al's method tend to preserve some of the original illuminations on the foreground object and struggle to remove them completely. In addition, D-HT produces over-smooth results. This may be due to the weighted average operation in the self-attention mechanism. Lalonde and Efros' method produces greenish results that differ from the ground truth. Our results are more realistic and closer to the ground truth in terms of both color and shading.

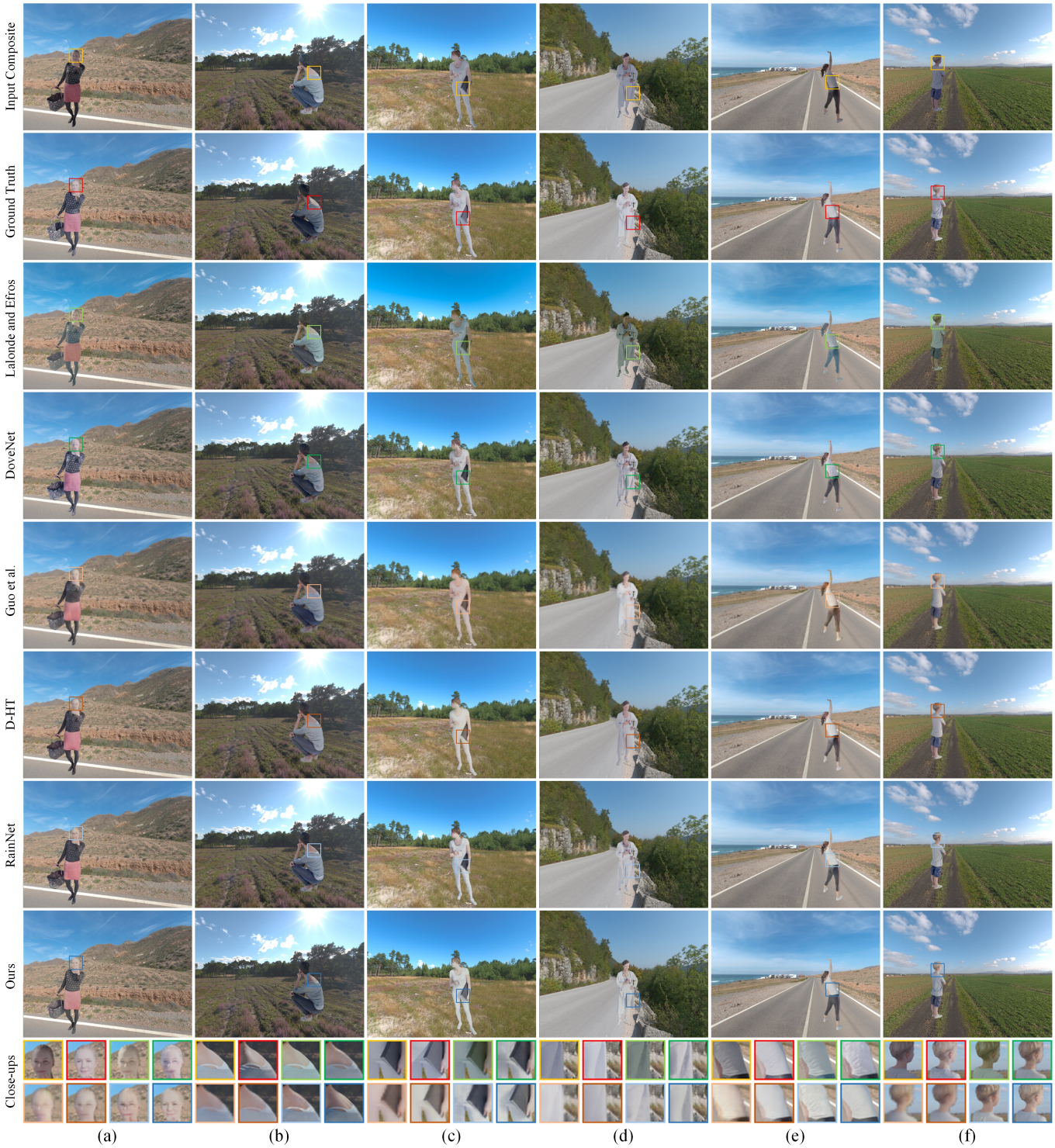


Fig. 2. More qualitative results of different methods on the *sunny* test set. We show representative examples with close-up details focusing on shading variation, color and brightness.

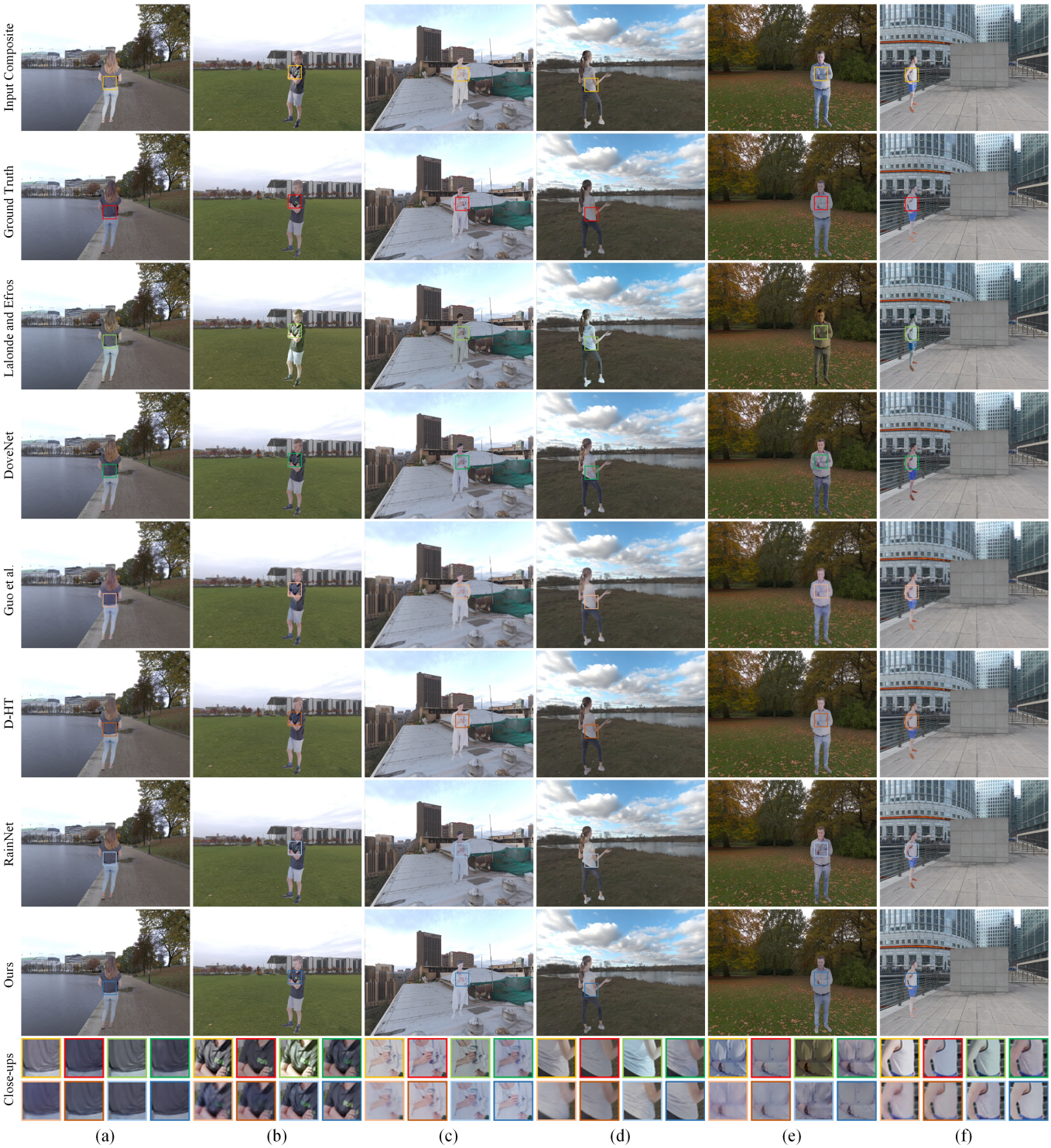


Fig. 3. More qualitative results of different methods on the *cloudy* test set. We show representative examples with close-up details focusing on shading variation, color and brightness.

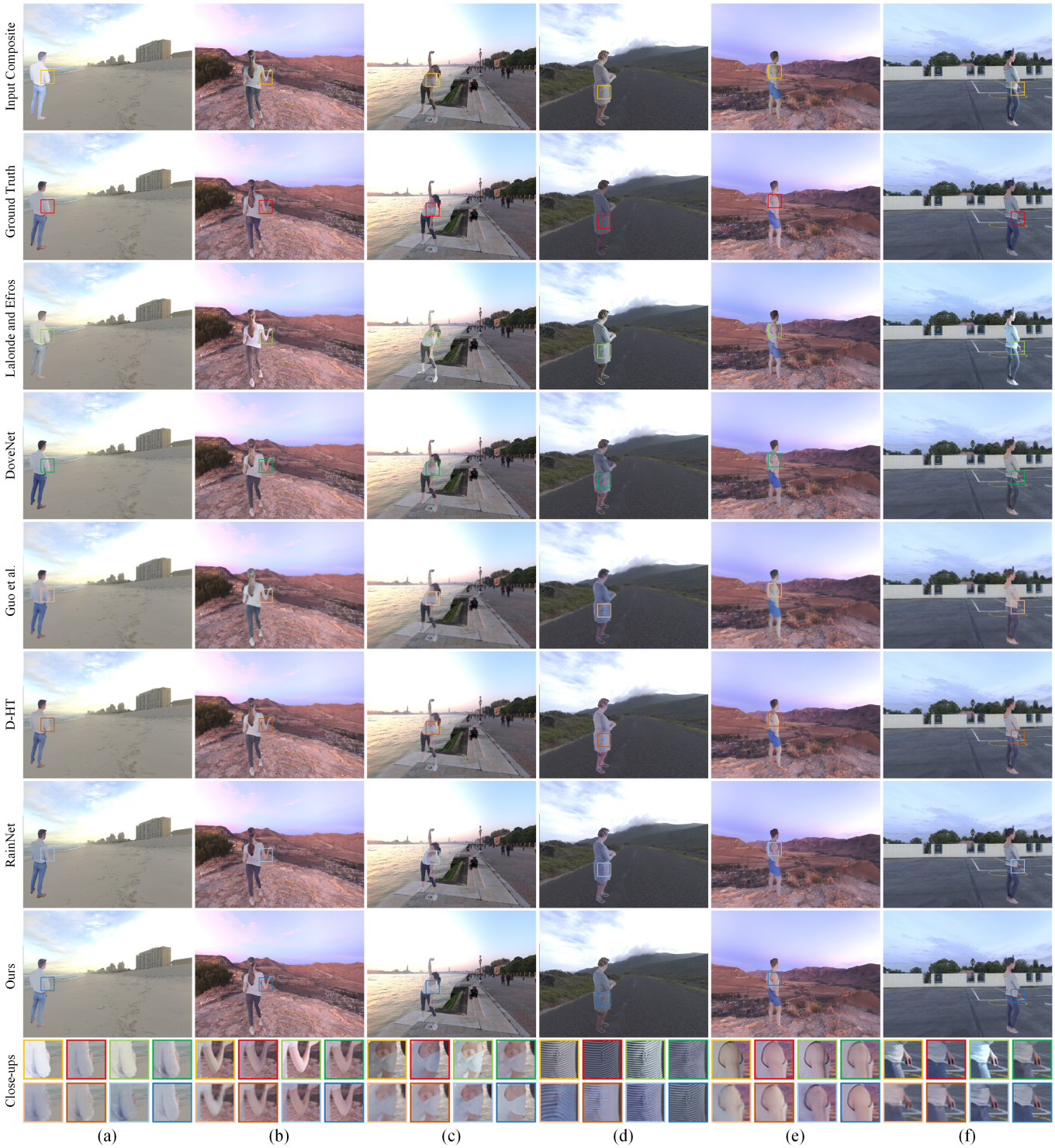


Fig. 4. More qualitative results of different methods on the *sunrise/sunset* test set. We show representative examples with close-up details focusing on shading variation, color and brightness.

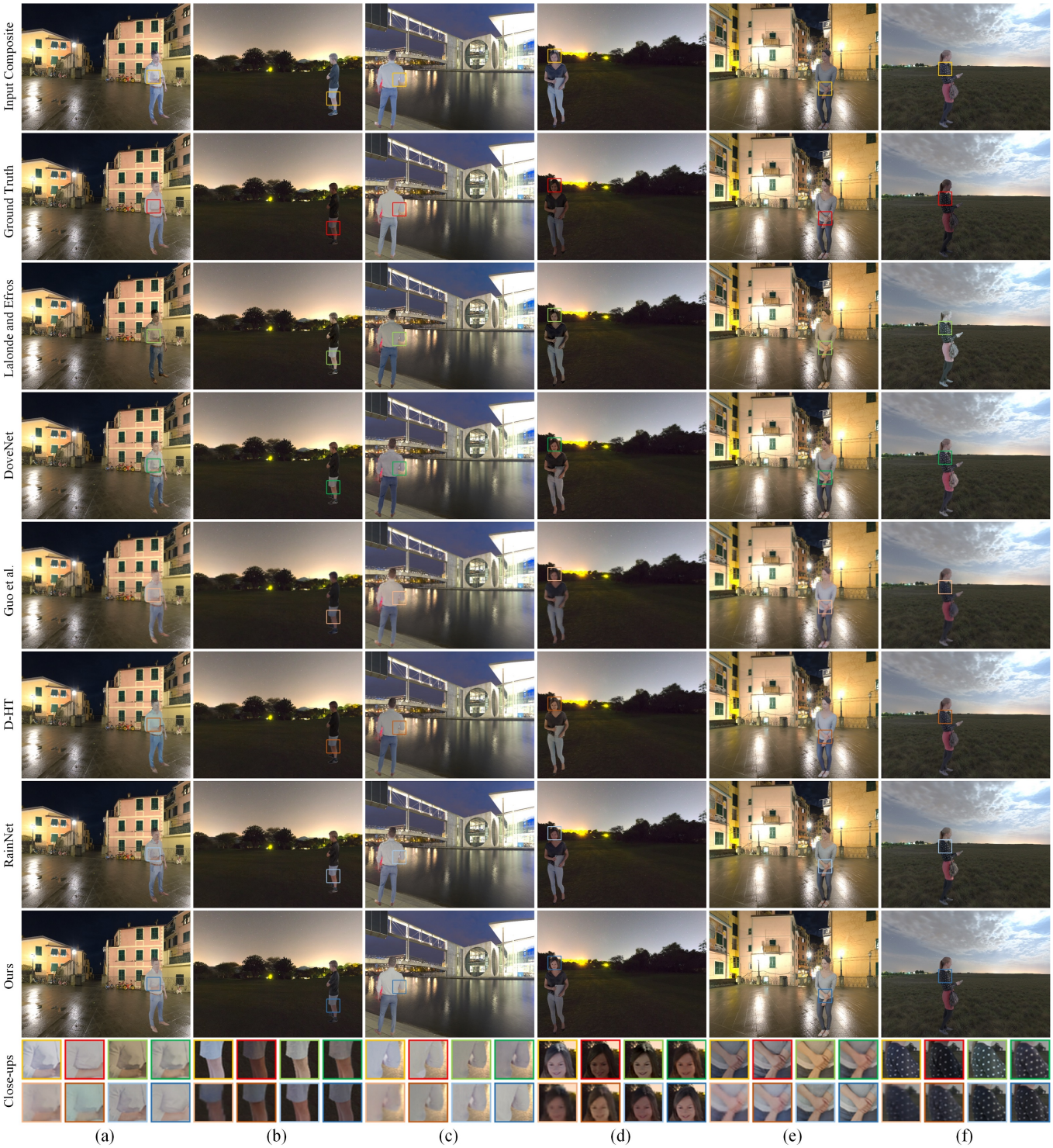


Fig. 5. More qualitative results of different methods on the *night* test set. We show representative examples with close-up details focusing on shading variation, color and brightness.

TABLE III
ABLATION STUDY ON GAB.

	fPSNR↑	fSSIM↑
Shading Bases Module w/o GAB	30.75	0.928
Shading Bases Module	**31.35**	**0.936**

B. More Results from User Study

In Fig. 6, we present more results from user study. First of all, the state-of-the-art methods [2], [3], [6], [7] mainly change the brightness and color of the foreground, but cannot change the shading of the foreground. In contrast, our method can not only perceive the illumination in the background image but also generate the corresponding foreground shading, as shown in close-up details of Fig. 6(a)(b)(c). Second, these methods tend to transfer the brightness and color of background objects to the foreground without perceiving the illumination information in the scene. Note that the pixel values in the background image reflect the combined effect of background objects and illumination. From a physical point of view, we should extract the illumination information in the background scene rather than only the brightness and color information of the background images. Taking the Fig. 6(e) as an example, Lalonde [6], RainNet [3] and DIH-GAN [7] transfer the color of the grass to the foreground to make it appear green, which is unreasonable.

C. Ablation Study on GAB

Quantitative results of ablation study on GAB are shown in Tab. III. It can be seen that the performance of the Shading Bases Module ($K = 32$) with GAB is better than that without GAB, in which fPSNR increases by 0.6dB and fSSIM increases by 0.008. Note that the reported metrics (fPSNR, fSSIM) are obtained by comparing the results generated by our Shading Module against the corresponding ground truth shading images.

REFERENCES

[1] Z. Guo, H. Zheng, Y. Jiang, Z. Gu, and B. Zheng, "Intrinsic image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 16 367–16 376.
[2] W. Cong, J. Zhang, L. Niu, L. Liu, Z. Ling, W. Li, and L. Zhang, "DoveNet: Deep image harmonization via domain verification," in *CVPR*, 2020.
[3] J. Ling, H. Xue, L. Song, R. Xie, and X. Gu, "Region-aware adaptive instance normalization for image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9361–9370.
[4] Z. Guo, D. Guo, H. Zheng, Z. Gu, B. Zheng, and J. Dong, "Image harmonization with transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 870–14 879.
[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[6] J.-F. Lalonde and A. A. Efros, "Using color compatibility for assessing image realism," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.
[7] Z. Bao, C. Long, G. Fu, D. Liu, Y. Li, J. Wu, and C. Xiao, "Deep image-based illumination harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 542–18 551.

Fig. 6. More qualitative results of different methods on the real data. We show representative examples with close-up details focusing on shading variation. For example, as shown in close-up details of Fig. 6(a)(b)(c), our proposed method is able to generate plausible shading that conforms to the background illumination. In contrast, the SOTAs still preserve the original illumination effects. Note that Bao et al. [7] only provide the results with a 256x256 resolution.