Supplementary Material for
Iterative Filter Adaptive Network for Single Image Defocus Deblurring

Junyong Lee Hyeongseok Son Jaesung Rim Sunghyun Cho Seungyong Lee

POSTECH

{junyonglee, sonh, jsrim123, s.cho, leesy}@postech.ac.kr

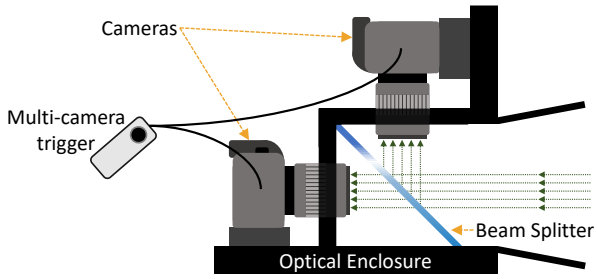
https://github.com/codeslake/IFAN

1. RealDOF Test Set

We present the *Real Depth of Field (RealDOF)* test set for quantitative and qualitative evaluations of single image defocus deblurring. Our RealDOF test set contains image pairs, each of which consists of a defocused image and its corresponding all-in-focus image that have been concurrently captured for the same scene.

To simultaneously capture such pairs of images, we built a dual-camera system (Fig. 1), where two cameras are vertically and horizontally aligned to each other on a vertical camera rig. The rig is designed to physically align the cameras as precisely as possible to ensure accurate alignment between the captured images. On the rig, we install a beam splitter between the cameras whose lenses are facing toward the beam splitter. The rig is encased with an optical enclosure that blocks light coming from outside of the viewing direction. In the system, we use the same models for the cameras (Sony A7R IV) and lenses (Sony 135mm F1.8). The cameras are connected with a multi-camera trigger for simultaneous capturing of images. We set the cameras with the minimum shutter speed of 1/125 seconds to avoid motion blur. ISO is adjusted accordingly for the images to pick up the same exposure.

Using the dual-camera system, we first capture defocused and all-in-focus images (namely, target image pairs). For a target image pair, one camera captures a defocused image with a wide aperture (F1.8 – 5.6), and the other camera obtains an all-in-focused image with a narrow aperture (F16). Images are captured in a 14-bit raw (Sony ARW) with the resolution of 9504×6344. The captured images are then processed to an sRGB using Adobe Lightroom and encoded with a lossless 16-bit TIFF format. Then, the encoded images are resized to 2376×1586, and geometric and photometric alignments are performed. As it is ambiguous to geometrically align blurry and sharp frames, we additionally capture all-in-focus image pairs (namely, reference image pairs) of the same scene, motivated by [4]. For the reference image pairs, we set both cameras with the same aperture (F16).



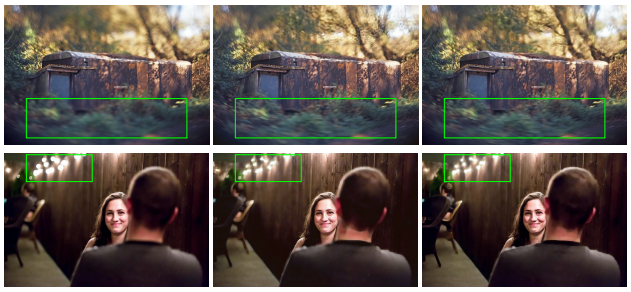
(a) Diagram for our dual-camera system.

(b) Our dual-camera system.

Figure 1. Our image acquisition system for the RealDOF test set.

Although the vertical rig used for our system is designed to accurately align the cameras, physical misalignment may still exist. Besides, the cameras may move slightly over time due to shakes. To handle such physical misalignments, we apply geometric alignments to captured images. We first compute a homography matrix from the reference image pair. As in [4], we use the enhanced correlation coefficients method [2], which is robust to photometric misalignment. Then, the matrix is applied to the corresponding target image pair, where the all-in-focused image is geometrically aligned to the defocused image using the matrix.

We use the same models for the two cameras and their lenses, but captured images still may exhibit exposure differences. We address this issue with photometric alignment based on a linear model as in [4]. Specifically, we compute linear photometric parameters from a target image pair, and then apply the parameters to the all-in-focus image to match its exposure to that of the defocused image. The final RealDOF test set contains geometrically and photometrically aligned target pairs of defocused and all-in-focused images.



(a) Input (b) DPDNet$_S$ (c) Ours

Figure 2. Failure cases. The input images are from the CUHK blur detection dataset [5]. From left to right: a defocused input image, deblurred results of DPDNet$_S$ [1] and our method.

2. Failure Cases

Our network works best with typical isotropic defocus blur, and may not properly handle blur with irregular shape (*e.g.* swirly bokeh as in the first row of Fig. 2) or strong highlight (*i.e.* glitter bokeh as in the second row).

3. Our Model with Different Input Types

16-bit images Our final model is trained on 8-bit images, as most standard encodings still rely on 8-bit images. Nonetheless, we also show the capability of our model in handling high bit-depth images, as the final model of DPDNet is targeted for 16-bit images. Table 1 shows a quantitative comparison on the DPDD dataset between our model and DPDNets that are trained and tested for 16-bit images. Our model outperforms DPDNets for all the deblurring metrics.

Dual-pixel images The deblurring performance of our model further increases if we explicitly feed dual-pixel images. Table 2 and Figs. 3, 4, and 5 show quantitative and qualitative comparisons of our model fed with dual-pixel images (Ours$_D$). Ours$_D$ has the same architecture of our final model taking a single image, except that the filter encoder in IFAN is fed with dual-pixel stereo images (I_B^l and I_B^r) that are concatenated along the channel dimension. In the figures, Ours$_{dual}$ shows better performance in handling spatially-varying (the red and green boxes at different focal planes in the figures) and large (green boxes) defocus blur.

4. Effect of Noise Augmentation Level

Table 3 shows the effect of the noise level used to augment training images. For training each model in the table, defocused images are randomly augmented with Gaussian noise, controlled by a random standard deviation within a range $[0, \sigma]$. We can infer from the table that compared to a model trained with a low noise level, a model trained with a higher noise level is better in restoring overall image contents (higher PSNR), but worse in recovering textures (higher LPIPS).

Model	Evaluations on the DPDD Dataset [1]			
	PSNR↑	SSIM↑	MAE$_{(\times 10^{-1})}$↓	LPIPS↓
DPDNet$_{D-16}$	25.13	0.786	0.406	0.224
DPDNet$_{S-16}$	24.41	0.751	0.434	0.278
Ours$_{S-16}$	**25.38**	**0.791**	**0.394**	**0.213**

Table 1. Quantitative comparison of 16-bit-based models.

Model	Evaluations on the DPDD Dataset [1]			
	PSNR↑	SSIM↑	MAE$_{(\times 10^{-1})}$↓	LPIPS↓
DPDNet$_D$	25.23	0.787	0.401	0.224
Ours$_D$	**25.99**	**0.804**	**0.373**	**0.207**

Table 2. Quantitative comparison of dual-pixel-based models.

σ	Evaluations on the DPDD Dataset [1]			
	PSNR↑	SSIM↑	MAE$_{(\times 10^{-1})}$↓	LPIPS↓
0.07	25.37	**0.789**	0.394	**0.217**
0.14	25.38	0.789	0.395	0.221
0.21	**25.39**	0.787	**0.394**	0.224

Table 3. Comparison between models trained with different noise augmentation levels. σ indicates the standard deviation of Gaussian noise used to augment defocused images in the training set.

5. Additional Qualitative Results

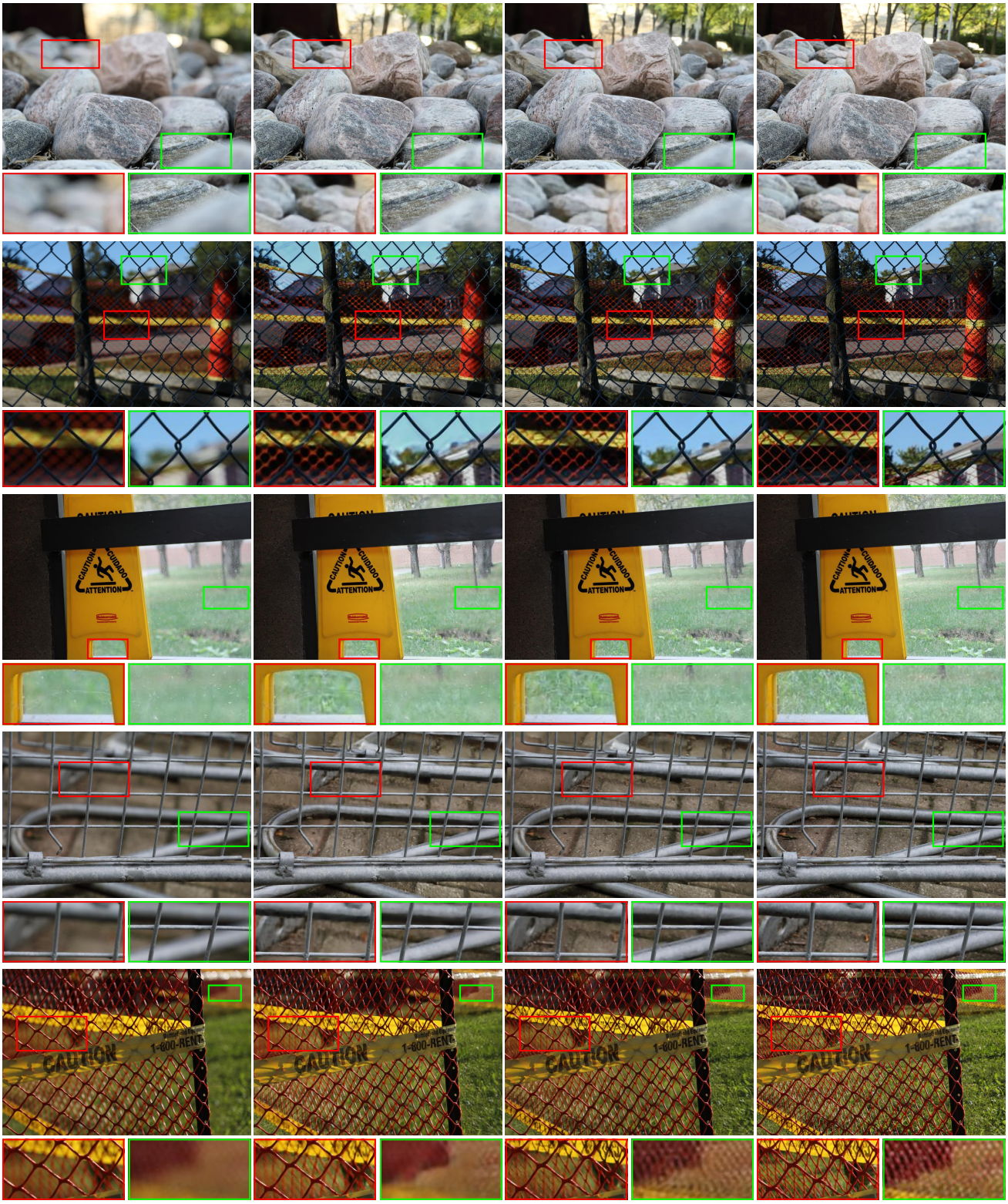
We show qualitative results on the DPDD test set [1] (Figs. 6, 7, and 8). For the qualitative results of real-world defocused images in different datasets, we show the results of the proposed RealDOF test set (Figs. 9 and 10), Pixel dual-pixel test set [1] (Figs. 11 and 12), and the CUHK dataset [5] (Figs. 13 and 14).

For the CUHK dataset, for readers to visually understand the capability of our method in handling spatially-varying and large blur, we visualize Gaussian point-spread-function (PSF) for each sampled coordinate of defocused images. Each PSF is generated using σ at the corresponding coordinates of a defocus map, which is computed by [3] and contains per-pixel standard deviations for a Gaussian kernel. Note that each PSF is displayed in a 31×31 grid and normalized for the visualization.

6. Network Architectures

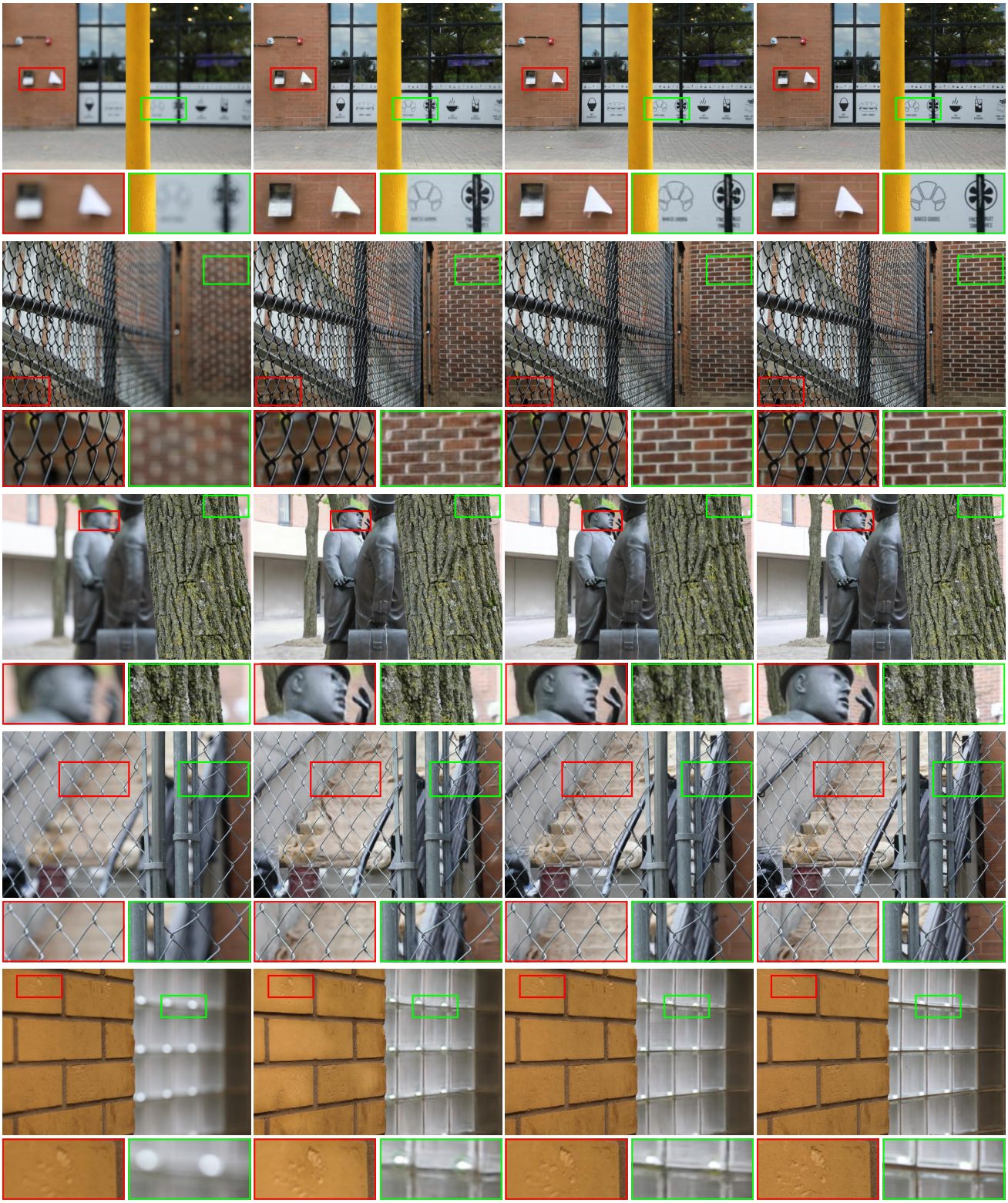
The detailed network architectures of the models used for the ablation study (Sec. 4.1. of the main paper) are shown in Tables 4, 5, 6, 7, and Fig. 15.

For the tables, `type`, `input`, `act`, `k`, `c`, `s`, `p` and `N` in columns denote the type of a layer, input for the layer, activation function, kernel size, out-channels, stride, padding, and repeating number, respectively. For leaky-ReLU (*lrelu*), we use 0.1 for its negative slope. For the layer types, we have *conv*, *dconv*, *identity*, *cat*, and *sum*, which denote convolution, deconvolution, identity, concatenation, and element-wise summation layers, respectively.



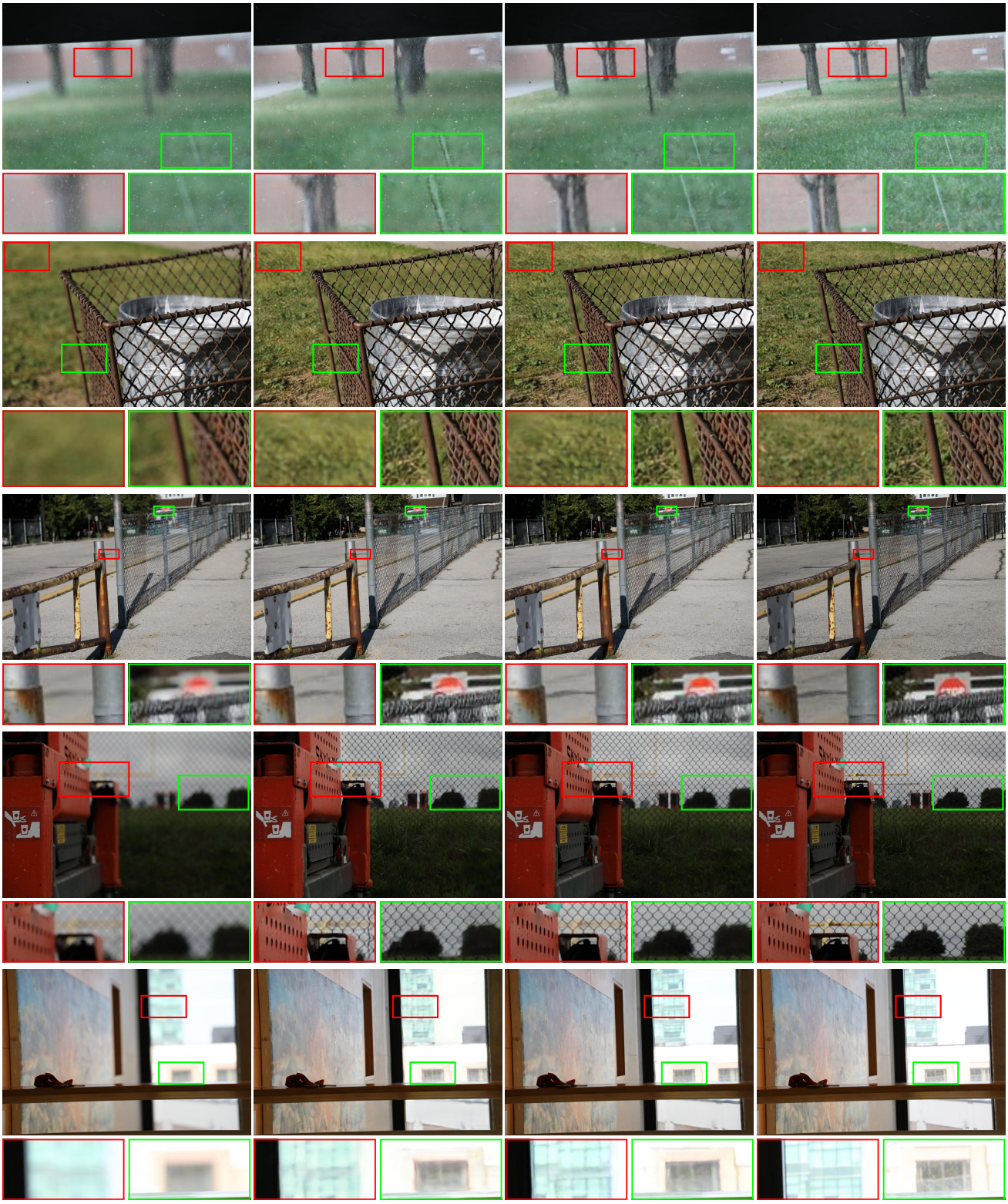
(a) Input (b) DPDNet$_D$ [1] (c) Ours$_D$ (d) GT

Figure 3. **Qualitative comparison of dual-pixel image-based models on the DPDD dataset** [1]. The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.



(a) Input (b) DPDNet$_D$ [1] (c) Ours$_D$ (d) GT

Figure 4. **Qualitative comparison of dual-pixel image-based models on the DPDD dataset** [1]. The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

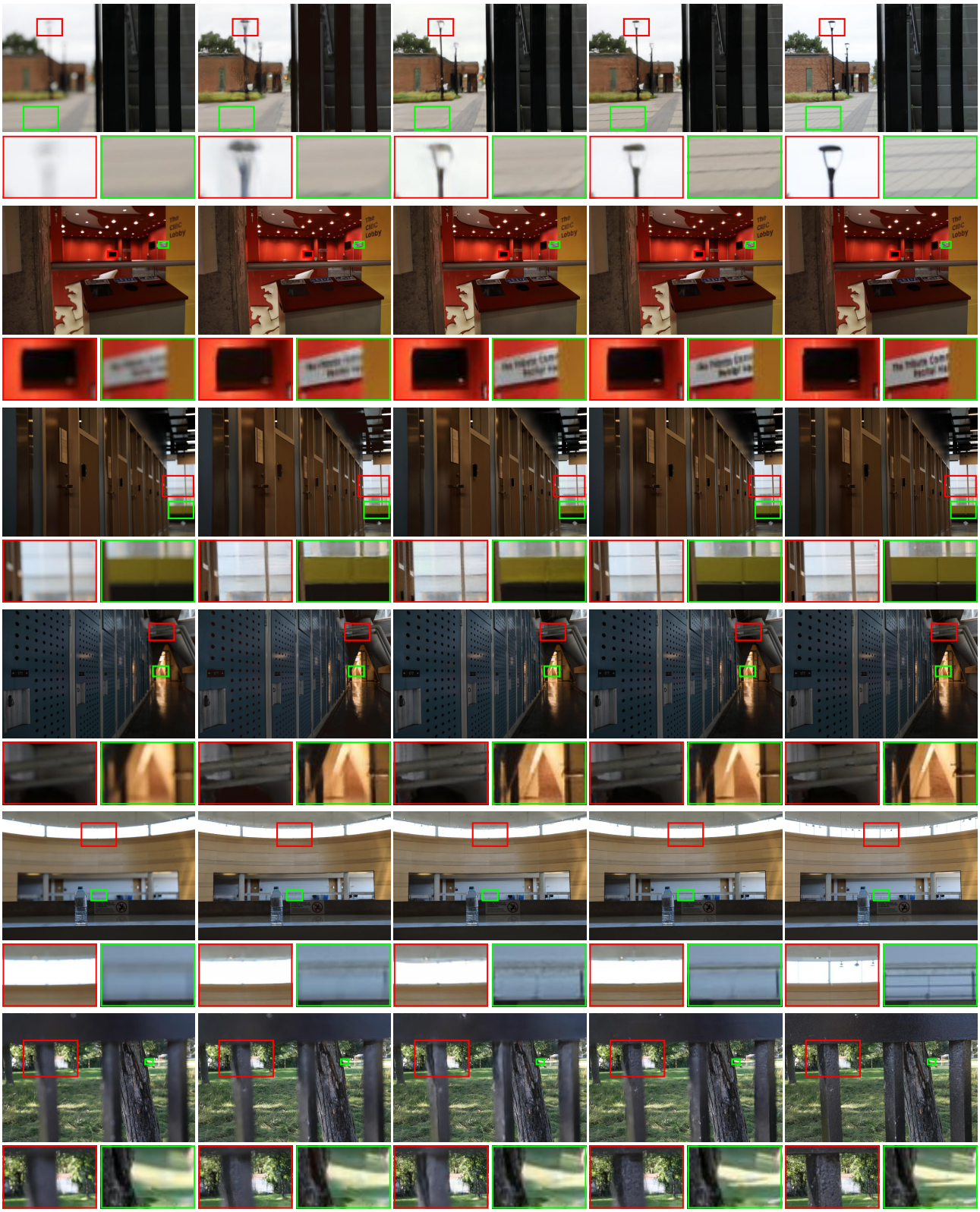


(a) Input (b) DPDNet$_D$ [1] (c) Ours$_D$ (d) GT

Figure 5. **Qualitative comparison of dual-pixel image-based models on the DPDD dataset** [1]. The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

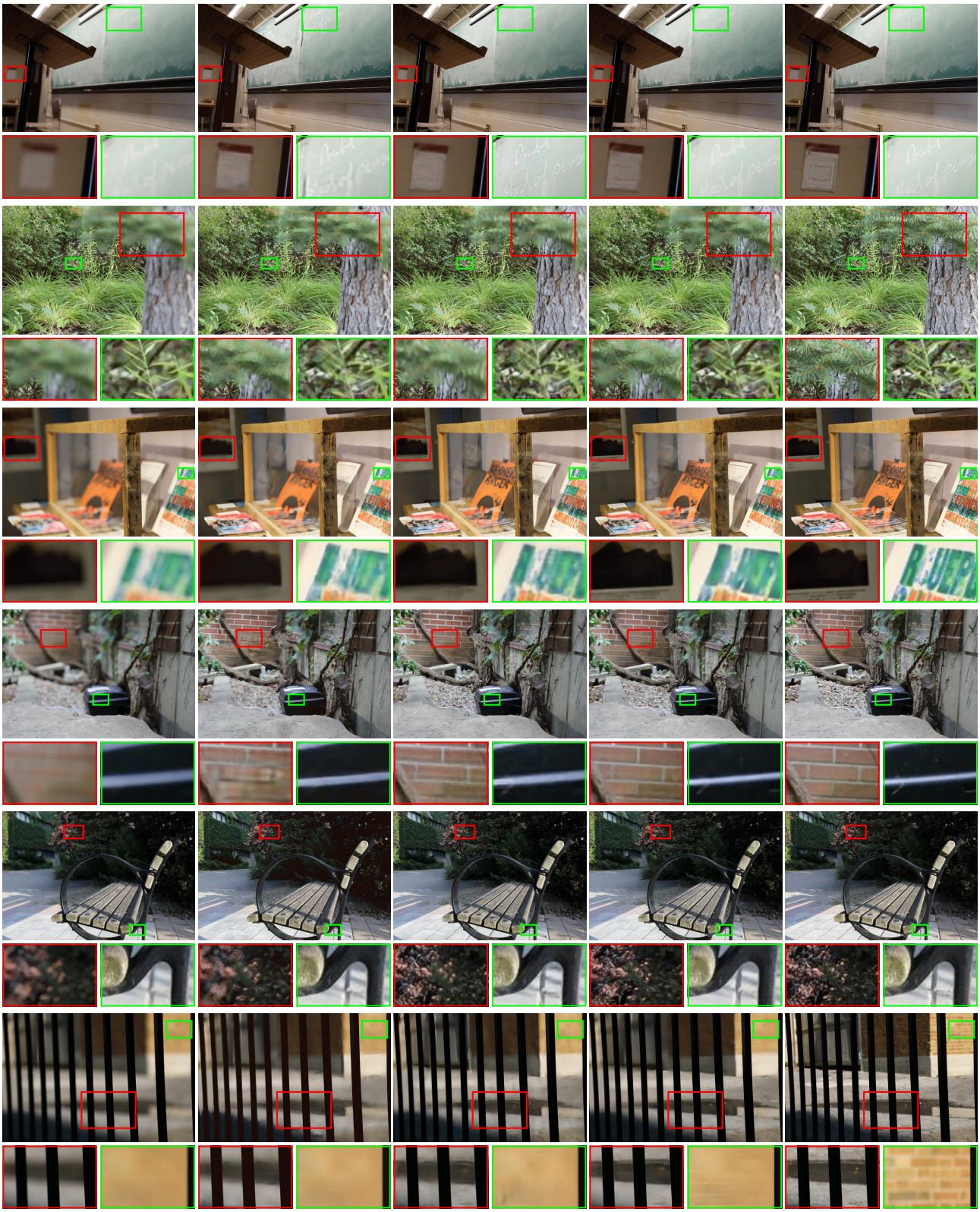
(a) Input (b) DPDNet$_S$ [1] (c) DPDNet$_D$ [1] (d) Ours (e) GT

Figure 6. **Additional qualitative results on the DPDD dataset [1].** The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.



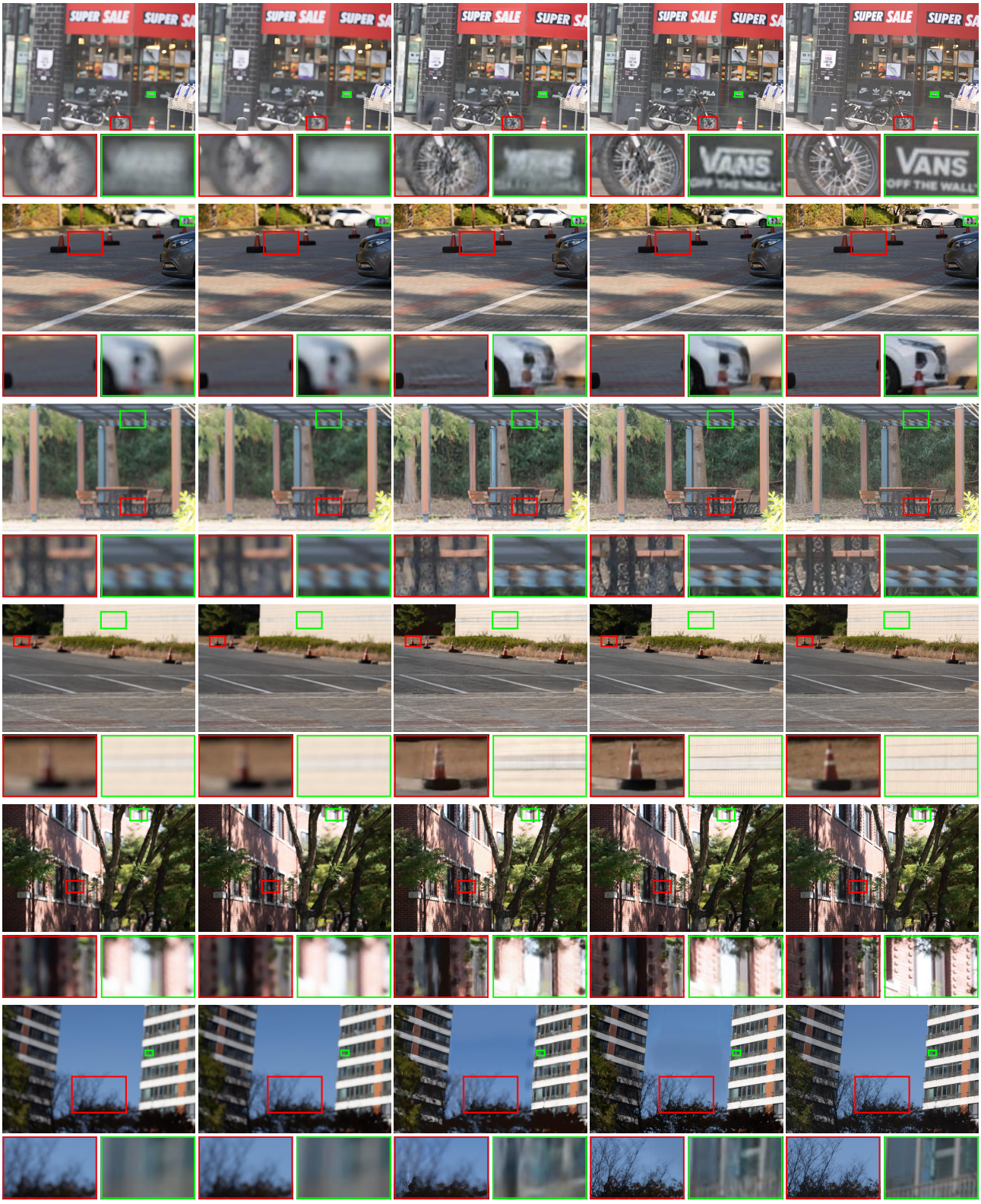
(a) Input (b) DPDNet$_S$ [1] (c) DPDNet$_D$ [1] (d) Ours (e) GT

Figure 7. **Additional qualitative results on the DPDD dataset [1].** The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.



(a) Input (b) DPDNet$_S$ [1] (c) DPDNet$_D$ [1] (d) Ours (e) GT

Figure 8. **Additional qualitative results on the DPDD dataset [1].** The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

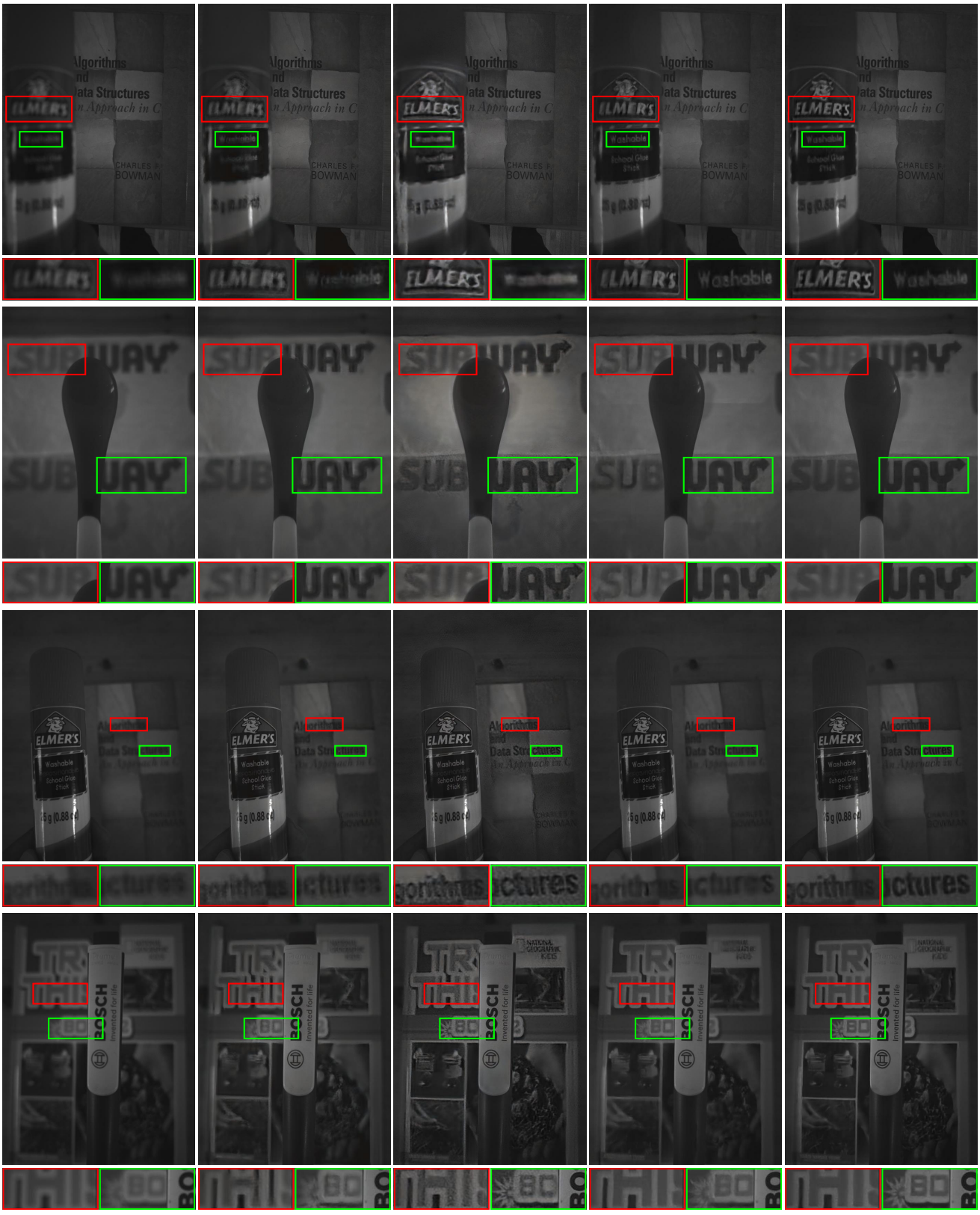


(a) Input (b) DMENet [3] (c) DPDNet$_S$ [1] (d) Ours (e) GT

Figure 9. **Additional qualitative comparison results on the proposed RealDOF test set.** The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

(a) Input (b) DMENet [3] (c) DPDNet$_S$ [1] (d) Ours (e) GT

Figure 10. **Additional qualitative comparison results on the proposed RealDOF test set.** The first and last columns show defocused input images and their ground-truth all-in-focus images, respectively. Between the columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

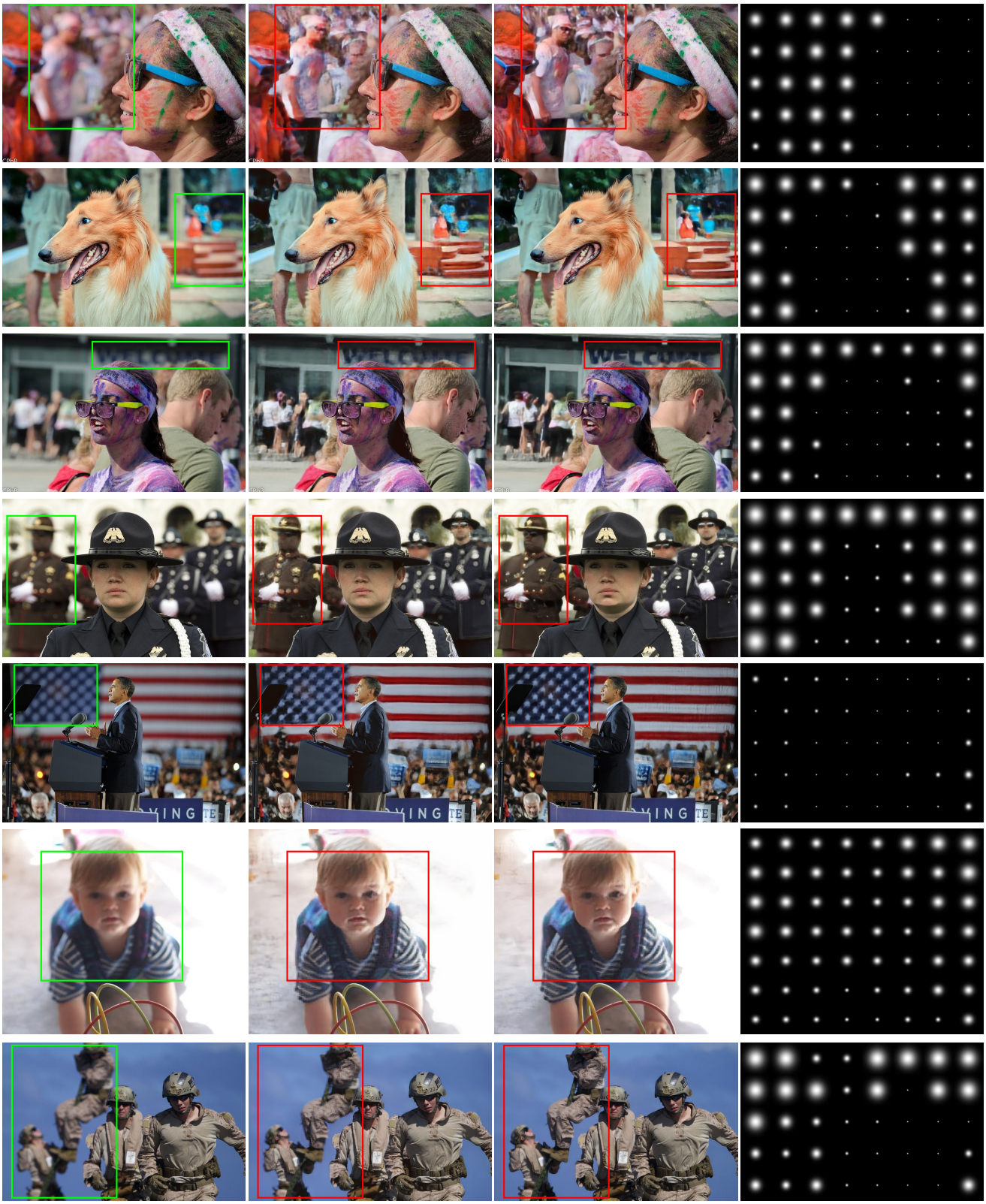


(a) Input (b) DPDNet$_S$ [1] (c) DPDNet$_D$ [1] (d) Ours (e) Ours$_D$

Figure 11. **Qualitative comparison on the Pixel Dual-Pixel dataset [1].** The first columns show defocused input images, and for the other columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.

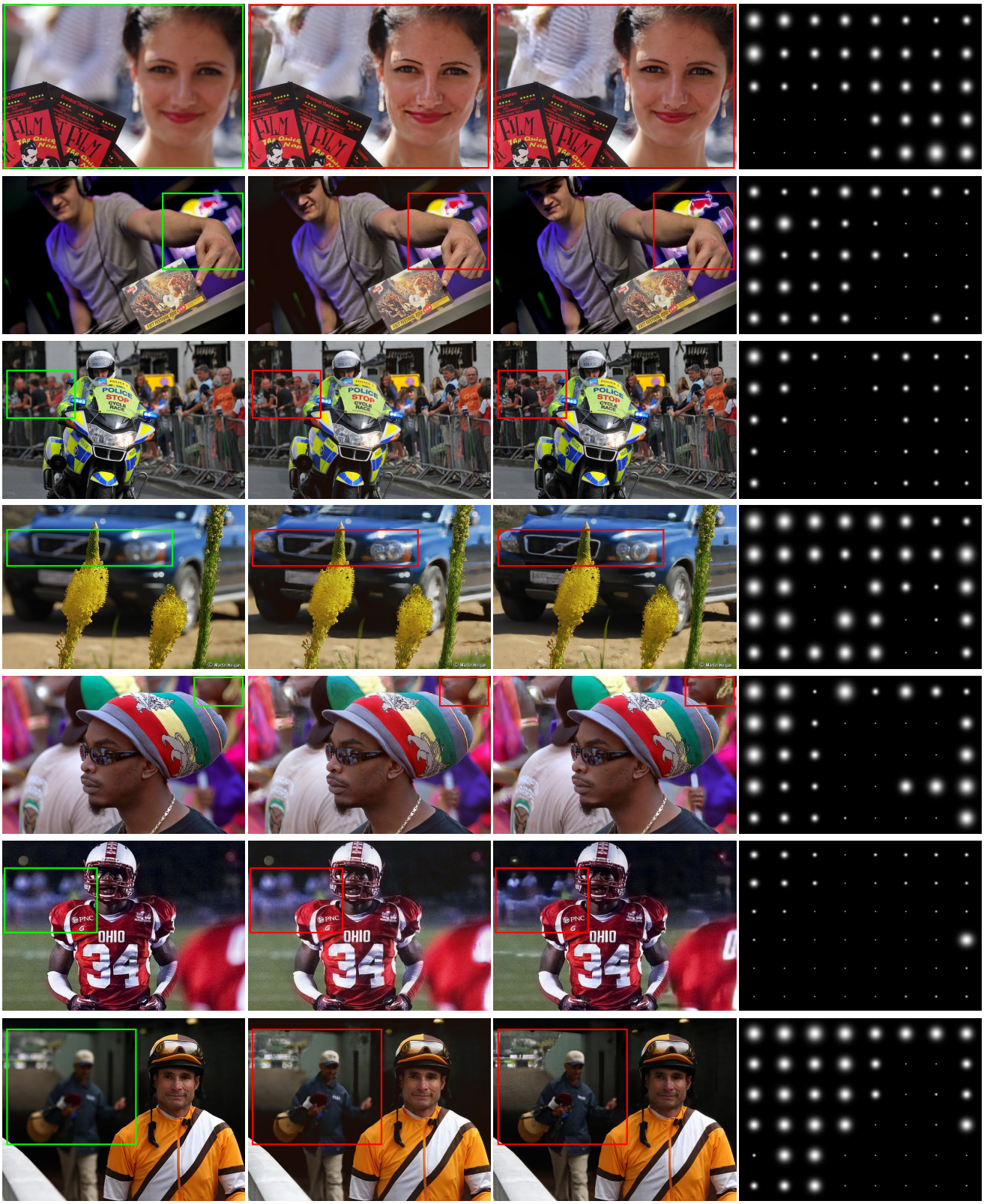
(a) Input (b) DPDNet$_S$ [1] (c) DPDNet$_D$ [1] (d) Ours (e) Ours$_D$

Figure 12. **Qualitative comparison on the Pixel Dual-Pixel dataset [1].** The first columns show defocused input images, and for the other columns, we show deblurring results of different methods. Images in the red and green boxes are zoomed-in cropped patches.



(a) Input (b) DPDNet$_S$ [1] (c) Ours (d) PSF visualization of (a)

Figure 13. **Qualitative comparison on the CUHK dataset [5].** The first and last columns show defocused input images and their per-pixel Gaussian PSFs visualized on sampled coordinates, respectively. Between the columns, we show deblurring results of different methods. Green and red boxes highlight the regions with large defocus blur and their deblurred results, respectively.



(a) Input (b) DPDNet$_S$ [1] (c) Ours (d) PSF visualization of (a)

Figure 14. **Qualitative comparison on the CUHK dataset [5].** The first and last columns show defocused input images and their per-pixel Gaussian PSFs visualized on sampled coordinates, respectively. Between the columns, we show deblurring results of different methods. Green and red boxes highlight the regions with large defocus blur and their deblurred results, respectively.

↓ For both training and testing

	type	input	act	k	c	s	p	N	output
Feature Encoder (F.E.)	*conv*	I_B	*lrelu*	3	32	1	1	3	conv_1
	conv	conv_1	*lrelu*	3	64	2	1	1	conv_2
	conv	conv_2	*lrelu*	3	64	1	1	2	conv_2
	conv	conv_2	*lrelu*	3	128	2	1	1	conv_3
	conv	conv_3	*lrelu*	3	128	1	1	2	conv_3
	conv	conv_3	*lrelu*	3	128	2	1	1	conv_4
	conv	conv_4	*lrelu*	3	128	1	1	2	e_B
IFAN	*F.E.*	I_B	-	-	-	-	-	-	$e_\mathbf{F}$
	conv	$e_\mathbf{F}$	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	*lrelu*	3	1	1	1	1	conv
	cat	$e_\mathbf{B}$	-	-	-	-	-	-	cat
	conv	conv	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	15	res
	conv	res	*lrelu*	3	128	1	1	1	e_{BS}
Reconstructor	*conv*	e_{BS}	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	15	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	*UP**	(res, conv_3)	-	-	128	-	-	-	up
	*UP**	(up, conv_2)	-	-	64	-	-	-	up
	*UP**	(up, conv_1)	-	-	32	-	-	-	up
	conv	up	*lrelu*	3	3	1	1	1	conv
	sum	I_B	-	-	-	-	-	-	I_{BS}

↓ Pre-defined blocks

RES*	*identity*	input	-	-	-	-	-	-	conv'
	conv	conv'	*lrelu*	3	c	1	1		conv
	conv	conv	-	3	c	1	1	N	conv
	sum	conv'	*lrelu*	-	-	-	-		conv'
	sum	input	-	-	-	-	-	-	res
UP*	*dconv*	input[0]	*lrelu*	4	c	2	1	1	dconv
	sum	input[1]	-	-	-	-	-	-	sum
	*RES**	sum	-	-	c	-	-	1	res
	*RES**	res	-	-	c	-	-	1	up

Table 4. Detailed network architecture of the baseline model used for the ablation study.

↓ For both training and testing

	type	input	act	k	c	s	p	N	output
Feature Encoder (F.E.)	*conv*	I_B	*lrelu*	3	32	1	1	3	conv_1
	conv	conv_1	*lrelu*	3	64	2	1	1	conv_2
	conv	conv_2	*lrelu*	3	64	1	1	2	conv_2
	conv	conv_2	*lrelu*	3	128	2	1	1	conv_3
	conv	conv_3	*lrelu*	3	128	1	1	2	conv_3
	conv	conv_3	*lrelu*	3	128	2	1	1	conv_4
	conv	conv_4	*lrelu*	3	128	1	1	2	e_B
IFAN	*F.E.*	I_B	-	-	-	-	-	-	$e_{\mathbf{F}}$
	conv	$e_{\mathbf{F}}$	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	-	3	1	1	1	1	$d^{r\to l}$
	conv	$d^{r\to l}$	*lrelu*	3	128	1	1	1	e_d
	cat	$e_{\mathbf{F}}$	-	-	-	-	-	-	cat
	conv	cat	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	4	res
	conv	res	*lrelu*	3	192	1	1	1	conv
	conv	res	*lrelu*	3	128	1	1	1	conv
	cat	$e_{\mathbf{B}}$	-	-	-	-	-	-	cat
	conv	conv	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	5	res
	*RES**	res	-	-	128	-	-	4	res
	conv	res	*lrelu*	3	128	1	1	1	e_{BS}
Reconstructor	*conv*	e_{BS}	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	3	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	*UP**	(res, conv_3)	-	-	128	-	-	-	up
	*UP**	(up, conv_2)	-	-	64	-	-	-	up
	*UP**	(up, conv_1)	-	-	32	-	-	-	up
	conv	up	*lrelu*	3	3	1	1	1	conv
	sum	I_B	-	-	-	-	-	-	I_{BS}

↓ Pre-defined blocks

RES*	*identity*	input	-	-	-	-	-	-	conv'
	conv	conv'	*lrelu*	3	c	1	1		conv
	conv	conv	-	3	c	1	1	N	conv
	sum	conv'	*lrelu*	-	-	-	-		conv'
	sum	input	-	-	-	-	-	-	res
UP*	*dconv*	input[0]	*lrelu*	4	c	2	1	1	dconv
	sum	input[1]	-	-	-	-	-	-	sum
	*RES**	sum	-	-	c	-	-	1	res
	*RES**	res	-	-	c	-	-	1	up

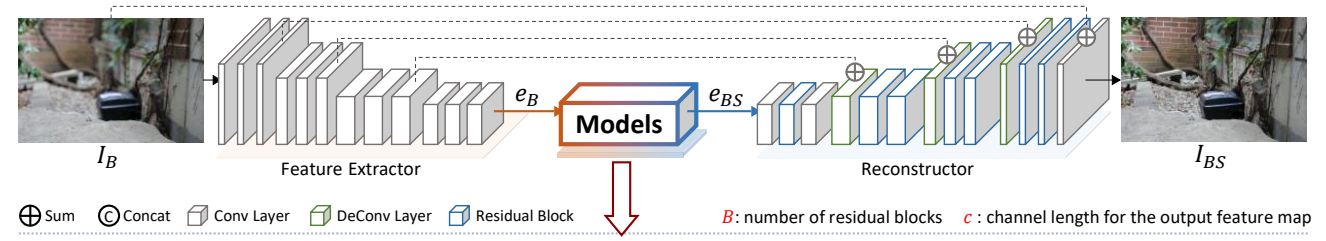
Table 5. Detailed network architecture of the baseline model embedded with the disparity map estimator used for the ablation study.

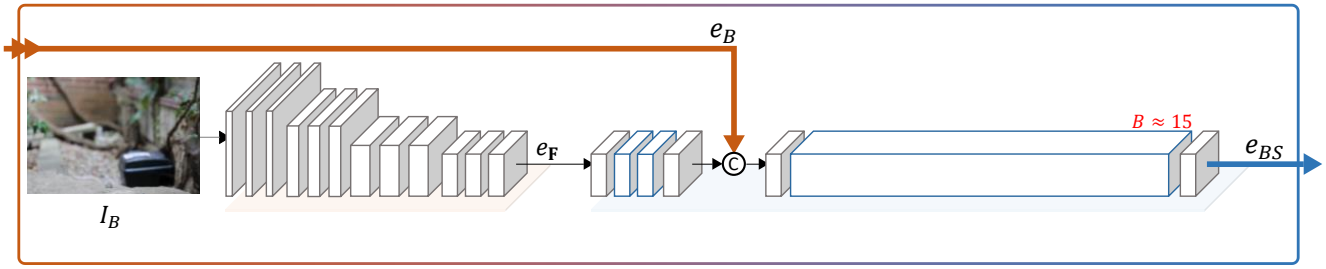
	type	input	act	k	c	s	p	N	output
									↓ For both training and testing
Feature Encoder (F.E.)	*conv*	I_B	*lrelu*	3	32	1	1	3	conv_1
	conv	conv_1	*lrelu*	3	64	2	1	1	conv_2
	conv	conv_2	*lrelu*	3	64	1	1	2	conv_2
	conv	conv_2	*lrelu*	3	128	2	1	1	conv_3
	conv	conv_3	*lrelu*	3	128	1	1	2	conv_3
	conv	conv_3	*lrelu*	3	128	2	1	1	conv_4
	conv	conv_4	*lrelu*	3	128	1	1	2	e_B
IFAN	*F.E.*	I_B	-	-	-	-	-	-	$e_{\mathbf{F}}$
	conv	$e_{\mathbf{F}}$	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	3	res
	conv	res	*lrelu*	3	129	1	1	1	conv
	conv	conv	*lrelu*	3	128	1	1	1	conv
	cat	$e_{\mathbf{F}}$	-	-	-	-	-	-	cat
	conv	cat	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	conv	conv	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	-	1	15,232	1	0	1	$\mathbf{F}_{deblur}$
	IAC	$(e_B, \mathbf{F}_{deblur})$	*lrelu*	3	128	1	1	17	e_{BS}
Reconstructor	*conv*	e_{BS}	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	3	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	*UP**	(res, conv_3)	-	-	128	-	-	-	up
	*UP**	(up, conv_2)	-	-	64	-	-	-	up
	*UP**	(up, conv_1)	-	-	32	-	-	-	up
	conv	up	*lrelu*	3	3	1	1	1	conv
	sum	I_B	-	-	-	-	-	-	I_{BS}
									↓ Training only
RBN	*conv*	$\mathbf{F}_{deblur}$	*lrelu*	3	32	1	1	1	conv
	*RES**	conv	-	-	32	-	-	2	res
	*RES**	res	-	-	32	-	-	2	res
	conv	res	-	3	357	1	1	1	$\mathbf{F}_{reblur}$
	IAC	$(I_S, \mathbf{F}_{reblur})$	-	3	3	1	1	17	$\hat{I}_{SB}$
	sum	I_S	-	-	-	-	-	-	I_{SB}
									↓ Pre-defined blocks
RES*	*identity*	input	-	-	-	-	-	-	conv'
	conv	conv'	*lrelu*	3	c	1	1		conv
	conv	conv	-	3	c	1	1	N	conv
	sum	conv'	*lrelu*	-	-	-	-		conv'
	sum	input	-	-	-	-	-	-	res
UP*	*dconv*	input[0]	*lrelu*	4	c	2	1	1	dconv
	sum	input[1]	-	-	-	-	-	-	sum
	*RES**	sum	-	-	c	-	-	1	res
	*RES**	res	-	-	c	-	-	1	up

Table 6. Detailed network architecture of the model with the filter predictor and IAC used for the ablation study.

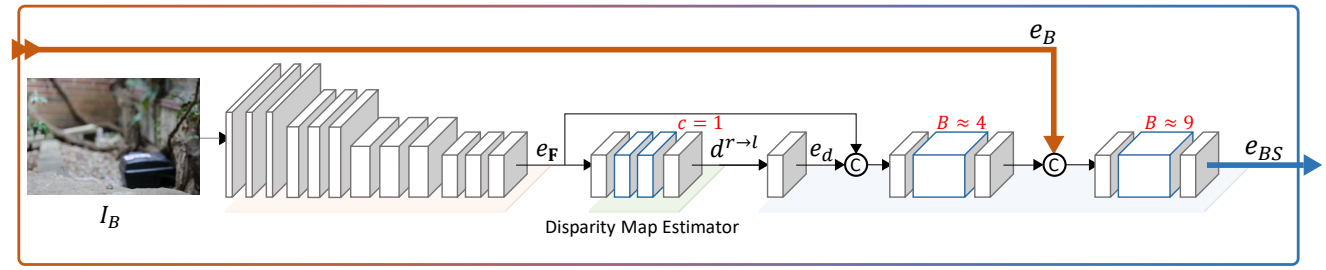
									↓ For both training and testing
Feature Encoder (F.E.)	type	input	act	k	c	s	p	N	output
	conv	I_B	*lrelu*	3	32	1	1	3	conv_1
	conv	conv_1	*lrelu*	3	64	2	1	1	conv_2
	conv	conv_2	*lrelu*	3	64	1	1	2	conv_2
	conv	conv_2	*lrelu*	3	128	2	1	1	conv_3
	conv	conv_3	*lrelu*	3	128	1	1	2	conv_3
	conv	conv_3	*lrelu*	3	128	2	1	1	conv_4
	conv	conv_4	*lrelu*	3	128	1	1	2	e_B
IFAN	*F.E.*	I_B	-	-	-	-	-	-	$e_{\mathbf{F}}$
	conv	$e_{\mathbf{F}}$	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	-	3	1	1	1	1	$d^{r\to l}$
	conv	$d^{r\to l}$	*lrelu*	3	128	1	1	1	e_d
	cat	$e_{\mathbf{F}}$	-	-	-	-	-	-	cat
	conv	cat	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	conv	conv	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	2	res
	*RES**	res	-	-	128	-	-	2	res
	conv	res	-	1	15,232	1	0	1	$\mathbf{F}_{deblur}$
	IAC	$(e_B, \mathbf{F}_{deblur})$	*lrelu*	3	128	1	1	17	e_{BS}
Reconstructor	*conv*	e_{BS}	*lrelu*	3	128	1	1	1	conv
	*RES**	conv	-	-	128	-	-	3	res
	conv	res	*lrelu*	3	128	1	1	1	conv
	*UP**	(res, conv_3)	-	-	128	-	-	-	up
	*UP**	(up, conv_2)	-	-	64	-	-	-	up
	*UP**	(up, conv_1)	-	-	32	-	-	-	up
	conv	up	*lrelu*	3	3	1	1	1	conv
	sum	I_B	-	-	-	-	-	-	I_{BS}
									↓ Training only
RBN	*conv*	$\mathbf{F}_{deblur}$	*lrelu*	3	32	1	1	1	conv
	*RES**	conv	-	-	32	-	-	2	res
	*RES**	res	-	-	32	-	-	2	res
	conv	res	-	3	357	1	1	1	$\mathbf{F}_{reblur}$
	IAC	$(I_S, \mathbf{F}_{reblur})$	-	3	3	1	1	17	$\hat{I}_{SB}$
	sum	I_S	-	-	-	-	-	-	I_{SB}
									↓ Pre-defined blocks
RES*	*identity*	input	-	-	-	-	-	-	conv′
	conv	conv′	*lrelu*	3	c	1	1		conv
	conv	conv	-	3	c	1	1	N	conv
	sum	conv′	*lrelu*	-	-	-	-		conv′
	sum	input	-	-	-	-	-	-	res
UP*	*dconv*	input[0]	*lrelu*	4	c	2	1	1	dconv
	sum	input[1]	-	-	-	-	-	-	sum
	*RES**	sum	-	-	c	-	-	1	res
	*RES**	res	-	-	c	-	-	1	up

Table 7. Detailed network architecture of our final model.

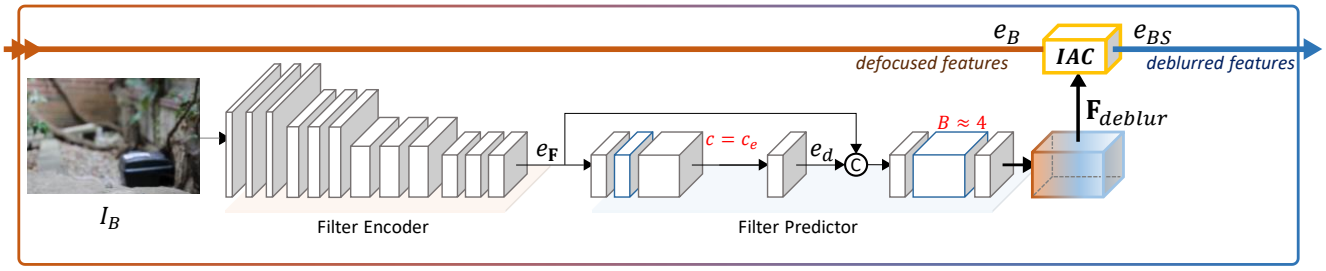




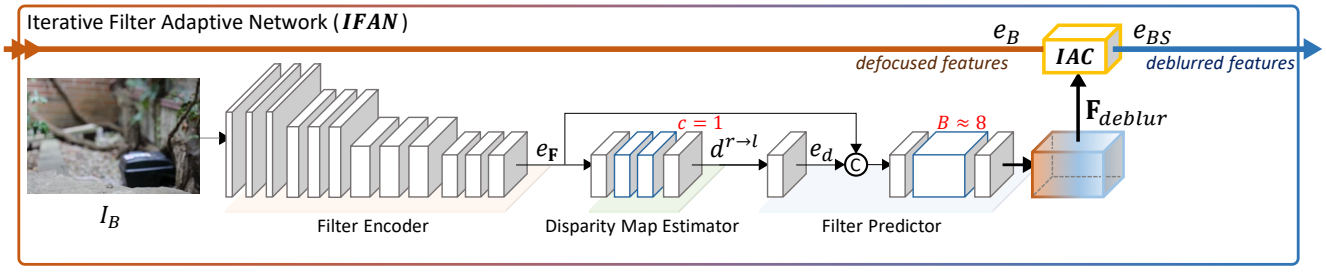
(a) the baseline model (Table 4)



(b) the baseline model embedded with the disparity map estimator (Table 5)



(c) the model with filter predictor and IAC (Table 6)



(e) our final model with filter predictor, IAC, and disparity map estimator (Table 7)

Figure 15. **Network architectures used for the ablation study (Table 1 and Fig. 5 in the main paper).** From top to bottom: architectures for (a) the baseline model, (b) baseline model embedded with the disparity map estimator, (c) proposed IFAN without the disparity map estimator, and (d) proposed IFAN.

References

[1] Abdullah Abuolaim and Michael Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

[2] Georgios Evangelidis and Emmanouil Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE TPAMI*, 30(10):1858–1865, 2008. 1

[3] Junyong Lee, Sungkill Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019. 2, 9, 10

[4] Jaesung Rim, Lee Haeyun, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 1

[5] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *CVPR*, 2014. 2, 13, 14