Supplementary File for: Intrinsic Image Harmonization

Zonghui Guo[1] Haiyong Zheng[1,*] Yufeng Jiang[1] Zhaorui Gu[1] Bing Zheng[1,2]
[1]Underwater Vision Lab (http://ouc.ai), Ocean University of China
[2]Sanya Oceanographic Institution, Ocean University of China
{guozonghui,jiangyufeng7526}@stu.ouc.edu.cn, {zhenghaiyong,guzhaorui,bingzh}@ouc.edu.cn

A. Implementation Details

The reflectance and illumination are disentangled by encoder-decoder networks, while lighting and guiding are regressed using simple encoder networks. The reflectance encoder-decoder network uses 4-layer ResBlocks and 2-layer guiding blocks, and the illumination encoder-decoder network adopts 4-layer lighting ResBlocks and 2-layer guiding blocks. The final activation function is $\tanh$ for reflectance, illumination, and guiding, while no activation but average pooling with fully-connected layer for lighting. Reflectance and illumination encoder-decoder outputs are normalized to $[0, 1]$ to recover $\hat{\mathbf{H}}$. We train our model using Adam optimizer [5] with parameters of $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $\alpha = 0.0001$. We resize input images as 256×256 for training and testing, and our model produces harmonized images with the same size. Light latent code is set as an 8-dimentional vector and inharmony-free feature maps are $32 \times 32 \times 256$ volume. We empirically set $\lambda_{RH} = 0.1$, $\lambda_{IS} = 0.01$, $\lambda_{IH} = 0.1$ and $\lambda_{IF} = 1$ in our experiments. Our model does not use the occluded background information for training and testing.

We report the implementation details of our autoencoder-based architecture for intrinsic image harmonization in Tables A (reflectance intrinsic image harmonization), B (illumination intrinsic image harmonization), C (light learning), and D (inharmony-free learning). We use standard encoder networks for both light and inharmony-free learning, and encoder-decoder networks for reflectance and illumination intrinsic image harmonization. Symbols of the operators are defined as follows:

- Conv(c_{in}, c_{out}, k, s, p): convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s, and padding p.
- Linear(f_{in}, f_{out}): linear transformation with f_{in} input features and f_{out} output features.
- ResBlock(c_{in}, c_{out}, k, s, p): residual block [4] with c_{in} input channels, c_{out} output channels, kernel size k, stride s, and padding p.
- GuidingBlock(c_{in}, c_{out}, k, s, p): our guiding block with c_{in} input channels, c_{out} output channels, kernel size k, stride s, and padding p.
- LightingResBlock(c_{in}, c_{out}, k, s, p): our lighting residual block with c_{in} input channels, c_{out} output channels, kernel size k, stride s, and padding p.
- Upsample(s): nearest-neighbor upsampling with a scale factor of s.
- IN(n): instance normalization [8] with n dimensions.
- LN(n): layer normalization [1] with n dimensions.
- LReLU(α): Leaky ReLU [6] with a negative slope of α.

B. Patch Covariance

We compute patch covariance for modeling patch relations by:

$$K_{(p^{fg},p^{bg})} = \frac{1}{C-1}\sum_{i=1}^{C}(p_i^{fg} - p_\mu^{fg})(p_i^{bg} - p_\mu^{bg})^T, \quad (1)$$

where $p \in \mathbb{R}^{HW \times C}$; C, H and W represent the number of channels, height and width, respectively; p^{fg} and p^{bg} denote the transformed foreground and background feature maps (from $\mathbb{R}^{C \times H \times W}$); and $p_\mu \in \mathbb{R}^{HW \times 1}$ is the mean, computed across channel dimension independently for each spatial location.

*Corresponding author (zhenghaiyong@ouc.edu.cn).

This work was supported in part by the Finance Science and Technology Project of Hainan Province under Grant ZDKJ202017, and the National Natural Science Foundation of China under Grant 61771440 and Grant 41776113.

Encoder	Output size
Conv(3, 64, 7, 1, 3) + IN(64) + LReLU(0.2)	256
Conv(64, 128, 4, 2, 1) + IN(128) + LReLU(0.2)	128
Conv(128, 256, 4, 2, 1) + IN(256) + LReLU(0.2)	64
Bottleneck	**Output size**
ResBlock(256, 256, 3, 1, 1) $\times 4$	64
GuidingBlock(256, 256, 3, 1, 1) $\times 2$	64
Decoder	**Output size**
Upsample(2)	128
Conv(256, 128, 3, 1, 1) + LN(128) + LReLU(0.2)	128
Upsample(2)	256
Conv(128, 64, 3, 1, 1) + LN(64) + LReLU(0.2)	256
Conv(64, 3, 7, 1, 3) + Tanh $\rightarrow$ *output*	256

Table A. Network architecture for reflectance intrinsic image harmonization.

Encoder	Output size
Conv(3, 64, 7, 1, 3) + IN(64) + LReLU(0.2)	256
Conv(64, 128, 4, 2, 1) + IN(128) + LReLU(0.2)	128
Conv(128, 256, 4, 2, 1) + IN(256) + LReLU(0.2)	64
Bottleneck	**Output size**
LightingResBlock(256, 256, 3, 1, 1) $\times 4$	64
GuidingBlock(256, 256, 3, 1, 1) $\times 2$	64
Decoder	**Output size**
Upsample(2)	128
Conv(256, 128, 3, 1, 1) + LN(128) + LReLU(0.2)	128
Upsample(2)	256
Conv(128, 64, 3, 1, 1) + LN(64) + LReLU(0.2)	256
Conv(64, 3, 7, 1, 3) + Tanh $\rightarrow$ *output*	256

Table B. Network architecture for illumination intrinsic image harmonization.

Encoder	Output size
Conv(3, 64, 7, 1, 3) + LReLU(0.2)	256
Conv(64, 128, 4, 2, 1) + LReLU(0.2)	128
Conv(128, 256, 4, 2, 1) + LReLU(0.2)	64
AdaptiveAvgPool2d(1)	1
MLP	**Output size**
Linear(256, 8) $\rightarrow$ *output*	1

Table C. Network architecture for light learning.

Encoder	Output size
Conv(3, 32, 7, 1, 3) + LReLU(0.2)	256
Conv(32, 64, 4, 2, 1) + LReLU(0.2)	128
Conv(64, 128, 4, 2, 1) + LReLU(0.2)	64
Conv(128, 256, 4, 2, 1) + LReLU(0.2)	32
Conv(128, 256, 3, 1, 1) + Tanh $\rightarrow$ *output*	32

Table D. Network architecture for inharmony-free learning.

C. Evaluation Metrics

In addition to MSE and SSIM [9], we also report foreground MSE (fMSE) and foreground SSIM (fSSIM) to measure how well the foreground is harmonized. MSE and SSIM essentially evaluate harmonization performance over all pixels across the dataset (dataset-level), while fMSE and fSSIM measure the harmonization over each single image (with different sizes of foreground) averaging on the dataset (image-level). We argue that image-level fMSE and fSSIM are more suitable to evaluate the harmonization generalization ability since many pixels (background) are unchanged and the sizes of foreground are different in terms of each image. Given the real image $\mathbf{H}$ and the harmonized image $\hat{\mathbf{H}}$, we provide the details of these four metrics as follows.

C.1. MSE vs. fMSE

We compute MSE by:

$$\begin{aligned}\text{MSE}(\hat{\mathbf{H}}, \mathbf{H}) &= \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{3K}\sum_{k=1}^{K}\left\|\hat{\mathbf{H}}_k^{(n)} - \mathbf{H}_k^{(n)}\right\|_2\right) \\ &= \frac{1}{3KN}\sum_{n=1}^{N}\sum_{k=1}^{K}\left\|\hat{\mathbf{H}}_k^{(n)} - \mathbf{H}_k^{(n)}\right\|_2, \end{aligned} \quad (2)$$

where K is the pixel number of image (k is the pixel index), N is the image number of dataset (n is the image index), and 3 means three RGB channels of image.

And we compute our fMSE by:

$$\begin{aligned}&\text{fMSE}(\hat{\mathbf{H}}, \mathbf{H}) = \\ &\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{3K_{fg}^{(n)}}\sum_{k=1}^{K}\left\|\hat{\mathbf{H}}_k^{(n)}\mathbf{M}_k^{(n)} - \mathbf{H}_k^{(n)}\mathbf{M}_k^{(n)}\right\|_2\right), \end{aligned} \quad (3)$$

where $K_{fg}^{(n)}$ is the foreground pixel number of n-th image, and $\mathbf{M}$ denotes the foreground mask.

Refer to Table 1 in the paper, it is worth mentioning that our method is superior to DoveNet in fMSE, but inferior to DoveNet in MSE on Hday2night, mainly because that MSE evaluates harmonization performance at the dataset level while fMSE reflects harmonization ability at the image level which is more valuable and generalized, for instance, one method may obtain lower MSE yet higher fMSE because it harmonizes some images with big foreground very better while harmonizes some images with small foreground very worse, indicating unstable performance.

C.2. SSIM vs. fSSIM

We compute SSIM by:

$$\text{SSIM}(\hat{\mathbf{H}}, \mathbf{H}) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{J}\sum_{j=1}^{J}\mathcal{S}_{\text{SSIM}}\left(\hat{\mathbf{H}}_j^{(n)}, \mathbf{H}_j^{(n)}\right)\right), \quad (4)$$

where J is the window number of image (j is the window index), N is the image number of dataset (n is the image index), and $\mathcal{S}_{\text{SSIM}}$ is the structural similarity function referring to [9].

$$\text{fSSIM}(\hat{\mathbf{H}}, \mathbf{H}) = \frac{1}{N}\sum_{n=1}^{N} \left(\frac{1}{J_{fg}^{(n)}} \left(\sum_{j=1}^{J} \mathcal{S}_{\text{SSIM}} \left(\hat{\mathbf{H}}_j^{(n)} \odot \mathbf{M}_j^{(n)}, \mathbf{H}_j^{(n)} \odot \mathbf{M}_j^{(n)} \right) - J_{bg}^{(n)} \right) \right), \tag{5}$$

where $J_{fg}^{(n)}$ and $J_{bg}^{(n)}$ are the foreground and background window number of n-th image respectively, $\mathbf{M}$ denotes the foreground mask, and $\odot$ indicates element-wise product.

Refer to Table 1 in the paper, surprisingly, composite images have highest SSIM scores representing best structural similarity to real images, suggesting that, (1) the inharmony of composite images is not caused by structure or semantics, so that the illumination may play an important role, and (2) all listed methods may destroy image structure during harmonization, among which our method is least destructive yet makes most harmonious. Noting that, in terms of fSSIM, our method performs best against all other methods as well as composite images, also because that SSIM evaluates at dataset level while fSSIM evaluates at image level (similar to MSE vs. fMSE), thus yielding inconsistent trend changes due to different size of foreground for each image.

D. Additional Quantitative Results of iHarmony4+HVIDIT

We report the quantitative comparison results of image harmonization models retrained by merging our HVIDIT into iHarmony4 [2] in Table E.

E. Additional Qualitative Results

We show additional qualitative comparison results of image harmonization in Figures A and B. And we show harmonized results with normal masks and inverted masks in Figures C and D. We also show light latent representation results by changing light latent code in Figure E, and visual results transferring the light from one source image to another target image in Figure F.

F. Results on Real Composite Images

We finally show all visual comparison results of different methods to harmonize 99 real composite images in Figures G–Q.

References

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

[2] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. DoveNet: Deep image harmonization via domain verification. In *CVPR*, pages 8394–8403, 2020. 3, 4

[3] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE TIP*, 29:4759–4771, 2020. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[6] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICMLW*, pages 1–6, 2013. 1

[7] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, pages 3789–3797, 2017. 4

[8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, pages 6924–6932, 2017. 1

[9] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 2, 3

[10] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. 4

Dataset	Metric	Composite	Retinex-Net [10]	DIH [7]	S^2AM [3]	DoveNet [2]	Ours (base)	Ours (base+lighting)	Ours (base+guiding)	Ours
HCOCO	PSNR↑	33.94	33.09	33.58	34.92	35.75	36.01	**37.08**	36.82	36.96
	MSE↓	69.37	69.11	55.84	37.32	36.05	29.77	23.42	27.92	**22.03**
	fMSE↓	996.59	988.08	803.19	568.14	547.78	483.75	420.19	470.37	**410.21**
	SSIM↑	**0.9853**	0.9544	0.9460	0.9451	0.9550	0.9751	0.9762	0.9756	0.9811
	fSSIM↑	0.8257	0.8249	0.8225	0.8454	0.8482	0.8388	0.8592	0.8402	**0.8621**
HAdobe5k	PSNR↑	28.16	27.34	31.53	32.67	33.84	34.16	35.02	34.95	**35.06**
	MSE↓	345.54	337.33	101.04	64.28	57.15	49.82	41.17	46.11	**40.15**
	fMSE↓	2051.61	2006.31	605.36	473.09	376.42	346.37	251.16	304.12	**248.13**
	SSIM↑	**0.9483**	0.8859	0.8721	0.8900	0.8853	0.9354	0.9390	0.9368	0.9362
	fSSIM↑	0.7294	0.7240	0.7766	0.8059	0.8144	0.8071	0.8377	0.8042	**0.8384**
HFlickr	PSNR↑	28.32	28.03	28.99	30.46	30.54	30.72	31.17	30.89	**31.23**
	MSE↓	264.35	265.63	167.90	116.11	125.74	125.11	100.76	109.02	**97.69**
	fMSE↓	1574.37	1565.72	1103.85	757.69	813.34	795.95	708.86	753.19	**700.51**
	SSIM↑	**0.9618**	0.9299	0.9130	0.9179	0.9281	0.9491	0.9592	0.9507	0.9604
	fSSIM↑	0.8031	0.7986	0.7981	0.8242	0.8247	0.8032	0.8287	0.8089	**0.8299**
Hday2night	PSNR↑	34.01	33.16	33.91	34.66	34.43	34.25	35.05	34.87	**35.76**
	MSE↓	109.65	110.25	75.51	**51.11**	57.17	90.14	55.58	80.13	51.16
	fMSE↓	1409.98	1405.23	1002.55	848.48	1001.27	1301.06	841.33	1052.11	**776.41**
	SSIM↑	**0.9606**	0.8995	0.8862	0.8908	0.8972	0.9293	0.9428	0.9309	0.9382
	fSSIM↑	0.6353	0.6321	0.6433	0.6467	0.6414	0.6010	0.6481	0.6053	**0.6529**
HVIDIT	PSNR↑	38.53	36.32	36.62	36.24	36.80	40.55	40.31	40.29	**41.55**
	MSE↓	53.12	53.01	45.55	45.82	35.36	33.16	22.51	25.57	**20.16**
	fMSE↓	1604.41	1603.21	1207.03	1230.92	1186.19	934.63	861.09	925.01	**800.92**
	SSIM↑	**0.9921**	0.9321	0.9310	0.9206	0.9585	0.9900	0.9912	0.9908	0.9914
	fSSIM↑	0.7612	0.7161	0.7512	0.7401	0.7440	0.7136	0.7635	0.7560	**0.7686**
All	PSNR↑	31.92	31.08	32.65	33.86	34.68	35.09	35.97	35.78	**35.99**
	MSE↓	167.39	165.09	80.37	53.88	51.88	46.76	37.17	42.48	**35.61**
	fMSE↓	1386.12	1381.32	800.73	594.90	541.74	512.05	411.74	470.30	**390.03**
	SSIM↑	**0.9723**	0.9308	0.9202	0.9248	0.9318	0.9611	0.9640	0.9620	0.9660
	fSSIM↑	0.7904	0.7881	0.8009	0.8242	0.8349	0.8167	0.8532	0.8313	**0.8727**

Note: we retrain the models on iHarmony4+HVIDIT to obtain the results for comparison.

Table E. Quantitative comparison on iHarmony4+HVIDIT. The ↑ indicates the higher the better, and ↓ indicates the lower the better. The best results are denoted in boldface. We compute fMSE and fSSIM at image level for better harmonization reflection.

Figure A. Additional qualitative comparison results of image harmonization. Red boxes in composite images mark foreground.

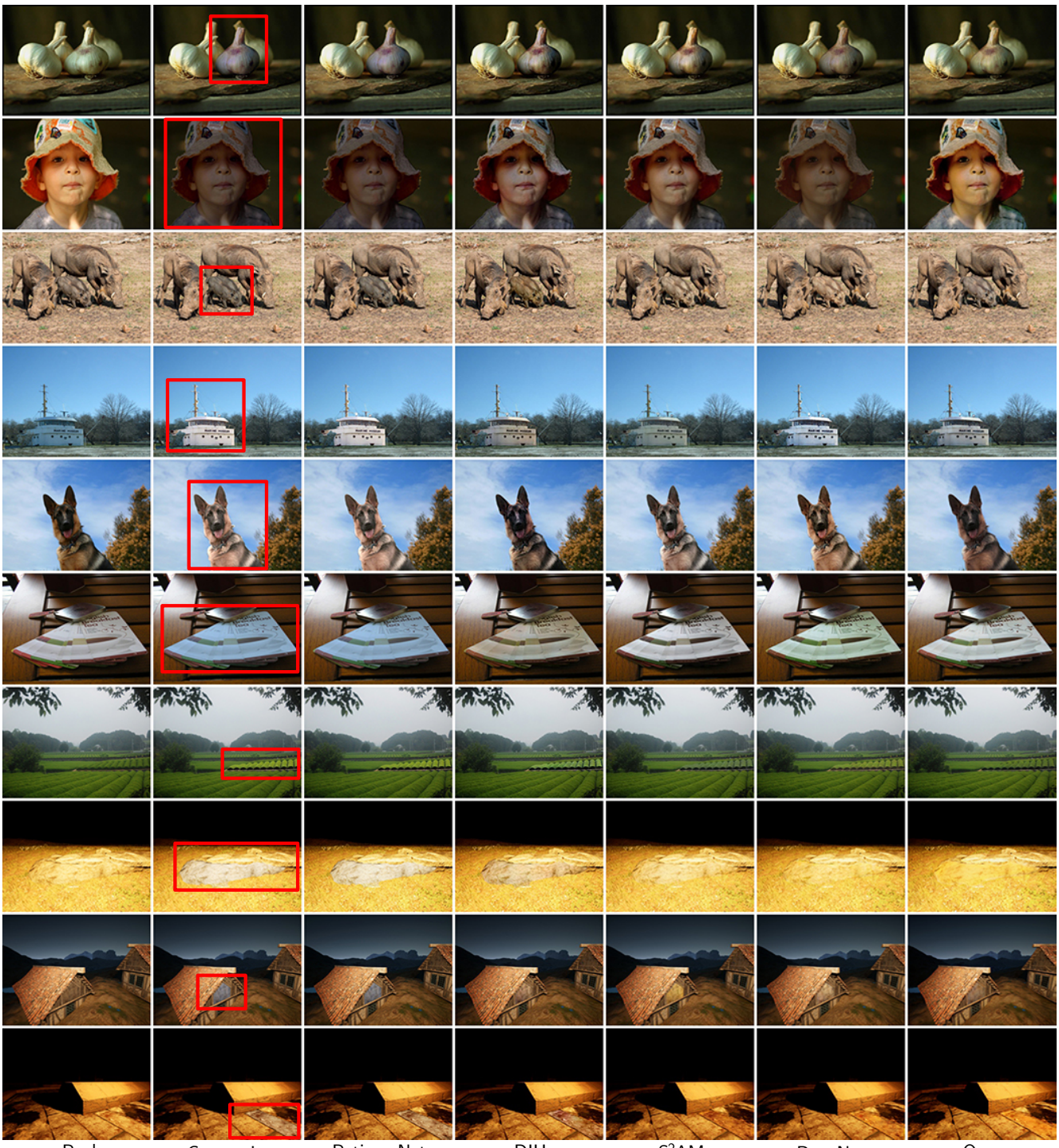

Figure B. Additional qualitative comparison results of image harmonization. Red boxes in composite images mark foreground.

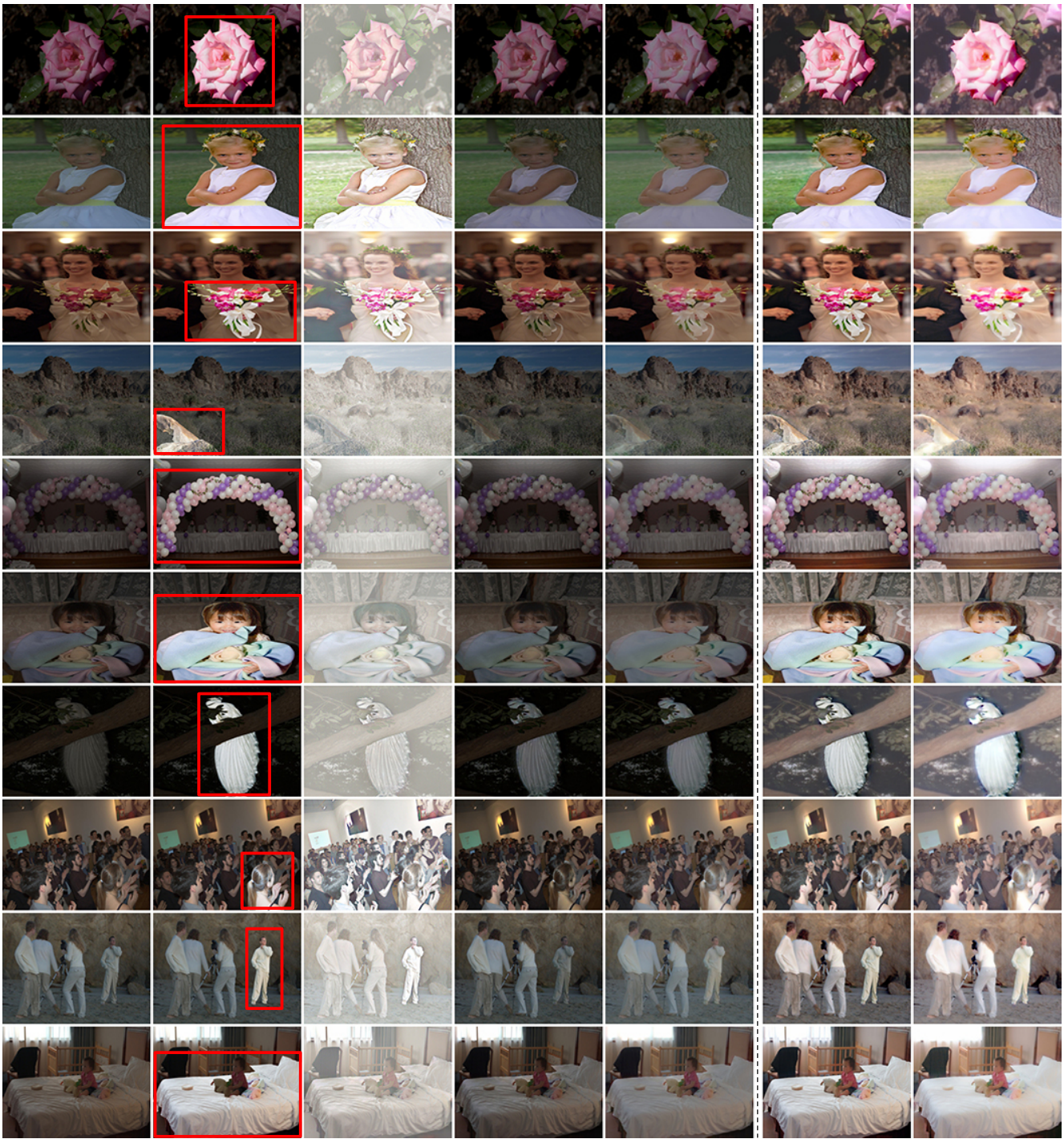

Figure C. Additional qualitative comparison results of image harmonization with normal masks and inverted masks. Red boxes in composite images mark foreground.

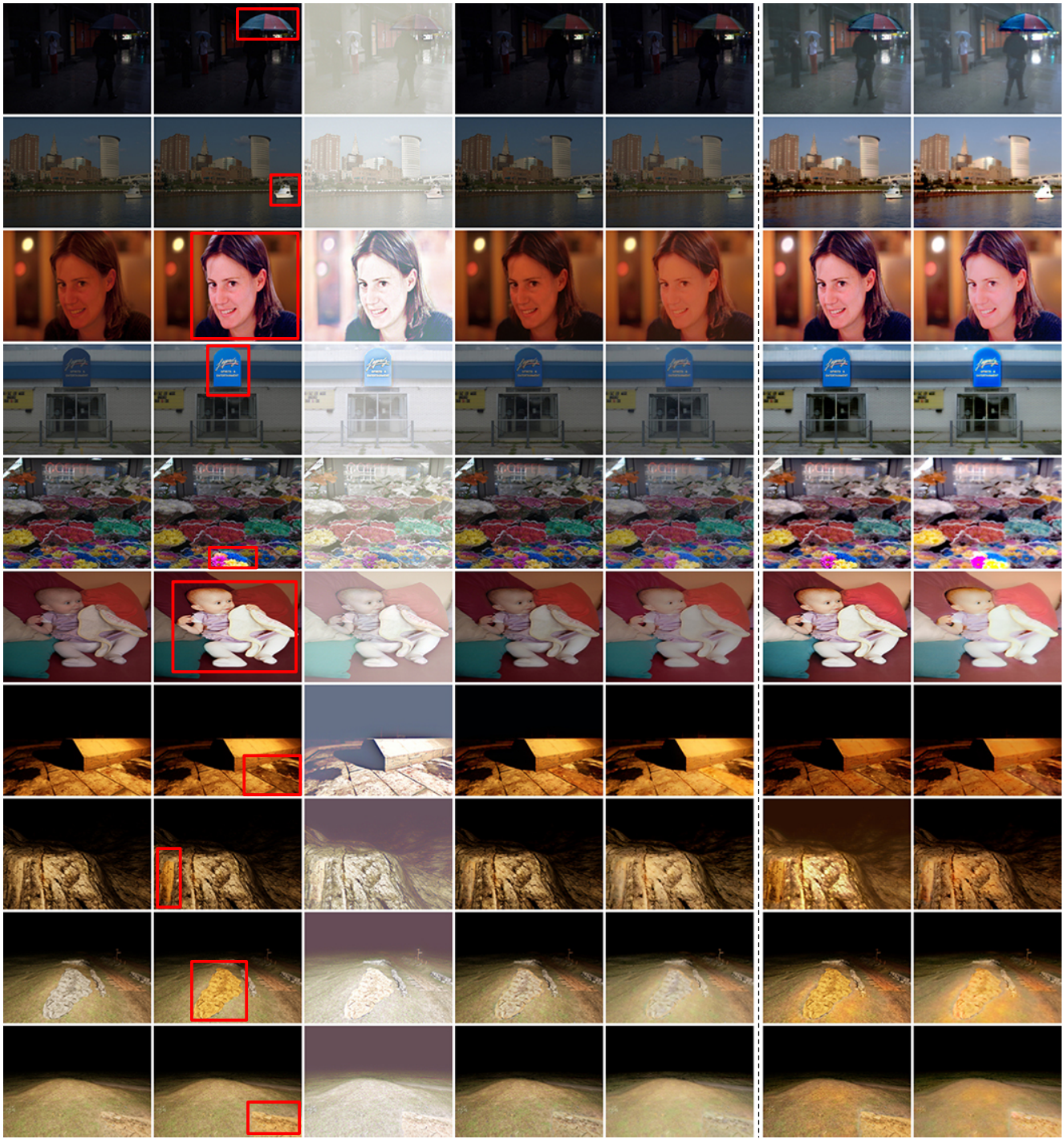

Figure D. Additional qualitative comparison results of image harmonization with normal masks and inverted masks. Red boxes in composite images mark foreground.

Figure E. Additional qualitative light latent representation results by changing light latent code.

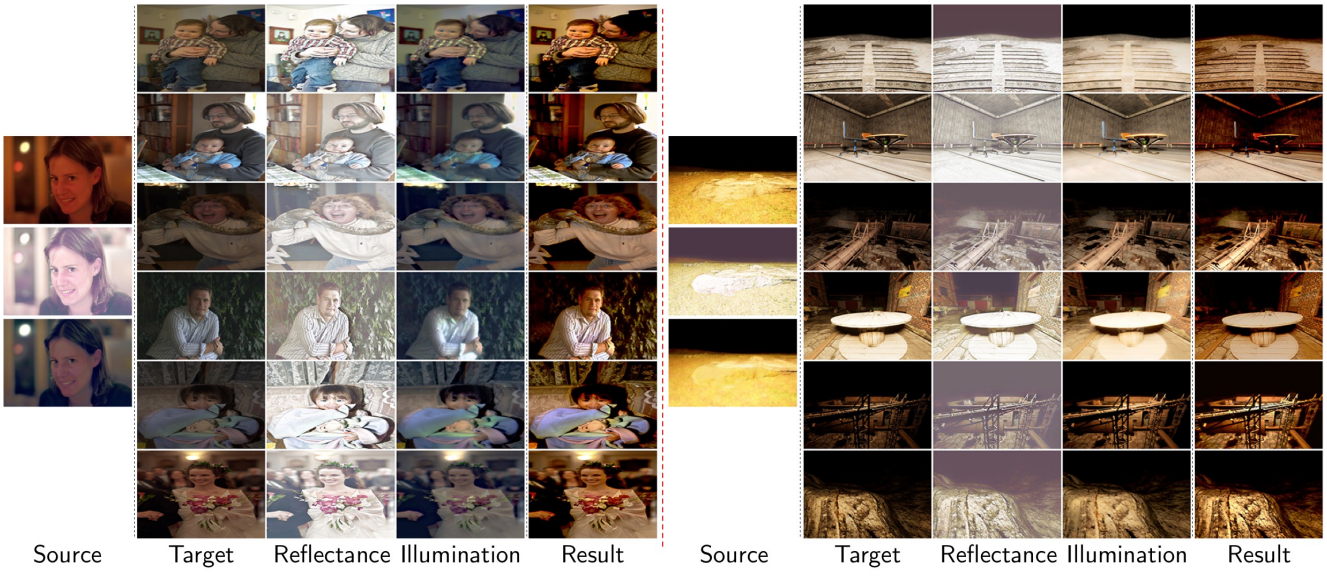

Figure F. Additional qualitative results transferring the light from one source image to another target image.

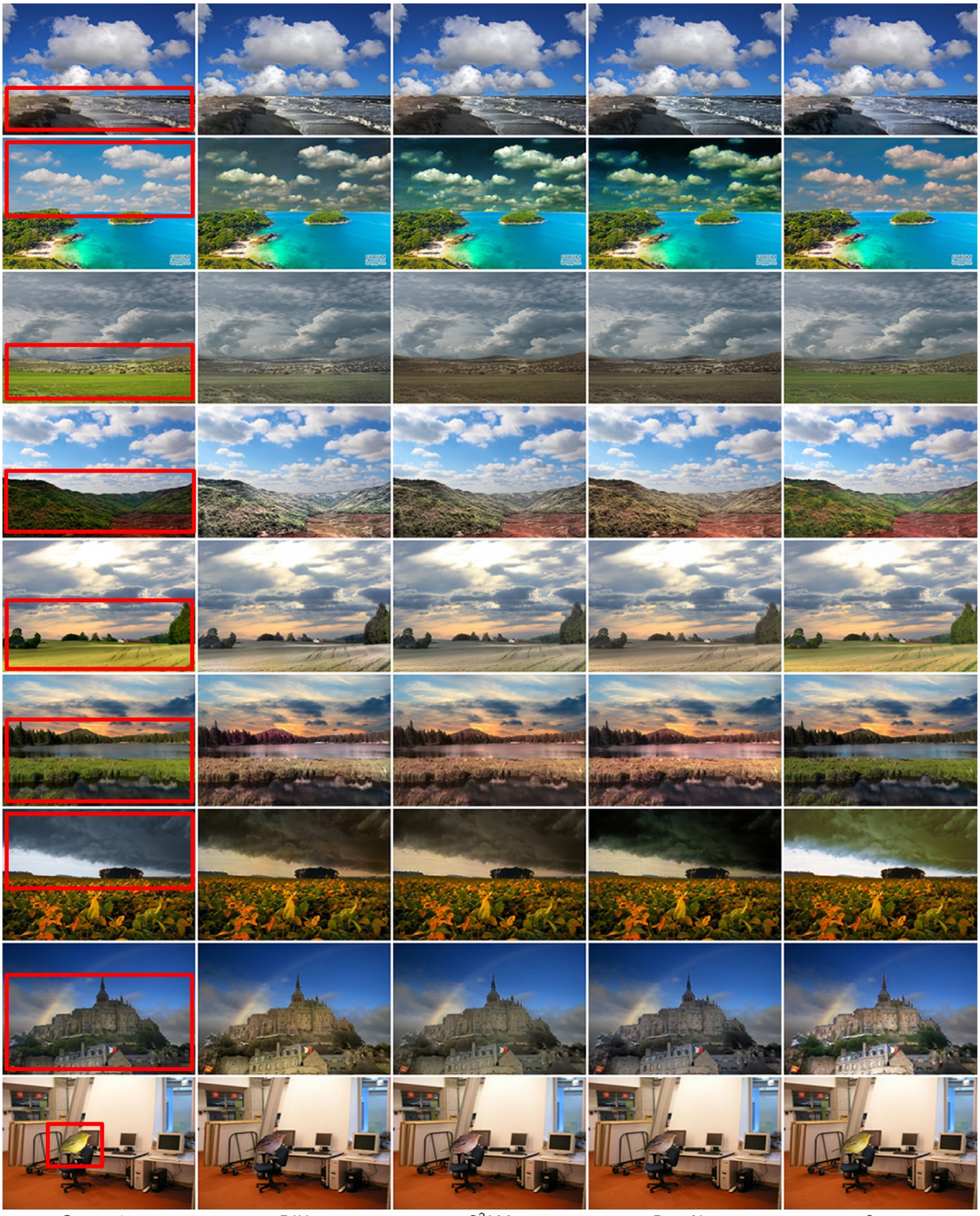

Figure G. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

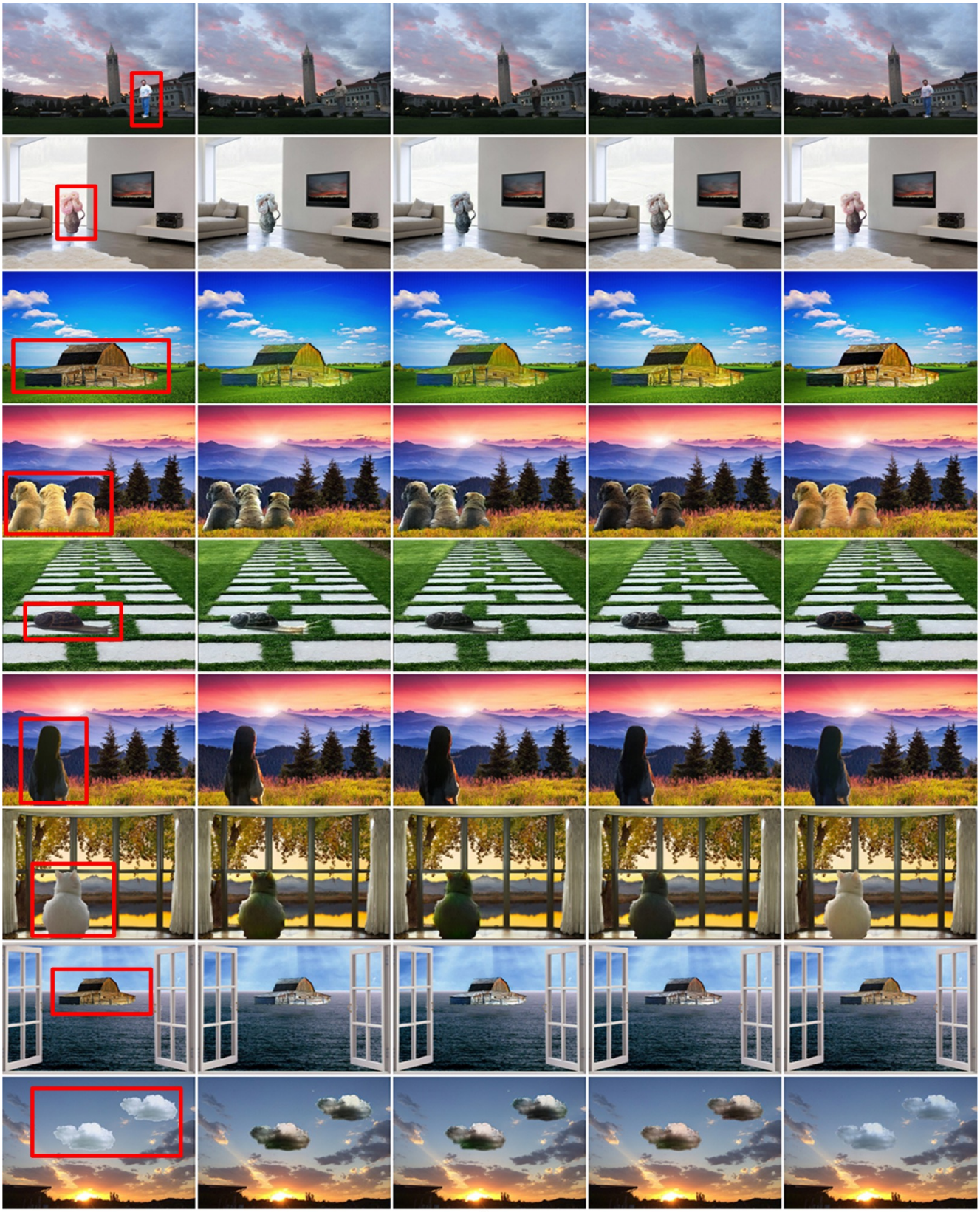

Figure H. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

Figure I. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

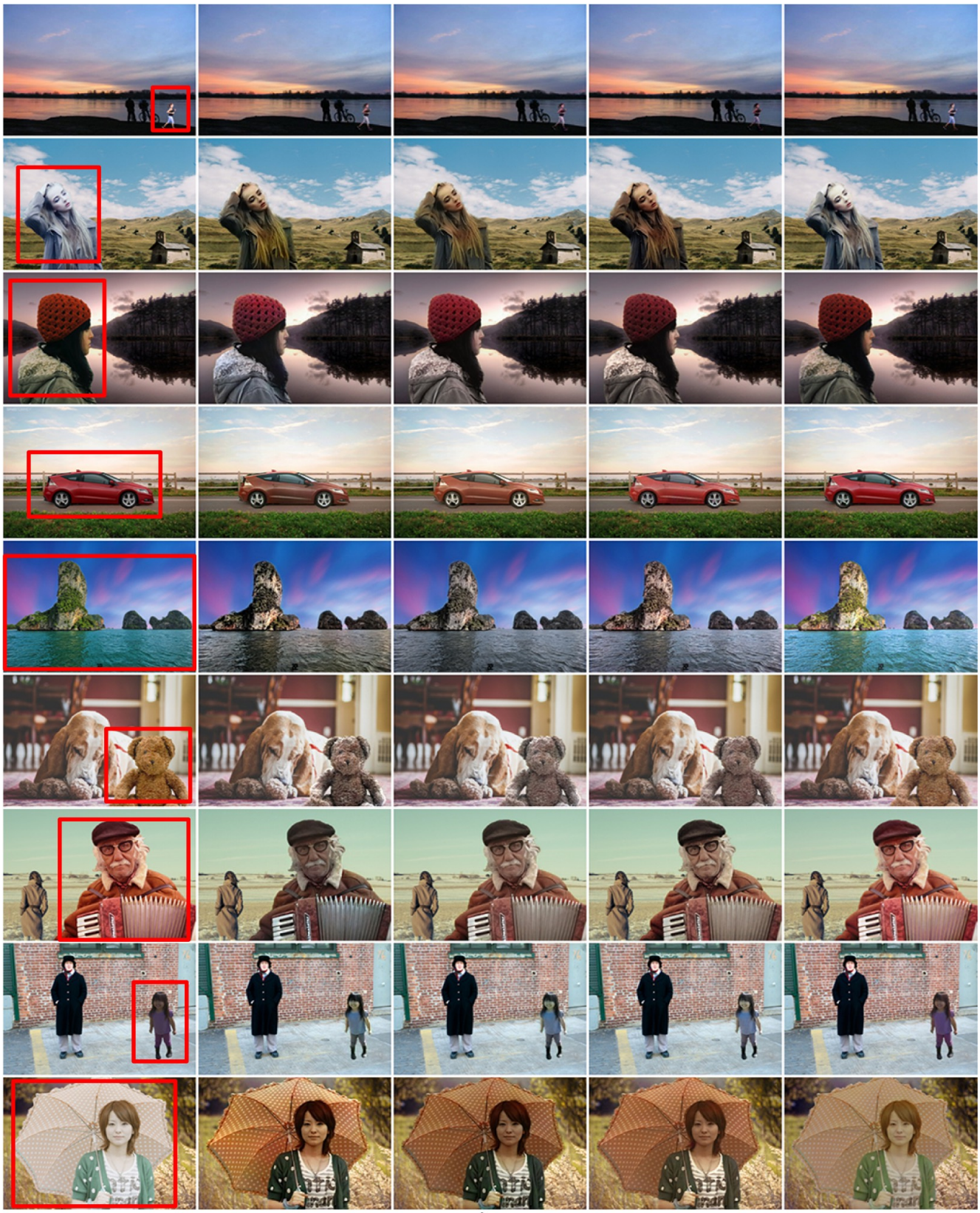

Figure J. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

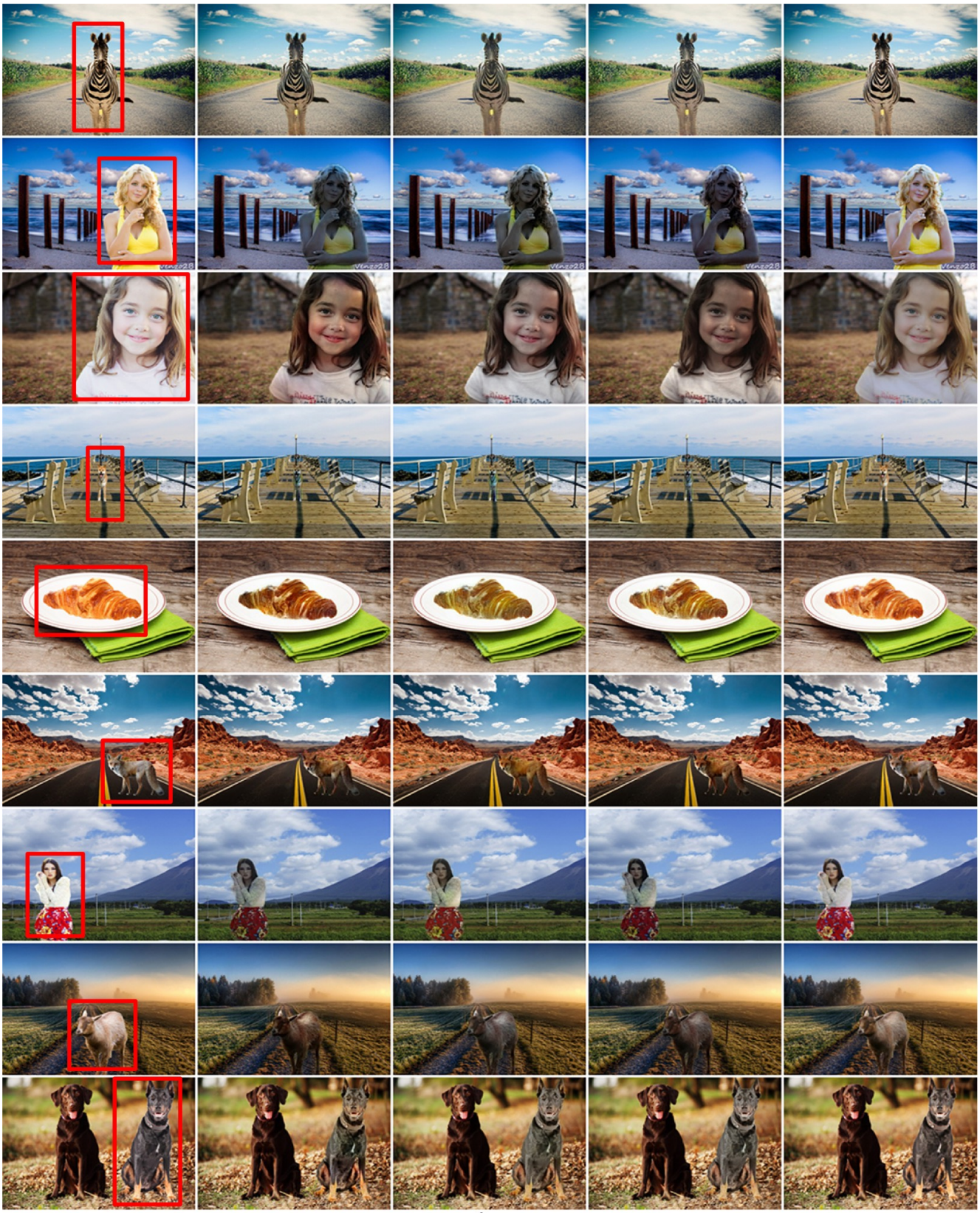

Figure K. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

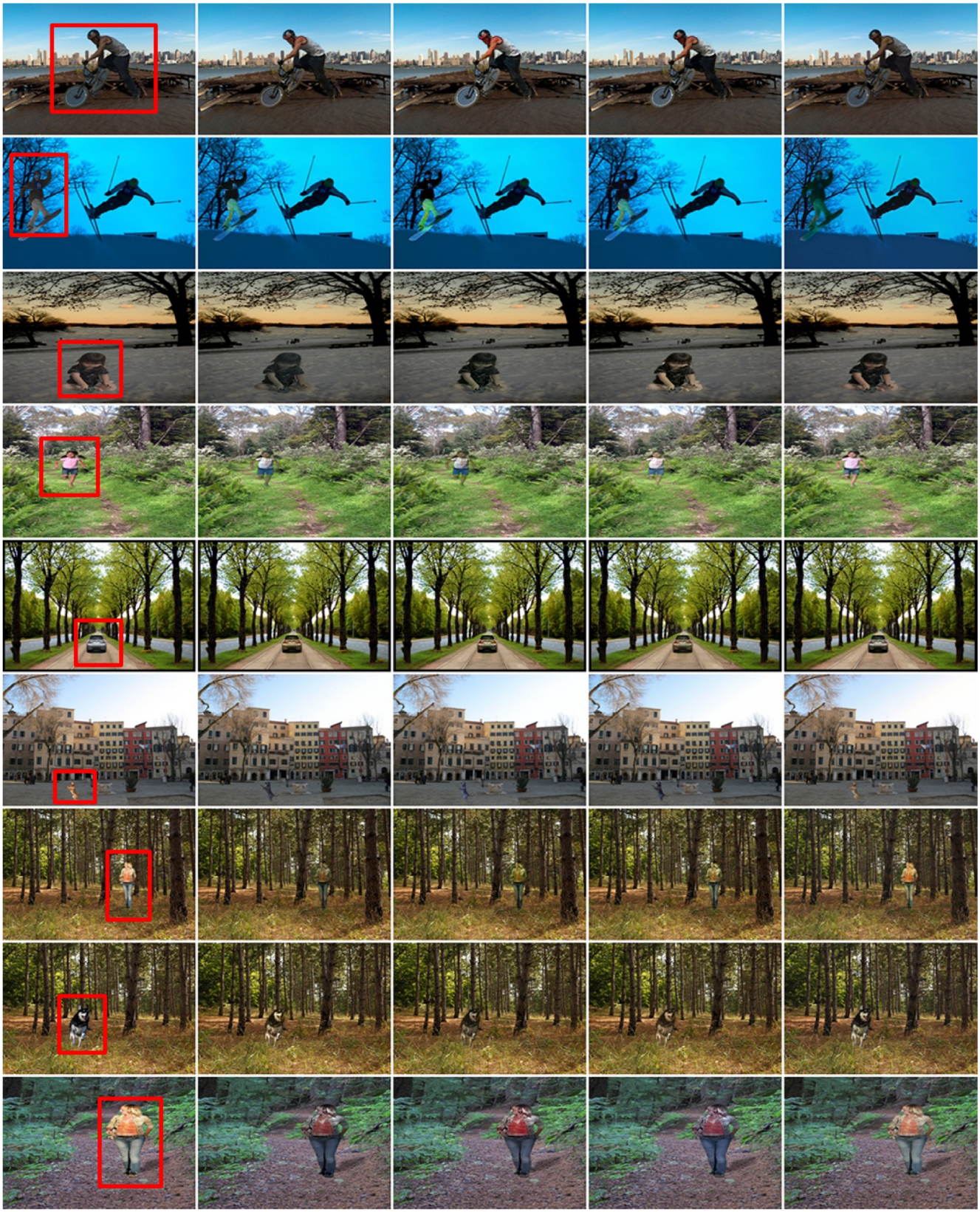

Figure L. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

Figure M. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

Figure N. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

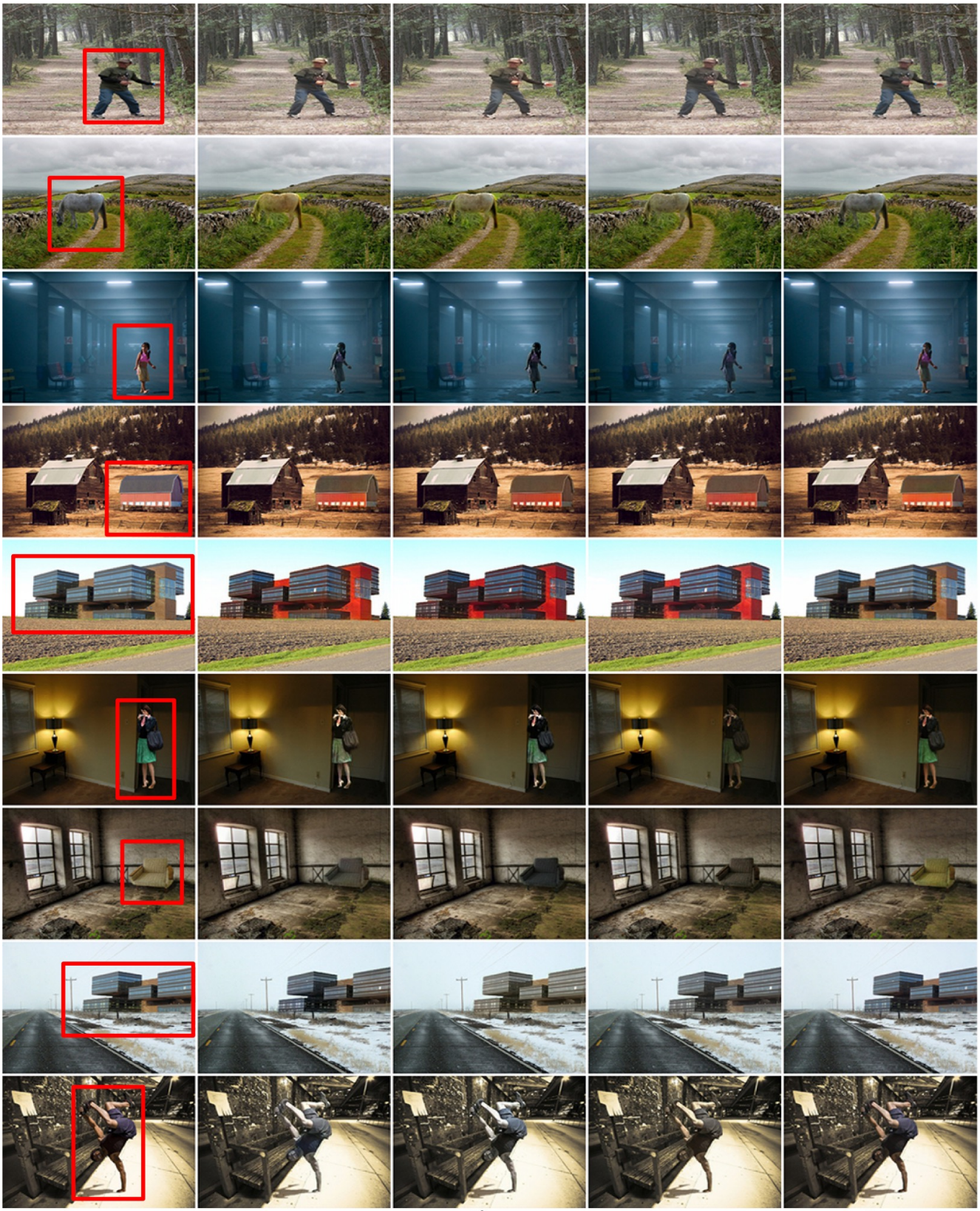

Figure O. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

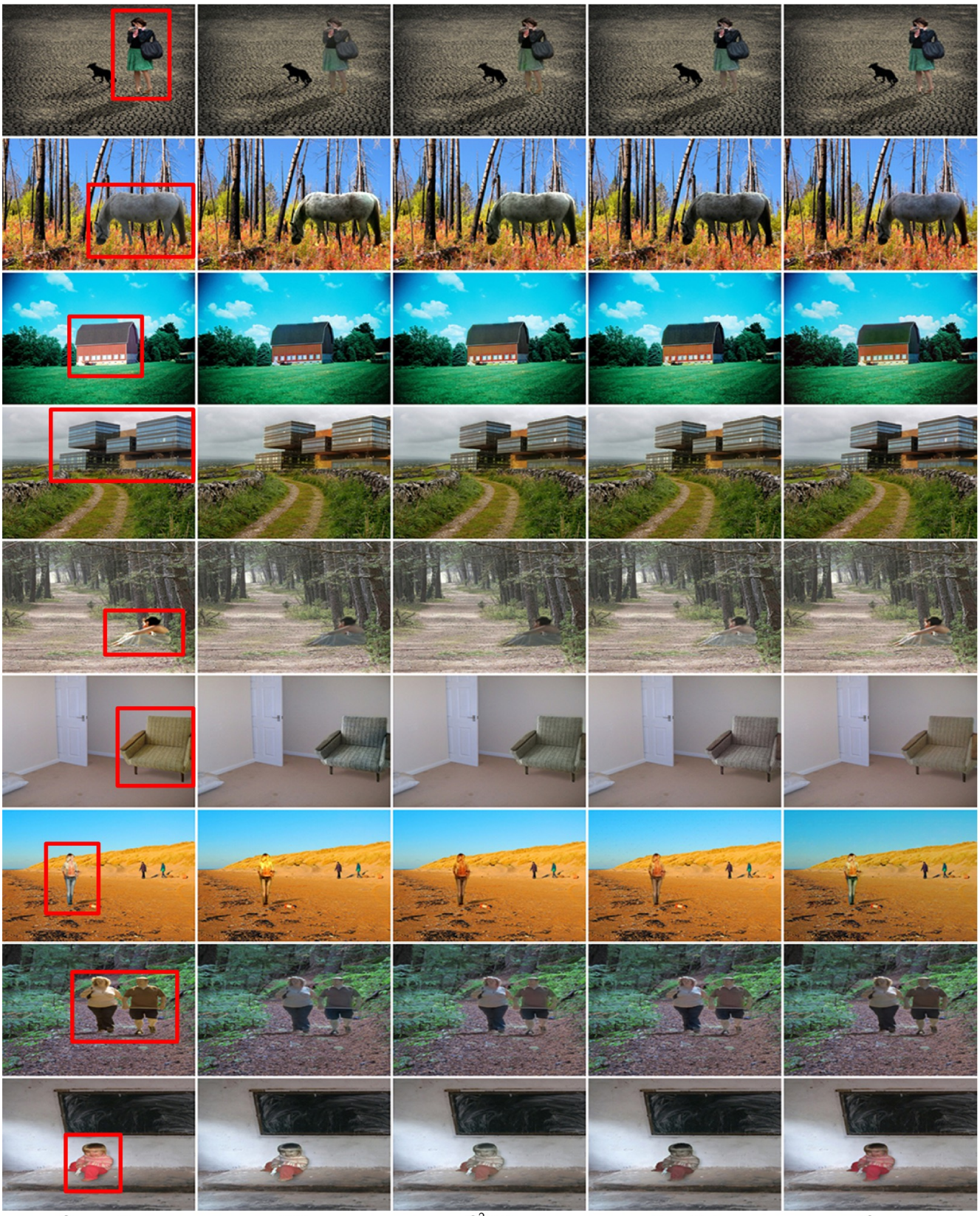

Figure P. Visual comparison results on real composite images. Red boxes in composite images mark foreground.

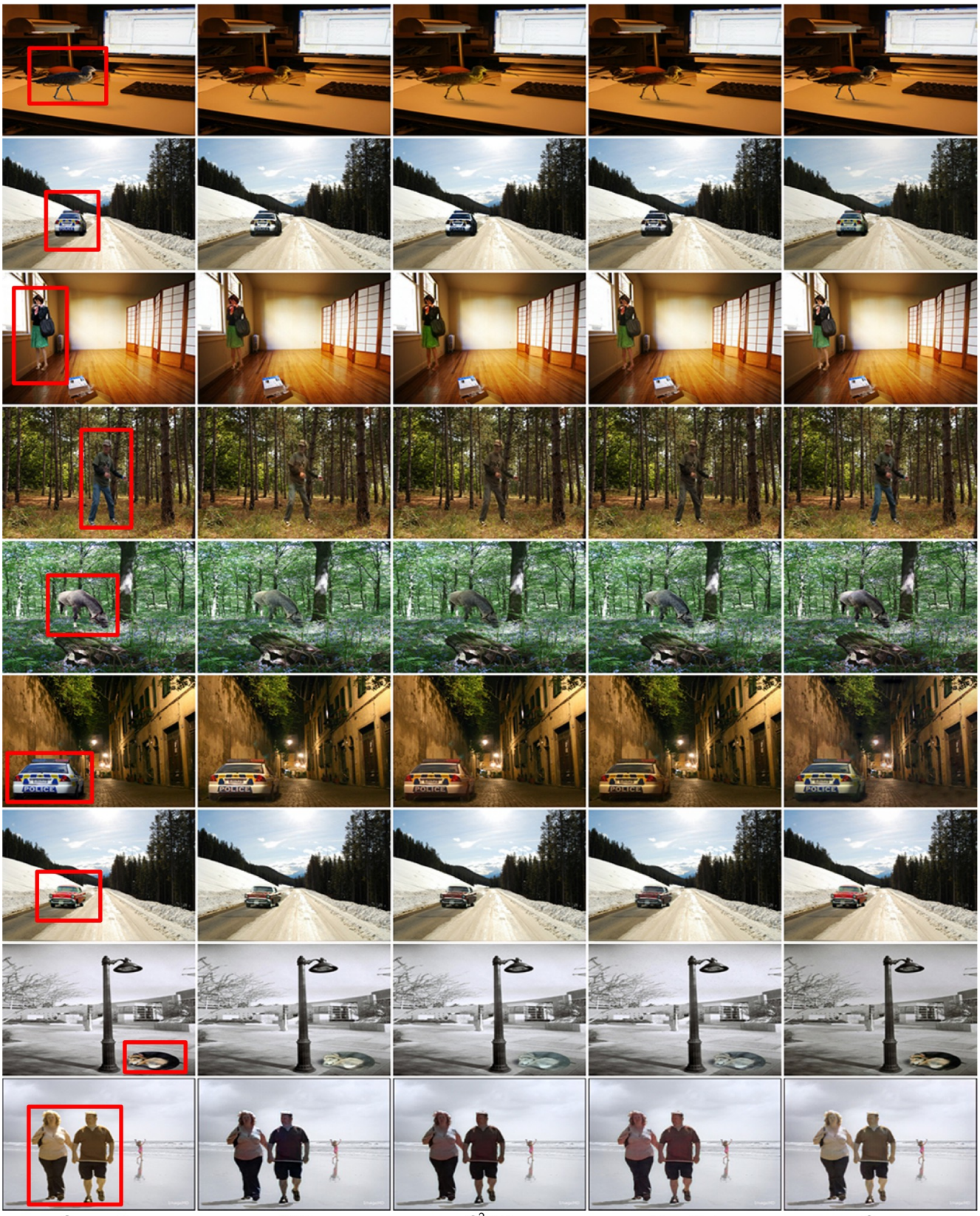

Figure Q. Visual comparison results on real composite images. Red boxes in composite images mark foreground.