Supplementary Material for Learning to Deblur using Light Field Generated and Real Defocus Images

Lingyan Ruan[1]* Bin Chen[2]* Jizhou Li[3] Miuling Lam[1]†
[1]City University of Hong Kong [2]Max-Planck-Institut für Informatik [3]Stanford University
http://lyruan.com/Projects/DRBNet

In this supplementary material, we show more details on network computational cost (Sec. 1), PSF visualization of conventional digital camera and light field camera (Sec. 2), failure cases (Sec. 3), detailed network architecture (Sec. 4), effect of LFDOF (Sec. 5), qualitative results (AIFNet *vs*. Ours) (Sec. 6) and additional qualitative comparison between IFANet [2] and our method tested on DPDD [1], RealDOF [2], CUHK [5] and PixelDP [1] (Sec. 7).

1. Computational Cost

We list the computational cost in terms of the number of network parameters (M), the number of multiply accumulated operations (MAC) on an image of size 1280×720 and the average computation time on 50 RealDOF images. DPDNet$_S$ is implemented with Keras framework on top of TensorFlow. AIFNet and KPAC are using TensorFlow. IFANet and ours are based on PyTorch framework. As shown in Tab. S1, our final model achieves the best performance while attaining competitive running time. Besides, we list the performance of our baseline model, which outperforms existing methods in terms of computation time.

Model	Evaluations		Computational Costs		
	PSNR↑	SSIM↑	Params (M)	MACs (B)	Time (s)
DPDNet$_S$ [1]	22.870	0.670	31.03	770	0.69
AIFNet [4]	23.093	0.680	41.55	1747	1.25
IFANet [2]	24.709	0.749	10.48	363	0.014
KPAC [6]	23.984	0.716	**2.06**	**113**	0.22
Our$_b$	25.327	0.749	6.96	326	**0.004**
Our	**25.745**	**0.771**	11.69	693	0.011

Tab. S1. Compuational costs comparison.

2. PSF Visualization

We estimate and visualize the PSF obtained by both conventional digital camera and light field camera using the algorithm proposed by Mannan and Langer [3] and Xin *et al*. [7]. Specifically, we display an image (3840 × 2160) that contains 8 × 13 disk patterns (Fig. S1) on a monitor which is imaged at a distance of 50cm using a Canon EOS R5 and a Lytro Illum. Multiple photos with different defocus blur levels are captured by adjusting the focal distance of the Canon camera to the depth behind and in front of the monitor. While for light field camera, only one capture is needed. Refocusing is then performed using Lytro Power Tool. The synthetic f number of $f/1.4$ is used.

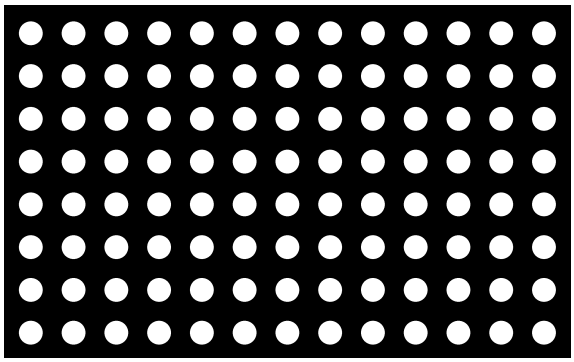

Fig. S1. 8 × 13 disk calibration pattern.

After capturing, we binarize the images and detect the centers of all disks, based on which a homography with respect to the calibration patterns (Fig. S1) can be calculated. We warp the images to eliminate any camera pose misalignment during image capturing. We show more PSF estimation results of Canon EOS R5 in Fig. S3 and S4 and that of Lytro in Fig. S5 and S6. From top to bottom, the depth is increased to cover three typical focus cases: front focus, in focus and back focus.

3. Failure Cases

Our network performs well in removing defocus blur but it cannot handle motion blur and saturated bokeh. We show some failure cases in Fig. S2.

∗ denotes equal contribution and † denotes corresponding author.

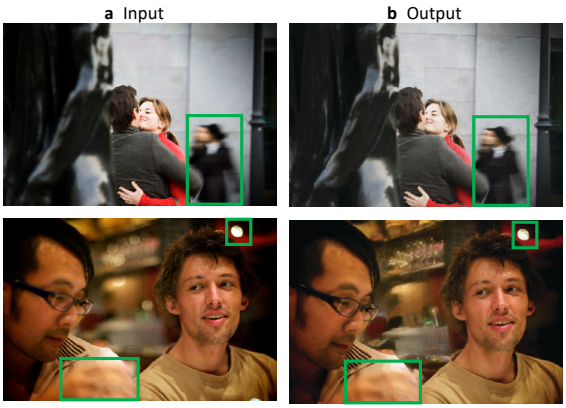

Fig. S2. Failure cases. The input images are from CUHK defocus blur detection dataset [5].

4. Detailed Network Architecture

The detailed parameters and network structure are shown in Tab. S2 and S3.

Encoder								
Type	Input	In	Out	K	S	Act.	N	output
conv	x	3	32	3	1	lrelu	×1	c1
conv	c1	32	32	3	1	lrelu	×2	c2
conv	c2	32	64	3	2	lrelu	×1	c3
conv	c3	64	64	3	1	lrelu	×2	c4
conv	c4	64	128	3	2	lrelu	×1	c5
conv	c5	128	128	3	1	lrelu	×2	c6
conv	c6	128	256	3	2	lrelu	×1	c7
conv	c7	256	256	3	1	lrelu	×1	c8
res-8	c8	256	256	3	1	lrelu	×2	r1
conv	r1	256	256	3	1	lrelu	×1	$d_{\downarrow_8}$
Decoder								
upconv	$d_{\downarrow_8}$	256	128	4	2	lrelu	×1	c9
sum	(c6, c9)	-	-	-	-	-	×1	c10
res-4	c10	128	128	3	1	lrelu	×2	$d_{\downarrow_4}$
upconv	$d_{\downarrow_4}$	128	64	4	2	lrelu	×1	c11
sum	(c4, c11)	-	-	-	-	-	×1	c12
res-2	c12	64	64	3	1	lrelu	×2	$d_{\downarrow_2}$
upconv	$d_{\downarrow_2}$	64	32	4	2	lrelu	×1	c13
sum	(c2, c13)	-	-	-	-	-	×1	c14
res-1	c14	32	32	3	1	lrelu	×2	$d_{\downarrow_1}$
res-i								
identity	input	-	-	-	-	-	-	c_{in}
conv	c_{in}	i×32	i×32	3	1	lrelu	×1	c_r
conv	c_r	i×32	i×32	3	1	-	×1	c_r
sum	(c_{in}, c_r)	-	-	-	-	-	×1	c_r

Tab. S2. Network architecture.

DRB $\downarrow_8$								
conv	$x_{\downarrow_8}$	3	64	3	1	lrelu	×1	$c1_{\downarrow_8}$
conv	$c1_{\downarrow_8}$	64	128	3	1	lrelu	×1	$c2_{\downarrow_8}$
conv	$c2_{\downarrow_8}$	128	256	3	1	lrelu	×1	$c3_{\downarrow_8}$
concat	($d_{\downarrow_8}$, $c3_{\downarrow_8}$)	-	-	-	-	-	×1	$cat_{\downarrow_8}$
conv	$cat_{\downarrow_8}$	512	256	3	1	lrelu	×1	$ck1_{\downarrow_8}$
conv	$ck1_{\downarrow_8}$	256	128	3	1	lrelu	×1	$ck2_{\downarrow_8}$
conv	$ck2_{\downarrow_8}$	128	7×7	1	1	-	×1	$K_{\downarrow_8}$
dynconv	($\hat{x}_{\downarrow_8}$,$K_{\downarrow_8}$)	-	-	-	-	-	×1	$\Delta\hat{x}_{\downarrow_8}$
conv	$cat_{\downarrow_8}$	512	256	3	1	lrelu	×1	$cr1_{\downarrow_8}$
conv	$cr1_{\downarrow_8}$	256	64	1	1	lrelu	×1	$cr2_{\downarrow_8}$
conv	$cr2_{\downarrow_8}$	64	3	1	1	lrelu	×1	$\Delta r_{\downarrow_8}$
sum	($x_{\downarrow_8}$, $\Delta\hat{x}_{\downarrow_8}$, $\Delta r_{\downarrow_8}$)	-	-	-	-	-	×1	$\hat{y}_{\downarrow_8}$
$\uparrow_2$	$\hat{y}_{\downarrow_8}$	-	-	-	-	-	×1	$\hat{x}_{\downarrow_4}$
DRB $\downarrow_4$								
conv	$\hat{x}_{\downarrow_4}$	3	64	3	1	lrelu	×1	$c1_{\downarrow_4}$
conv	$c1_{\downarrow_4}$	64	128	3	1	lrelu	×1	$c2_{\downarrow_4}$
conv	$c2_{\downarrow_4}$	128	128	3	1	lrelu	×1	$c3_{\downarrow_4}$
concat	($d_{\downarrow_4}$, $c3_{\downarrow_4}$)	-	-	-	-	-	×1	$cat_{\downarrow_4}$
conv	$cat_{\downarrow_4}$	256	128	3	1	lrelu	×1	$ck1_{\downarrow_4}$
conv	$ck1_{\downarrow_4}$	128	128	3	1	lrelu	×1	$ck2_{\downarrow_4}$
conv	$ck2_{\downarrow_4}$	128	7×7	1	1	-	×1	$K_{\downarrow_4}$
dynconv	($\hat{x}_{\downarrow_4}$,$K_{\downarrow_4}$)	-	-	-	-	-	×1	$\Delta\hat{x}_{\downarrow_4}$
conv	$cat_{\downarrow_4}$	256	128	3	1	lrelu	×1	$cr1_{\downarrow_4}$
conv	$cr1_{\downarrow_4}$	128	64	3	1	lrelu	×1	$cr2_{\downarrow_4}$
conv	$cr2_{\downarrow_4}$	64	3	1	1	lrelu	×1	$\Delta r_{\downarrow_4}$
sum	($\hat{x}_{\downarrow_4}$, $\Delta\hat{x}_{\downarrow_4}$, $\Delta r_{\downarrow_4}$)	-	-	-	-	-	×1	$\hat{y}_{\downarrow_4}$
$\uparrow_2$	$\hat{y}_{\downarrow_4}$	-	-	-	-	-	×1	$\hat{x}_{\downarrow_2}$
DRB $\downarrow_2$								
conv	$\hat{x}_{\downarrow_2}$	3	64	3	1	lrelu	×1	$c1_{\downarrow_2}$
conv	$c1_{\downarrow_2}$	64	64	3	1	lrelu	×1	$c2_{\downarrow_2}$
conv	$c2_{\downarrow_2}$	64	64	3	1	lrelu	×1	$c3_{\downarrow_2}$
concat	($d_{\downarrow_2}$, $c3_{\downarrow_2}$)	-	-	-	-	-	×1	$cat_{\downarrow_2}$
conv	$cat_{\downarrow_2}$	128	64	3	1	lrelu	×1	$ck1_{\downarrow_2}$
conv	$ck1_{\downarrow_2}$	64	64	3	1	lrelu	×1	$ck2_{\downarrow_2}$
conv	$ck2_{\downarrow_2}$	64	7×7	1	1	-	×1	$K_{\downarrow_2}$
dynconv	($\hat{x}_{\downarrow_2}$,$K_{\downarrow_2}$)	-	-	-	-	-	×1	$\Delta\hat{x}_{\downarrow_2}$
conv	$cat_{\downarrow_2}$	128	64	3	1	lrelu	×1	$cr1_{\downarrow_2}$
conv	$cr1_{\downarrow_2}$	64	64	3	1	lrelu	×1	$cr2_{\downarrow_2}$
conv	$cr2_{\downarrow_2}$	64	3	1	1	lrelu	×1	$\Delta r_{\downarrow_2}$
sum	($\hat{x}_{\downarrow_2}$, $\Delta\hat{x}_{\downarrow_2}$, $\Delta r_{\downarrow_2}$)	-	-	-	-	-	×1	$\hat{y}_{\downarrow_2}$
$\uparrow_2$	$\hat{y}_{\downarrow_2}$	-	-	-	-	-	×1	$\hat{x}_{\downarrow_1}$
DRB								
conv	$\hat{x}_{\downarrow_1}$	3	64	3	1	lrelu	×1	$c1_{\downarrow_1}$
conv	$c1_{\downarrow_1}$	64	64	3	1	lrelu	×1	$c2_{\downarrow_1}$
conv	$c2_{\downarrow_1}$	64	32	3	1	lrelu	×1	$c3_{\downarrow_1}$
concat	($d_{\downarrow_1}$, $c3_{\downarrow_1}$)	-	-	-	-	-	×1	$cat_{\downarrow_1}$
conv	$cat_{\downarrow_1}$	64	32	3	1	lrelu	×1	$ck1_{\downarrow_1}$
conv	$ck1_{\downarrow_1}$	64	64	3	1	lrelu	×1	$ck2_{\downarrow_1}$
conv	$ck2_{\downarrow_1}$	64	7×7	1	1	-	×1	$K_{\downarrow_1}$
dynconv	($\hat{x}_{\downarrow_1}$,$K_{\downarrow_1}$)	-	-	-	-	-	×1	$\Delta\hat{x}_{\downarrow_1}$
conv	$cat_{\downarrow_1}$	64	64	3	1	lrelu	×1	$cr1_{\downarrow_1}$
conv	$cr1_{\downarrow_1}$	64	64	3	1	lrelu	×1	$cr2_{\downarrow_1}$
conv	$cr2_{\downarrow_1}$	64	3	1	1	lrelu	×1	$\Delta r_{\downarrow_1}$
sum	($\hat{x}_{\downarrow_1}$, $\Delta\hat{x}_{\downarrow_1}$, $\Delta r_{\downarrow_1}$)	-	-	-	-	-	×1	$\hat{y}_{\downarrow_1}$

Tab. S3. Multi-scale dynamic residual block architecture.

5. Effect of LFDOF

Fig. S7 and S8 show the effect of LFDOF dataset on the network performance supplementing Tab. 3 and Fig. 7 in the paper (tested on RealDOF [2]). Specifically, Fig. S7

clearly shows that the network trained on LFDOF is able to restore very sharp content (second column), but the deblurred result is different from the ground truth (last column) owing to the domain difference. Despite showing clear texture, it may fail in some regions as shown in Fig. S8 due to the blur discrepancy between two datasets. It is proven that training on LFDOF followed by DPDD can largely improve the network performance in all the regions (fourth column in Fig. S7 and S8) and successfully adapt the real image domain.

6. Qualitative Result (AIFNet *vs*. Ours)

We show qualitative results on the LFDOF [4] test set (Fig. S23 and S22) supplementing Tab. 8 in the main text. AIFNet [4] and ours are both using the same training and testing sets in LFDOF, while our method presents more details and textures than AIFNet does.

7. Addition Qualitative Results

We show more qualitative results on DPDD [1] (Figs. S9, S10, S11 and S12), RealDOF [2] (Figs. S13, S14, S15 and S16), CUHK [5] (Figs. S17, S18, S19 and S20) and PixeDP [1] (Fig. S21). Due to the limit of file size, we only show the comparison between our method and IFANet as it is the state-of-the-art method.

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, pages 111–126, 2020. 1, 3

[2] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021. 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24

[3] Fahim Mannan and Michael S Langer. Blur calibration for depth from defocus. In *Conference on Computer and Robot Vision*, pages 281–288, 2016. 1

[4] Lingyan Ruan, Bin Chen, Jizhou Li, and Miu-Ling Lam. Aifnet: All-in-focus image restoration network using a light field-based dataset. *IEEE Transactions on Computational Imaging*, 7:675–688, 2021. 1, 3

[5] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. 1, 2, 3

[6] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2642–2650, 2021. 1

[7] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T Barron, Pratul P Srinivasan, Jiawen Chen, Ioannis Gkioulekas, and Rahul Garg. Defocus map estimation and deblurring from a single dual-pixel image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2228–2238, 2021. 1

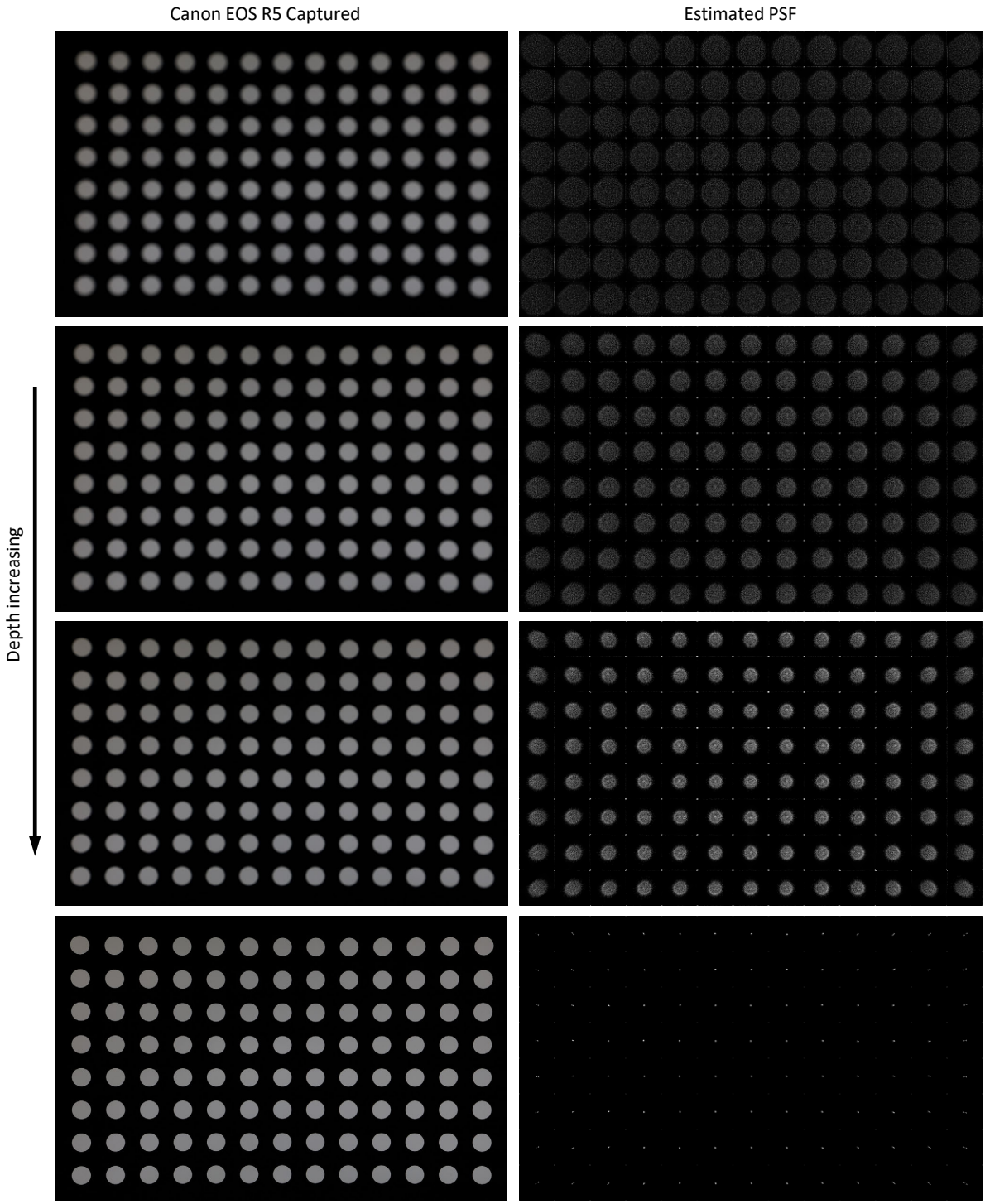

Fig. S3. **PSF estimation of Canon EOS R5 focus in front of the screen (first three rows) and on screen (last row).** Left: captured defocus images. Right: estimated PSF. Depth increasing from top to bottom.

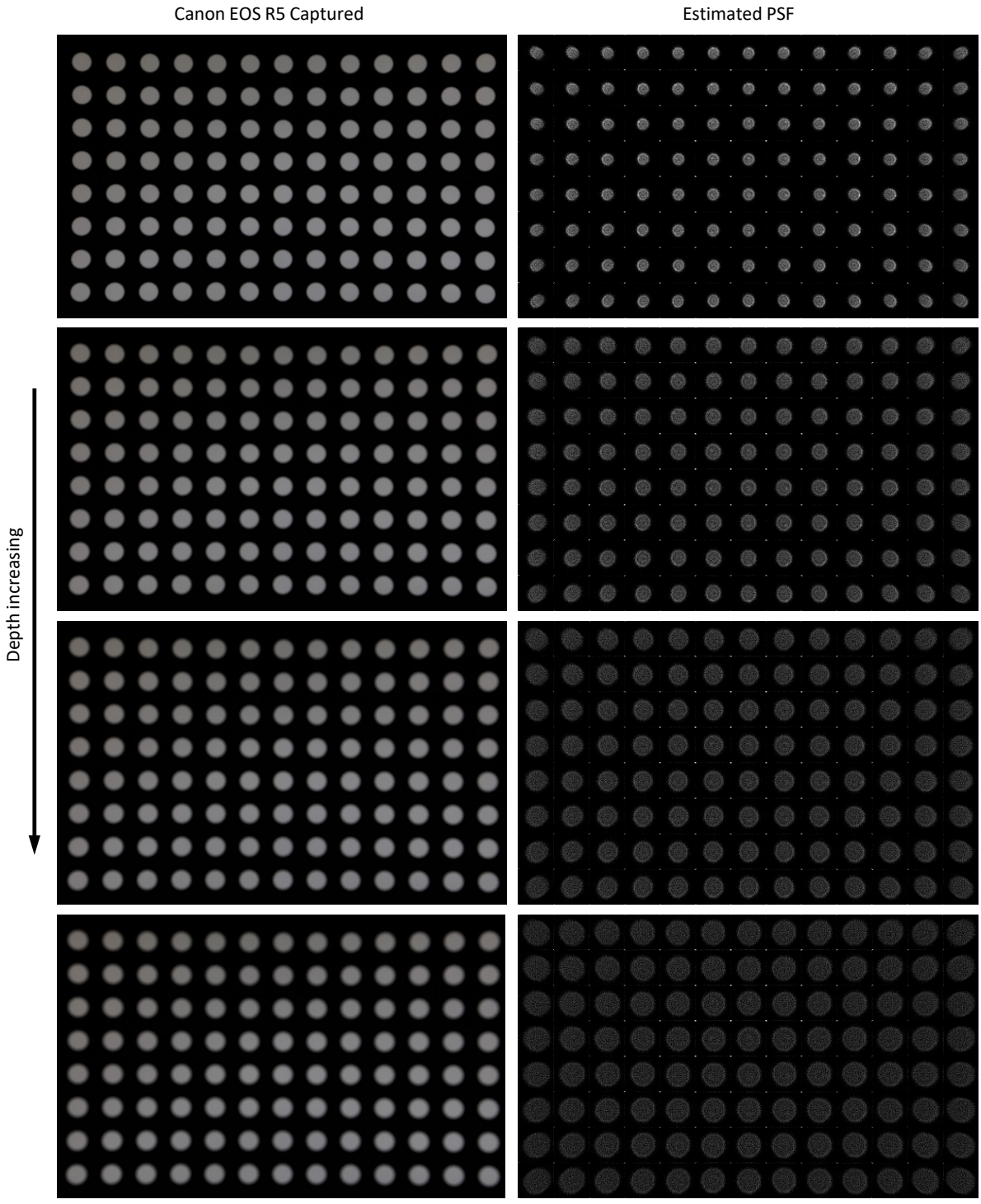

Fig. S4. **PSF estimation of Canon EOS R5 focus behind the screen.** Left: captured defocus images. Right: estimated PSF. Depth increasing from top to bottom.

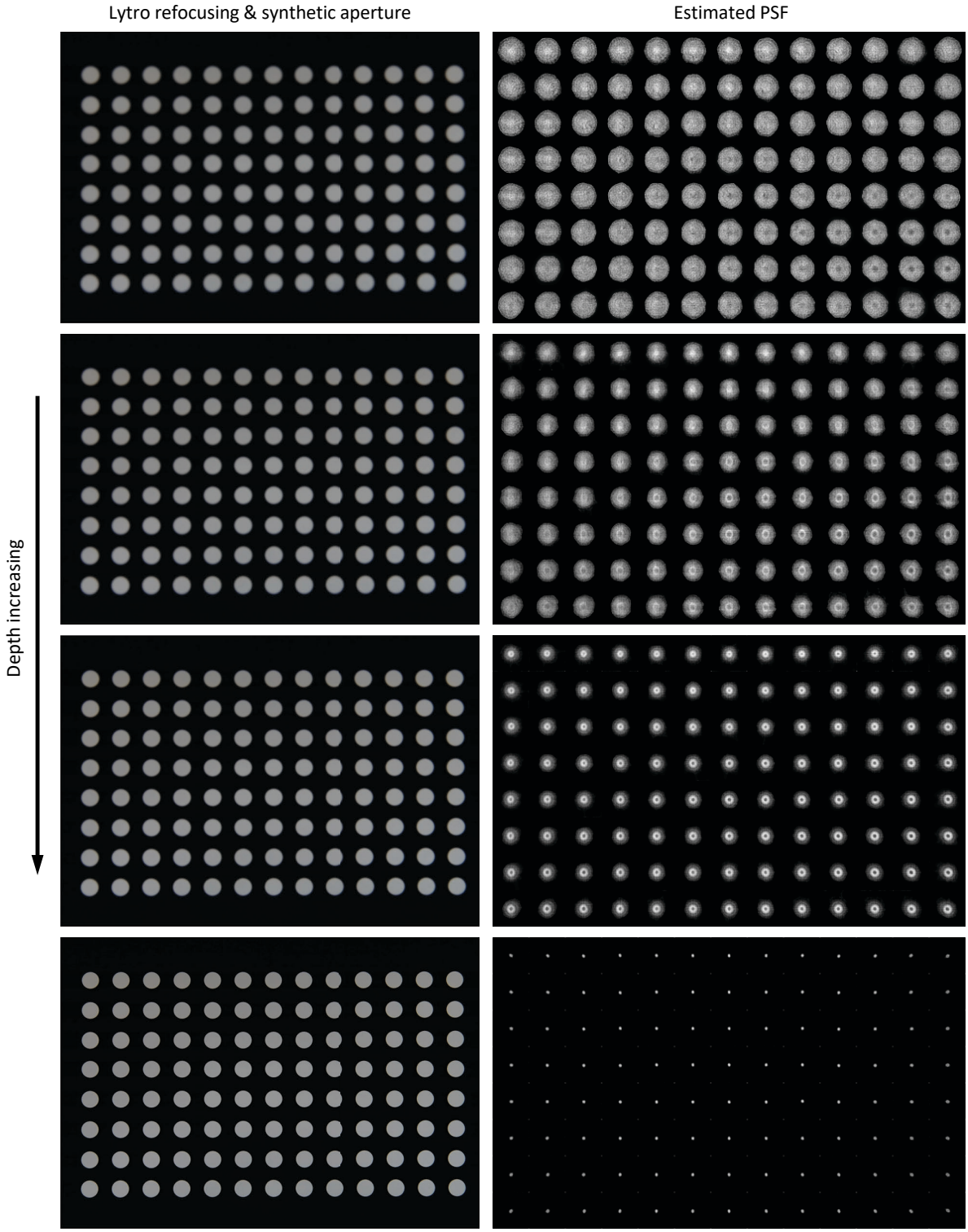

Fig. S5. **PSF estimation of Lytro refocuses to the depth in front of the screen (first three rows) and on screen (last row).** Left: refocused defocus images. Right: estimated PSF. Depth increasing from top to bottom.

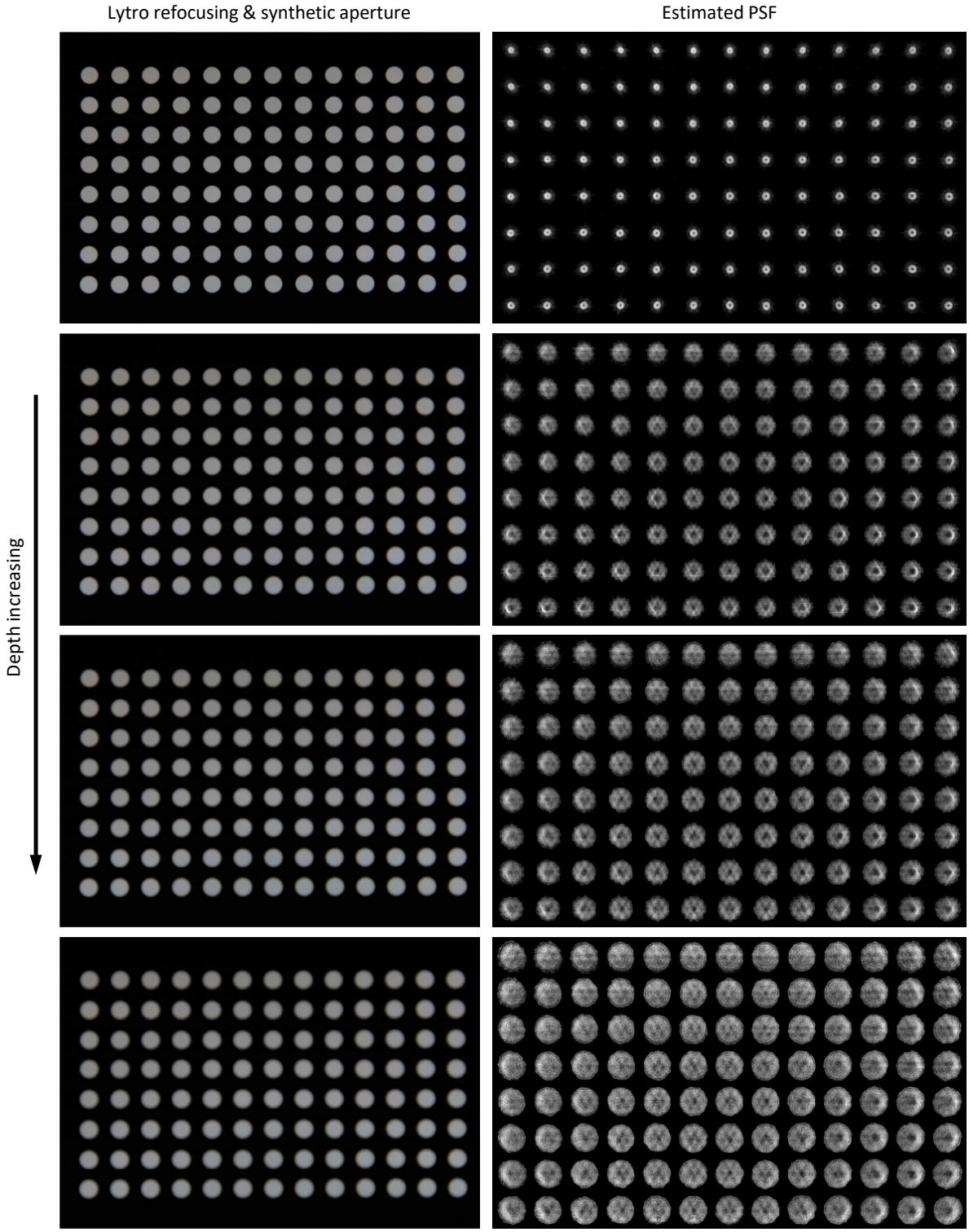

Fig. S6. **PSF estimation of Lytro refocuses to the depth behind the screen.** Left: refocused defocus images. Right: estimated PSF. Depth increasing from top to bottom.

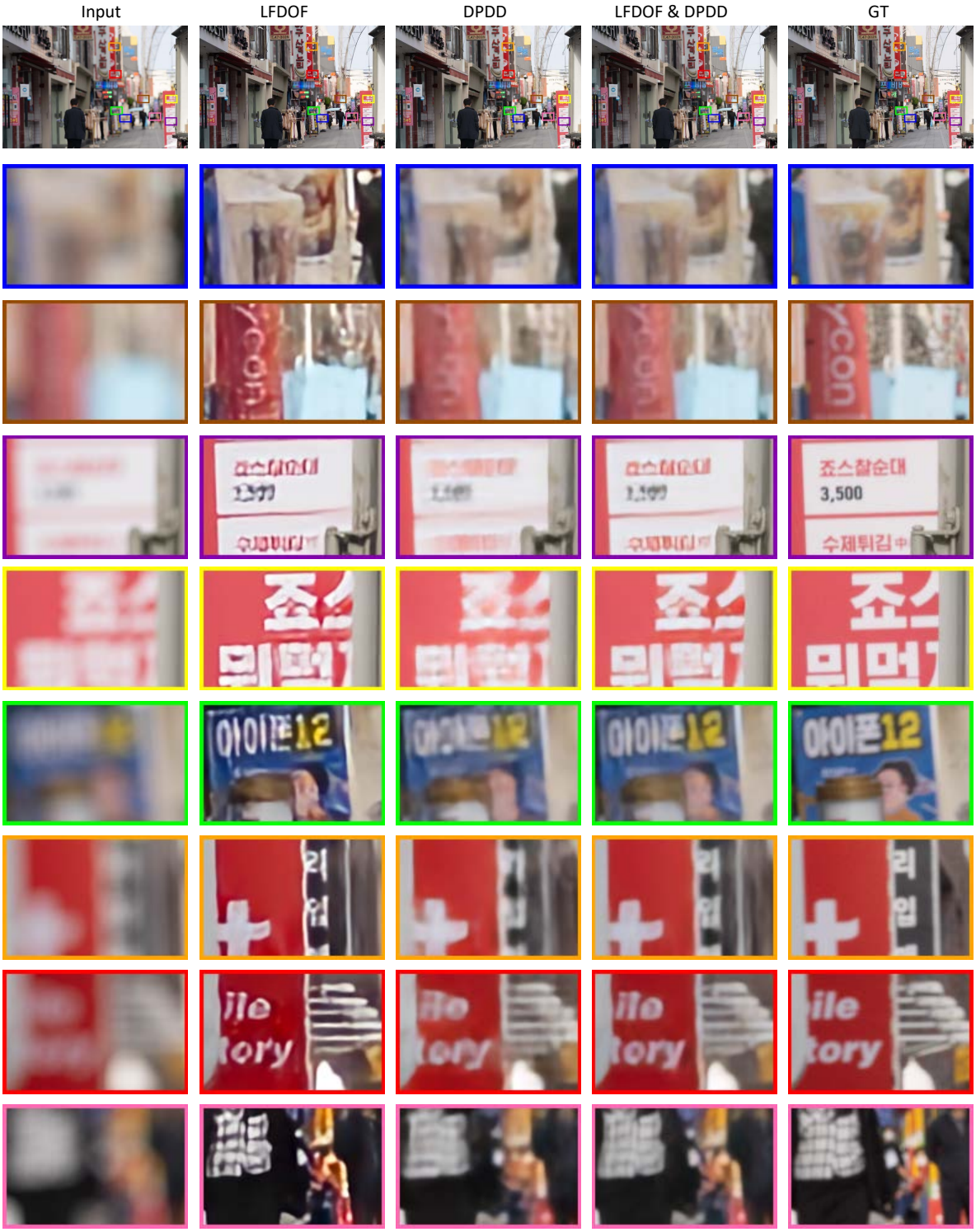

Fig. S7. **Qualitative comparison among our network trained on LFDOF, DPDD and both datasets.** The image sample is from RealDOF dataset [2]. The network trained on LFDOF gives much sharper details and is then adapted to the real image domain when fine-tuned by DPDD dataset. From left to right: defocused input, deblurred result evaluated on trained on LFDOF, DPDD, LFDOF & DPDD respectively, and ground truth.

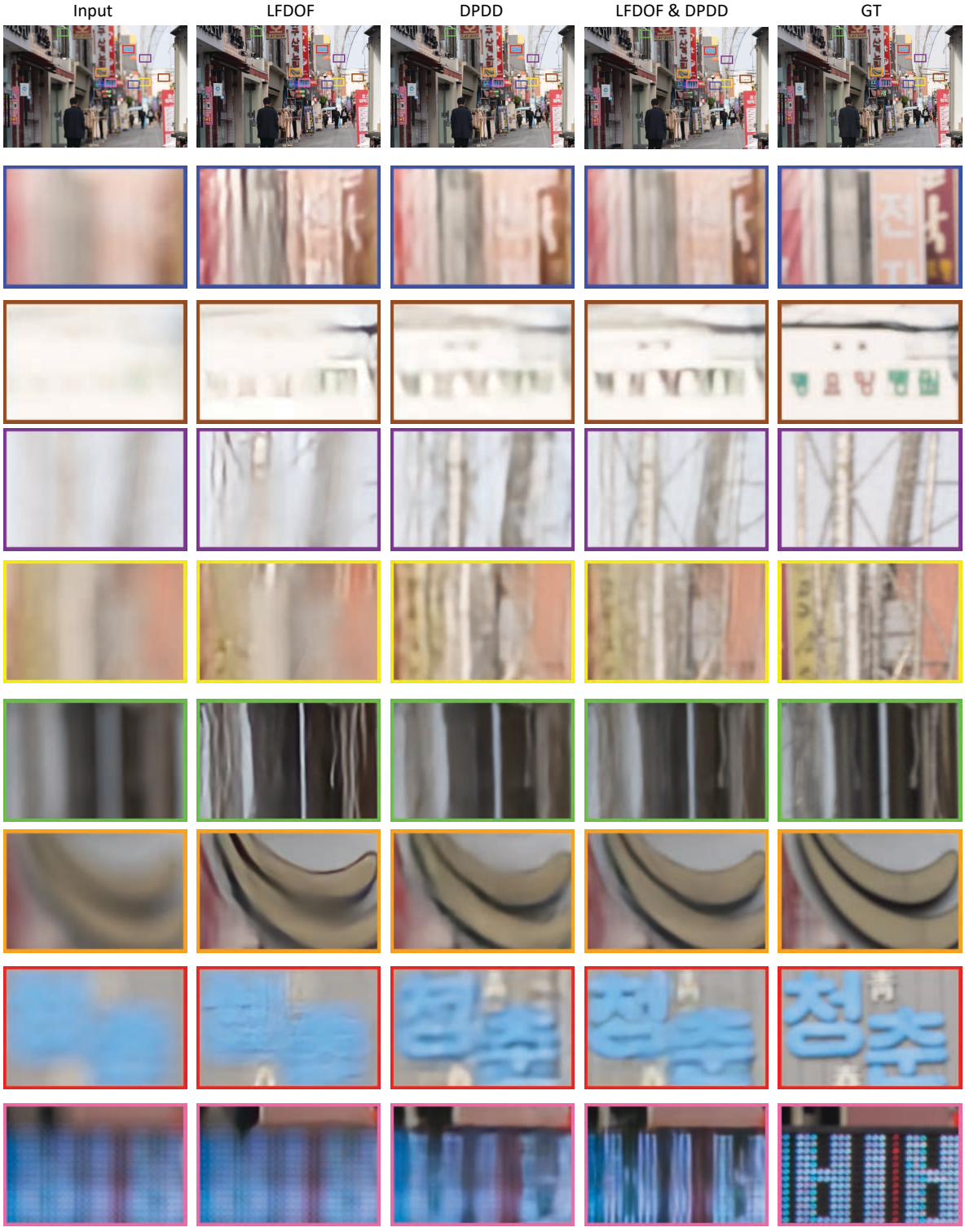

Fig. S8. **Qualitative comparison among our network trained on LFDOF, DPDD and both datasets.** The image sample is from RealDOF dataset [2]. The network trained on LFDOF fails to handle some regions due to the blur discrepancy and performs much better when fine-tuned by DPDD than the one trained on DPDD only. From left to right: defocused input, deblurred result evaluated on trained on LFDOF, DPDD, LFDOF & DPDD respectively, and ground truth.

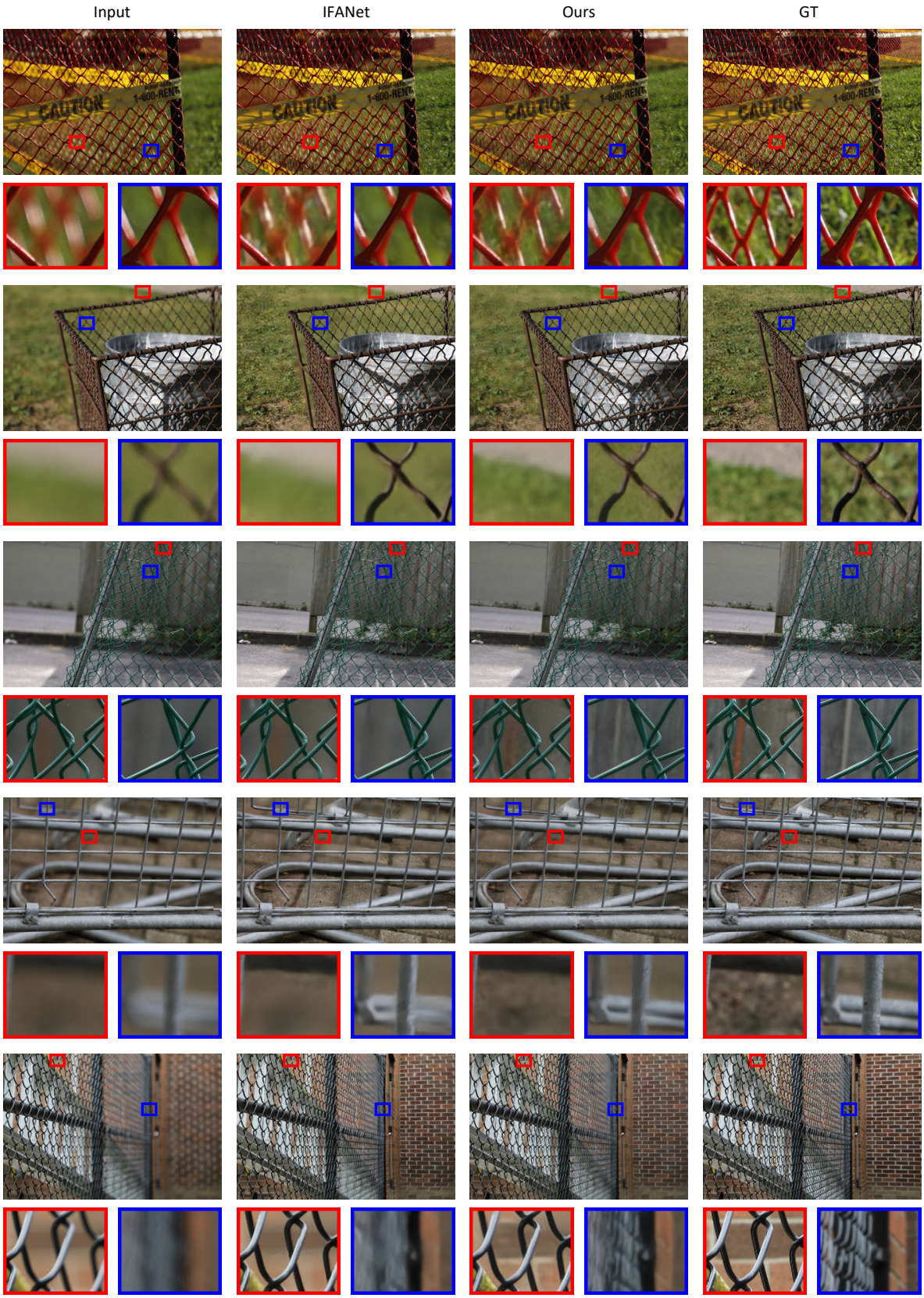

Fig. S9. **Qualitative comparison between IFAN [2] and our method evaluated on DPDD dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

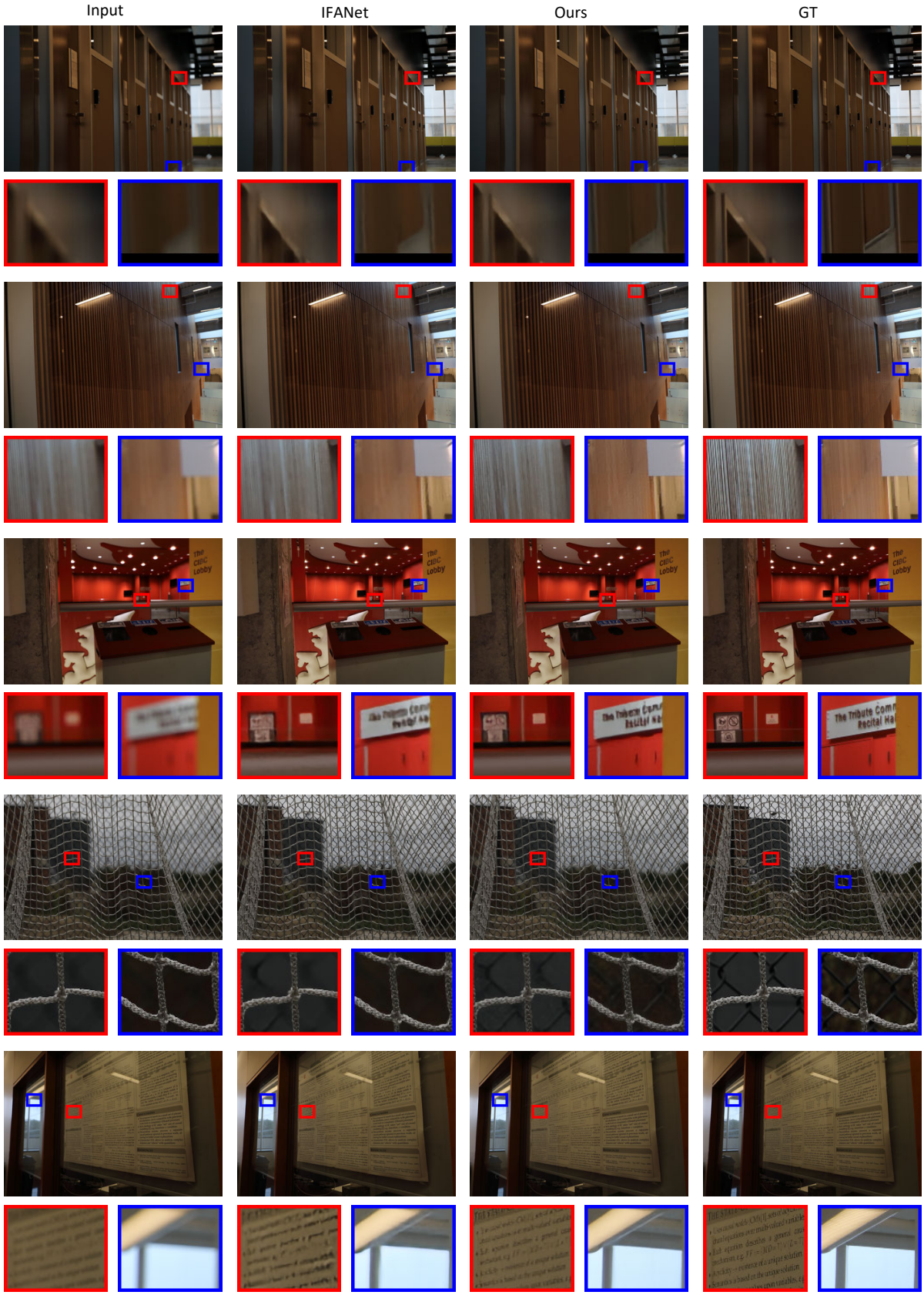

Fig. S10. **Qualitative comparison between IFAN [2] and our method evaluated on DPDD dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

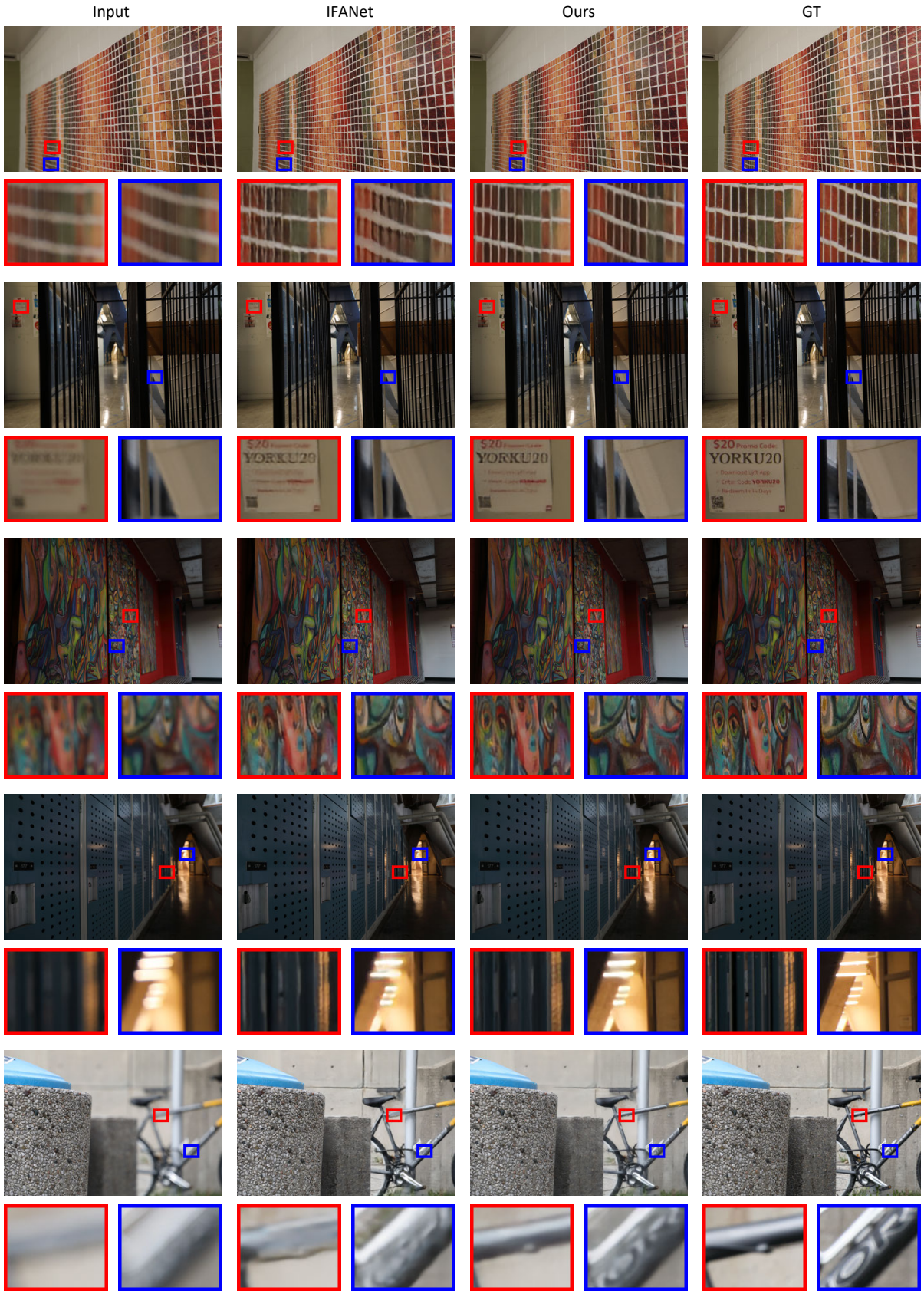

Fig. S11. **Qualitative comparison between IFAN [2] and ours evaluated on DPDD dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

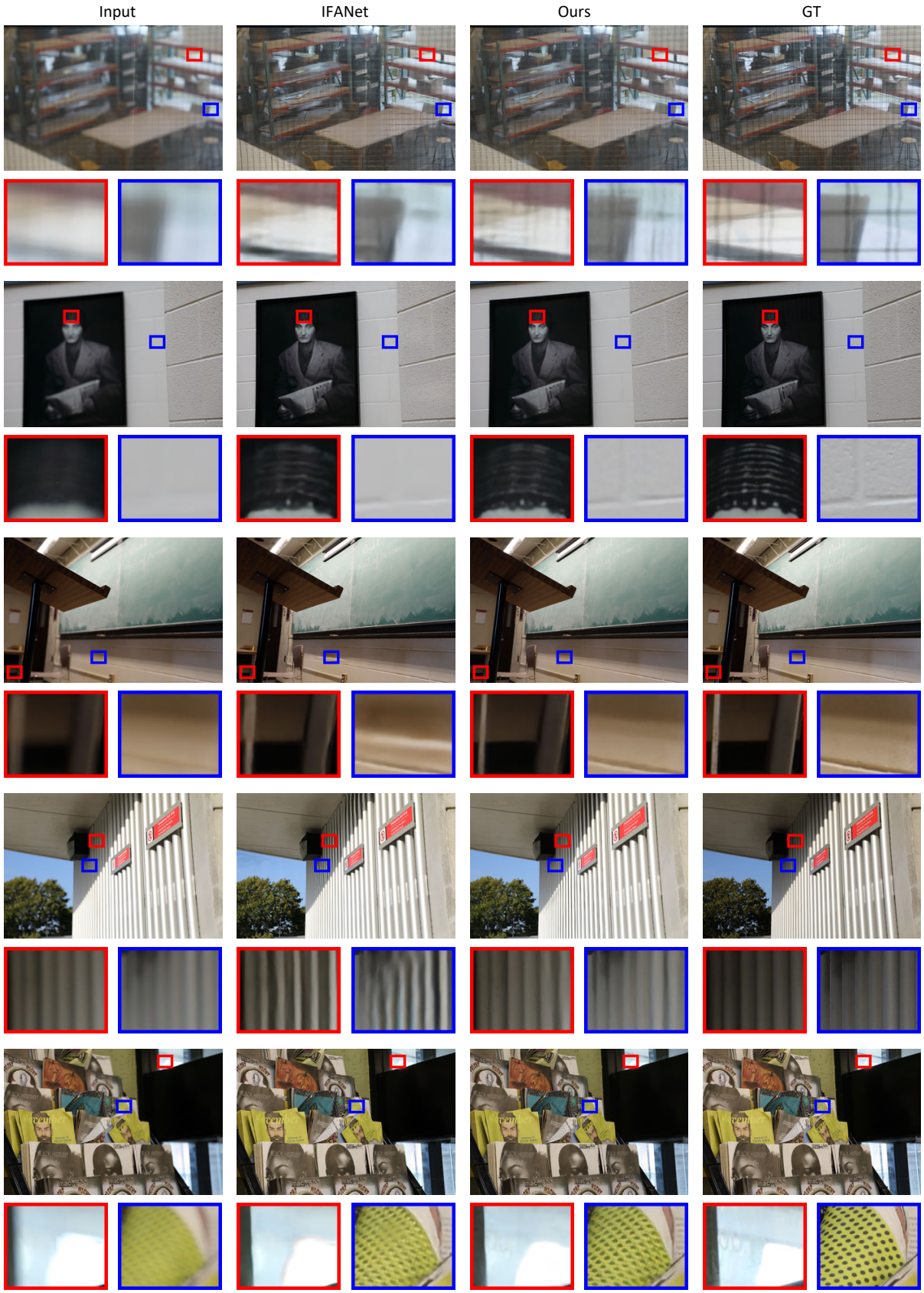

Fig. S12. **Qualitative comparison between IFAN [2] and ours evaluated on DPDD dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

Fig. S13. **Qualitative comparison between IFAN [2] and ours evaluated on RealDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

Fig. S14. **Qualitative comparison between IFAN [2] and ours evaluated on RealDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

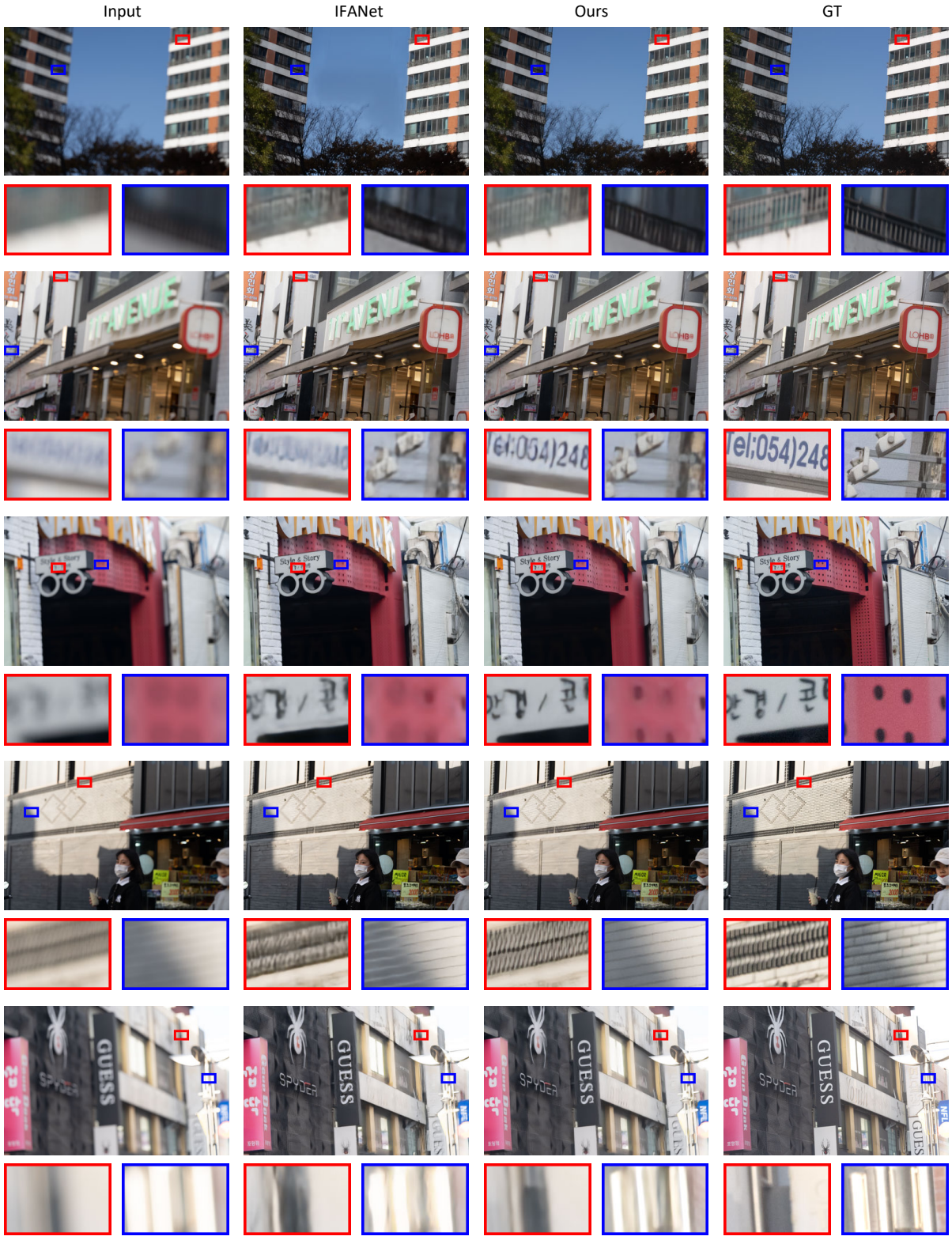

Fig. S15. **Qualitative comparison between IFAN [2] and ours evaluated on RealDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

Fig. S16. **Qualitative comparison between IFAN [2] and ours evaluated on RealDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

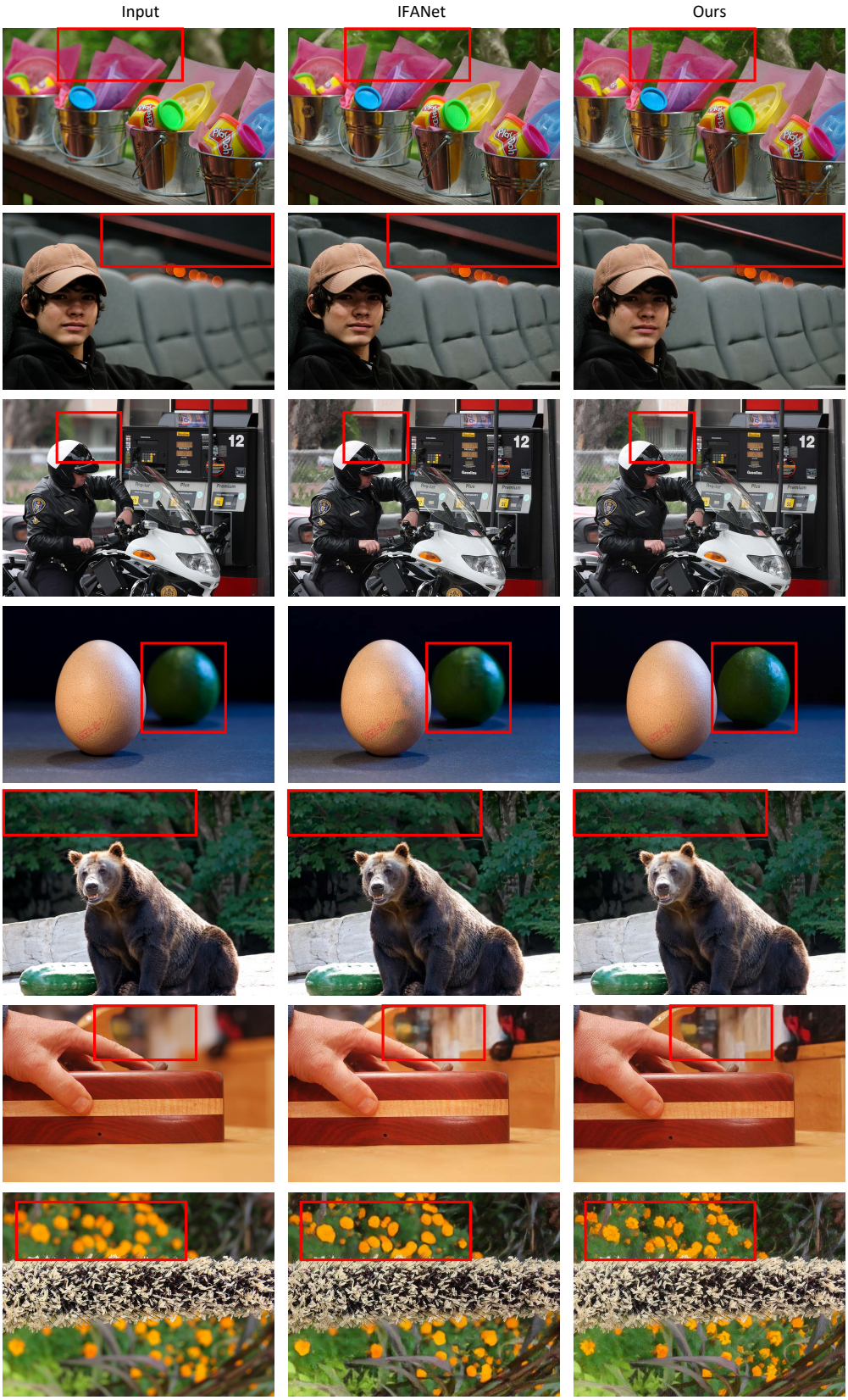

Fig. S17. **Qualitative comparison between IFAN [2] and ours evaluated on CUHK dataset.** From left to right: defocused input, deblurred by IFAN and ours. No ground truth is available from CUHK dataset.

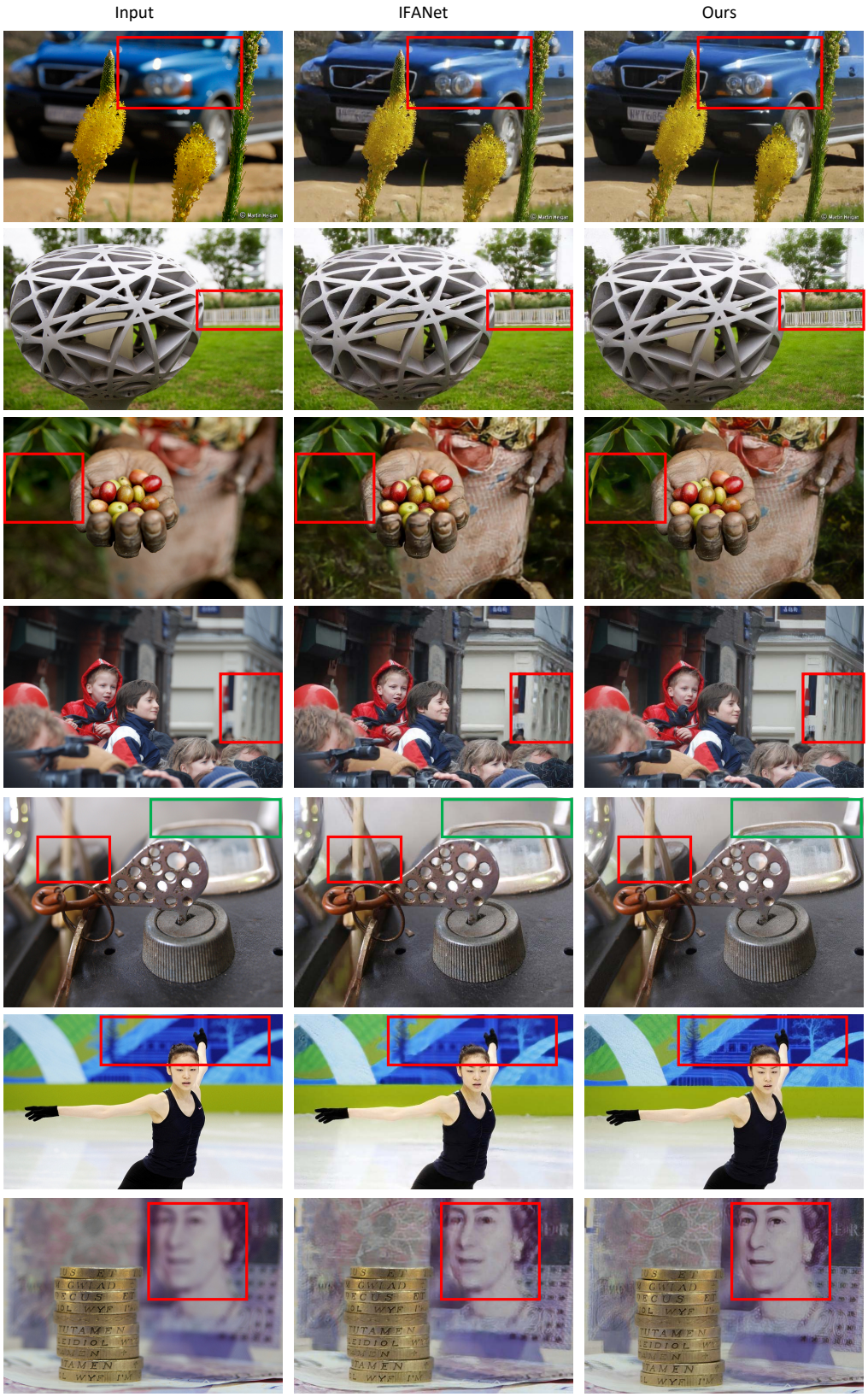

Fig. S18. **Qualitative comparison between IFAN [2] and ours evaluated on CUHK dataset.** From left to right: defocused input, deblurred by IFAN and ours. No ground truth is available from CUHK dataset.

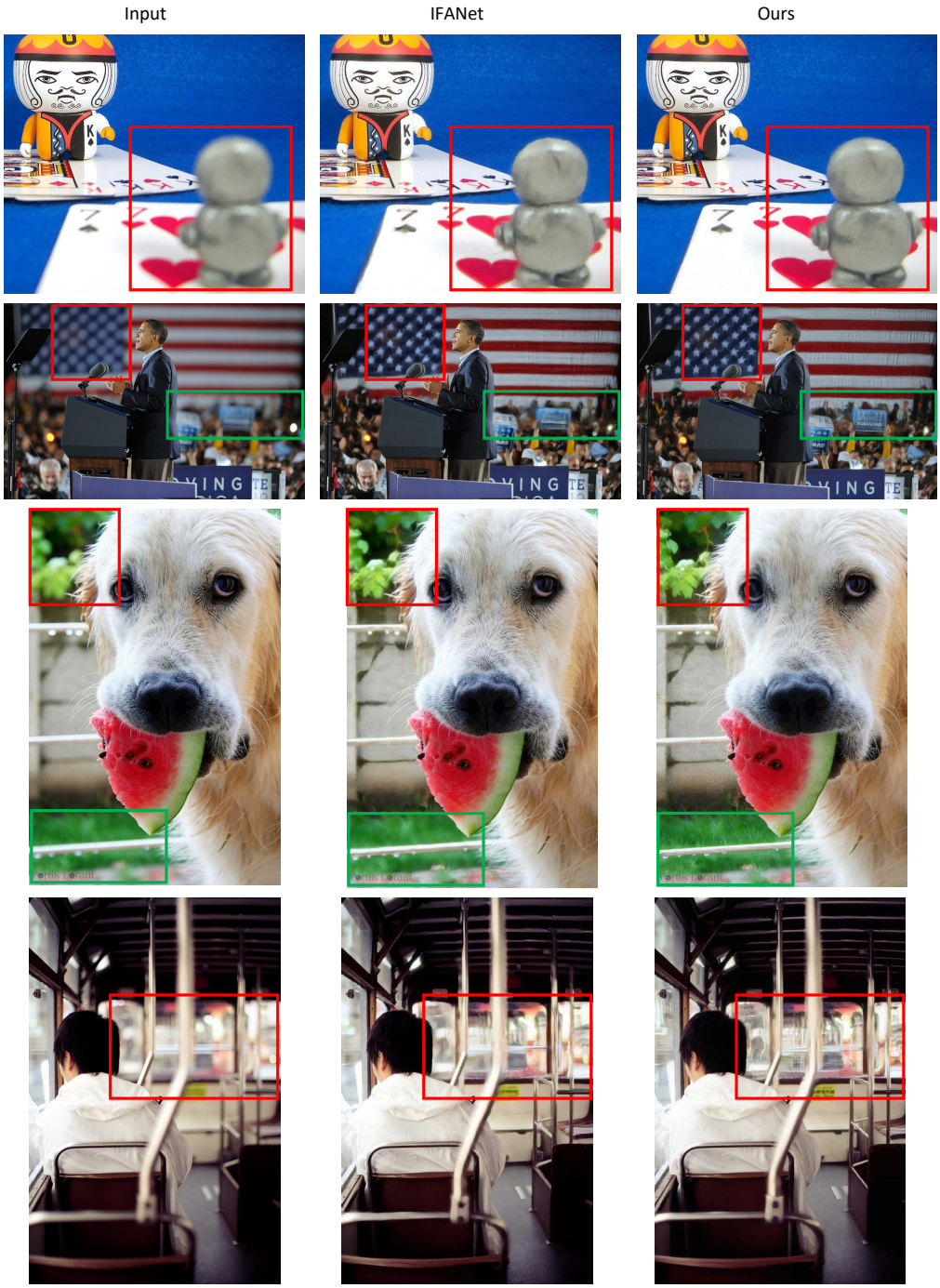

Fig. S19. **Qualitative comparison between IFAN [2] and ours evaluated on CUHK dataset.** From left to right: defocused input, deblurred by IFAN and ours. No ground truth is available from CUHK dataset.

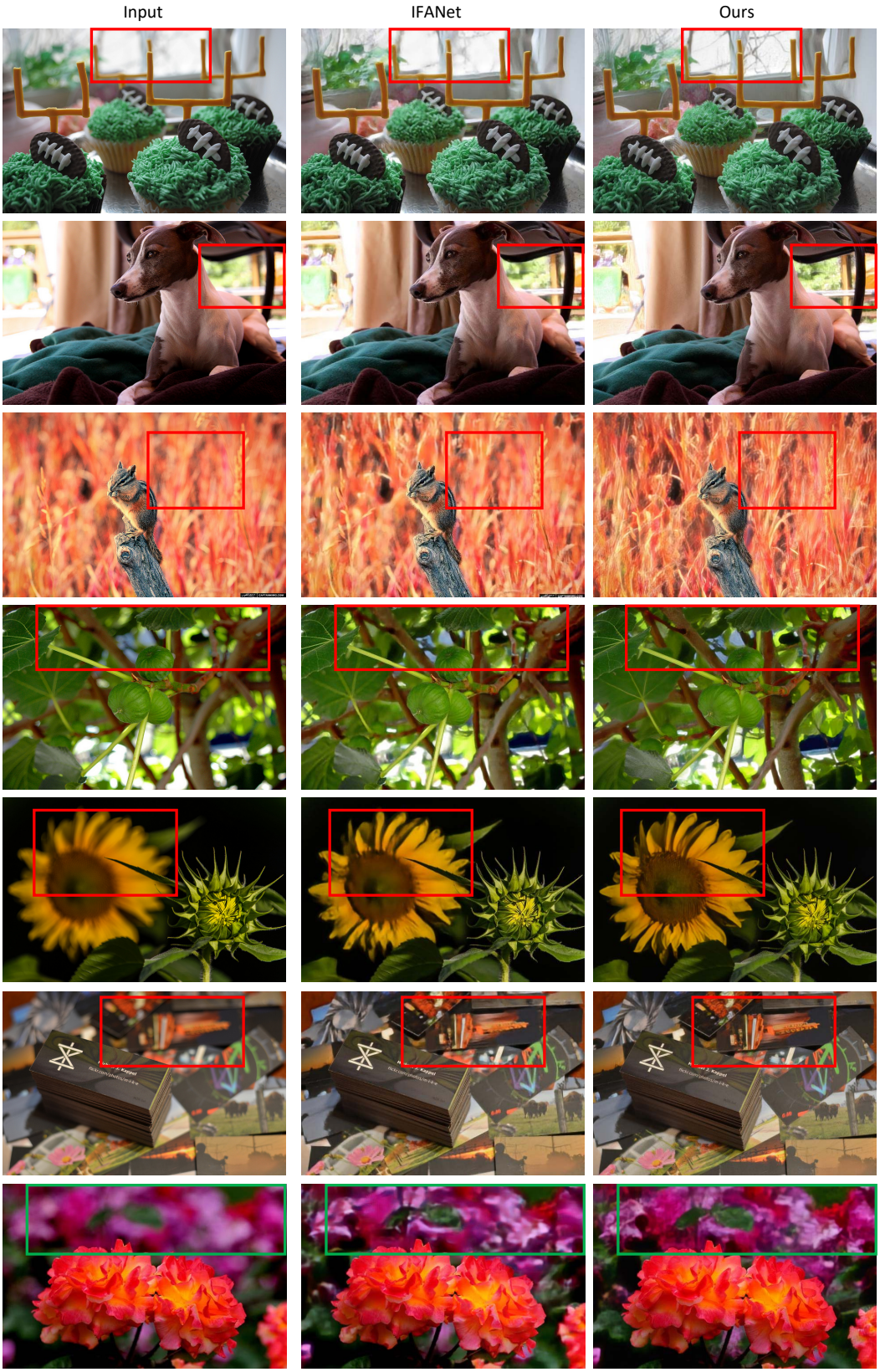

Fig. S20. **Qualitative comparison between IFAN [2], and ours evaluated on CUHK dataset.** From the left to right: defocused input, deblurred by IFAN and ours. No ground truth is available from CUHK dataset.

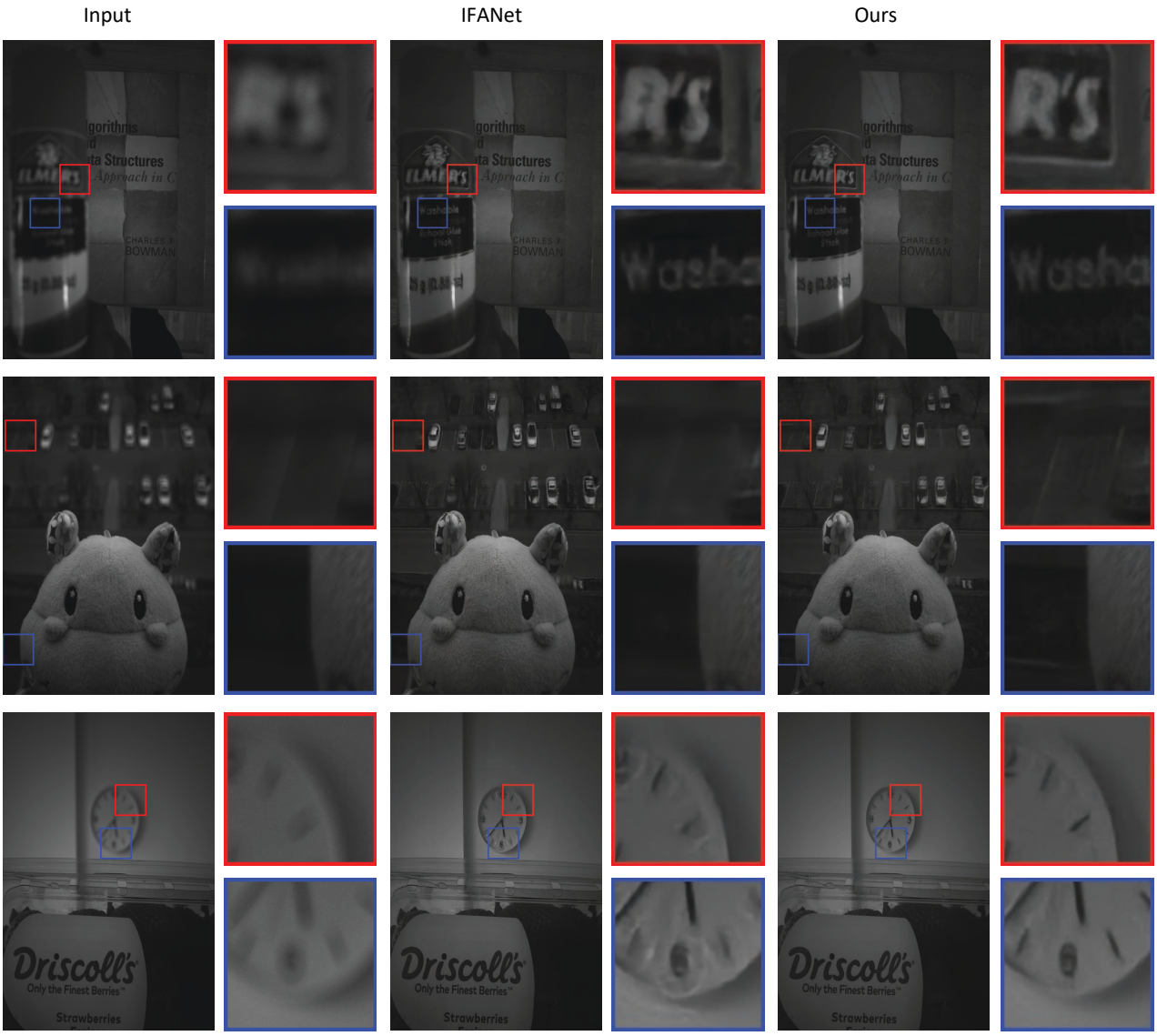

Fig. S21. **Qualitative comparison between IFAN [2] and ours evaluated on PixelDP dataset.** From left to right: defocused input, deblurred by IFAN and ours. No ground truth is available from PixelDP dataset.

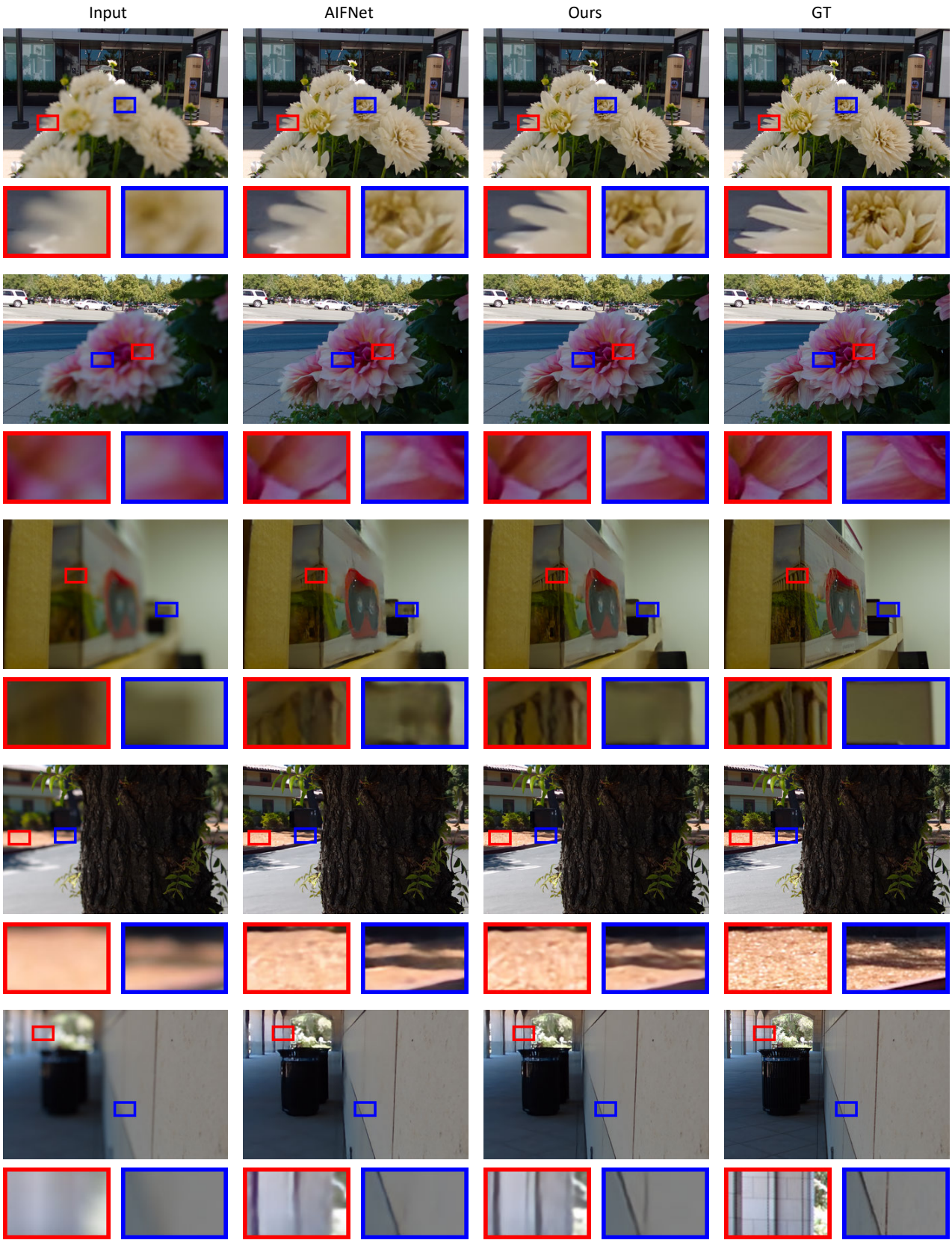

Fig. S22. **Qualitative comparison between IFAN [2] and ours evaluated on LFDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.

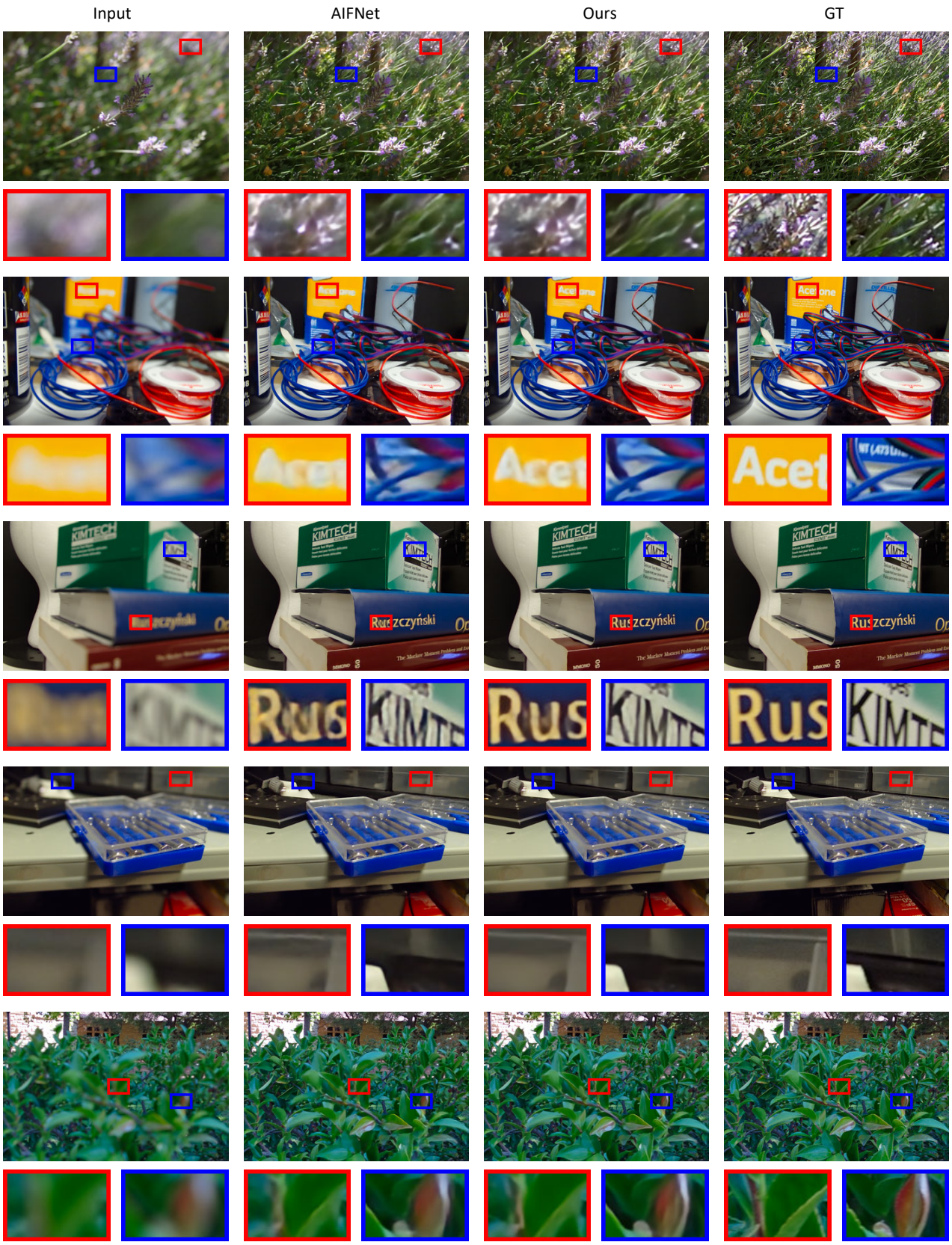

Fig. S23. **Qualitative comparison between IFAN [2] and ours evaluated on LFDOF dataset.** From left to right: defocused input, deblurred by IFAN and ours, and ground truth.