Multi-scale Residual Low-Pass Filter Network for Image Deblurring Supplemental Material

Jiangxin Dong Jinshan Pan* Zhongbao Yang Jinhui Tang
Nanjing University of Science and Technology

Overview

In this supplemental material, we first present the detailed network architecture of the proposed approach in Section 1. Then, we provide the visualization of the learned low-pass filters in Section 3 and discuss the robustness of the proposed method to various learnable low-pass filters in Section 4. Section 5 shows more visual comparisons against other approaches.

1. Network architecture of the Multi-scale Residual Low-Pass Filter Network

Figure 2(a) of the main paper shows the network architecture of the proposed multi-scale residual low-pass filter network. To extract useful features from the input image of the finest scale, we use a network of one convolutional layer followed by the ReLU activation. For the feature extraction in the coarser scale, we first employ a network of four convolutional layers, each layer followed by the ReLU activation. Then we concatenate the output with the input and apply one convolutional layer on the concatenation to generate the features. In addition, Figure 1 shows the detailed architecture of each RLPF module in Figure 2(a) of the main paper.

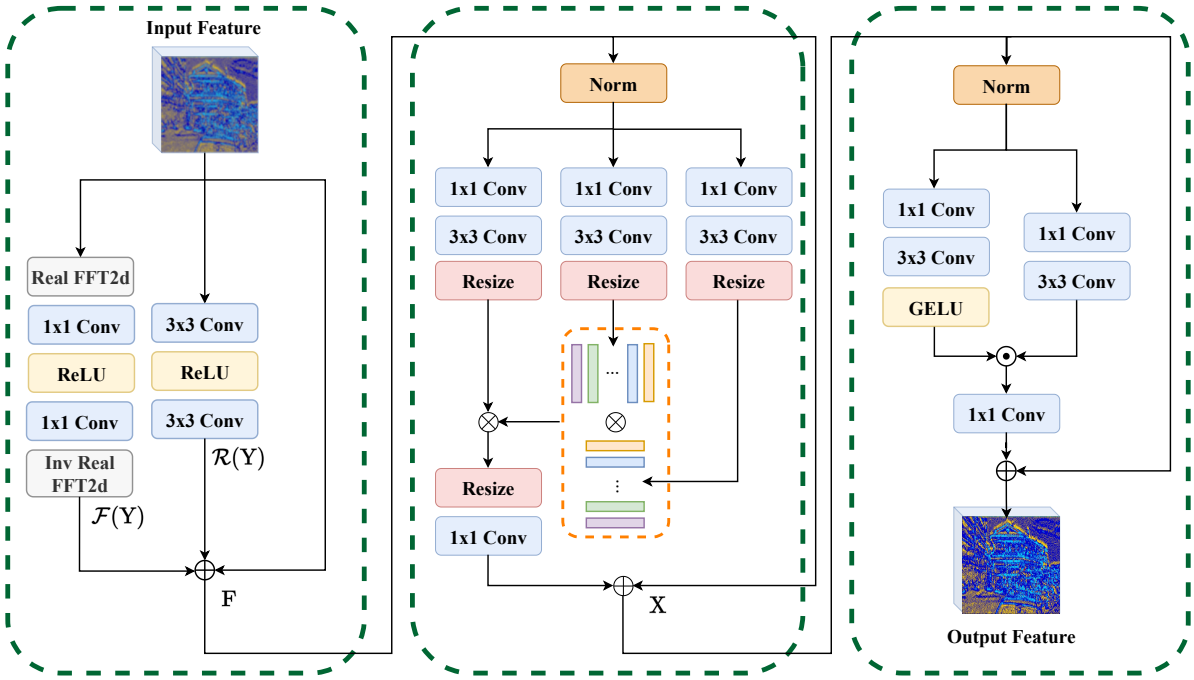


Figure 1. Network architecture of the proposed RLPF module for image deblurring.

*Corresponding author

Table 1. Effectiveness of the proposed low-pass filter and wavelet-based feature fusion for image deblurring. All baseline methods are trained using the same settings as the proposed method for fair comparisons.

	Low-pass filter (LPF)			Feature fusion (FF)		GoPro
	Gaussian filter (GF)	Bilateral filter (BF)	Learnable filter (LF)	Bilinear-based (BFF)	Wavelet-based (WFF)	PSNR (dB)/SSIM
MRLPFNet$_{\text{w/ GF\&WFF}}$	✔	✗	✗	✗	✔	33.1/0.9620
MRLPFNet$_{\text{w/ BF\&WFF}}$	✗	✔	✗	✗	✔	33.3/0.9635
MRLPFNet	✗	✗	✔	✗	✔	**33.5/0.9650**

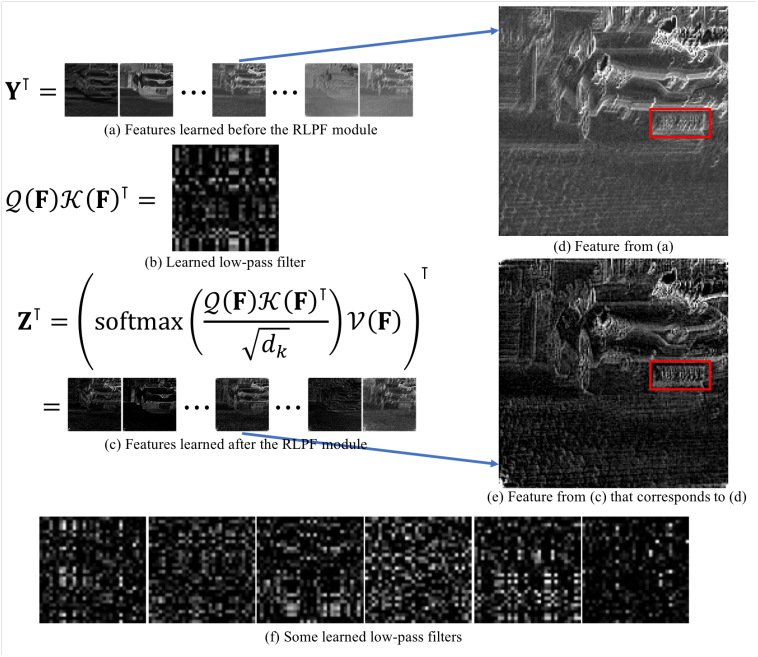


Figure 2. Visualization of some learned low-pass filters and features learned before and after the proposed RLPF module. $\mathbf{F}$ is obtained by Eq. (3) in the main paper.

2. Effect of the Learned Low-Pass Filter

In this section, we aim to complement the results in the main paper (in particular Table 4 of the main paper) to demonstrate the effectiveness of the proposed low-pass filter and wavelet-based feature fusion. We further investigate whether using the classical low-pass filters, e.g., Gaussian filter and bilateral filter, can achieve competitive performances compared to the learnable low-pass filter. For this purpose, we individually replace the learnable low-pass filter with the Gaussian filter (MRLPFNet$_{\text{w/ GF\&WFF}}$ for short) and the bilateral filter (MRLPFNet$_{\text{w/ BF\&WFF}}$ for short) in our implementation. As Table 1 shows, using the image-adaptive bilateral filter performs better than using the translation-invariant Gaussian filter, as the former one can adaptively model the spatially-variant contents and preserve more useful low-frequency information in contrast with the latter one. Whereas, the proposed method using the learnable low-pass filter outperforms these baselines by about 0.2dB, which demonstrates the effectiveness of embedding the learnable low-pass filter into an end-to-end network for better image deblurring.

3. Visualization of Learned Low-Pass Filters

To intuitively illustrate what the proposed RLPF module learns, we show some features learned before the RLPF module in Figure 2(a). Some learned low-pass filters are shown in Figure 2(b). The corresponding features learned after the RLPF module are shown in Figure 2(c). Specifically, we amplify two corresponding features in Figure 2(d) and (e). Compared to the feature learned before the RLPF module in Figure 2(d), the feature in Figure 2(e) obtained after the RLPF module with the learned low-pass filter mainly contains the image structures and less blur, e.g. the license plate enclosed in the red boxes, which shows the effectiveness of the RLPF module. In addition, we visualize more learned low-pass filters in Figure 2(f).

4. Robustness to Various Learnable Low-Pass Filters

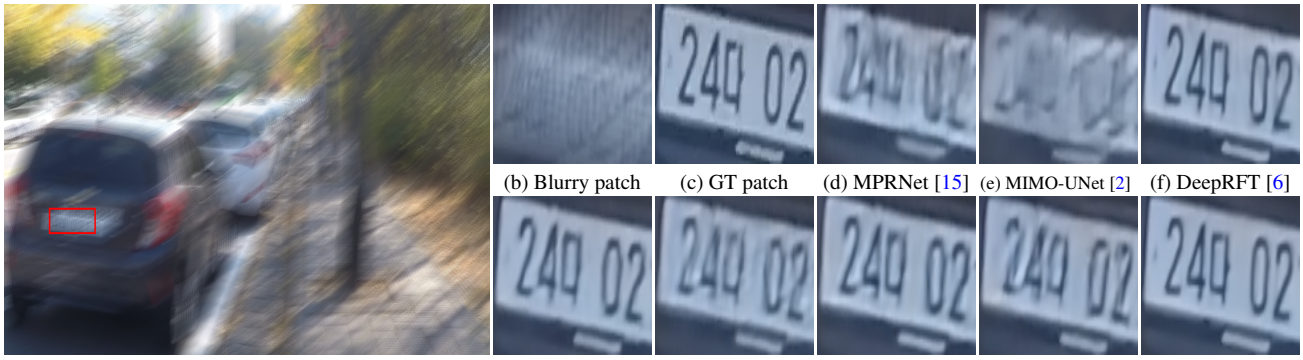
In Section 3 of the main paper, we discuss the relations between the low-pass filters and the scaled dot-production attention in Transformers. In the experiments, we use the standard scaled dot-production attention in the Transformer according to [14]. Note that there are lots of variants of Transformers. To demonstrate the robustness of the proposed method to the methods that are used for the scaled dot-production attention, we further use the window-based scheme by [5] to compute the scaled dot-production attention as the learnable low-pass filter. Table 2 shows that the proposed method is robust to different self-attention mechanisms to some extent. In addition, we note that using the low-pass filter improves the deblurring performance, where the PSNR gains are at least 0.41dB.

Table 2. Robustness of the proposed method to different learnable low-pass filters, evaluated on the GoPro dataset [7].

Methods	MRLPFNet w/o low-pass filter	MRLPFNet w/ [5]	**MRLPFNet**
PSNRs	33.01	33.42	**33.50**
SSIMs	0.9616	0.9649	**0.9650**

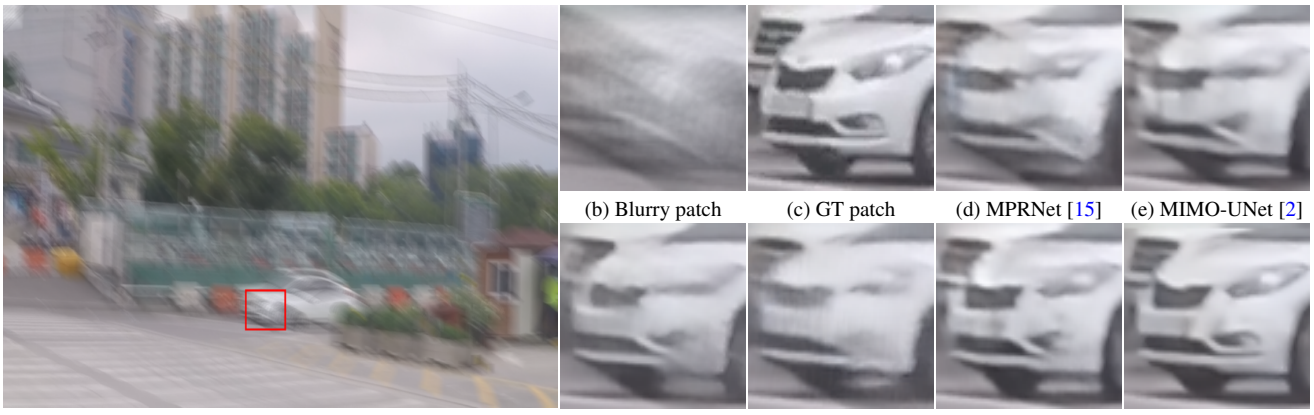
5. More Experimental Results

In this section, we provide more visual comparisons with state-of-the-art methods to demonstrate the effectiveness of the proposed method in Figures 3-13.



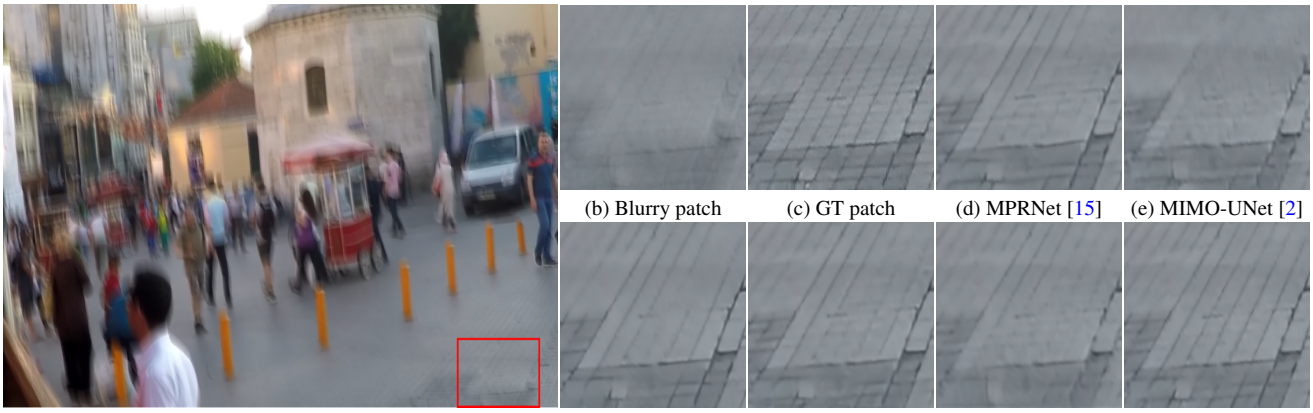
(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) DeepRFT [6] (g) NAFNet [1] (h) Restormer [14] (i) Restormer-L [3] (j) Stripformer [11] (k) MRLPFNet

Figure 3. Example from the GoPro dataset [7]. The competing methods [15, 2, 1, 14, 3, 11] are less effective in generating clear images, where the restored characters in (d)-(e) and (g)-(j) still contain the blur effects. In contrast, the proposed approach obtains a better-deblurred image with clearer license plate numbers.



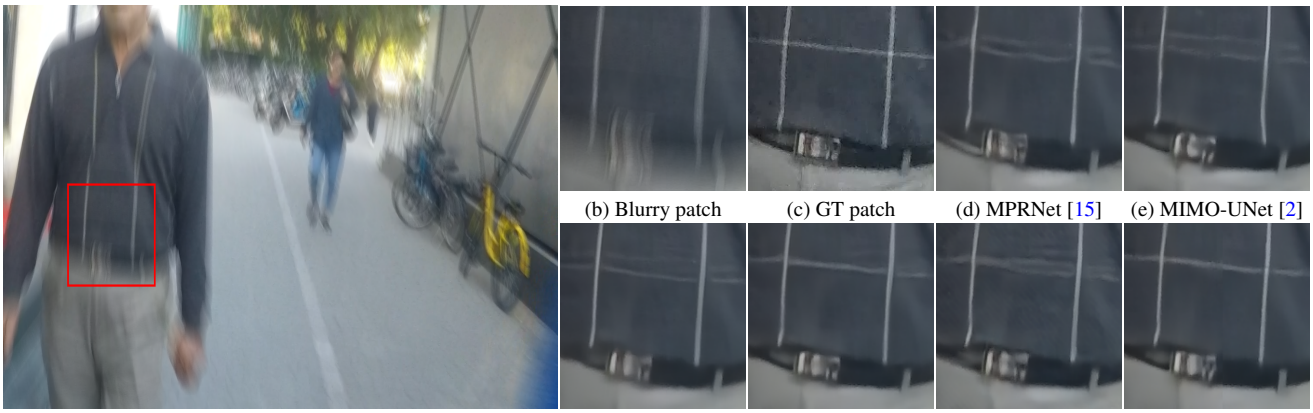
(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) MAXIM [12] (g) Restormer [14] (h) Uformer [13] (i) MRLPFNet

Figure 4. Example from the GoPro dataset [7]. The results generated by [15, 2] have severe artifacts as shown in (d)-(e). For the evaluated methods [12, 14, 13], the restored cars in (f)-(h) still contain some blur effects. However, the proposed method is able to generate a much clearer image as shown in (i).



(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) MAXIM [12] (g) Restormer [14] (h) Uformer [13] (i) MRLPFNet

Figure 5. Example from the GoPro dataset [7]. The result restored by [2] still has some blur effects as shown in (e). The methods [15, 12, 14, 13] can remove the blur, but some details are smoothed as shown in (d) and (f)-(h). In contrast, the proposed approach is more effective in preserving fine-scale structures as shown in (i).

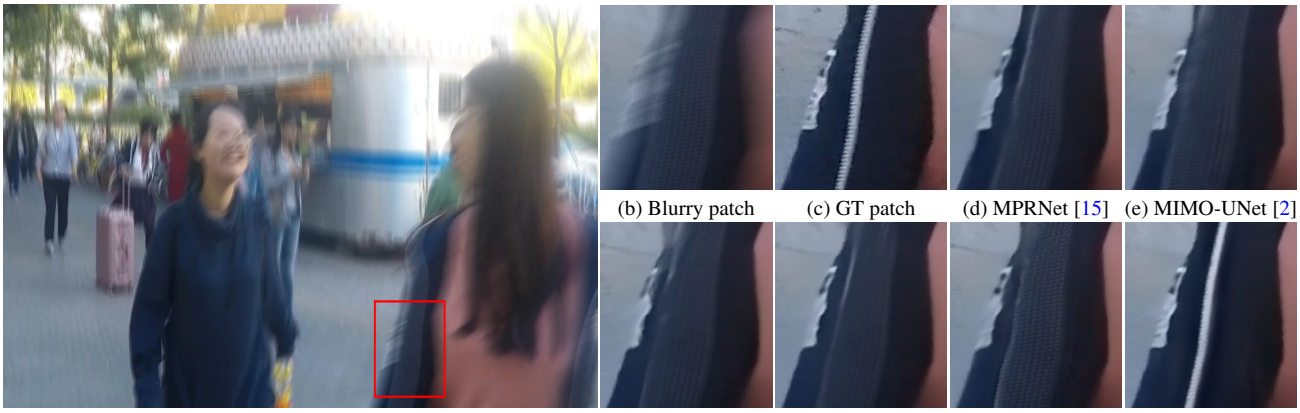


(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) Restormer [14] (g) Restormer-L [3] (h) Stripformer [11] (i) MRLPFNet

Figure 6. Example from the HIDE dataset [9]. The results (f)-(g) generated by [14, 3] have some blur effects. The deblurred images by [15, 2, 11] have obvious visual distortion, where the stripes of the clothes are not restored well as shown in (d)-(e) and (h). In contrast, the proposed method generates a clearer image with better details as shown in (i).

(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) DeepRFT [6] (g) Restormer [14] (h) Stripformer [11] (i) MRLPFNet

Figure 7. Example from the HIDE dataset [9]. The evaluated methods [15, 2, 6, 14, 11] do not effectively remove the blur effects from the blurry input as shown in (d)-(h). In contrast, the proposed approach generates a better-deblurred image with clearer stripes of the T-shirt as shown in (i).

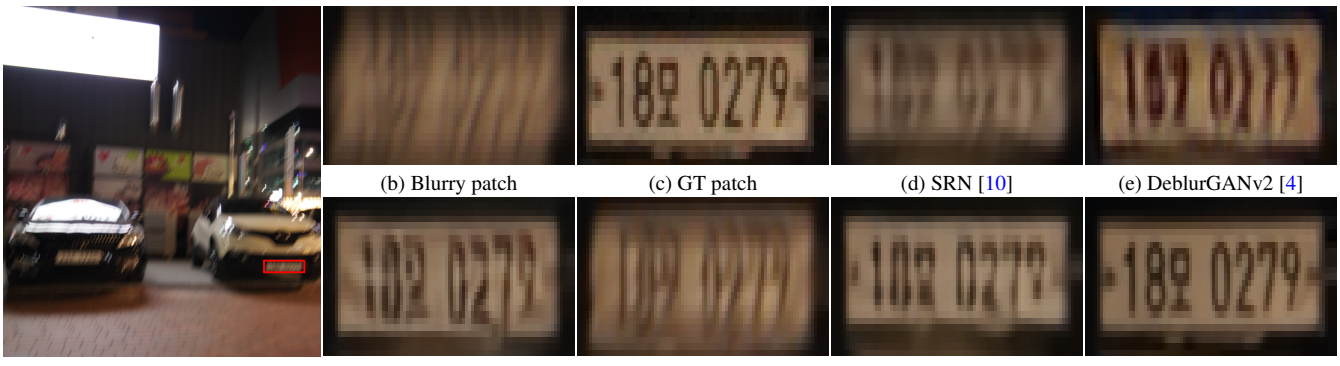


(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) MIMO-UNet [2] (f) Restormer [14] (g) Restormer-L [3] (h) Stripformer [11] (i) MRLPFNet

Figure 8. Example from the HIDE dataset [9]. Compared to the results restored by [15, 2, 14, 3, 11] in (d)-(h), the proposed method is more effective in generating a clear image and preserving fine-scale structures as shown in (i).

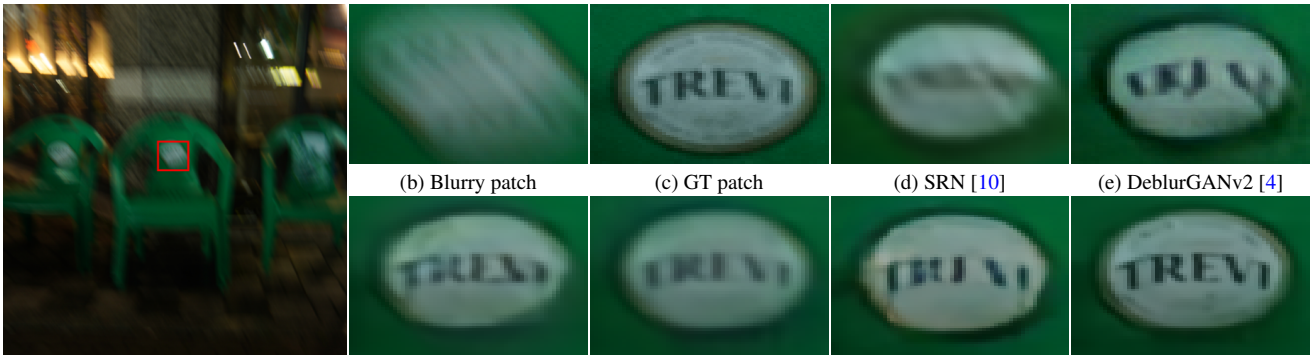
(a) Blurry input (b) Blurry patch (c) GT patch (d) MPRNet [15] (e) Restormer [14] (f) MPRNet-L [3] (g) Restormer-L [3] (h) Stripformer [11] (i) MRLPFNet

Figure 9. Example from the HIDE dataset [9]. The methods [14, 3] do not effectively remove the blur effects as shown in (e)-(f). The details in (d) and (g)-(h) obtained by [15, 3, 11] are over-smoothed. In contrast, the proposed method generates a clearer image with better structural details as shown in (i).



(a) Blurry input (b) Blurry patch (c) GT patch (d) SRN [10] (e) DeblurGANv2 [4] (f) MIMO-UNet+ [2] (g) DeepRFT+ [6] (h) Stripformer [11] (i) MRLPFNet

Figure 10. Example from the RealBlur dataset [8]. The evaluated methods [10, 4, 2, 6, 11] are less effective in generating clear results as shown in (d)-(h). In contrast, the proposed approach is able to restore a clear result (i) from the blurry input, where the license plate numbers are recovered well.

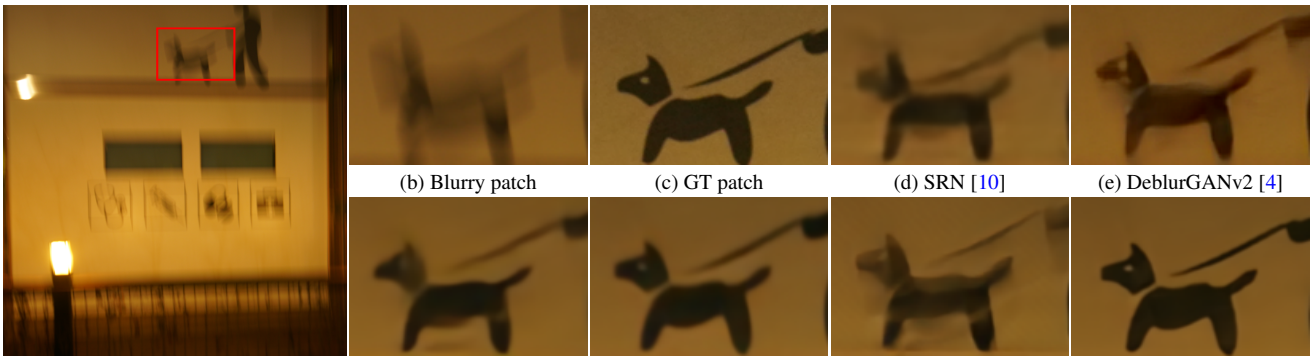


(a) Blurry input (b) Blurry patch (c) GT patch (d) SRN [10] (e) DeblurGANv2 [4]
(f) MIMO-UNet+ [2] (g) DeepRFT+ [6] (h) Stripformer [11] (i) MRLPFNet

Figure 11. Example from the RealBlur dataset [8]. Compared to the results in (d)-(h), our approach is more effective in obtaining a better-deblurred image with clearer characters as shown in (i).

(a) Blurry input (b) Blurry patch (c) GT patch (d) SRN [10] (e) DeblurGANv2 [4]
(f) MIMO-UNet+ [2] (g) DeepRFT+ [6] (h) Stripformer [11] (i) MRLPFNet

Figure 12. Example from the RealBlur dataset [8]. The results by [10, 4] in (d)-(e) contain significant artifacts. The methods [2, 6, 11] oversmooth fine-scale structures in (f)-(h). Compared to existing methods, the proposed approach can effectively preserve finer details in (i), where clearer license plate numbers are recovered.



(a) Blurry input (b) Blurry patch (c) GT patch (d) SRN [10] (e) DeblurGANv2 [4]
(f) MIMO-UNet+ [2] (g) DeepRFT+ [6] (h) Stripformer [11] (i) MRLPFNet

Figure 13. Example from the RealBlur dataset [8]. Compared to the results in (d)-(h), the proposed method recovers a much clearer image as shown in (i).

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, 2022. 3

[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 *ICCV*, pages 4641–4650, 2021. 3, 4, 5, 6

[3] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Revisiting global statistics aggregation for improving image restoration. In *ECCV*, 2022. 3, 4, 5

[4] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, pages 8877–8886, 2019. 5, 6

[5] 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. In *ICCV*, pages 10012–10022, 2021. 3

[6] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring. *CoRR*, abs/2111.11745, 2021. 3, 4, 5, 6

[7] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 257–265, 2017. 3, 4

[8] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, pages 184–201, 2020. 5, 6

[9] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, pages 5571–5580, 2019. 4, 5

[10] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018. 5, 6

[11] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *ECCV*, 2022. 3, 4, 5, 6

[12] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. MAXIM: multi-axis MLP for image processing. In *CVPR*, pages 5759–5770, 2022. 3, 4

[13] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, pages 17662–17672, 2022. 3, 4

[14] 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 *CVPR*, pages 5718–5729, 2022. 3, 4, 5

[15] Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, pages 14821–14831, 2021. 3, 4, 5