Appendix: Multiscale Structure Guided Diffusion for Image Deblurring

Mengwei Ren[†‡*] Mauricio Delbracio[‡] Hossein Talebi[‡] Guido Gerig[†] Peyman Milanfar[‡]
[†]New York University [‡]Google Research

1. Additional Results

1.1. Effectiveness of the guidance on GoPro, HIDE and REDS

We include additional perception-distortion plots for GoPro [14], HIDE [17] and REDS [13] datasets in Fig. 1, as supplementary for Section 4.3 of the main paper, to verify the effectiveness of the proposed guidance.

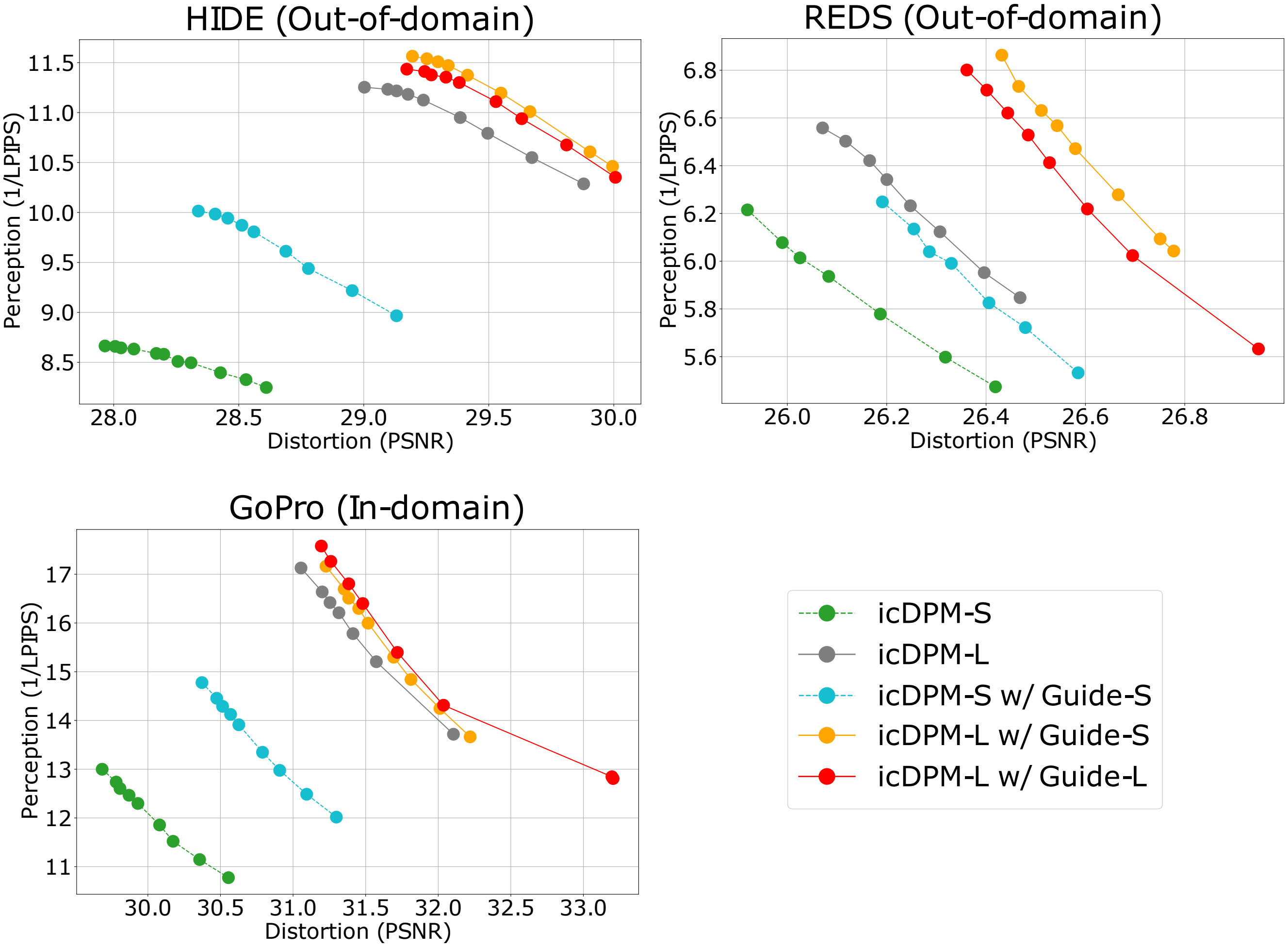


Figure 1. Additional perception-distortion plots as supplementary for Sec. 4.3 in the main paper. All models are trained only on GoPro [14]. The guidance mechanism allows for consistent better perceptual qualities and lower distortions compared to image-conditioned diffusion probablistic model (icDPM) under different network capacities ('icDPM-S w/ Guide-S' > 'icDPM-S'; 'icDPM-L w/ Guide-S/L' > 'icDPM-L'), both in-domain (GoPro) and out-of-domain (HIDE, REDS) '-S' and '-L' refer to small and large networks respectively.

1.2. Additional visual results

To supplement main paper Fig.6,7,8, we provide additional and enlarged qualitative results for all datasets below.
Realblur-J (out-of-domain) deblurring examples are shown in Fig. 16, 17, 18, 19.
REDS (out-of-domain) deblurring examples are shown in Fig. 4, 5, 6, 7.
HIDE (out-of-domain) deblurring examples are shown in Fig. 8, 9, 10, 11.
GoPro (in-domain) deblurring examples are shown in Fig. 12, 13, 14, 15.

1.3. Failure cases

As discussed in the main paper, we acknowledge that the domain generalization of the model is still extensively bounded by the quality of the training set. In our experiments, we only train with GoPro [14] for the sake of benchmarking. However, the data diversity and representativity from GoPro is limited, i.e., it only contains daytime scenes, acquired outdoor under sufficient lighting conditions. Moreover, the synthesis of blur in GoPro by simple averaging of consecutive frames is less realistic [26]. Lastly, the ground truth images in GoPro dataset are rather low-quality, which may further hurt the out-of-domain performance. Therefore, it is expected that it will be extremely hard for the model to perform decent deblurring on scenes significantly different from GoPro, such as low-light images with saturated regions, in Realblur-J [15].

We include a few failure cases on such scenes in Fig. 20, 21, 22 23, where **all** methods fail to remove blur from the night scenes, especially with night streaks. We believe that in practice, more realistic training datasets [5] will further increase the model generalization.

2. Additional Ablation

Input concatenation During prototyping, we also explored the possibility of removing input-level concatenation, and only rely on the intermediate representations from regression as the condition of the diffusion model, similar as in [7] for super-resolution. Potentially, we expect such setting will further make the model domain-generalizable as it does not directly interact with images from different domains, although it may also risk losing detailed information from the input.

As proof of concept, we use the same multiscale regression networks, and compare the models with or without input concatenation. Further, since the diffusion model now only takes the intermediate representations as input, we reintroduce the RGB information by using our model variants (d) in Table 8. in the main paper (i.e., regression targets are downsampled RGB images instead of grayscale images). From Table 1, we observe that the input concatenation obtained a much better performance both in-domain and out-of-domain than without concatenation. Therefore, in our final model, we keep the input concatenation and only rely on the guidance features to provide additional information.

Table 1. Effects of input-level concatenation. From our model variant (d) in Table 8. of the main paper, we remove the input concatenation, loosely inspired by [7] (super-resolution). In the context of deblurring, we observe deteriorate results indicated in row 'w/o input concatenation', compared to the setting with additional input concatenation.

	In-domain		Out-of-domain	
	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
w/o input concatenation	25.20	0.230	28.29	0.177
w/ concatenation	30.65	0.090	28.45	0.141

Further cross-domain alignment. We also explored the potential effects of finetuning the DPMs with adversarial formulation where we used additional discriminators on the guidance features between different datasets (e.g., GoPro and Realblur-J) so that the features extracted from different domains become indistinguishable, similar to the feature alignment strategy in [20]. However, we do not observe extra benefits, and find that such finetuning may even hurt the performance as shown in Fig. 2. We speculate that it could be a result of training instability of GANs, or perhaps the suboptimal formulation under the image-conditioned DPM framework. We will leave this for future investigation.

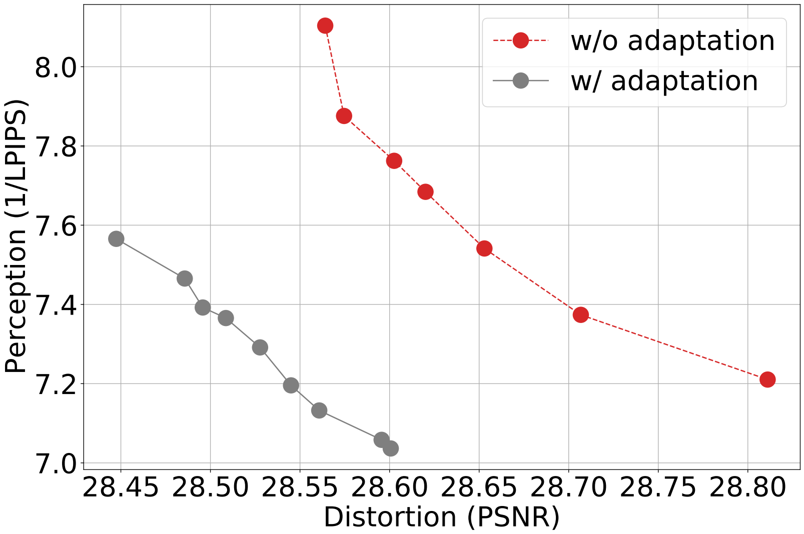
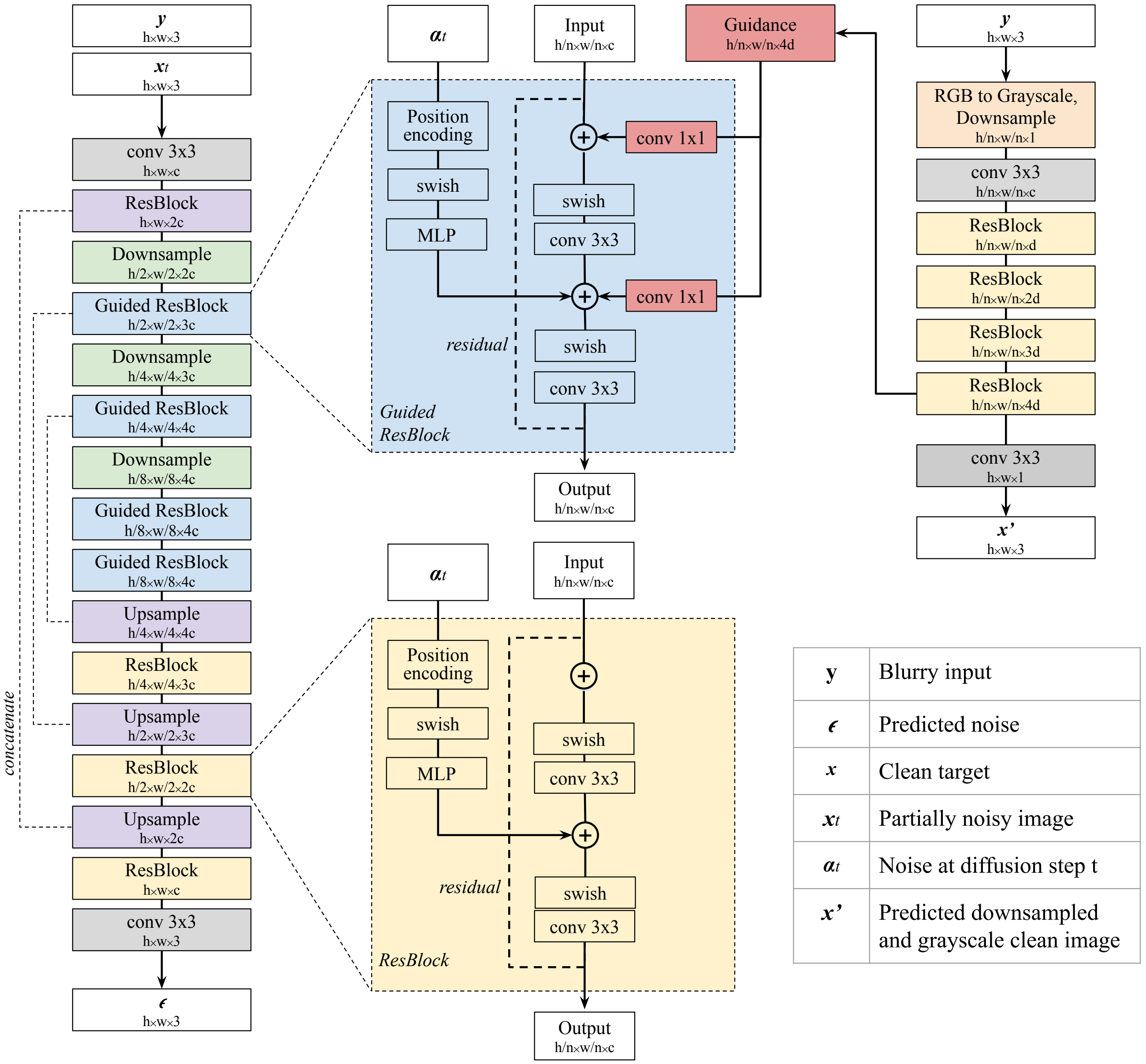


Figure 2. A comparison between our models with or without further domain adaptation with Realblur-J, on a GoPro trained model. Surprisingly, further adversarial domain adaptation on the guidance features between GoPro and Realblur-J hurt the performance.

3. Additional implementation details

3.1. Architectures

The architectural details for the diffusion network and the guidance network are illustrated in Fig. 3.



$\mathbf{y}$	Blurry input
ϵ	Predicted noise
x	Clean target
x_t	Partially noisy image
α_t	Noise at diffusion step t
x'	Predicted downsampled and grayscale clean image

Figure 3. The detailed architecture of the proposed method. **Left**: the image-conditioned diffusion network based on a fully-convolutional UNet similar to [22], where we replace the residual blocks from the UNet encoder with the proposed guided residual block. **Middle** column illustrates the difference between a standard residual block and the proposed guided block, where we additionally incorporate multiscale structure guidance. **Right**: The proposed guidance network for extracting the coarse structure features from the input at multiple resolutions. At each scale, the blurry image is first converted to grayscale, downsampled, and lastly fed into the network to predict its clean counterpart. The output from the last residual block is leveraged as the guidance feature.

3.2. Inference

As we use continuous noise level sampling during training, it enables the use of different noise schedulers during the inference to potentially obtain samples with different distortion-perception trade-off. We therefore perform a grid search over a set of different diffusion steps T, as well as the upper bound of the noise variance $1 - \alpha_T$. For efficiency, we also exclude certain combinations that do not produce reasonable sampling (i.e., sampling results are pure noise or blank image), and the final combinations are indicated in Table 2.

Table 2. The sampling parameters for inference.

		Maximum noise variance $1 - \alpha_T$					
		0.01	0.02	0.05	0.1	0.2	0.5
Steps (T)	20						✓
	30						✓
	50					✓	✓
	100				✓	✓	✓
	200			✓	✓	✓	✓
	500		✓	✓	✓	✓	
	1000	✓	✓	✓	✓		

3.3. Computational cost

In Table 3, we report floating point operations per second (FLOPs) under different model configurations, calculated based on an input image of $720 \times 1280 \times 3$. For diffusion networks (c)-(d), the FLOPS are calculated based on a single diffusion step. While optimizing sampling speed is out-of-scope of this work, we believe recent advance in speeding up DPM sampling [18, 3, 23, 9, 10, 11, 4, 12, 8] could be further incorporated into our framework.

Table 3. FLOPs under different model configurations, calculated based on a full-size input image of $720 \times 1280 \times 3$. For diffusion networks (c)-(d), the FLOPs are calculated based on a single diffusion step.

	Guidance network	Diffusion network	# Params	FLOPs
(a) icDPM-S	-	ch=32	6M	1200B
(b) icDPM-L	-	ch=64	27M	4800B
(c) icDPM-S w/ Guide-S	ch=32	ch=32	10M	2500B
(d) icDPM-L w/ Guide-S	ch=32	ch=64	30M	6100B
(e) icDPM-L w/ Guide-L	ch=64	ch=64	52M	10000B

3.4. Benchmark results

We performed a consistent computation over all benchmarks for fair comparisons. To acquire the benchmark results, we use the author provided results whenever possible. On the cross-domain set up of Realblur-J with GoPro trained only models, we use author provided results of DvSR [22], UFormer [21], Restormer [24]. For DeblurGAN-v2 [6] and MPRNet [25], we use official code repository along with the provided GoPro checkpoints for inference. On REDS [13], all results are obtained by running their official models with the GoPro checkpoints.

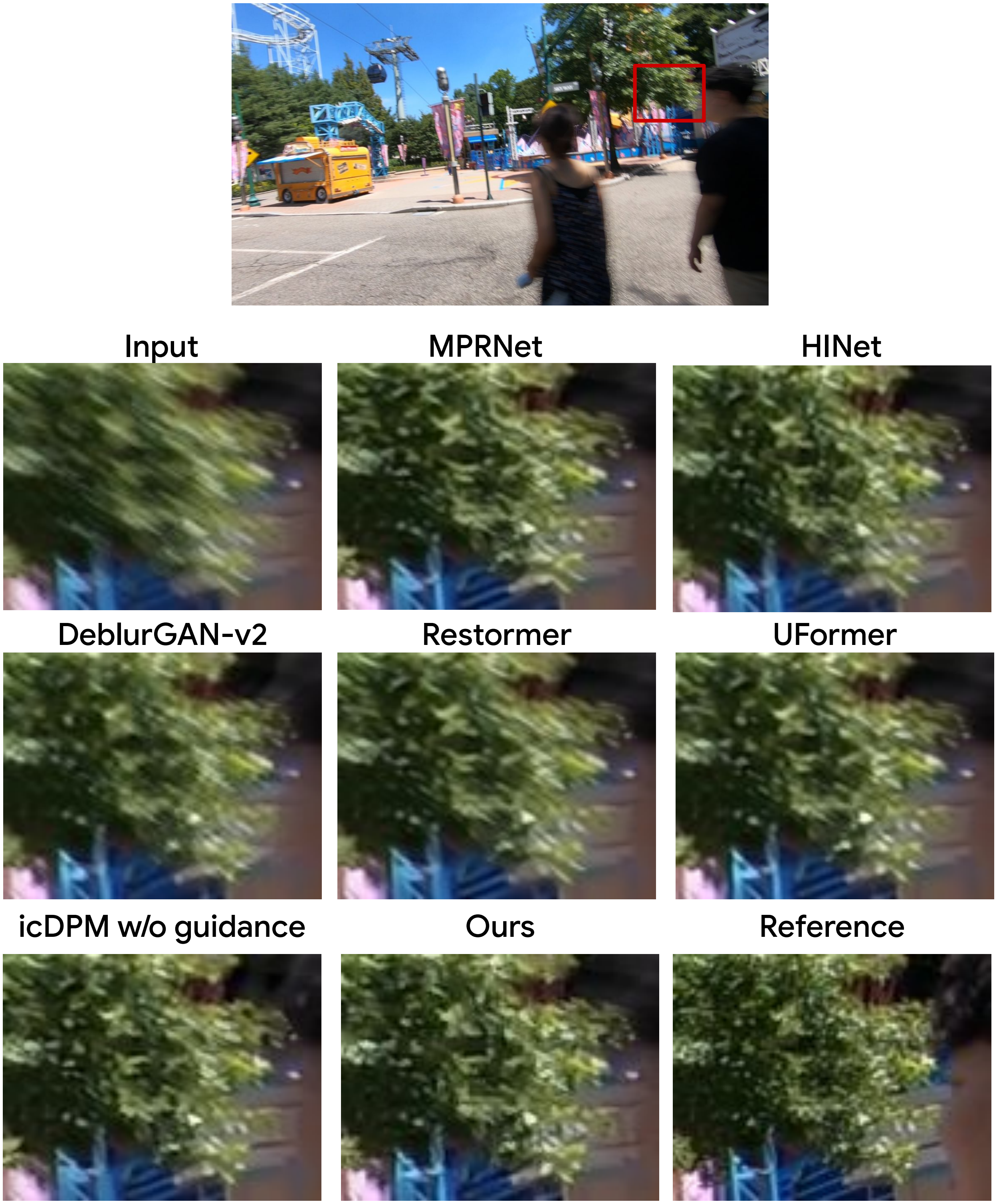


Figure 4. **REDS** [13] deblurring examples from MPRNet [25], HINet [1], DeblurGAN-v2 [6], Restormer [24], UFormer [21], icDPM without guidance and Ours (icDPM with guidance).

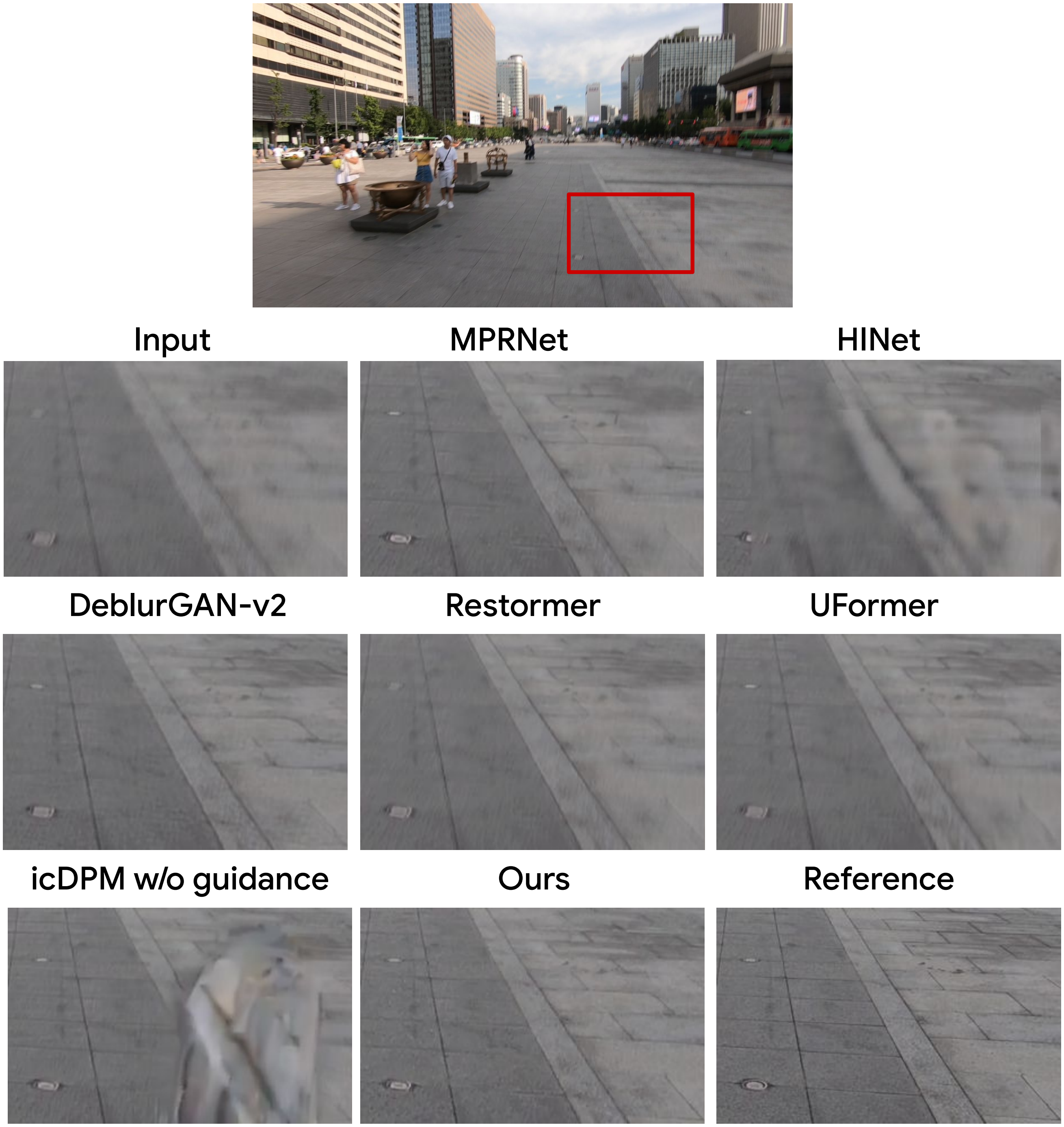


Figure 5. **REDS** [13] deblurring examples from MPRNet [25], HINet [1], DeblurGAN-v2 [6], Restormer [24], UFormer [21], icDPM without guidance and Ours (icDPM with guidance).

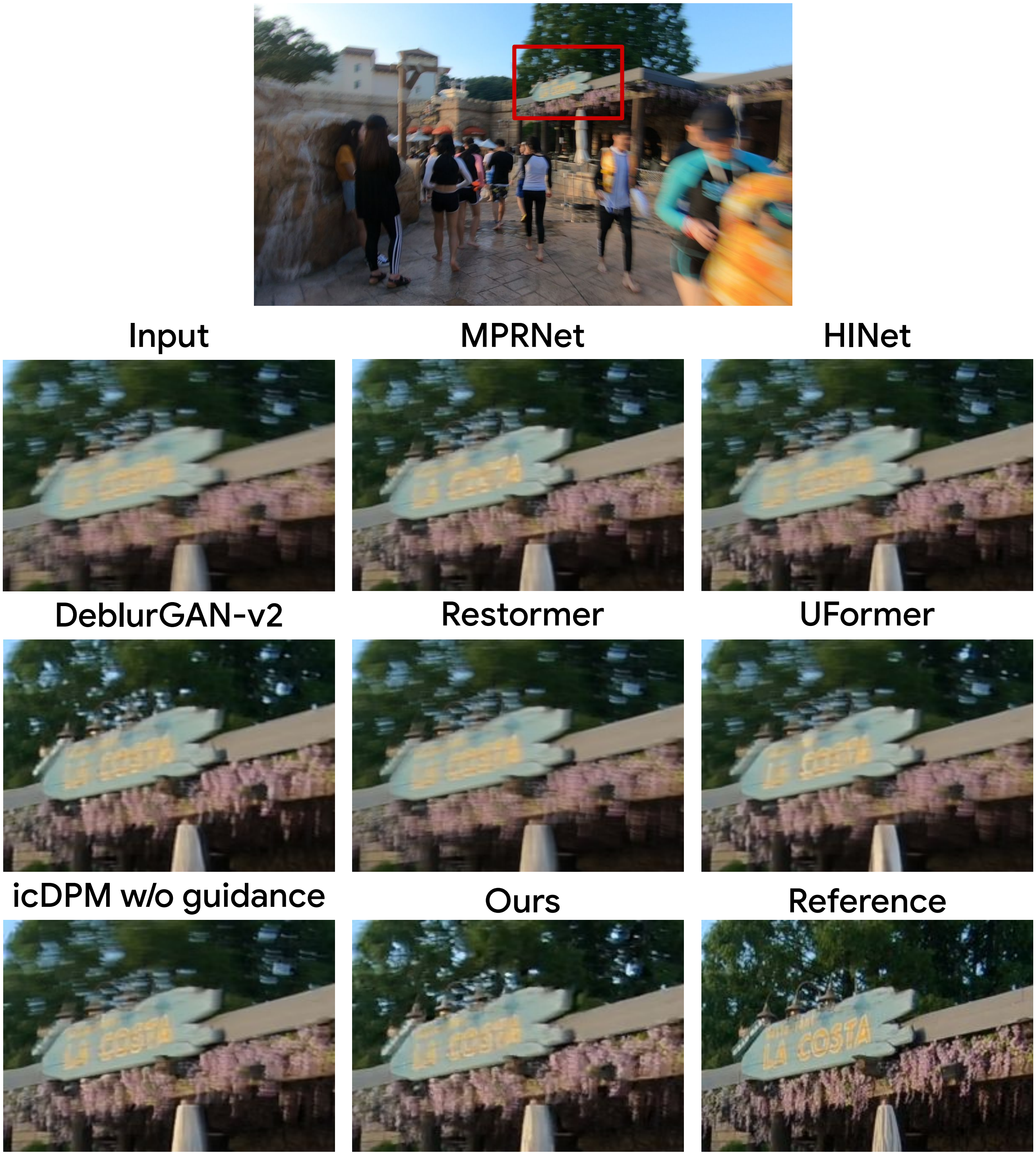


Figure 6. **REDS** [13] deblurring examples from MPRNet [25], HINet [1], DeblurGAN-v2 [6], Restormer [24], UFormer [21], icDPM without guidance and Ours (icDPM with guidance).

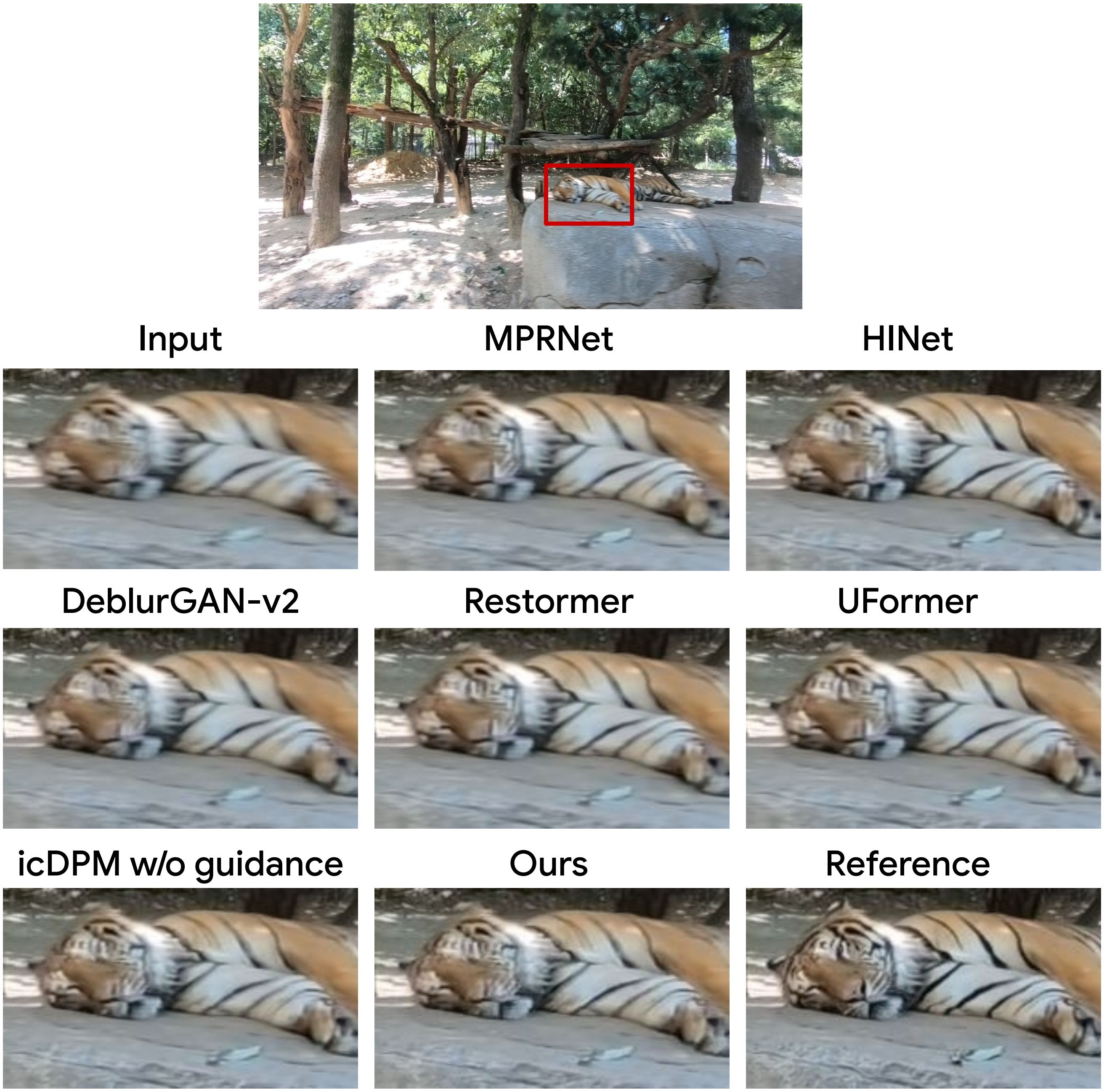


Figure 7. **REDS** [13] deblurring examples from MPRNet [25], HINet [1], DeblurGAN-v2 [6], Restormer [24], UFormer [21], icDPM without guidance and Ours (icDPM with guidance).

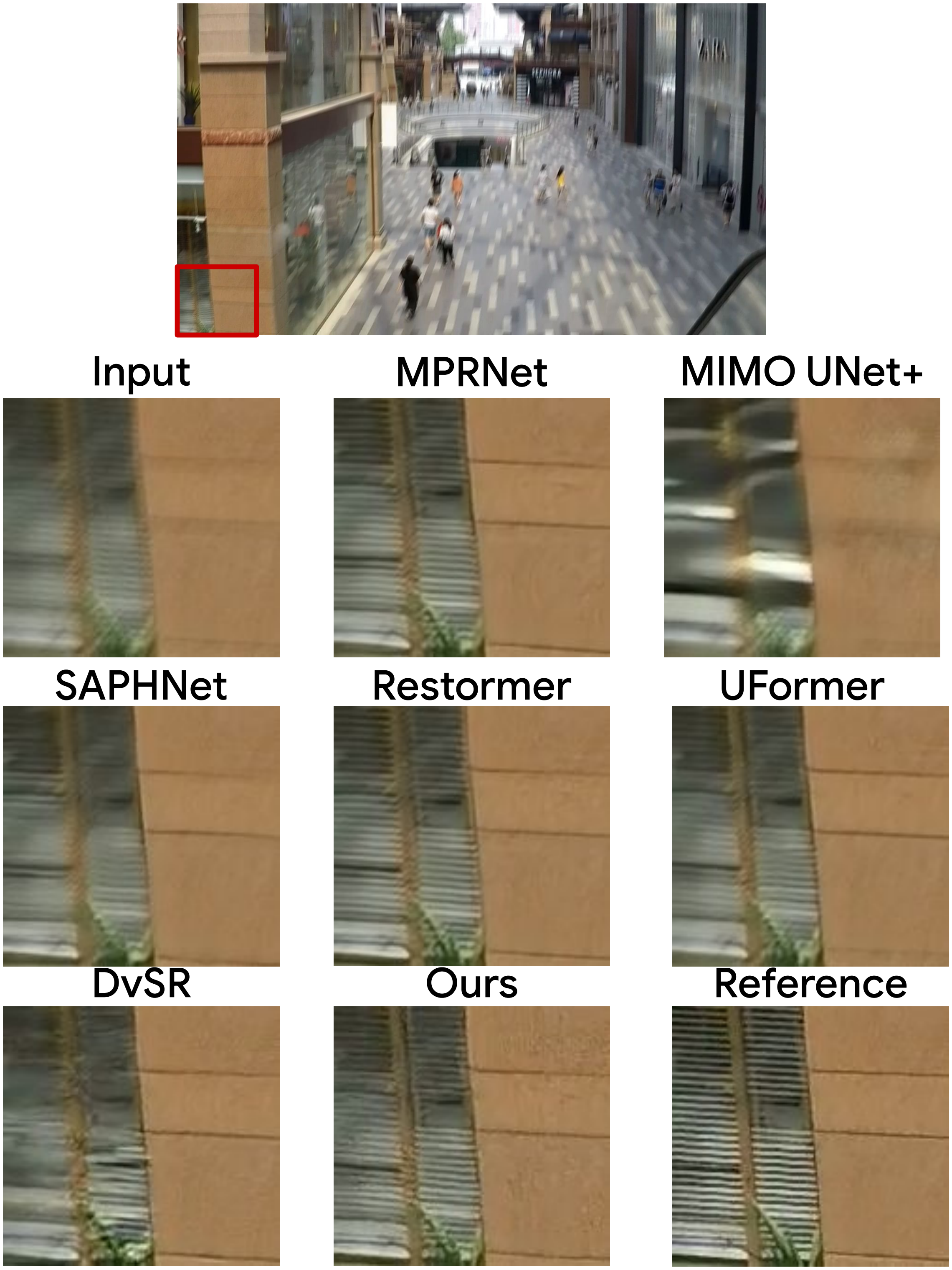


Figure 8. **HIDE** [17] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

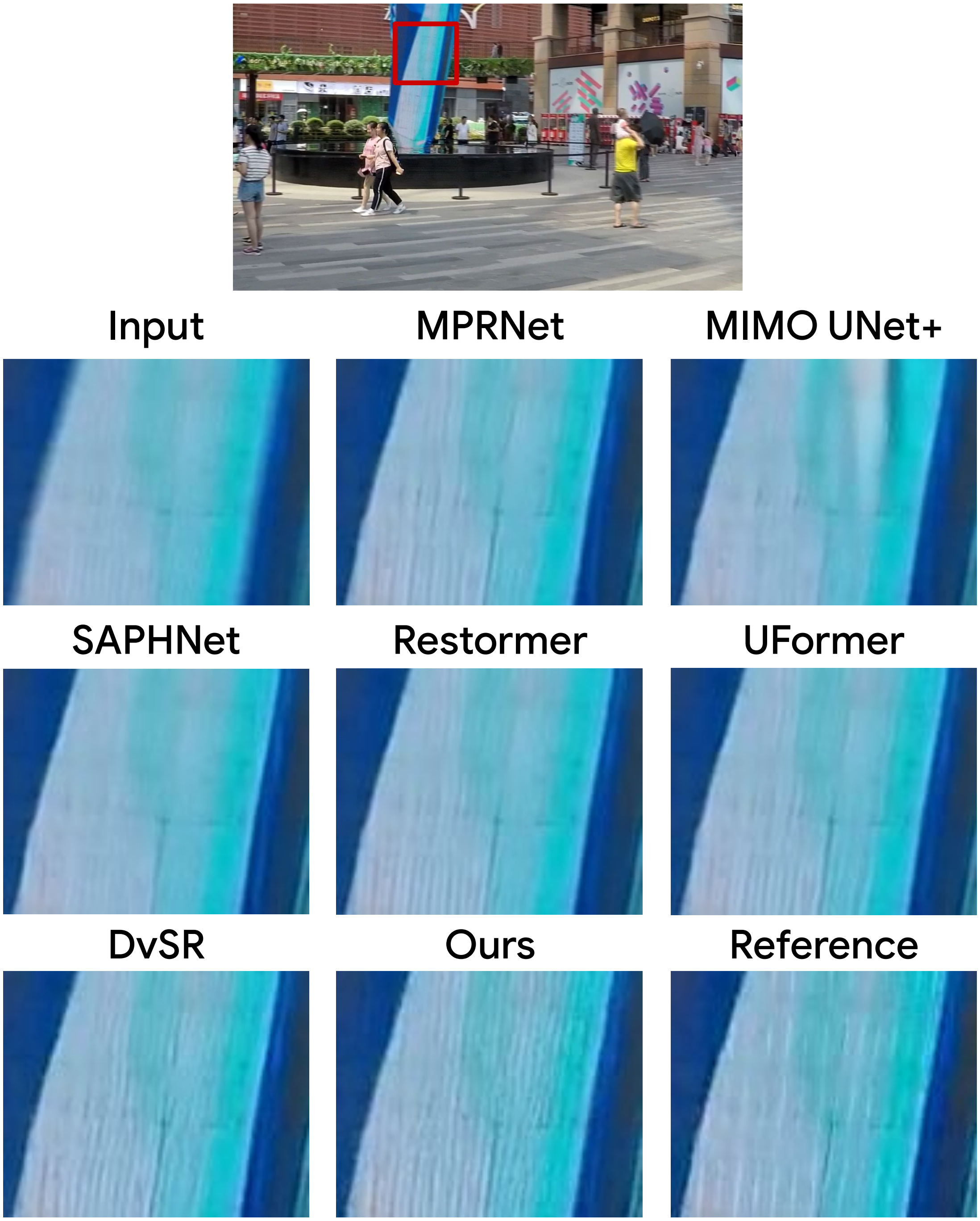


Figure 9. **HIDE** [17] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

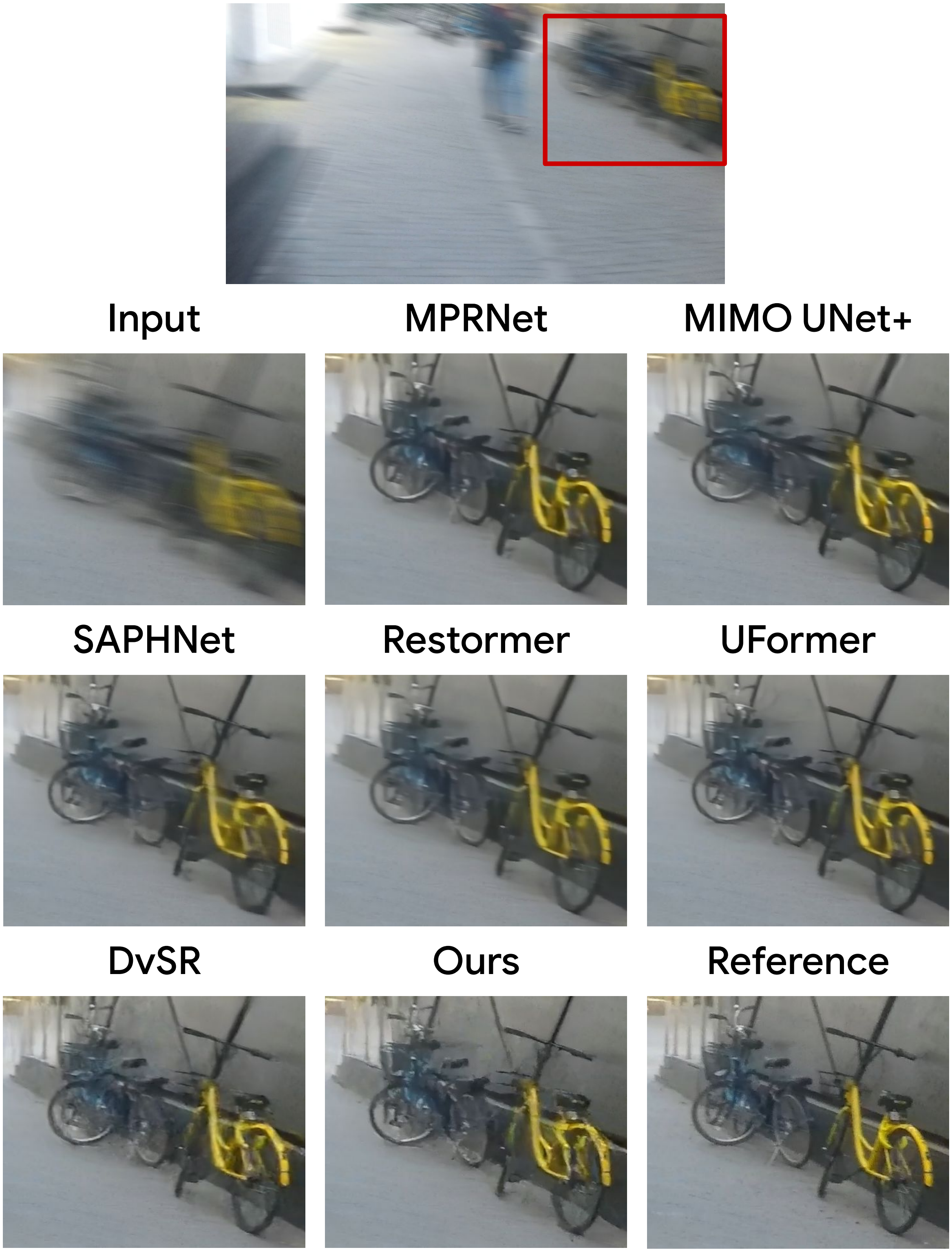


Figure 10. **HIDE** [17] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

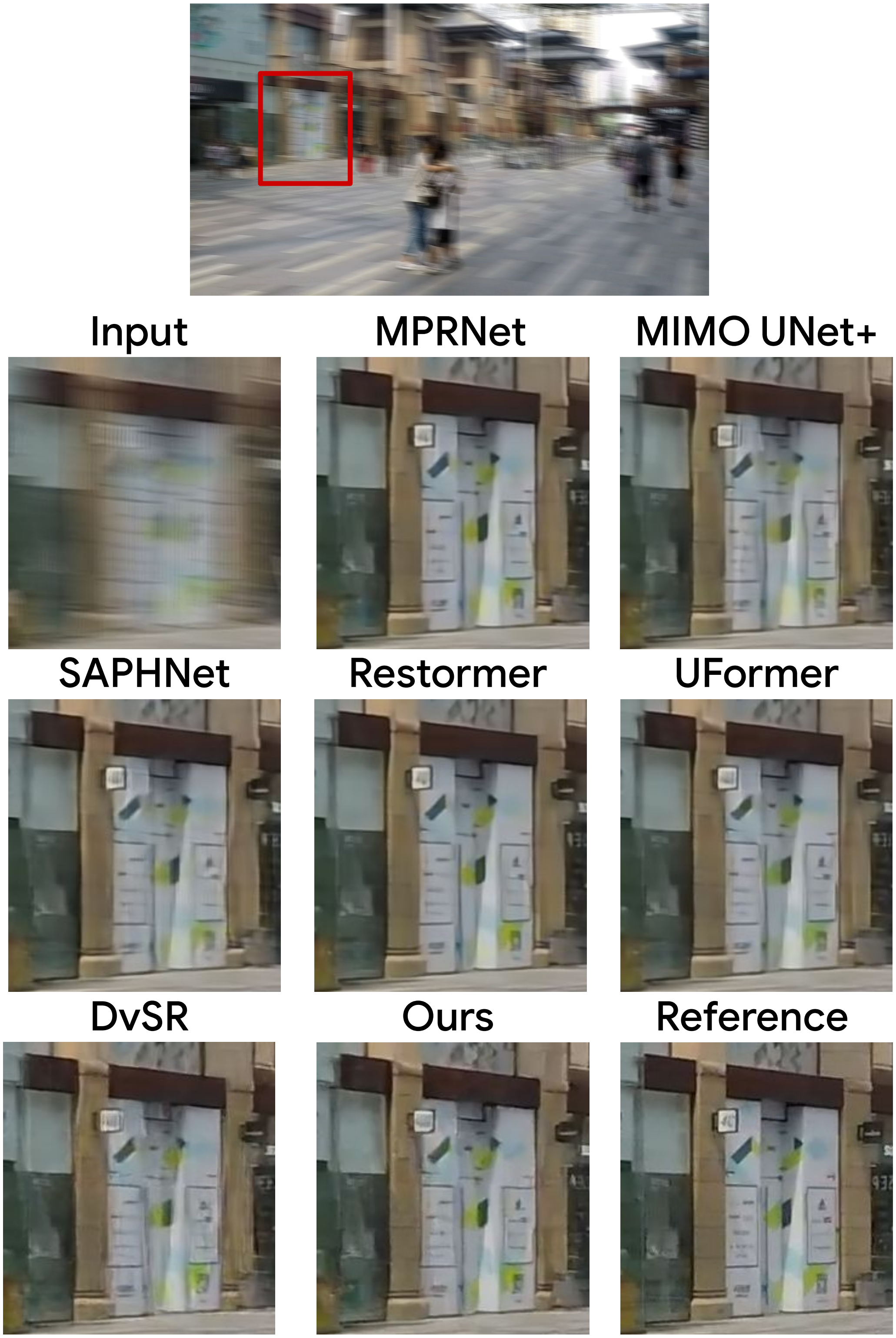


Figure 11. **HIDE** [17] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

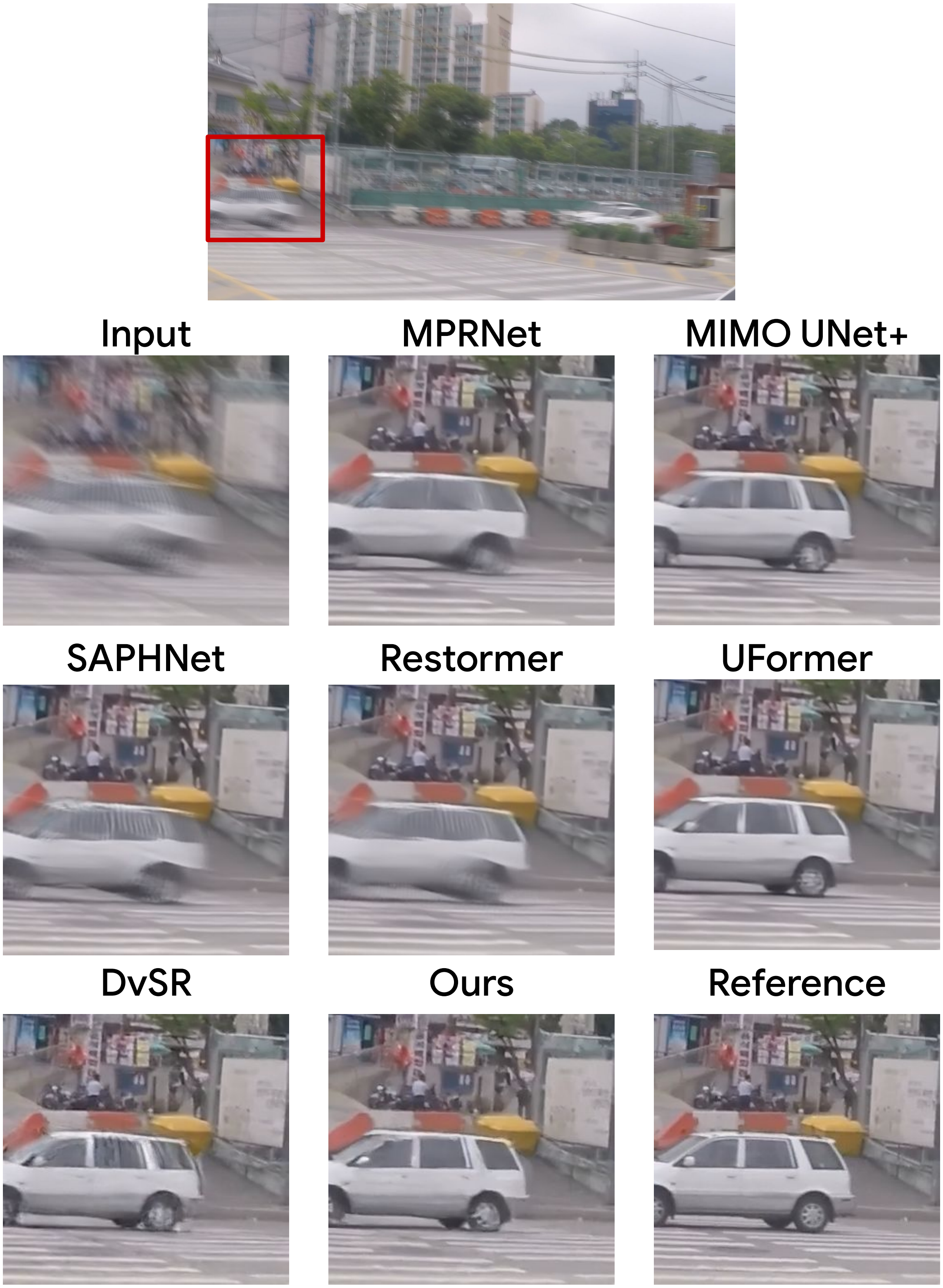


Figure 12. **GoPro** [14] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

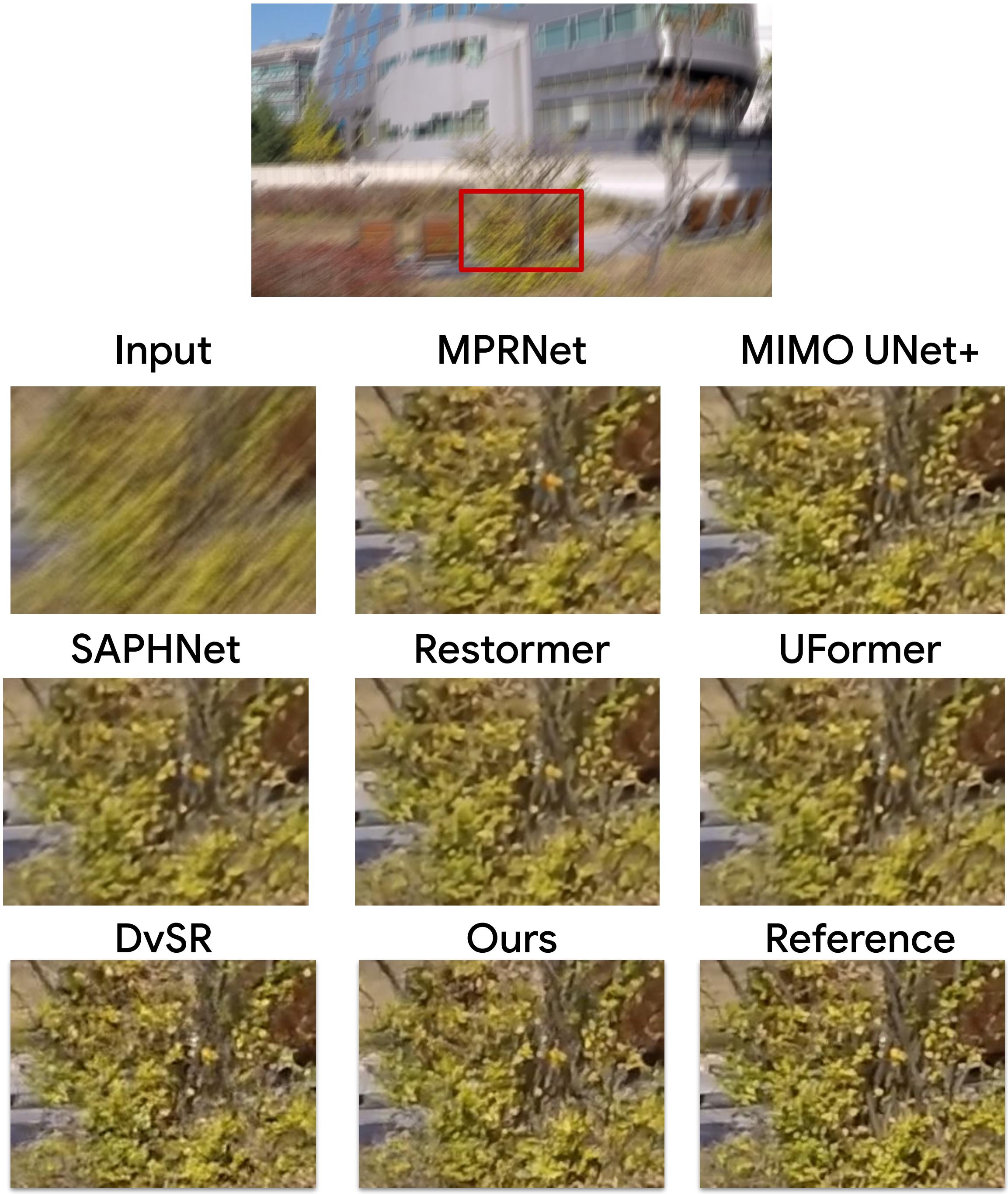


Figure 13. **GoPro** [14] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

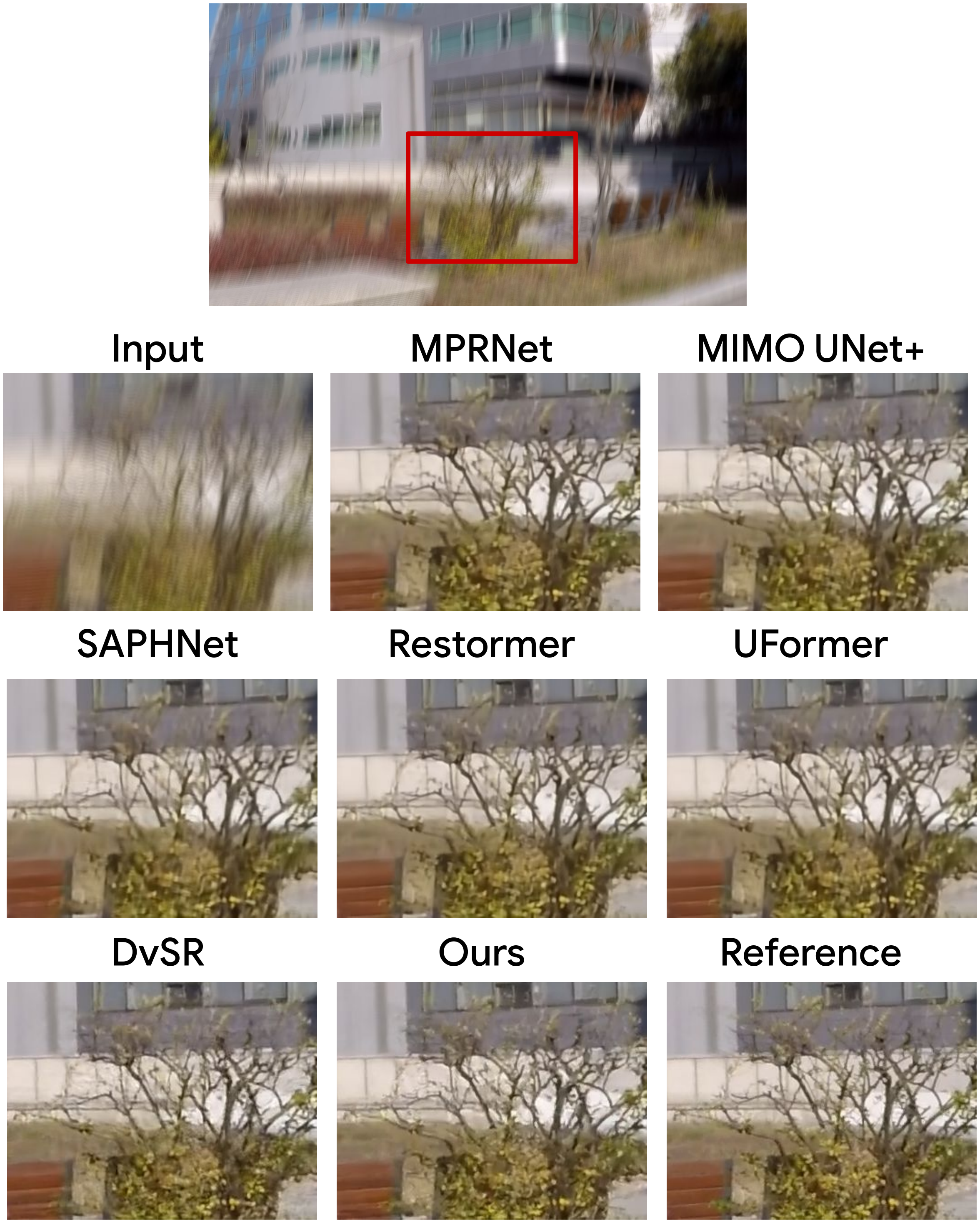


Figure 14. **GoPro** [14] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

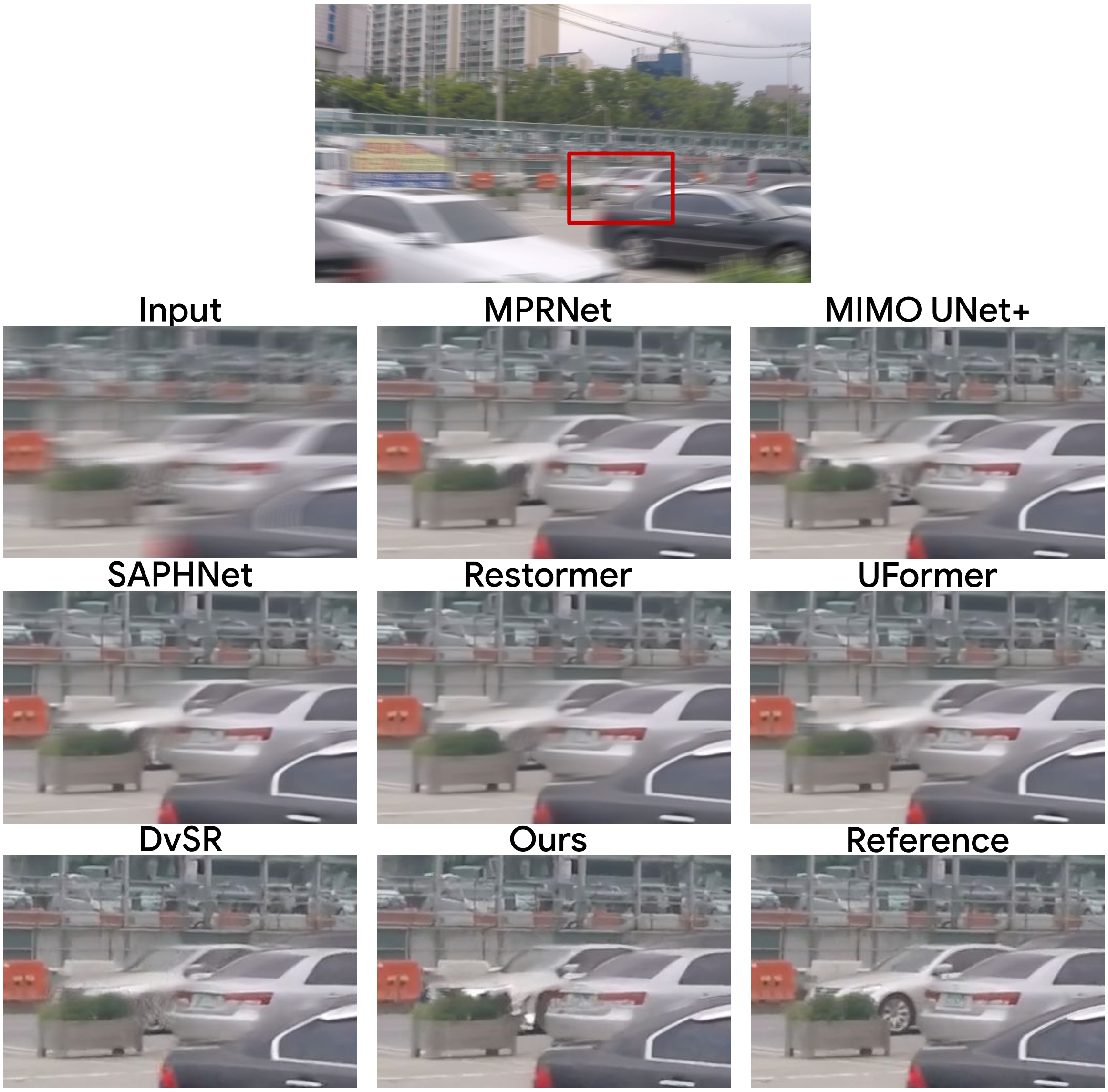


Figure 15. **GoPro** [14] deblurring examples from MPRNet [25], MIMO UNet+ [2], SAPHNet [19], Restormer [24], UFormer [21], DvSR [22] and Ours.

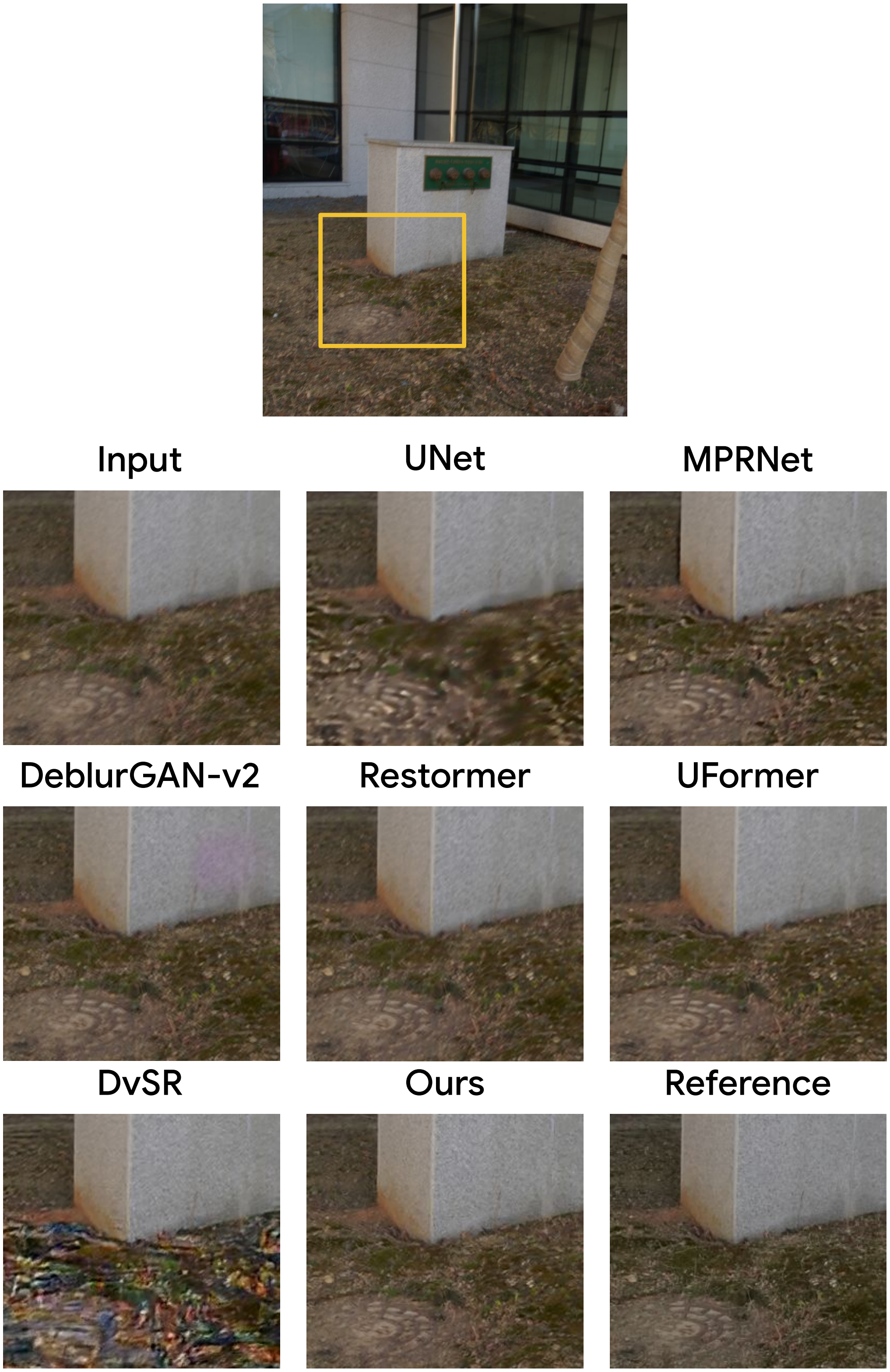


Figure 16. **Realblur-J** [15] deblurring examples from UNet [16], MPRNet [25], DeblurGAN-v2 [6], Restormer [24], UFormer [21], DvSR [22] and Ours.

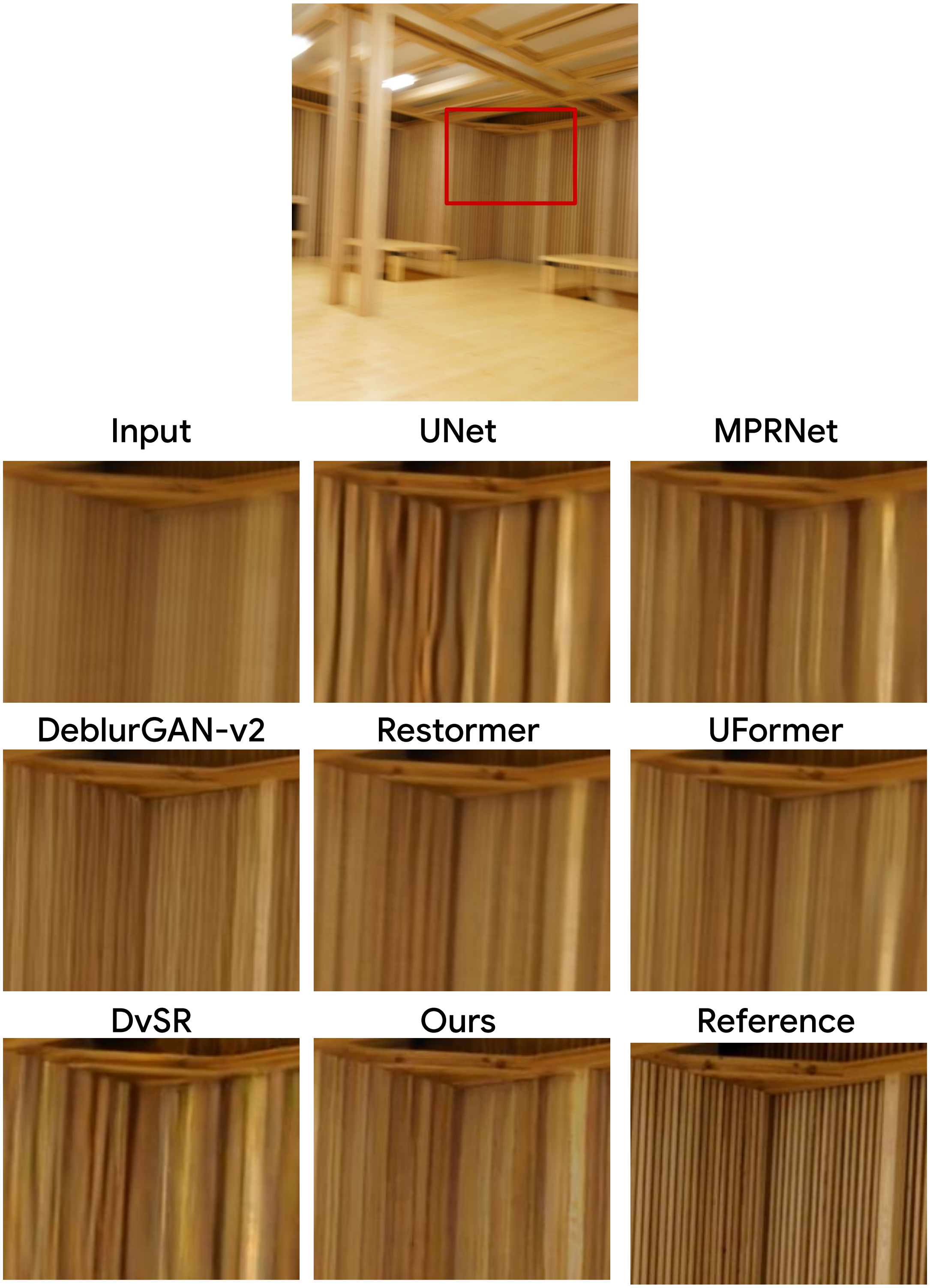


Figure 17. **Realblur-J** [15] deblurring examples from UNet [16], MPRNet [25], DeblurGAN-v2 [6], Restormer [24], UFormer [21], DvSR [22] and Ours.

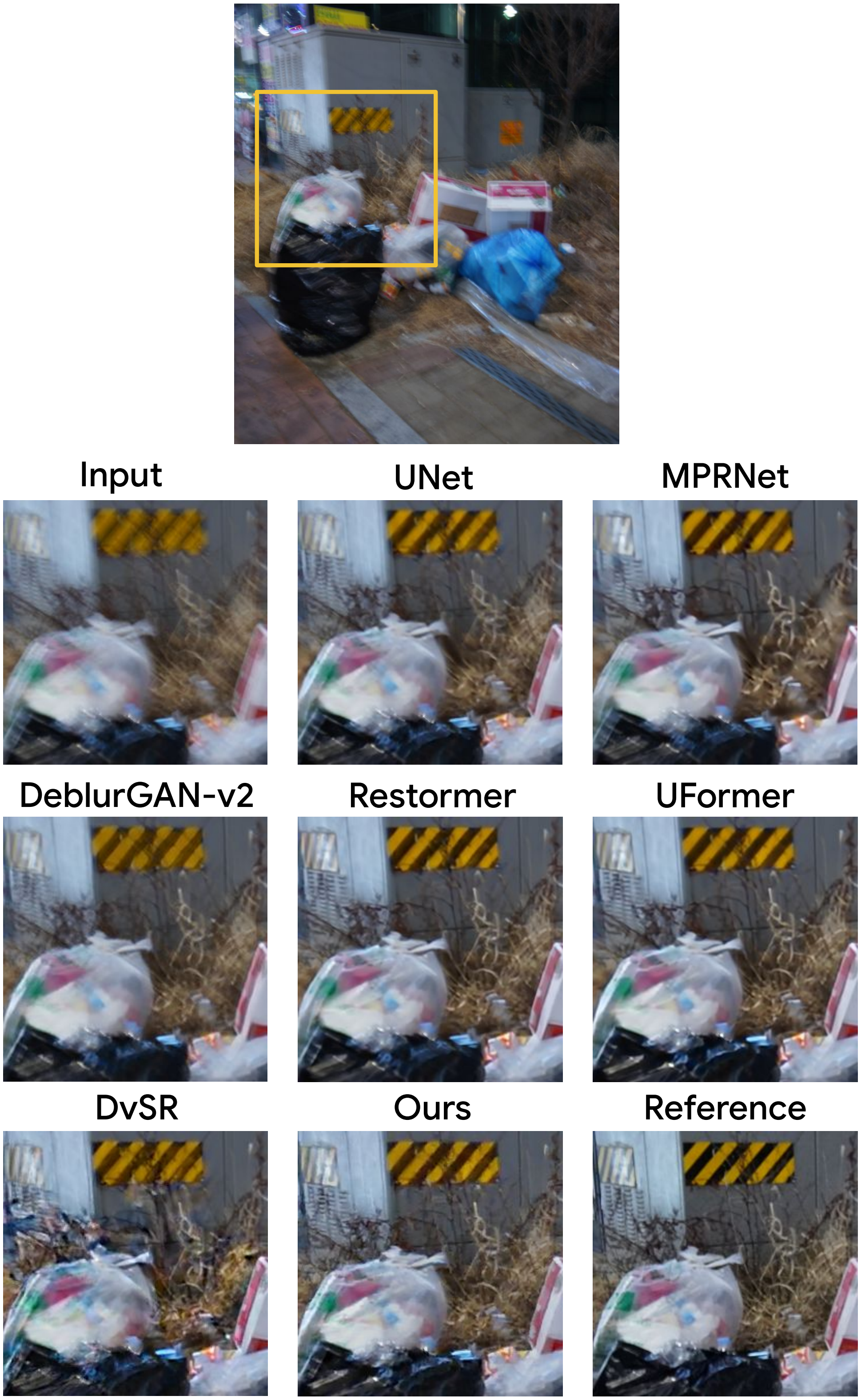


Figure 18. **Realblur-J** [15] deblurring examples from UNet [16], MPRNet [25], DeblurGAN-v2 [6], Restormer [24], UFormer [21], DvSR [22] and Ours.

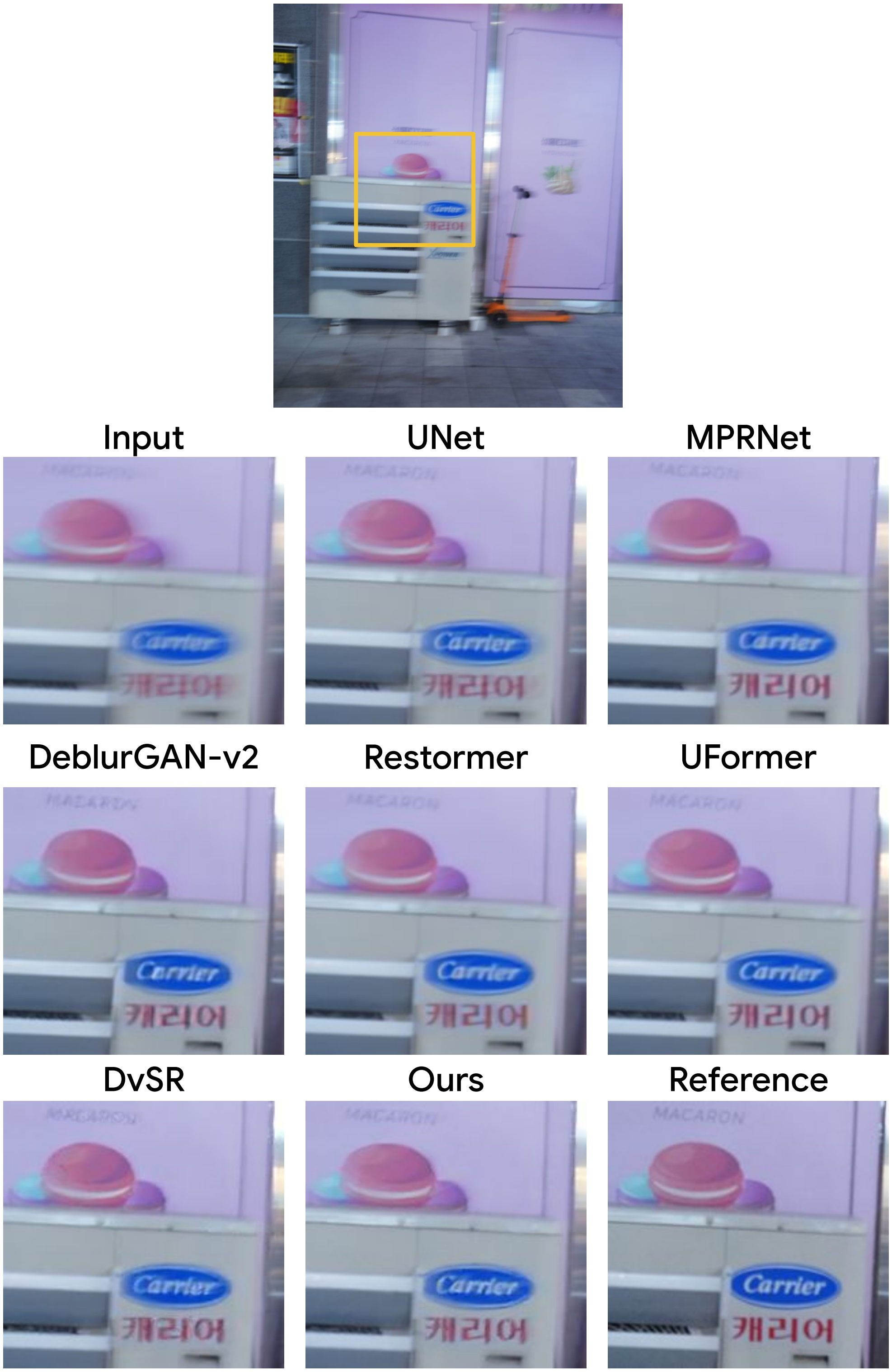


Figure 19. **Realblur-J** [15] deblurring examples from UNet [16], MPRNet [25], DeblurGAN-v2 [6], Restormer [24], UFormer [21], DvSR [22] and Ours.

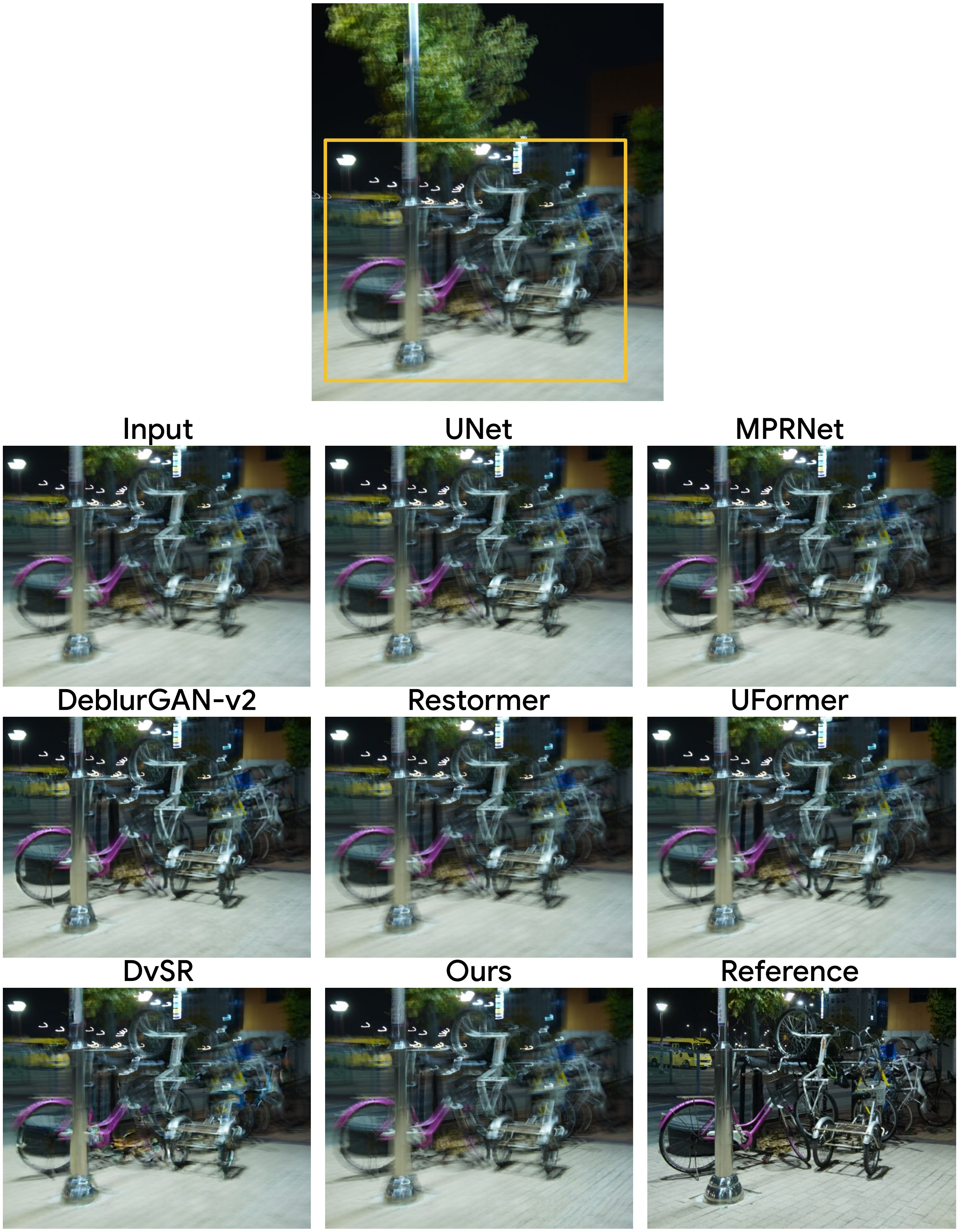


Figure 20. Failure case from **Realblur-J** [15] in low-light scenes.

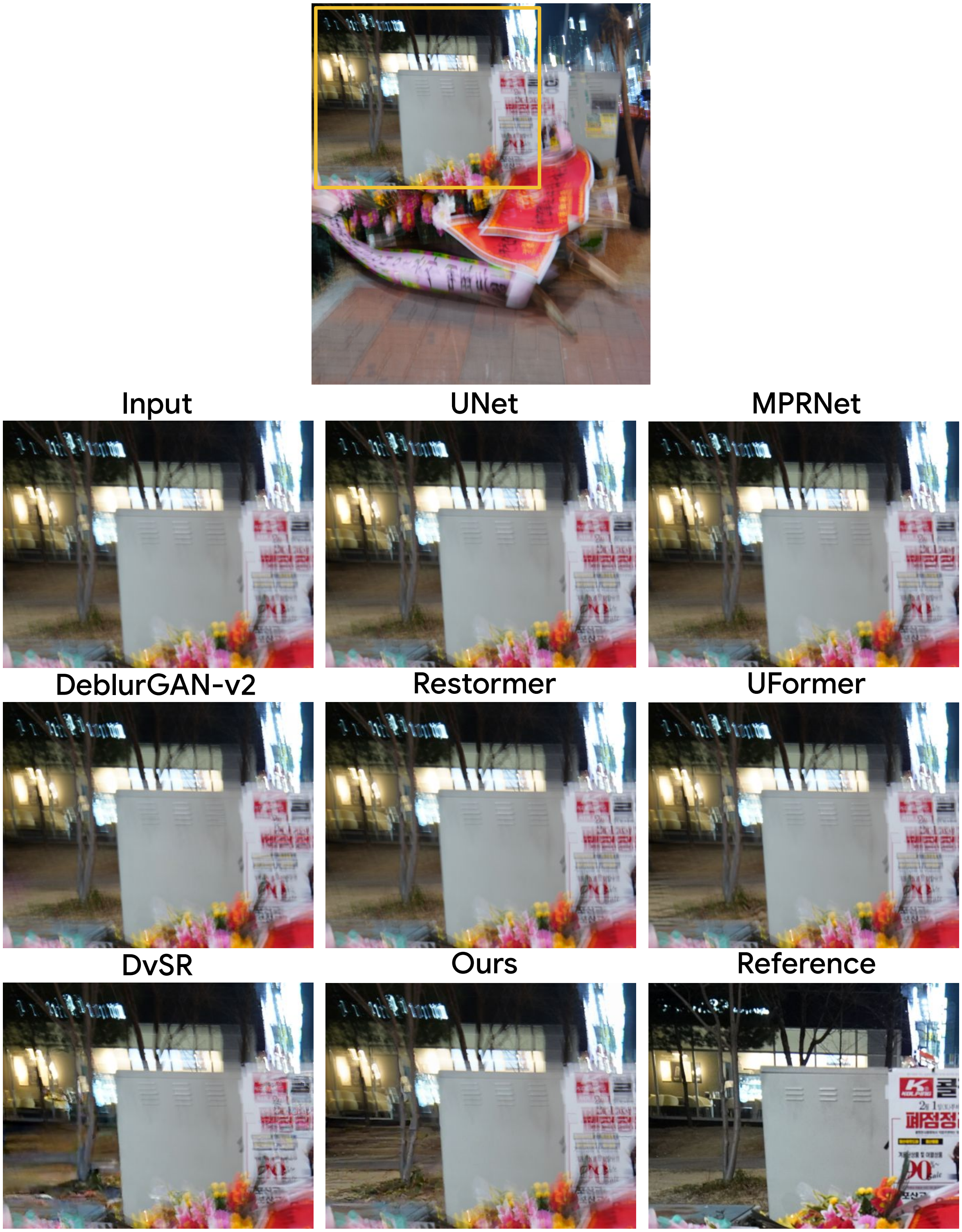


Figure 21. Failure case from **Realblur-J** [15] with strong light streaks.

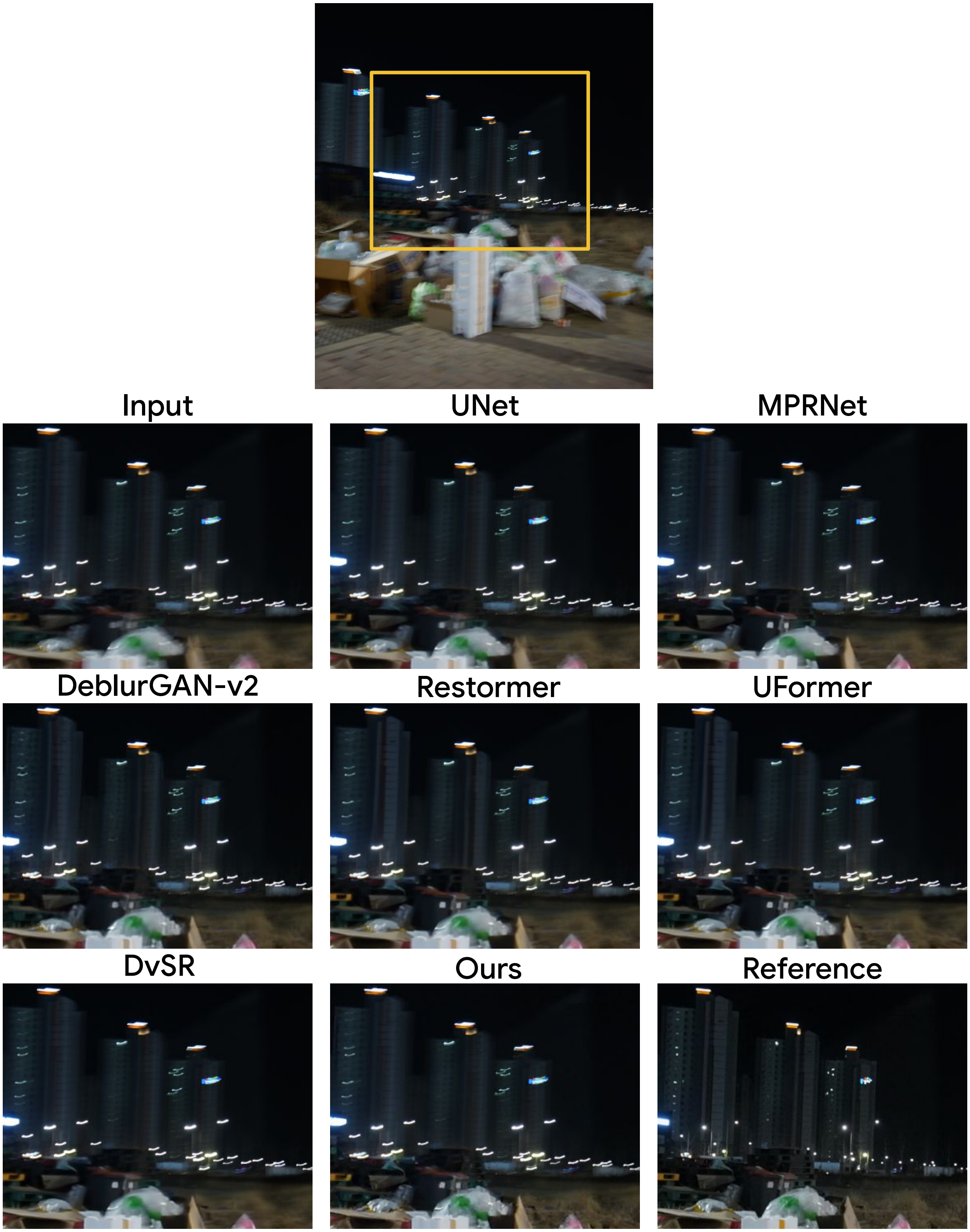


Figure 22. Failure case from **Realblur-J** [15] in night scenes.

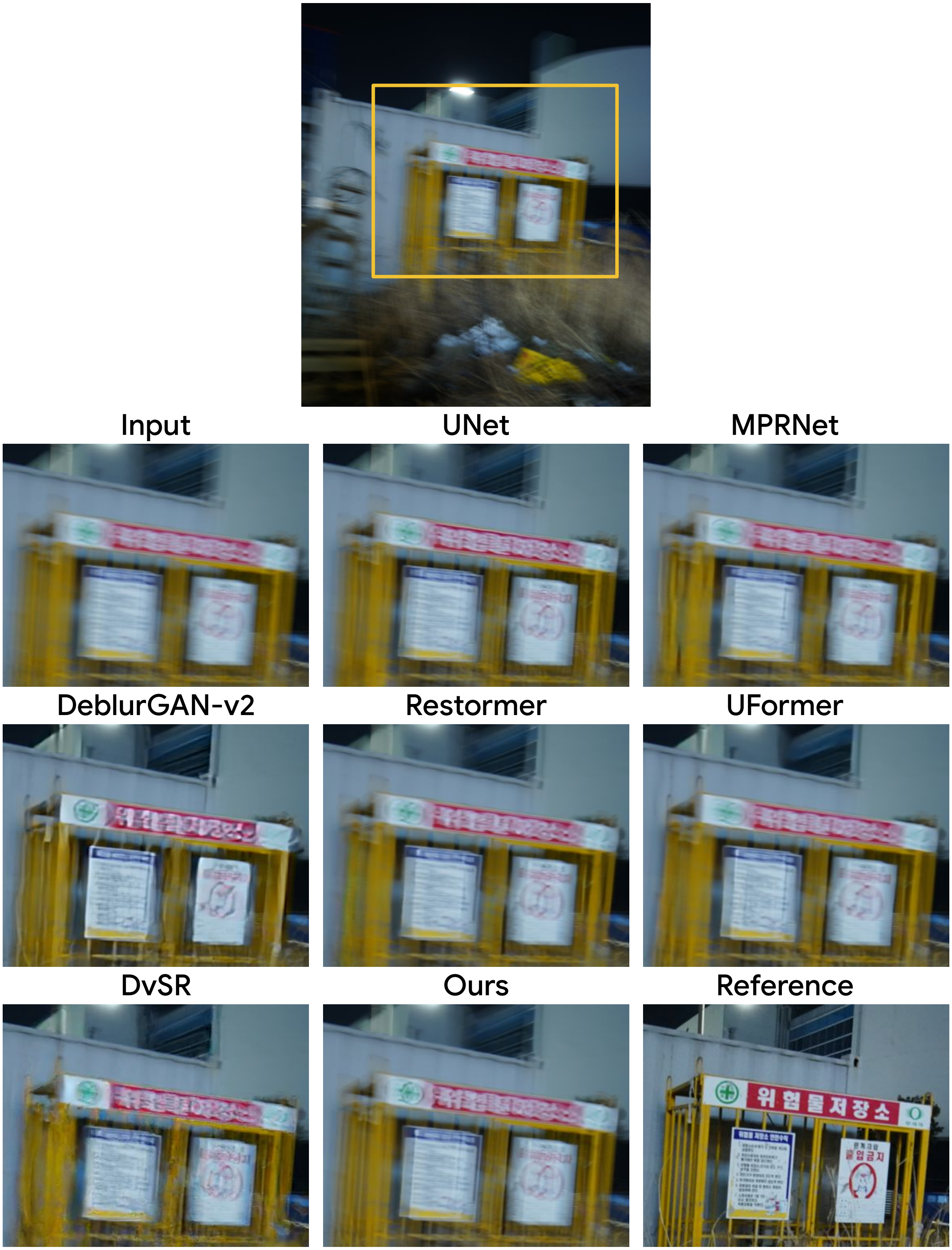


Figure 23. Failure case from **Realblur-J** [15] in low-light condition.

References

[1] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–192, June 2021. 6, 7, 8, 9

[2] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4641–4650, October 2021. 10, 11, 12, 13, 14, 15, 16, 17

[3] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022. 5

[4] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *arXiv preprint arXiv:2210.05475*, 2022. 5

[5] Jungeon Kim Junyong Lee Seungyong Lee Sunghyun Cho Jaesung Rim, Geonung Kim. Realistic blur synthesis for learning image deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[6] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 5, 6, 7, 8, 9, 18, 19, 20, 21

[7] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 2

[8] Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. Efficient spatially sparse inference for conditional gans and diffusion models. *arXiv preprint arXiv:2211.02048*, 2022. 5

[9] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021. 5

[10] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 5

[11] Hengyuan Ma, Li Zhang, Xiatian Zhu, and Jianfeng Feng. Accelerating score-based generative models with preconditioned diffusion sampling. In *European Conference on Computer Vision*, pages 1–16. Springer, 2022. 5

[12] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022. 5

[13] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, June 2019. 1, 5, 6, 7, 8, 9

[14] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 14, 15, 16, 17

[15] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, pages 184–201. Springer, 2020. 2, 18, 19, 20, 21, 22, 23, 24, 25

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 18, 19, 20, 21

[17] Ziyi Shen, Wenguan Wang, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision*, 2019. 1, 10, 11, 12, 13

[18] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 5

[19] Maitreya Suin, Kuldeep Purohit, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 10, 11, 12, 13, 14, 15, 16, 17

[20] Wei Wang, Haochen Zhang, Zehuan Yuan, and Changhu Wang. Unsupervised real-world super-resolution: A domain adaptation perspective. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4298–4307, 2021. 3

[21] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17683–17693, June 2022. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[22] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16293–16303, 2022. 4, 5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[23] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022. 5

[24] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[25] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[26] Shangchen Zhou, Chongyi Li, and Chen Change Loy. Lednet: Joint low-light enhancement and deblurring in the dark. *arXiv preprint arXiv:2202.03373*, 2022. 2