Supplementary Materials for: Uformer: A General U-Shaped Transformer for Image Restoration

1. Additional Ablation Study

1.1. Is Window Shift Important

Table A reports the results of whether to use the shifted window design [7] in Uformer. We observe that window shift brings an improvement of 0.01 dB for image denoising. We use the window shift as the default setting in our experiments.

Uformer-S	PSNR ↑
w/o window shift	39.76
w/ window shift	**39.77**

Table A. Effect of window shift.

1.2. Variants of Skip-Connections

To investigate how to deliver the learned low-level features from the encoder to the decoder, considering the self-attention computing in Transformer, we present three different skip-connection schemes, including concatenation-based skip-connection, cross-attention as skip-connection, and concatenation-based cross-attention as skip-connection.

Concatenation-based Skip-connection (Concat-Skip). Concat-Skip is based on the widely-used skip-connection in UNet [3, 11, 16]. To build our network, firstly, we concatenate the l-th stage flattened features $\mathbf{E}_l$ and each encoder stage with the features $\mathbf{D}_{K-l+1}$ from the $(K\text{-}l\text{+}1)$-th decoder stage channel-wisely. Here, K is the number of the encoder/decoder stages. Then, we feed the concatenated features to the W-MSA component of the first LeWin Transformer block in the decoder stage, as shown in Figure A(a).

Cross-attention as Skip-connection (Cross-Skip). Instead of directly concatenating features from the encoder and the decoder, we design Cross-Skip inspired by the decoder structure in the language Transformer [14]. As shown in Figure A(b), we first add an additional attention module into the first LeWin Transformer block in each decoder stage. The first self-attention module in this block (the shaded one) is used to seek the self-similarity pixel-wisely from the decoder features $\mathbf{D}_{K-l+1}$, and the second attention module in this block takes the features $\mathbf{E}_l$ from the encoder as the *keys* and *values*, and uses the features from the first module as the *queries*.

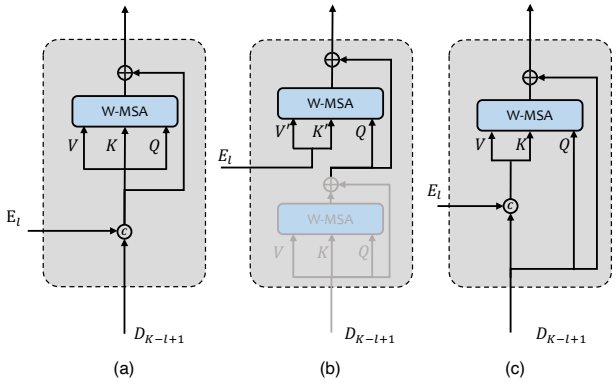


Figure A. Three skip-connection schemes: (a) Concat-Skip, (b) Cross-Skip, and (c) ConcatCross-Skip.

	GMACs	# Param	PSNR ↑
Uformer-S-*Concat*	43.86G	20.63M	**39.77**
Uformer-S-*Cross*	44.78G	27.95M	39.75
Uformer-S-*ConcatCross*	42.75G	27.28M	39.73

Table B. Different skip-connections.

Concatenation-based Cross-attention as Skip-connection (ConcatCross-Skip). Combining above two variants, we also design another skip-connection. As illustrated in Figure A(c), we concatenate the features $\mathbf{E}_l$ from the encoder and $\mathbf{D}_{K-l+1}$ from the decoder as the *keys* and *values*, while the *queries* are only from the decoder.

Table B compares the results of using different skip-connections in our Uformer: concatenating features (*Concat*), cross-attention (*Cross*), and concatenating *keys* and *values* for cross-attention (*ConcatCross*). For a fair comparison, we increase the channels in Uformer-S from 32 to 44 in variants *Cross* and *ConcatCross*. These three skip-connections achieve similar results, and concatenating features gets slightly better performance. We adopt the feature concatenation as the default setting in Uformer.

1.3. More comparisons of the modulator

There are many fine details that need to be restored in degraded images. Traditional models need to find some way

Uformer-T	Baseline	w/ IDM	w/ AdaIN	w/ ours
PSNR ↑	28.47	27.99	28.18	**28.63**

Table C. Comparisons of different modulators for deblurring.

to model these details in activations, which consumes network capacity and is not always successful. However, our modulator directly adds per-pixel learnable parameters on activations, which effectively recovers missing details, as demonstrated in Figure 4 of the main paper. Moreover, we design two variants: input-dependent modulator (IDM) and AdaIN. IDM is an input-dependent adjustment that adds a layer to predict modulator parameters. We report the comparison results in Table C. Our designed modulators achieve the best results. One possible reason for the failure of IDM is that the model uses IDM as part of the network structure to capture fine details rather than learning a general degradation pattern to adjust activations. Besides, AdaIN adjusts the activations too coarsely, which may be harmful to recover missing details.

2. Additional Experiment for Demoireing

We also conduct an experiment of moire pattern removal on the TIP18 dataset [13]. As shown in Table D, Uformer outperforms previous methods MopNet [5], MSNet [13], CFNet [6], UNet [11] by 1.53 dB, 2.29 dB, 3.19 dB, and 2.79 dB, respectively. And in Figure F, we show examples of visual comparisons with other methods. This experiment further demonstrates the superiority of Uformer.

	UNet [11]	CFNet [6]	MSNet [13]	MopNet [5]	**Uformer-B**
PSNR ↑	26.49	26.09	26.99	<u>27.75</u>	**29.28**
SSIM ↑	0.864	0.863	0.871	<u>0.895</u>	**0.917**

Table D. Results on the TIP18 dataset [13] for demoireing.

3. Additional Experimental Settings for Different Tasks

Denoising. The training samples are randomly cropped from the original images in SIDD [1] with size 128×128, which is also the common training strategy for image denoising in recent works [3, 16, 17]. And the training process lasts for 250 epochs with batch size 32. Then, the trained model is evaluated on the 256×256 patches of SIDD and 512×512 patches of the DND test images [9], following [3, 17]. The results on DND are online evaluated.

Motion deblurring. Following previous methods [18, 19], we train Uformer only on the GoPro dataset [8], and evaluate it on the test set of GoPro, HIDE [12], and RealBlur-R/-J [10]. The training patches are randomly cropped from the training set with size 256×256. The batch size is set to 32. For validation, we use the central crop with size 256×256. The number of training epochs is 3k. For evaluation, the trained model is tested on the full-size test images.

Defocus deblurring. Following the official patch segmentation algorithm [2] of DPD, we crop the training and validation samples to 60% overlapping 512×512 patches to train the model. We also discard 30% of the patches that have the lowest sharpness energy (by applying Sobel filter to the patches) as [2]. The whole training process lasts for 160 epochs with batch size 4. For evaluation, the trained model is tested on the full-size test images.

Deraining. We conduct deraining experiments on the SPAD dataset [15]. This dataset contains over 64k 256×256 images for training and 1k 512×512 images for evaluation. We train Uformer on two GPUs, with mini-batches of size 16 on the 256×256 samples. Since this dataset is large enough and the training process converges fast, we just train Uformer for 10 epochs in the experiment. Finally, we evaluate the performance on the test images following the default setting in [15].

Demoireing. We further validate the effectiveness of Uformer on the TIP18 dataset [13] for demoireing. Since the images in this dataset contain additional borders, following [5], we crop the central regions with the ratio of $[0.15, 0.85]$ in all training/validation/testing splits and resize them to 256×256 for training and evaluation. Since this task is sensitive to the down-sampling operation, we choose the bilinear interpolation same as the previous work [5][1]. The training epochs are 250.

4. More Visual Comparisons

As shown in Figures B-F in this supplementary materials, we give more visual results of our Uformer and others on the five tasks (denoising, motion deblurring, defocus deblurring, deraining, and demoireing) as the supplement of the visualization in the main paper.

5. Limitation and broader impacts

Thanks to the proposed architecture, Uformer achieves the state-of-the-art performance on a variety of image restoration tasks (image denoising, deblurring, and deraining). But we have not evaluated Uformer for more vision tasks such as image-to-image translation, image super-resolution, and so on. We look forward to investigating Uformer for more applications. Meanwhile, we notice that there are several negative impacts caused by abusing image restoration techniques. For example, it may cause human privacy issue with the restored images in surveillance. The techniques may destroy the original patterns for camera identification and

[1] The dataset we used is also downloaded from the Github Page of [5].

multi-media copyright [4], which hurts the authenticity for image forensics.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A High-Quality Denoising Dataset for Smartphone Cameras. In *CVPR*, 2018. 2, 4

[2] Abdullah Abuolaim and Michael S Brown. Defocus Deblurring Using Dual-Pixel Data. In *ECCV*. Springer, 2020. 2, 6

[3] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. NBNet: Noise Basis Learning for Image Denoising with Subspace Projection. In *CVPR*, 2021. 1, 2

[4] Davide Cozzolino and Luisa Verdoliva. Noiseprint: a CNN-based camera model fingerprint. *arXiv preprint arXiv:1808.08396*, 2018. 3

[5] Bin He, Ce Wang, Boxin Shi, and Ling-Yu Duan. Mop Moire Patterns Using MopNet. In *ICCV*, 2019. 2

[6] Bolin Liu, Xiao Shu, and Xiaolin Wu. Demoiréing of Camera-Captured Screen Images Using Deep Convolutional Neural Network. *arXiv preprint arXiv:1804.03809*, 2018. 2

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[8] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring. In *CVPR*, 2017. 2

[9] Tobias Plotz and Stefan Roth. Benchmarking Denoising Algorithms with Real Photographs. In *CVPR*, 2017. 2

[10] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms. In *ECCV*, 2020. 2

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*. Springer, 2015. 1, 2

[12] Ziyi Shen, Wenguan Wang, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-Aware Motion Deblurring. In *ICCV*, 2019. 2

[13] Yujing Sun, Yizhou Yu, and Wenping Wang. Moiré photo restoration using multiresolution convolutional neural networks. *TIP*, 27(8):4160–4172, 2018. 2

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017. 1

[15] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset. In *CVPR*, 2019. 2, 5, 6, 7

[16] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual Adversarial Network: Toward Real-world Noise Removal and Noise Generation. In *ECCV*, 2020. 1, 2

[17] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning Enriched Features for Real Image Restoration and Enhancement. In *ECCV*, 2020. 2

[18] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-Stage Progressive Image Restoration. In *CVPR*, 2021. 2

[19] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep Stacked Hierarchical Multi-patch Network for Image Deblurring. In *CVPR*, 2019. 2

Figure B. More visual results on the SIDD dataset [1] for image denoising. The PSNR value under each patch is computed on the corresponding whole image.

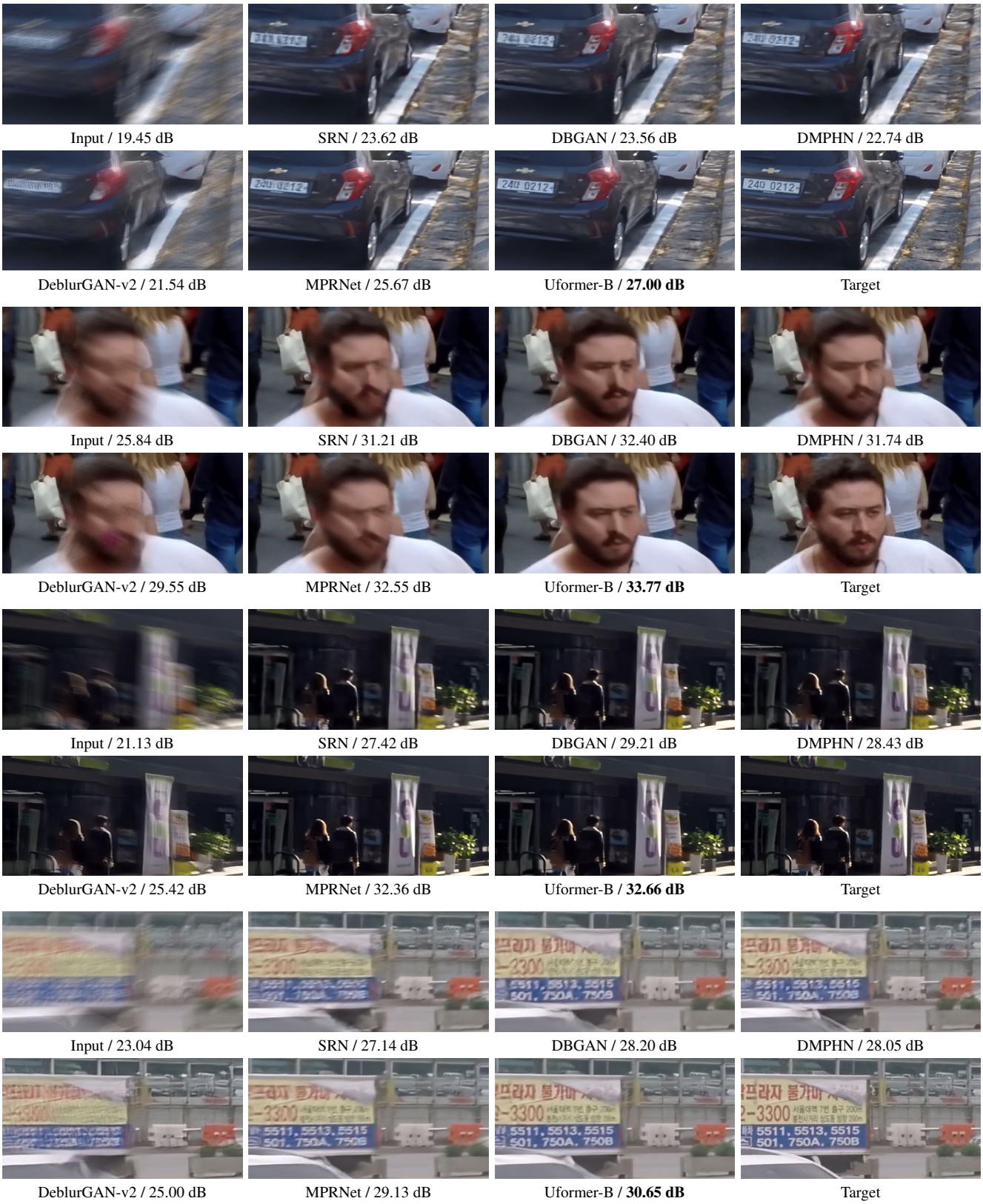


Figure C. More results on GoPro [15] for image motion deblurring. The PSNR value under each patch is computed on the corresponding whole image.

Figure D. More results on DPD [2] for image defocus deblurring. We report the performance of PSNR on the whole test image and show the zoomed region only for visual comparison.

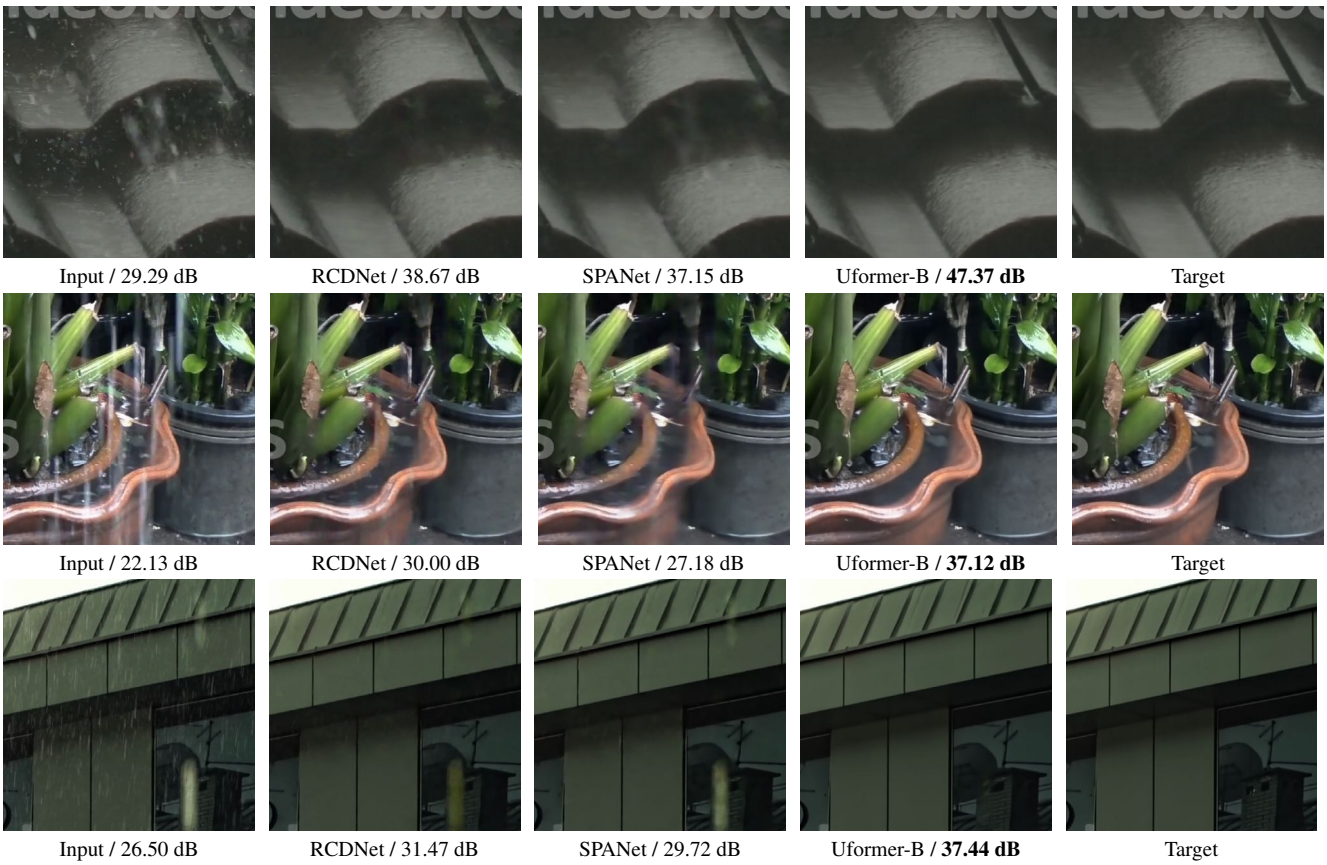
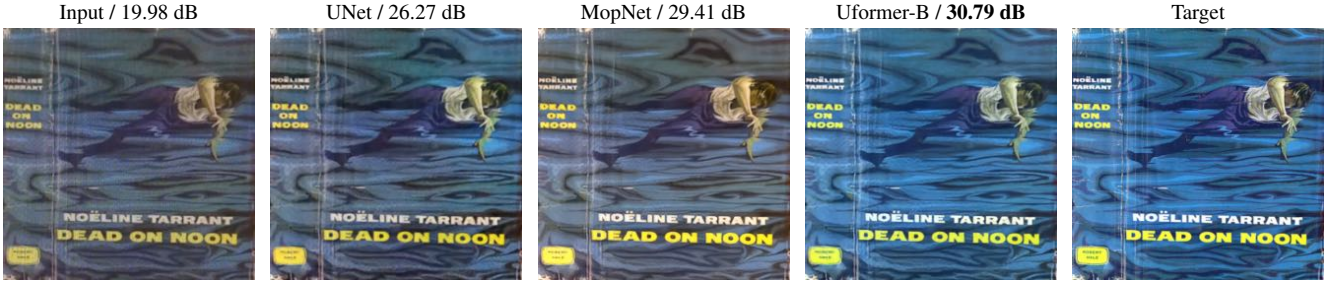


Figure E. More results on SPAD [15] for image deraining.

Input / 19.47 dB | UNet / 29.62 dB | MopNet / 30.45 dB | Uformer-B / **32.74 dB** | Target

Input / 14.99 dB | UNet / 26.78 dB | MopNet /25.44 dB | Uformer-B / **31.76 dB** | Target

Input / 19.98 dB | UNet / 26.27 dB | MopNet / 29.41 dB | Uformer-B / **30.79 dB** | Target



Input / 17.15 dB | UNet / 19.35 dB | MopNet / 18.84 dB | Uformer-B / **26.86 dB** | Target

Input / 15.20 dB | UNet / 28.17 dB | MopNet / 29.09 dB | Uformer-B / **30.63 dB** | Target

Figure F. Results on the TIP18 dataset [15] for image demoireing.