PCT-Net: Full Resolution Image Harmonization Using Pixel-Wise Color Transformations

Supplementary Material

Julian Jorge Andrade Guerreiro[1,*] Mitsuru Nakazawa[2] Björn Stenger[2]

[1]The University of Tokyo [2]Rakuten Institute of Technology, Rakuten Group, Inc.

julianguerreio@outlook.de, {mitsuru.nakazawa, bjorn.stenger}@rakuten.com

This document contains additional explanations and evaluations of the proposed image harmonization method. We provide additional information regarding evaluation metrics, explaining why we use the foreground mean squared error (fMSE) in section A. In section B, we show results on low-resolution (256×256 pixels) images, which has been the standard until recently, allowing us to compare our proposed approach with a larger number of methods. In section C, we investigate the parameter maps for the ViT backbone explaining the negative effect of employing a smoothing regularization. In section D, we explore different pixel-wise color transformations (PCTs) and compare them to the affine transform proposed in the paper. For completeness, we provide the images from the user study in section E, and additional qualitative results in section F.

A. Evaluation Metrics

Deciding on suitable metrics to evaluate and compare different image harmonization methods is critical. The Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are two commonly used metrics to asses the reconstruction error. Given a ground truth image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$ and a predicted image $\hat{\boldsymbol{I}} \in \mathbb{R}^{H\times W\times 3}$ of the width W and height H, we first define the Squared Error (SE) of an image as:

$$\delta(\hat{\boldsymbol{I}}, \boldsymbol{I}) := \sum_{i,j} (\hat{\boldsymbol{I}}_{i,j} - \boldsymbol{I}_{i,j})^2. \tag{1}$$

Based on the SE, we can easily define the MSE as the average over all pixels:

$$\text{MSE}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = \frac{\delta(\hat{\boldsymbol{I}}, \boldsymbol{I})}{HW}, \tag{2}$$

The PSNR is defined in terms of the MSE as:

$$\text{PSNR}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = 10 \log \frac{I_{\text{max}}^2}{\text{MSE}(\hat{\boldsymbol{I}}, \boldsymbol{I})}, \tag{3}$$

where I_{max} represents the largest possible pixel value, set to $I_{\text{max}} = 255$.

A.1. Considering the Foreground Mask

In the image harmonization task, we are only interested in a particular target region of the image indicated by a binary mask $\boldsymbol{M} \in \{0,1\}^{H\times W}$. However, since the mask regions vary in size across the dataset, images with larger target regions contribute more to the MSE than images with smaller regions. Therefore, instead of averaging across the entire image, the foreground Mean Squared Error (fMSE) considers only the area of the mask, denoted as $A(\boldsymbol{M})$. It is given as

$$\text{fMSE}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = \frac{\delta(\boldsymbol{M} \odot \hat{\boldsymbol{I}}, \boldsymbol{M} \odot \boldsymbol{I})}{A(\boldsymbol{M})}, \tag{4}$$

where $\odot$ represents the Hadamard product. If the predicted images are not modified outside the mask region, which we assume, then $\delta(\boldsymbol{M} \odot \hat{\boldsymbol{I}}, \boldsymbol{M} \odot \boldsymbol{I}) = \delta(\hat{\boldsymbol{I}}, \boldsymbol{I})$, and thus $\text{fMSE}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = \frac{HW}{A(\boldsymbol{M})}\text{MSE}(\hat{\boldsymbol{I}}, \boldsymbol{I})$.

Based on the same reasoning as the fMSE, [6] defines the fPSNR as:

$$\text{fPSNR}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = 10 \log \frac{I_{\text{max}}^2}{\text{fMSE}(\hat{\boldsymbol{I}}, \boldsymbol{I})}. \tag{5}$$

However, we argue that a fPSNR is not required for performance comparison across different datasets, as it only differs from the PSNR value by an added constant independent of the predicted image. Using $\text{fMSE}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = \alpha\text{MSE}(\hat{\boldsymbol{I}}, \boldsymbol{I})$ with $\alpha = \frac{HW}{A(\boldsymbol{M})}$, fPSNR is expressed as:

$$10 \log \frac{I_{\text{max}}^2}{\alpha\text{MSE}(\hat{\boldsymbol{I}}, \boldsymbol{I})} = \text{PSNR}(\hat{\boldsymbol{I}}, \boldsymbol{I}) - 10 \log \alpha. \tag{6}$$

As a result, averaging the fPSNR across the dataset, yields the same value as averaging the PSNR and subtracting the

Table 1. Image-wise vs. Pixel-wise averaged MSE and fMSE. *As the dimension as well as the number of target pixels can differ for different images, we compute average errors over images and over individual pixels. While the values change, the ranking of the methods remains unchanged. The best results are marked in bold, the second best are underlined.*

Method	$\text{MSE}_{\text{Images}} \downarrow$	$\text{MSE}_{\text{Pixels}} \downarrow$	$\text{fMSE}_{\text{Images}} \downarrow$	$\text{fMSE}_{\text{Pixels}} \downarrow$
Input	177.99	299.96	1462.45	2243.14
Harmonizer [8]	27.62	24.78	339.23	183.86
DCCF [15]	24.65	22.97	302.89	171.77
Ours (CNN)	<u>24.05</u>	<u>21.81</u>	<u>282.77</u>	<u>163.07</u>
Ours (ViT)	**18.80**	**19.32**	**238.27**	**144.44**

average of $10 \log \alpha$, a value defined by the dataset, not by the method. Therefore, the fPSNR does not provide any new insights when comparing two methods.

A.2. Averaging Across the Dataset

Given K different images in a dataset, two different averaging ways are to average K MSE values for each image, or to average over all pixels in the dataset. These two values are identical when all images are of the same size, which cannot be assumed in our case using full-resolution image datasets. In the main paper we show the average with respect to all images, consistent with the literature. In Table 1 we provide the results averaged over pixels. We observe that using pixel-average error metrics the ranking of the methods in terms of performance does not change.

B. Results on Low-Resolution Images

In addition to the full-resolution evaluation provided in the main paper, we compare our methods with other image harmonization methods on lower-resolution images. Like prior work, the lower-resolution images are obtained by downsampling the images in the iHarmony4 dataset [4] to 256×256 pixels. The results are summarized in Table 2. Again, our ViT-based method outperforms prior work on the complete iHarmony4 dataset and only shows slightly worse results on the HDay2night subset, which represents the smallest part in the dataset.

C. Parameter Maps

Since smoothing regularization did not improve the performance of our ViT-based model in the main paper, we further investigate the differences between the parameter maps of the model with and without smoothing regularization. We visualize the full-resolution parameter maps for an example image in Figure 1. Since the parameters have 12 dimensions, representing affine coefficients, we arrange the first 9 parameter maps representing matrix coefficients according to their position in the transformation matrix. We further add the three parameter maps representing the translation part on the right side. Therefore, the first row shows all the parameters that influence the first component of the image, which in our case is the red value. Since the values of the parameter map can vary, we provide two different visualization methods. The first one, normalizes the parameter values according to the largest value across all the parameter dimensions, while the second map normalizes the image according to the largest value in the respective channel. As can be seen in Fig. 1 (a) and (b), both parameter maps without and with smoothing regularization look quite similar. However, the parameter map with smoothing regularization, as expected, overall exhibits more smoothness among the parameter values. Taking a closer look at the parameter map without smoothing regularization, a block pattern emerges that can be explained by the decoder architecture. The encoder splits the image into smaller patches, and then the output patches are rearranged and processed by a single deconvolutional layer. This block pattern is visible in parameter maps, however, it is not visible in the final results. Since these blocks are caused by the architecture, they cannot be smoothed using a regularization loss. Instead, the smoohing regularization loss further restricts the potential output space and pushes the network to find a solution that is as uniform as possible. This decrease in flexibility leads to less details, which eventually hurts the overall performance.

D. More PCT functions

To investigate the general effect of PCT functions on the harmonization results, we test different functions in addition to the PCT functions introduced in the main paper. The results are summarized in Table 3, where the mathematical definitions for the multiplication, addition and linear PCT function can be found in Eq. 7, 8 and 9, respectively. The affine PCT function corresponds to the PCT function used in the main paper.

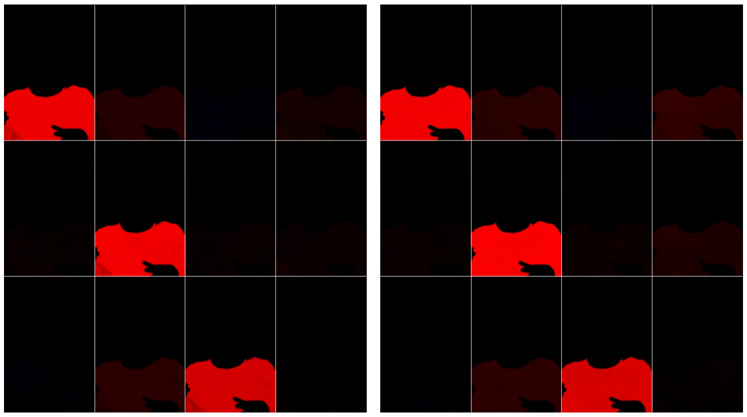
$$h_{\text{mul}}(\boldsymbol{p}; \boldsymbol{\theta}) = (p_1\theta_1, p_2\theta_2, p_3\theta_3)^T, \tag{7}$$

$$h_{\text{add}}(\boldsymbol{p}; \boldsymbol{\theta}) = (p_1 + \theta_1, p_2 + \theta_2, p_3 + \theta_3)^T, \tag{8}$$

$$h_{\text{linear}}(\boldsymbol{p}; \boldsymbol{\theta}) = \boldsymbol{W_\theta p}, \quad \boldsymbol{W_\theta} = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 \end{pmatrix}. \tag{9}$$

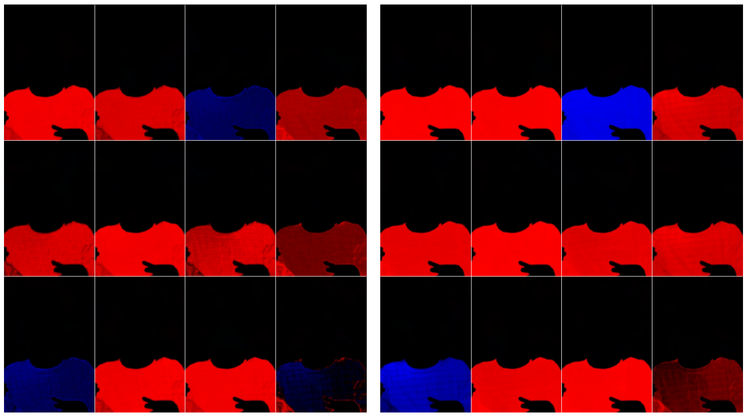
For both CNN and ViT backbones, the results shown in Table 3 underline that the affine PCT function introduced in the main paper clearly outperforms all other functions. It seems that by adding more parameters the network gains more flexibility which eventually improves performance. If

$$\left[\begin{array}{ccc|c} \theta_1 & \theta_2 & \theta_3 & \theta_{10} \\ \theta_4 & \theta_5 & \theta_6 & \theta_{11} \\ \theta_7 & \theta_8 & \theta_9 & \theta_{12} \end{array}\right]$$

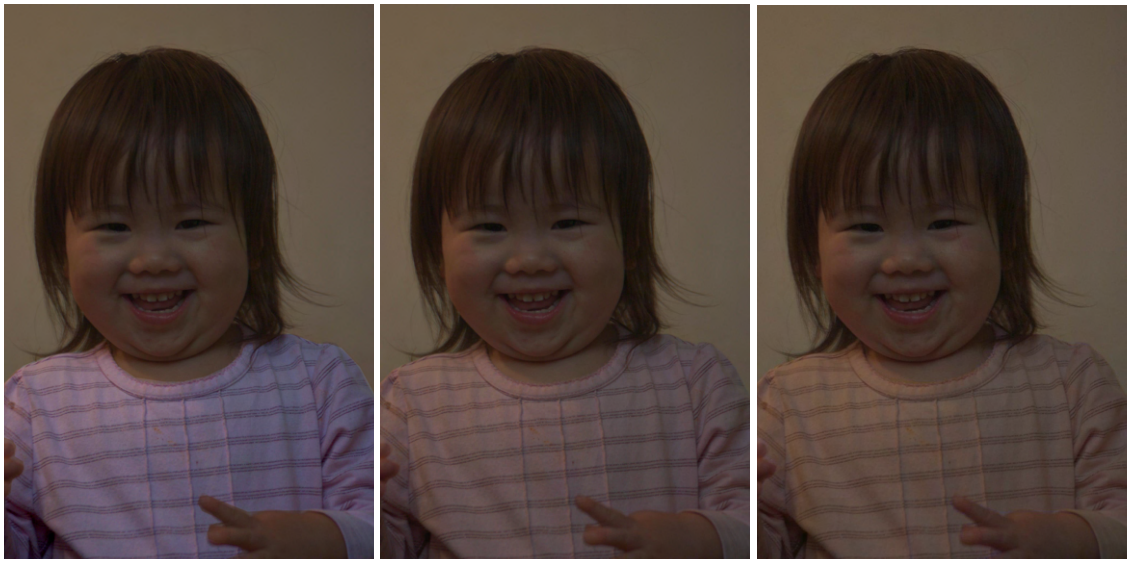


(a) Visualization of parameter maps normalized by the largest parameter value over all channels. *(Left) Arrangement of parameter maps. (Center) PCT-Net (ViT) without smoothing regularization. (Right) PCT-Net (ViT) with smoothing regularization.*

$$\left[\begin{array}{ccc|c} \theta_1 & \theta_2 & \theta_3 & \theta_{10} \\ \theta_4 & \theta_5 & \theta_6 & \theta_{11} \\ \theta_7 & \theta_8 & \theta_9 & \theta_{12} \end{array}\right]$$



(b) Visualization of parameter maps normalized by the largest parameter of each individual channel. *(Left) Arrangement of parameter maps. (Center) PCT-Net (ViT) without smoothing regularization. (Right) PCT-Net (ViT) with smoothing regularization.*



(c) Results. *(Left) Composite image. (Center) Output image without smoothing regularization. (Right) Output image with smoothing regularization.*

Figure 1. Parameter maps for ViT-based models without and with smoothing regularization *In (a) and (b), the 12 parameter maps of an example image are visualized while normalizing by the largest parameter value over all channels, and normalizing by the largest parameter of each individual channel, respectively. Positive parameter values are shown in red, while negative values are shown in blue. In (c), we show the output images created by applying the parameter map without and with smoothing regularization. Images are from [4].*

Table 2. Quantitative performance on the low-resolution iHarmony4 dataset (256×256 pixels). *Ours (ViT) outperforms all of the previous works for the entire dataset as well as in almost all subsets. The best results are marked in bold, the second best are underlined.*

Method	**HAdobe5k subset**			**HCOCO subset**			**HDay2night subset**			**HFlickr subset**			**All (7,404)**		
	fMSE↓	MSE↓	PSNR↑	fMSE↓	MSE↓	PSNR↑	fMSE↓	MSE↓	PSNR↑	fMSE↓	MSE↓	PSNR↑	fMSE↓	MSE↓	PSNR↑
Input	2051.61	345.54	28.16	996.59	69.37	33.94	1409.98	109.65	34.01	1574.37	264.35	28.32	1387.30	172.47	31.63
DoveNet [4]	380.39	52.32	34.34	551.01	36.72	35.83	1075.71	54.05	35.18	827.03	133.14	30.21	549.96	52.36	34.75
BargainNet [12]	279.66	39.94	35.34	397.85	24.84	37.03	835.63	50.98	35.67	698.40	97.32	31.34	405.23	37.82	35.88
Guo [7]	284.21	43.02	35.20	416.38	24.92	37.16	797.04	55.53	35.96	716.60	105.13	31.34	400.29	38.71	35.90
D-HT [6]	265.11	38.53	36.88	299.30	16.89	38.76	704.42	53.01	37.10	515.45	74.51	33.13	320.78	30.30	37.55
D-HT+ [5]	242.56	36.83	37.17	274.66	37.10	36.83	736.58	49.68	36.68	471.06	67.88	33.55	295.56	27.89	37.94
iSSAM [14]	173.96	21.88	38.08	266.19	16.48	39.16	590.97	40.59	37.72	443.65	69.67	33.56	264.96	24.44	38.19
iDIH+HRNet [14]	N/A	21.36	37.35	N/A	14.01	39.64	N/A	50.61	37.68	N/A	60.41	34.03	252.00	22.00	38.31
CDTNet [3]	N/A	20.62	38.24	N/A	16.25	39.15	N/A	36.72	37.95	N/A	68.61	33.55	252.05	23.75	38.23
S2CRNet [11]	N/A	34.91	36.42	N/A	23.22	38.48	N/A	51.67	36.81	N/A	98.73	32.38	N/A	35.58	37.18
Harmonizer [8]	170.05	21.89	37.64	298.42	17.34	38.77	**542.07**	**33.14**	37.56	434.06	64.81	33.63	280.51	24.26	37.84
DCCF [15]	172.49	23.43	37.18	272.10	14.87	39.52	655.46	45.09	**38.08**	411.56	61.42	33.84	265.52	22.05	38.50
Sg-MHH [13]	N/A	22.04	38.64	N/A	13.95	39.14	N/A	43.57	36.86	N/A	59.03	34.00	N/A	21.89	38.38
SCS-Co [8]	165.48	**21.01**	38.29	245.54	13.58	39.88	606.80	41.75	37.83	393.72	55.83	34.22	258.86	21.33	38.75
Ours (CNN)	182.71	23.76	38.83	275.61	16.61	39.35	732.89	49.92	37.34	425.65	66.15	33.82	273.50	24.83	38.54
Ours (ViT)	**157.24**	21.25	**39.97**	**208.26**	**10.72**	**40.78**	654.81	44.74	37.65	**341.10**	**44.30**	**35.13**	**216.25**	**18.16**	**39.85**

Table 3. Effect of PCT function on performance. *We investigate the impact of different PCT functions on the performance by evaluating on the iHarmony4 test set. The results are evaluated on the full resolution images as well as images that were downsampled to* 256×256 *pixels. The best results are marked in bold.*

Backbone	**PCT**	**Full resolution images**			**Downsampled images**		
		fMSE↓	MSE↓	PSNR↑	fMSE↓	MSE↓	PSNR↑
CNN	multiplication	391.80	34.29	37.01	334.32	30.59	37.75
	addition	333.82	29.52	37.24	283.99	25.86	38.27
	linear	333.97	29.22	37.47	290.70	26.53	38.19
	affine	**296.84**	**25.32**	**38.05**	**268.22**	**23.94**	**38.63**
ViT	multiplication	1228.88	141.57	32.04	1173.85	138.01	32.37
	addition	287.36	23.21	38.17	240.93	20.31	39.29
	linear	7594.78	757.18	23.14	741.32	23.32	317.15
	affine	**250.30**	**20.03**	**38.98**	**228.00**	**19.34**	**39.52**

we compare the results of our CNN-based model on the downsampled and full resolution images, we also notice that increasing the number of parameters helps reduce the degradation in performance that is observed when evaluating on full resolution. In the ViT-based model, we notice that using linear and multiplication PCT functions results in significantly worse performance. During training we observed that the optimizer got stuck in a local minimum and was not able to find a reasonable solution. This could probably be fixed by adjusting the learning rate in future experiments. However, considering the experiments conducted for the CNN-based model, we expect that the results would follow the trend observed for the CNN-based model.

E. Images User Study

We provide the 26 images that we used for our user study as well as the images which were predicted by the three different harmonization methods in Fig. 2– 6.

F. More Image Results

We provide more qualitative results in Figure 7 and Figure 8 comparing our results to Harmonizer [10] and DCCF [15] on the iHarmony4 test set. In Figure 7, we show randomly selected images from the HFlickr, Hday2night and HCOCO subsets. Results for the HAdobe5k subset, which contains all the high-resolution images, are shown in Figure 8.

References

[1] unsplash.com. `https://unsplash.com`. 6, 7, 8, 9, 10

[2] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6, 7, 8, 9, 10

[3] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18470–18479, June 2022. 4

[4] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8394–8403, June 2020. 2, 3, 4, 11, 12

[5] Zonghui Guo, Zhaorui Gu, Bing Zheng, Junyu Dong, and Haiyong Zheng. Transformer for image harmonization and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–19, 2022. 4

[6] Zonghui Guo, Dongsheng Guo, Haiyong Zheng, Zhaorui Gu, Bing Zheng, and Junyu Dong. Image harmonization

with transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14870–14879, October 2021. 1, 4

[7] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16367–16376, June 2021. 4

[8] Yucheng Hang, Bin Xia, Wenming Yang, and Qingmin Liao. Scs-co: Self-consistent style contrastive learning for image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19710–19719, June 2022. 2, 4

[9] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. Ssh: A self-supervised framework for image harmonization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4832–4841, October 2021. 6, 7, 8, 9, 10

[10] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. *arXiv preprint arXiv:2207.01322*, 2022. 4, 6, 7, 8, 9, 10, 11, 12

[11] Jingtang Liang, Xiaodong Cun, and Chi-Man Pun. Spatial-separated curve rendering network for efficient and high-resolution image harmonization. *arXiv preprint arXiv:2109.05750*, 2021. 4

[12] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9361–9370, June 2021. 4

[13] Xuqian Ren and Yifan Liu. Semantic-guided multi-mask image harmonization. *arXiv preprint arXiv:2207.11722*, 2022. 4

[14] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1620–1629, January 2021. 4

[15] Ben Xue, Shenghui Ran, Quan Chen, Rongfei Jia, Binqiang Zhao, and Xing Tang. Dccf: Deep comprehensible color filter learning framework for high-resolution image harmonization. *arXiv preprint arXiv:2207.04788*, 2022. 2, 4, 6, 7, 8, 9, 10, 11, 12

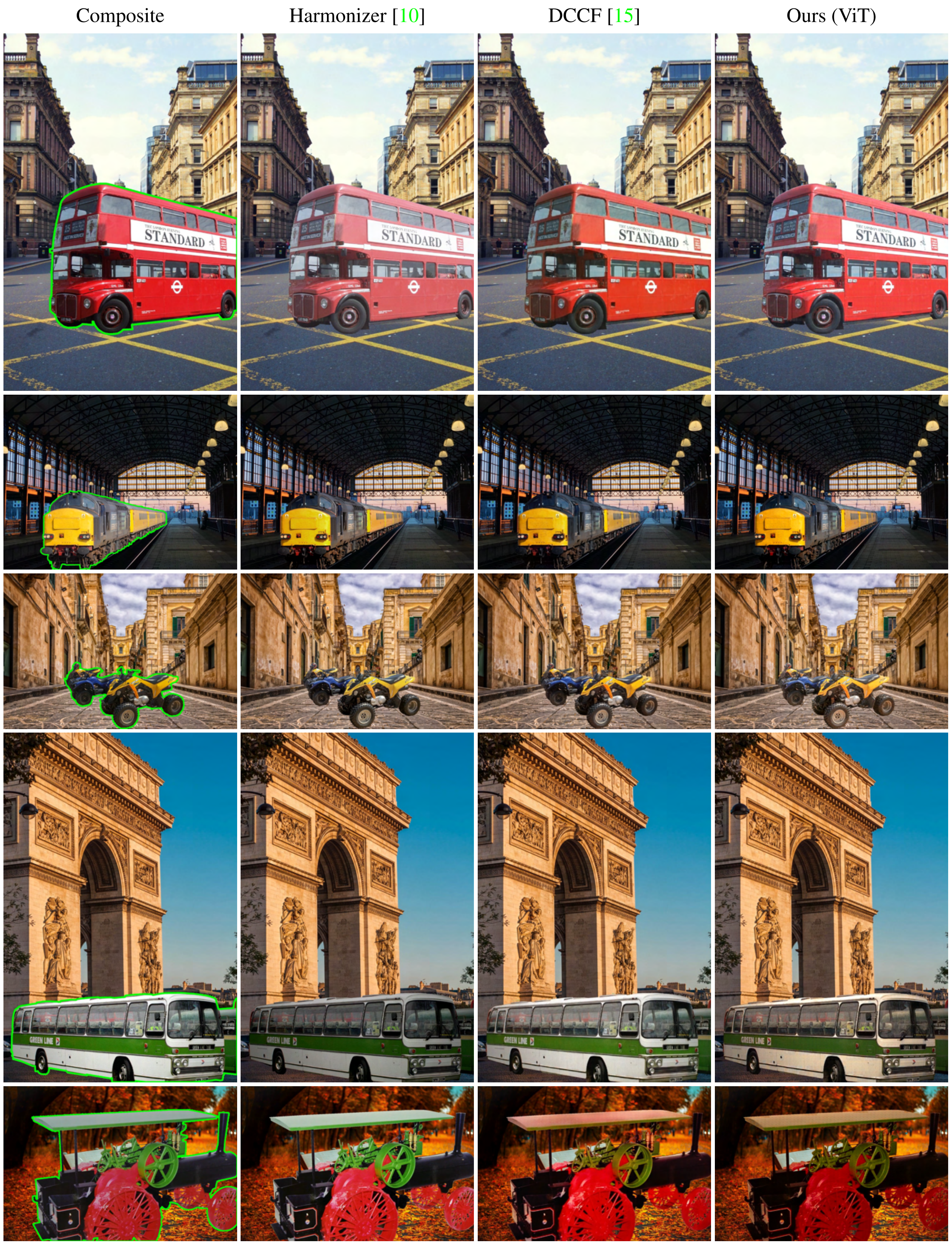


Figure 2. Images used for the user study (No.1-5 of the 26 images). *The foreground region of the composite image is outlined in green. Foreground images are from the BIG dataset [2] or the RealHM dataset [9]. Background images are from [1].*

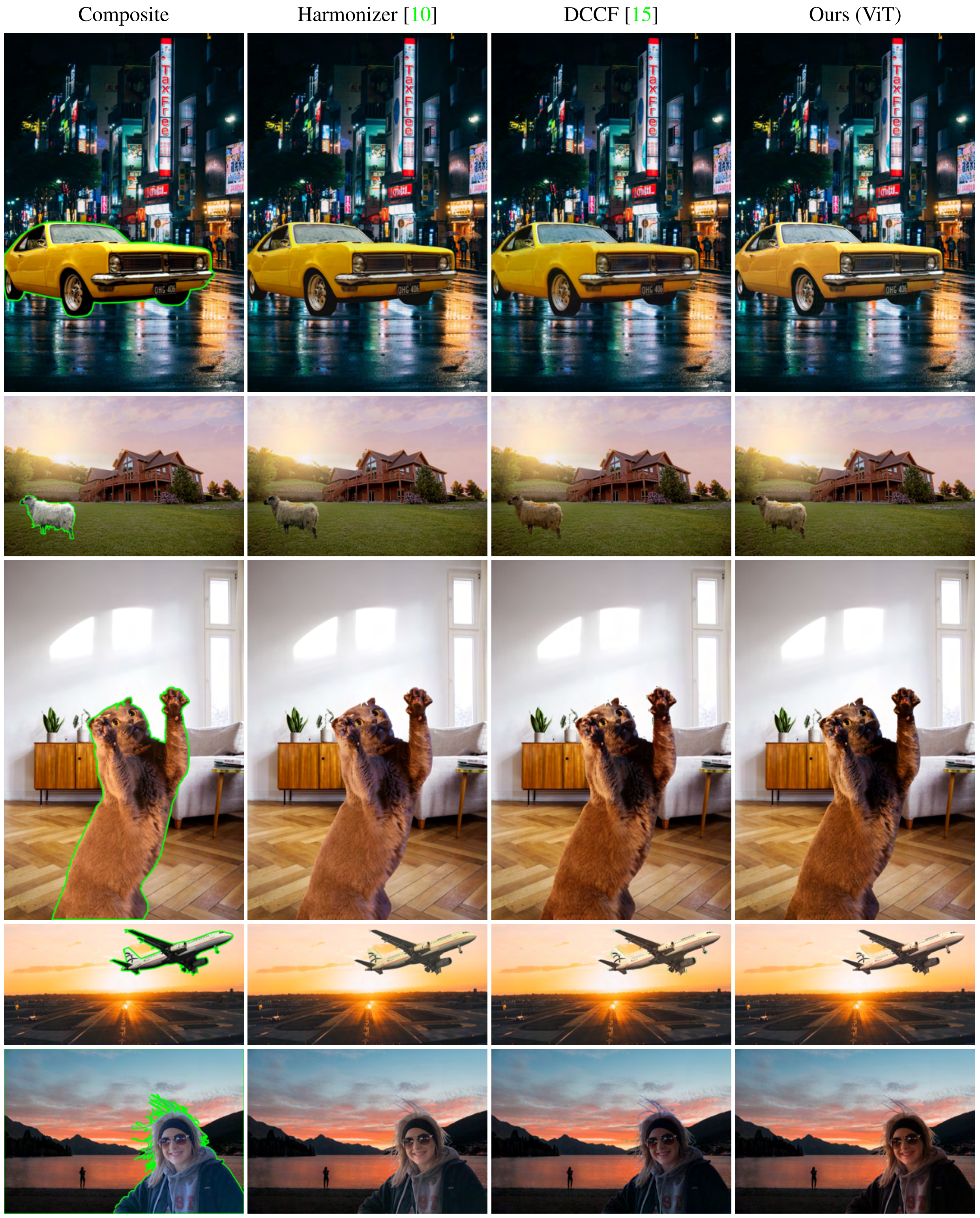


Figure 3. Images used for the user study (No.6-10). *The foreground region of the composite image is outlined in green. Foreground images are from the BIG dataset [2] or the RealHM dataset [9]. Background images are from [1].*

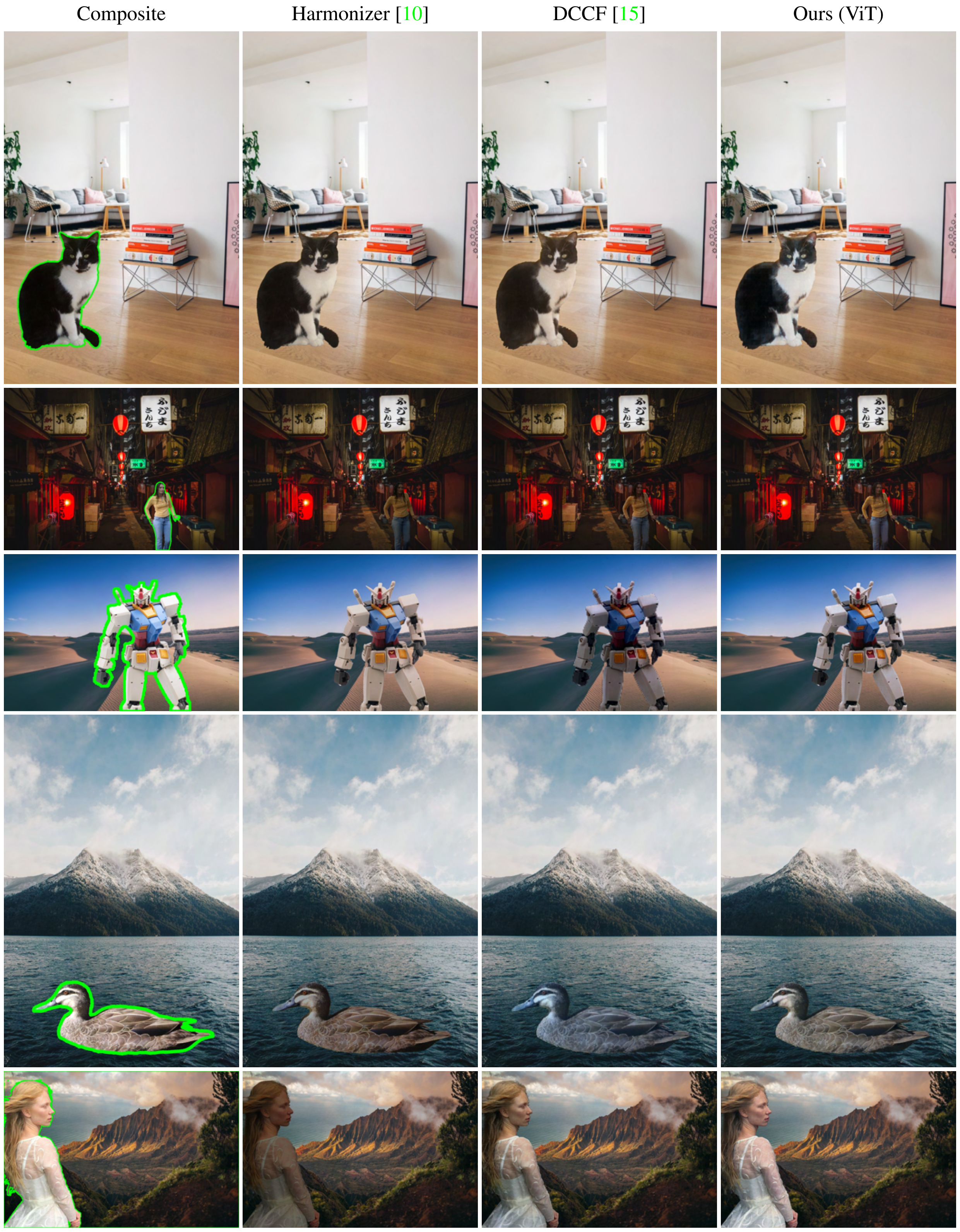


Figure 4. Images used for the user study (No.11-15). *The foreground region of the composite image is outlined in green. Foreground images are from the BIG dataset [2] or the RealHM dataset [9]. Background images are from [1].*

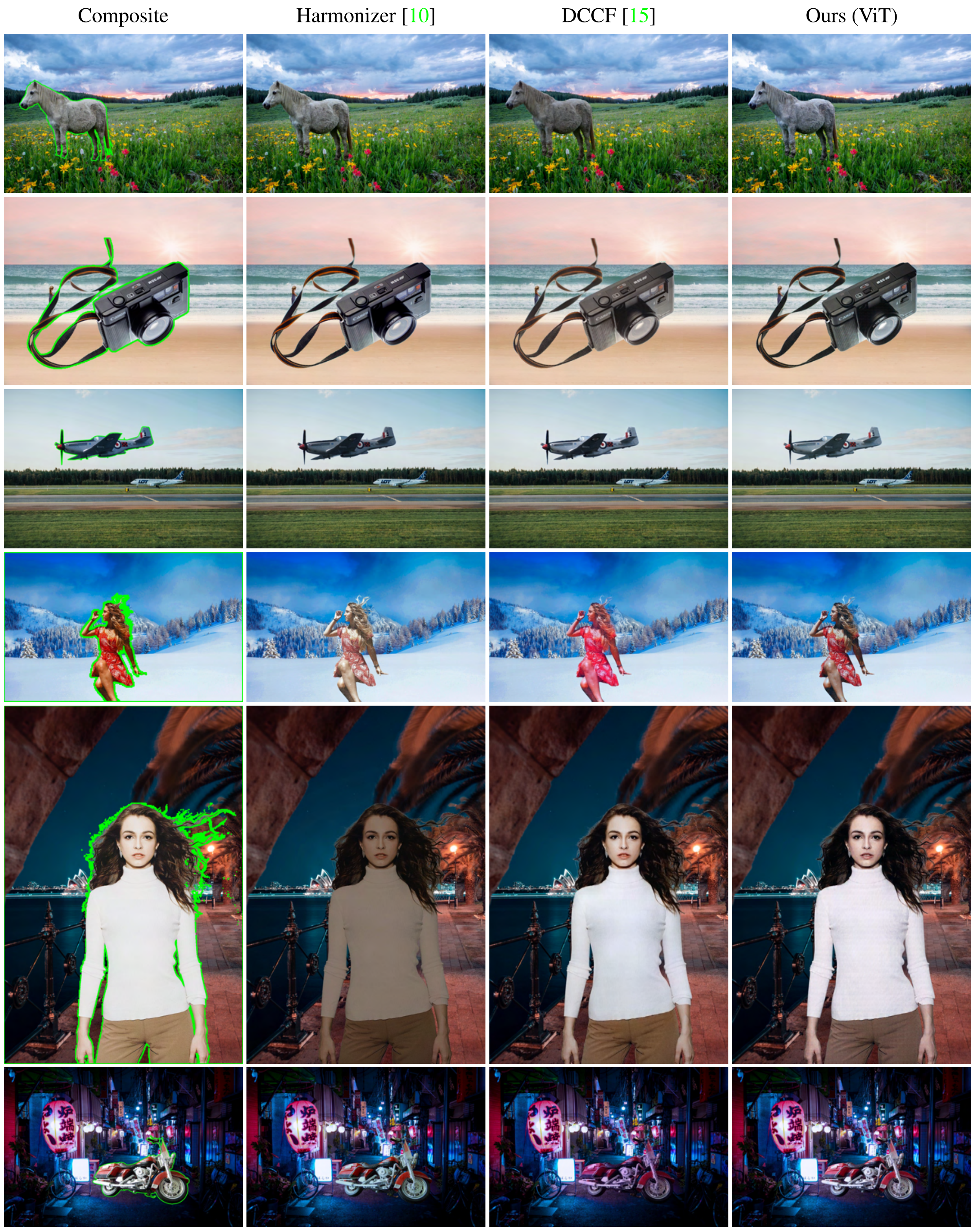


Figure 5. Images used for the user study (No.16-21). *The foreground region of the composite image is outlined in green. Foreground images are from the BIG dataset [2] or the RealHM dataset [9]. Background images are from [1].*

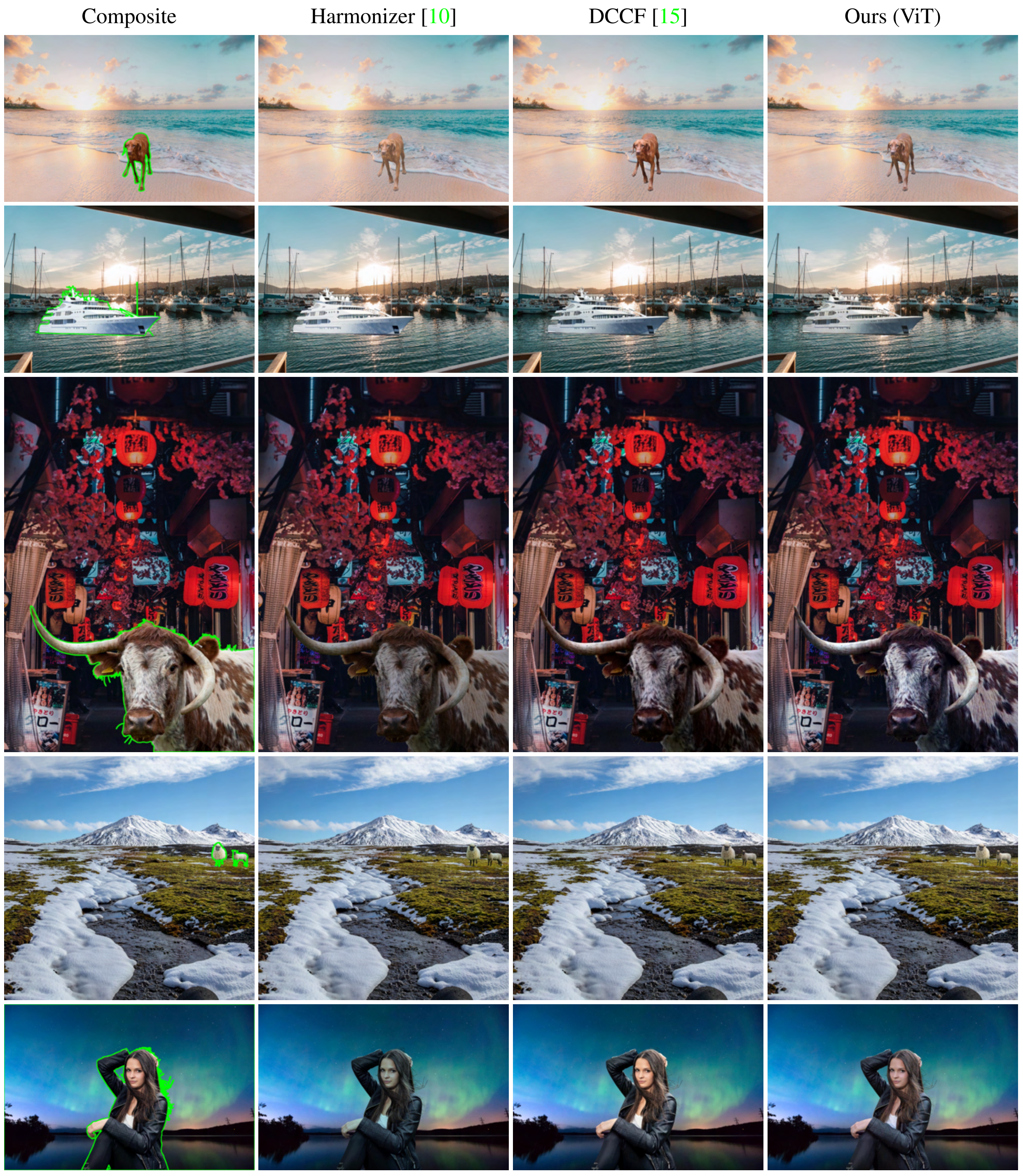


Figure 6. Images used for the user study (No.22-26). *The foreground region of the composite image is outlined in green. Foreground images are from the BIG dataset [2] or the RealHM dataset [9]. Background images are from [1].*

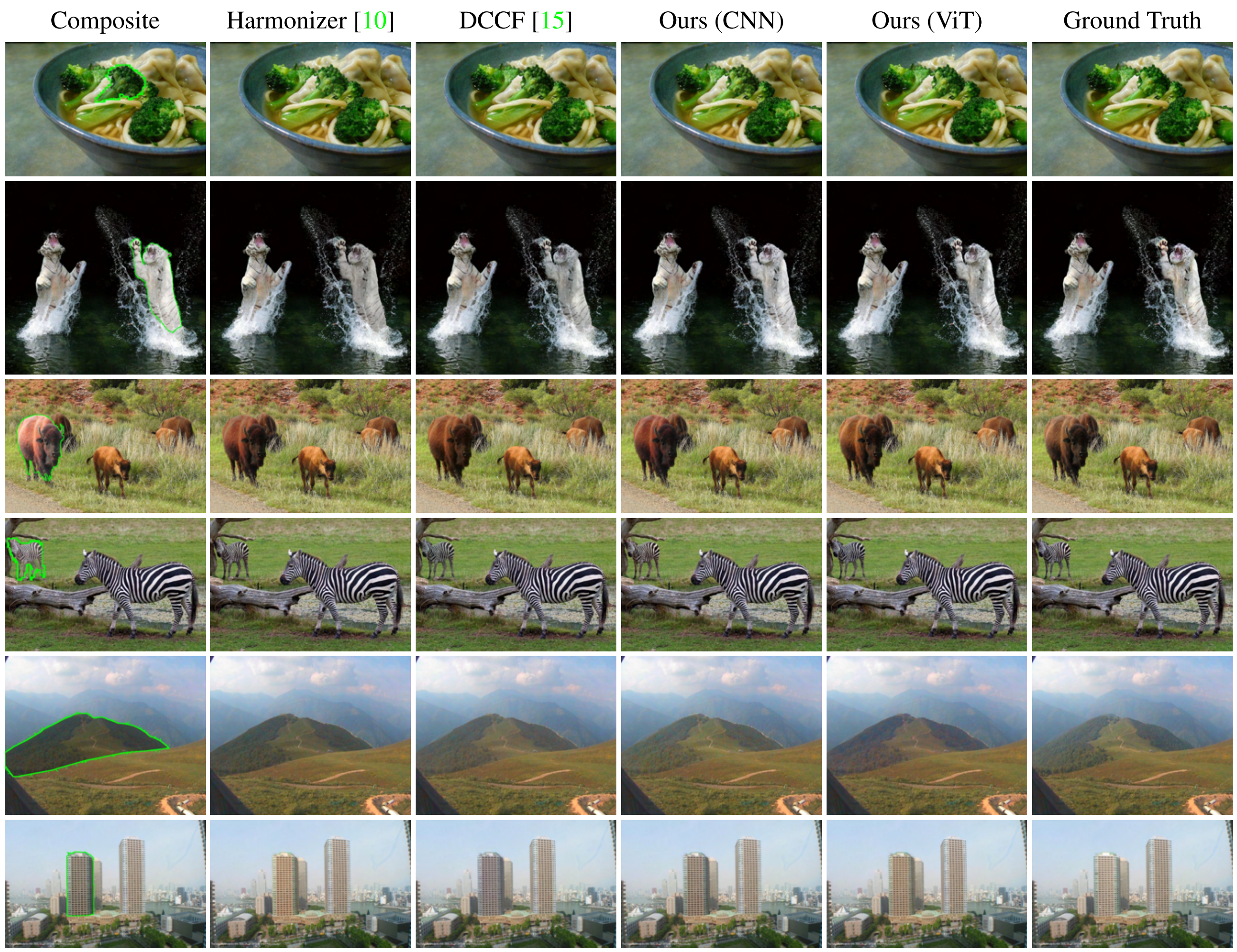


Figure 7. Qualitative results on the HCOCO, HFlickr and Hday2night test subset *We randomly select images from the HCOCO, HFlickr and Hday2night test subsets [4] and compare the output of different methods. The foreground region of the composite image is outlined in green.*

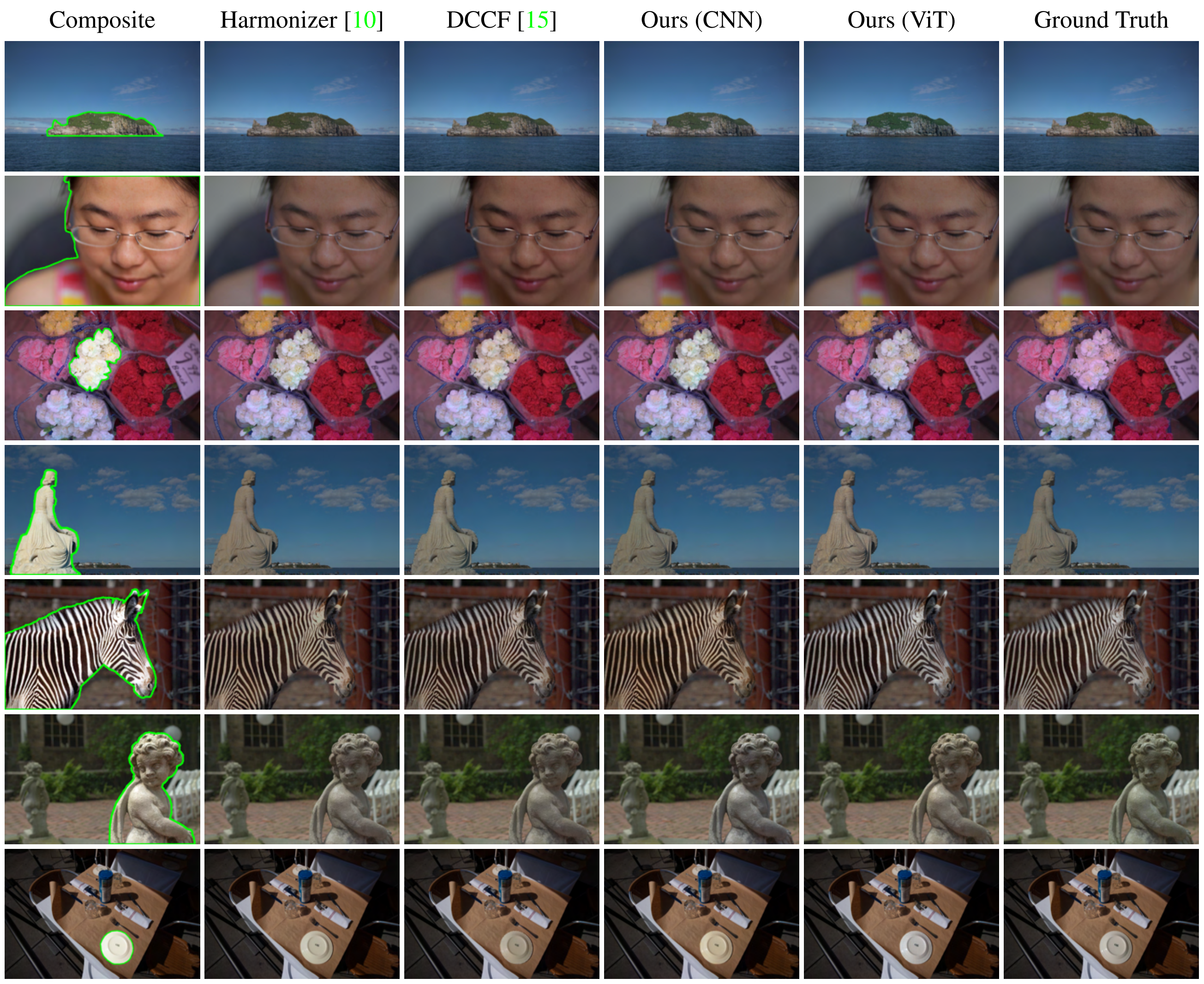


Figure 8. ***Qualitative results on the HAdobe5k test subset*** *We randomly select images from the HAdobe5k test subset [4] and compare the output of different methods. The foreground region of the composite image is outlined in green.*