Efficient Concertormer for Image Deblurring and Beyond

Supplementary Material

In this supplemental material, we first provide real noisy image denoising results and discuss limitations. Then we report training details of tasks other than single-image motion deblurring. Finally, we provide more visual comparisons.

8. Evaluations on Real Noisy Image Denoising

We further evaluate our method on the real noisy image denoising on the SIDD dataset. Table 8 shows that the proposed methods, i.e., *Concertormer* and *Concertormer*†[2], achieve comparable performance against state-of-the-art ones.

9. Limitation Analysis

We have demonstrated the efficiency of *Concertormer* in the main paper. Although we propose a building block that can be applied to existing restoration models to solve various image restoration tasks, the backbone restoration model still requires a careful design for better performance improvement when using the proposed *Concertormer*. For example, the improvement of the proposed method on image denoising is marginal as shown in Table 8.

Table 8. Real noisy image denoising. * denotes methods using additional training data. *Concertormer*† has more blocks in the latent layer.

Dataset	Metrics	DnCNN [91]	BM3D [15]	CBDNet* [23]	RIDNet* [3]	AINDNet* [30]	VDN [82]	SADNet* [7]	DANet+* [83]	CycleISP* [84]	MIRNet [85]	DeamNet* [56]	MPRNet [86]	DAGL [49]	Uformer [71]	Restormer [87]	NAFNet [12]	*Concertormer*	*Concertormer*†
SIDD [1]	PSNR SSIM	23.66 0.583	25.65 0.685	30.78 0.801	38.71 0.951	38.95 0.952	39.28 0.956	39.46 0.957	39.47 0.957	39.52 0.957	39.72 0.959	39.35 0.955	39.71 0.958	38.94 0.953	39.77 0.959	40.02 0.960	40.30 0.962	40.28 0.962	40.33 0.962

10. Evaluations on Single-Image Super-Resolution

Although we do not design a dedicated model for super-resolution, our method yields competitive results compared to the state-of-the-art works. We replace the W-MSA and SW-MSA modules in the SwinIR with Concerto SA (Fig. 3) for Table 9.

Table 9. Single-image super-resolution.

Method	scale	SET5		SET14		B100		URBAN100		MANGA109		Avg.	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR [39]	×2	38.11	0.960	33.92	0.920	32.32	0.901	32.93	0.935	39.10	0.977	35.28	0.939
NLSN [48]	×2	38.34	0.962	34.08	0.923	32.43	0.903	33.42	0.939	39.59	0.979	35.57	0.941
ENLCN [73]	×2	38.37	0.962	34.17	0.923	32.49	0.903	33.56	0.940	39.64	0.979	35.65	0.941
SwinIR [38]	×2	38.35	0.962	34.14	0.923	32.44	0.903	33.40	0.939	39.60	0.979	35.59	0.941
CSA	×2	38.33	0.962	34.29	0.924	32.45	0.903	33.43	0.940	39.65	0.979	35.63	0.942
EDSR [39]	×4	32.46	0.897	28.80	0.788	27.71	0.742	26.64	0.803	31.02	0.915	29.33	0.829
NLSN [48]	×4	32.59	0.900	28.87	0.789	27.78	0.744	26.96	0.811	31.27	0.918	29.49	0.832
ENLCN [73]	×4	32.67	0.900	28.94	0.789	27.82	0.745	27.12	0.814	31.33	0.919	29.58	0.833
SwinIR [38]	×4	32.72	0.902	28.94	0.791	27.83	0.746	27.07	0.816	31.67	0.923	29.65	0.836
CSA	×4	32.74	0.902	28.96	0.791	27.82	0.746	26.92	0.811	31.49	0.922	29.59	0.834

11. Running Time Analysis

In this section, we compare the deblurring methods of self-attention on HIDE [61]. As shown in Table 10, our methods generate the highest scores with relatively short running times. This table demonstrates the efficiency of our Concerto self-attention. The running times are calculated on one RTX 4080 Super GPU with one $256 \times 256 \times 3$ random input image.

[2] The architecture of *Concertormer* is detailed in Section 4.3, while for *Concertormer*†, the configuration of $L_1 - L_7$ is $[2, 4, 8, 16, 8, 4, 4]$.

Table 10. Running time analysis. We compare our model to self-attention-based deblurring methods. The HIDE [61] dataset is used for metrics. * denotes the model without Test-time Local Converter.

	Uformer [71]	Stripformer[67]	Restormer[87]	Restormer-local[14]	GRL[37]	FFTformer[32]	*Concertormer-lite**	*Concertormer**
PSNR	30.90	31.03	31.22	31.49	31.65	31.62	31.67	31.96
SSIM	0.953	0.940	0.942	0.945	0.947	0.946	0.947	0.950
FLOPs (G)	90.16	177.43	140	155.13	1,289	139.09	116.79	220.2
Param (M)	50.8	19.7	26.1	26.1	20.2	16.6	28.9	50.5
Time (ms)	22.47	25.67	57.60	183.28	344.60	103.26	45.03	83.46

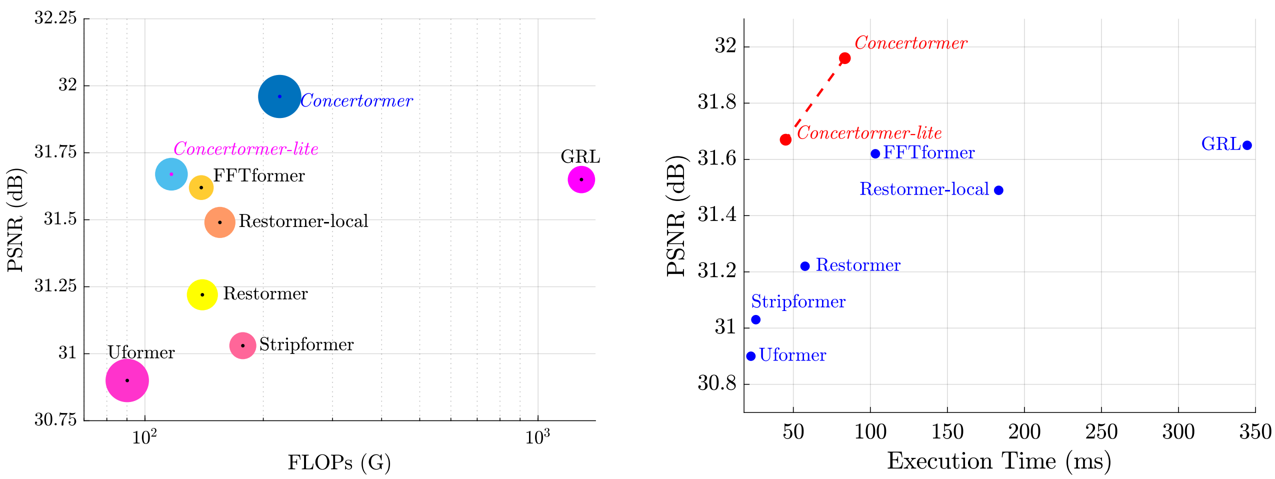


Figure 9. Visualization of model size and execution time.

12. Other Training Details

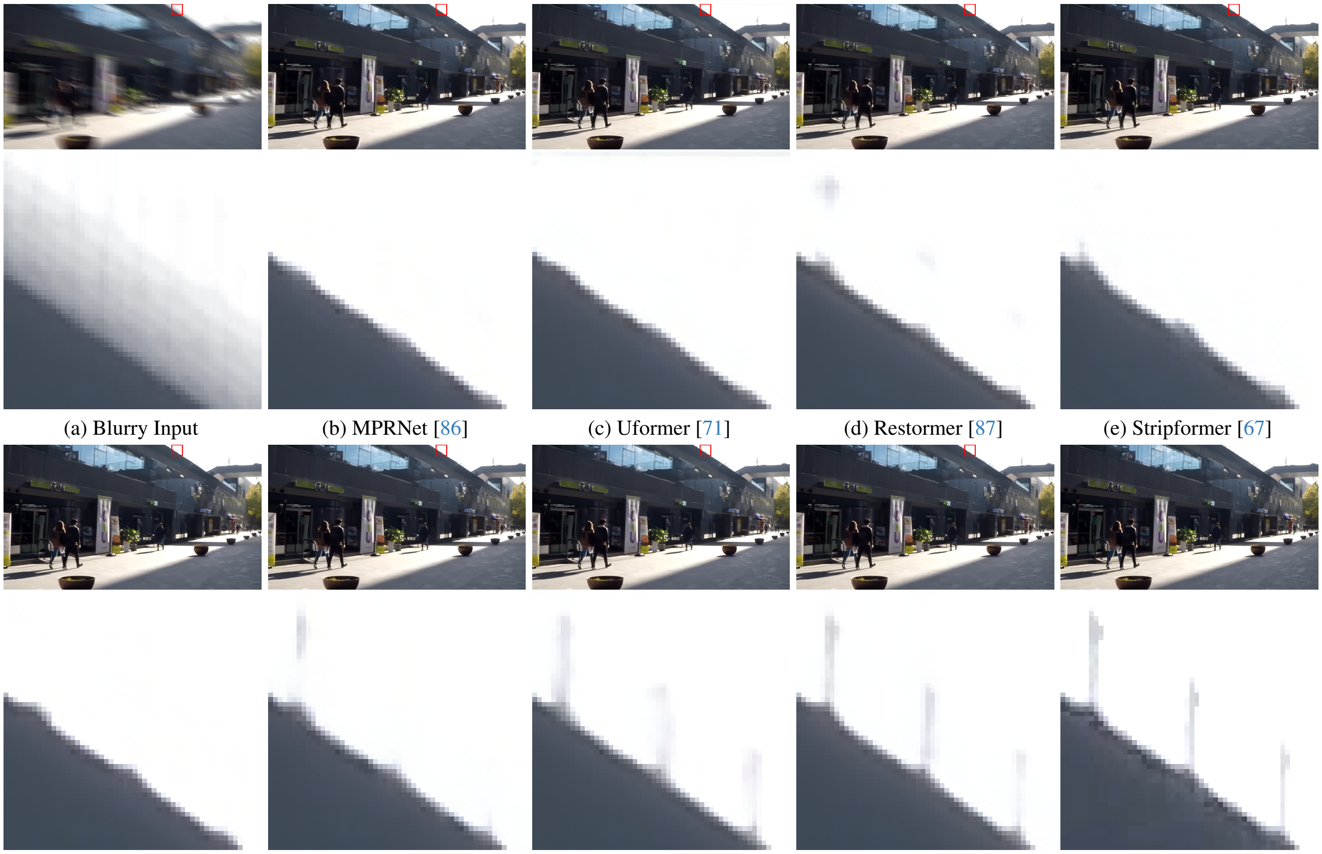
For real noisy image denoising on the SIDD dataset, we train the model for 400,000 iterations, following [12], as additional iterations do not yield further improvements. Since both the training and testing data consist of 256×256 pixel images, we do not employ progressive training; instead, we train the model exclusively on 256×256 patches.
For the deraining task, however, we adopt a progressive training strategy. The model is trained with 192×192 patches for 100,000 iterations, followed by 256×256 patches for 200,000 iterations, 320×320 patches for 100,000 iterations, and an additional 10,000 iterations using 128×128 patches.
As for REDS, we train the model as described in Section 5.1 with the configuration as Section 4.3.

13. More Visual Comparisons

(a) Blurry Input (b) MPRNet [86] (c) Uformer [71] (d) Restormer [87] (e) Stripformer [67]

(f) GRL [37] (g) FFTformer [32] (h) DiffIR [74] (i) *Concertormer* (j) GT

Figure 10. Visual comparisons of single-image motion deblurring on GoPro [50].



(a) Blurry Input (b) MPRNet [86] (c) Uformer [71] (d) Restormer [87] (e) Stripformer [67]

(f) GRL [37] (g) FFTformer [32] (h) DiffIR [74] (i) *Concertormer* (j) GT

Figure 11. Visual comparisons of single-image motion deblurring on GoPro [50].

(a) Blurry Input (b) MPRNet [86] (c) Uformer [71] (d) Restormer [87] (e) Stripformer [67]

(f) GRL [37] (g) FFTformer [32] (h) DiffIR [74] (i) *Concertormer* (j) GT

Figure 12. Visual comparisons of single-image motion deblurring on HIDE [61].

(a) Blurry Input (b) MPRNet [86] (c) Uformer [71] (d) Restormer [87] (e) Stripformer [67]

(f) GRL [37] (g) FFTformer [32] (h) DiffIR [74] (i) *Concertormer* (j) GT

Figure 13. Visual comparisons of single-image motion deblurring on HIDE [61].

(a) Blurry Input (b) DeblurGAN-v2 [33] (c) MAXIM-3S [69] (d) DeepRFT+ [46] (e) Stripformer [67]

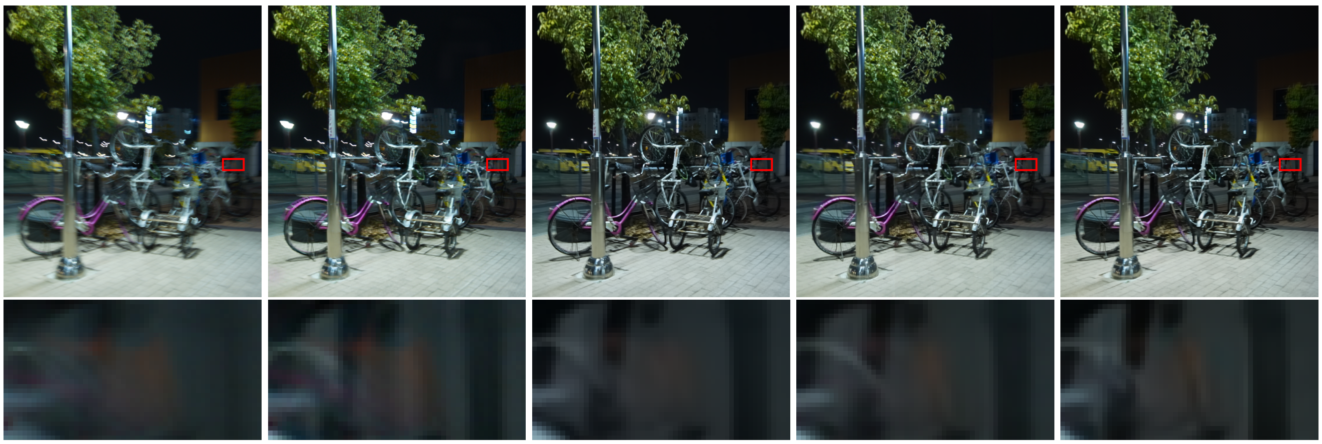
(f) FMIMO-UNet+ [47] (g) FFTformer [32] (h) GRL [37] (i) *Concertormer* (j) GT

Figure 14. Visual comparisons of single-image motion deblurring on REALBLUR_J [58].

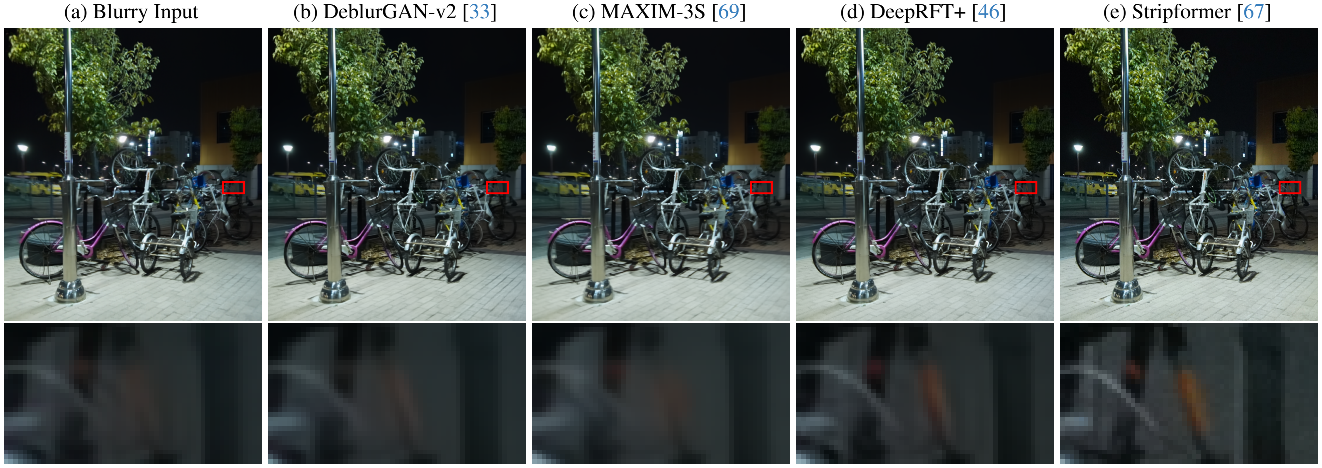
(a) Blurry Input (b) DeblurGAN-v2 [33] (c) MAXIM-3S [69] (d) DeepRFT+ [46] (e) Stripformer [67]

(f) FMIMO-UNet+ [47] (g) FFTformer [32] (h) GRL [37] (i) *Concertormer* (j) GT

Figure 15. Visual comparisons of single-image motion deblurring on REALBLUR_J [58].

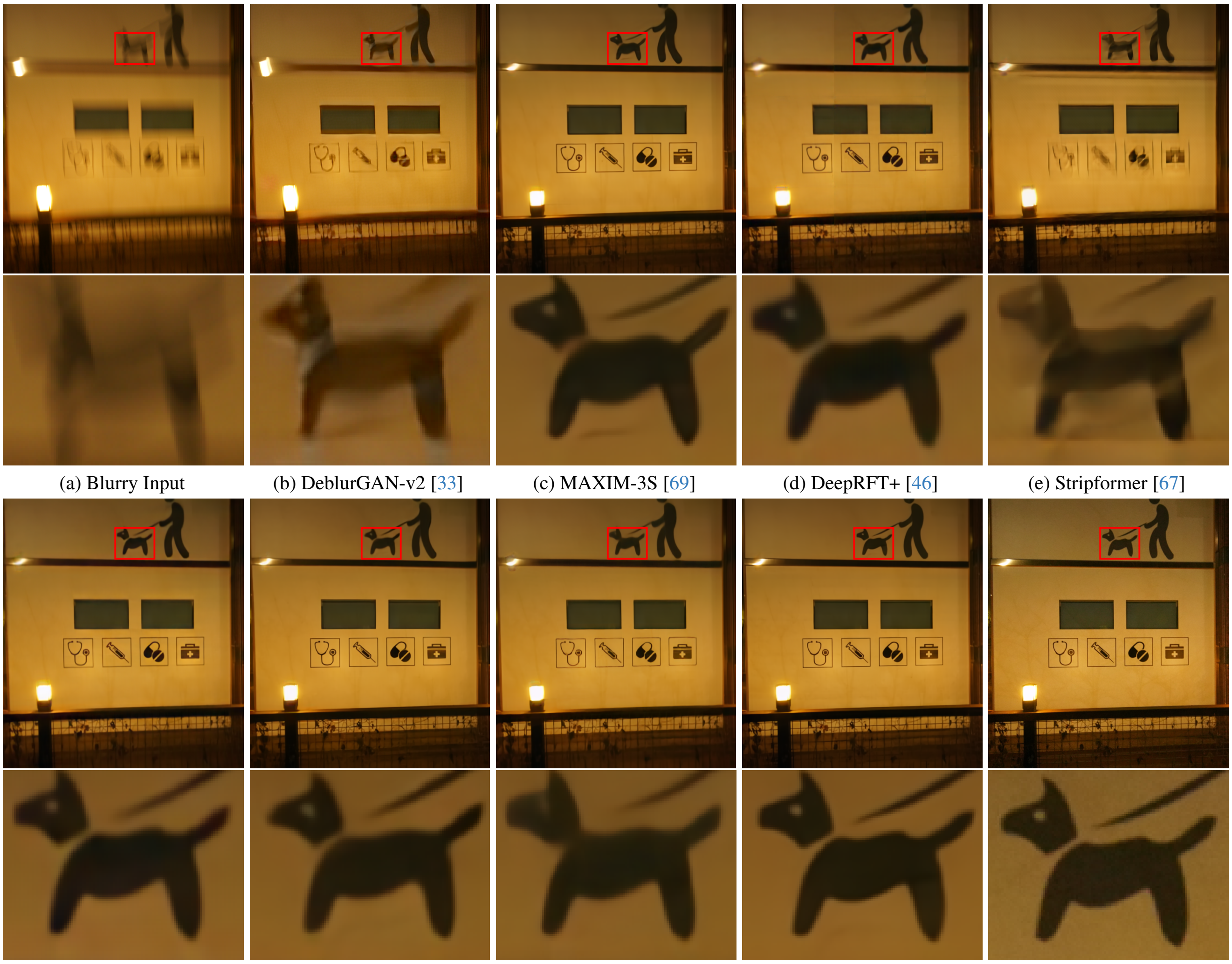


(a) Blurry Input (b) DeblurGAN-v2 [33] (c) MAXIM-3S [69] (d) DeepRFT+ [46] (e) Stripformer [67]



(f) FMIMO-UNet+ [47] (g) FFTformer [32] (h) GRL [37] (i) *Concertormer* (j) GT

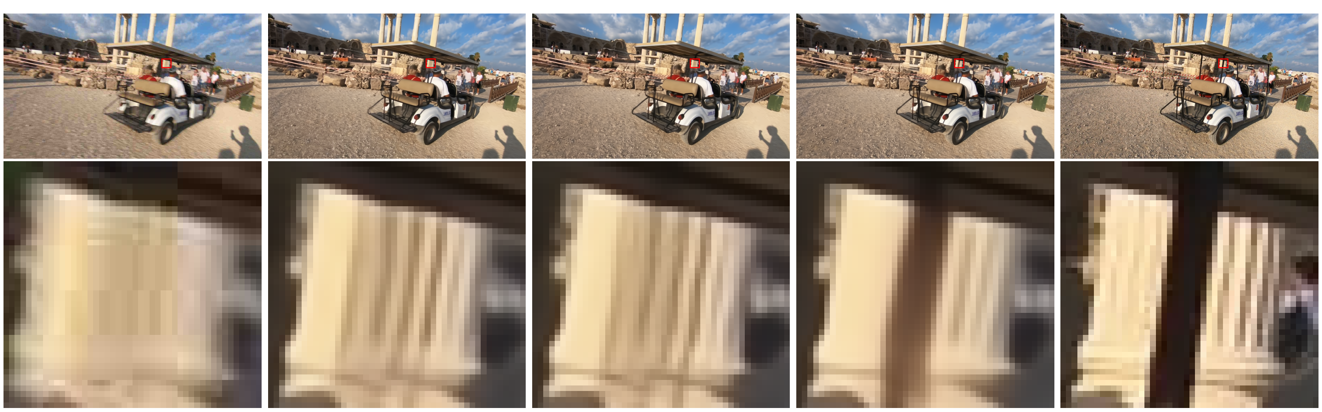
Figure 16. Visual comparisons of single-image motion deblurring on REALBLUR_J [58].



(a) Blurry Input (b) DeblurGAN-v2 [33] (c) MAXIM-3S [69] (d) DeepRFT+ [46] (e) Stripformer [67]

(f) FMIMO-UNet+ [47] (g) FFTformer [32] (h) GRL [37] (i) *Concertormer* (j) GT

Figure 17. Visual comparisons of single-image motion deblurring on REALBLUR_J [58].

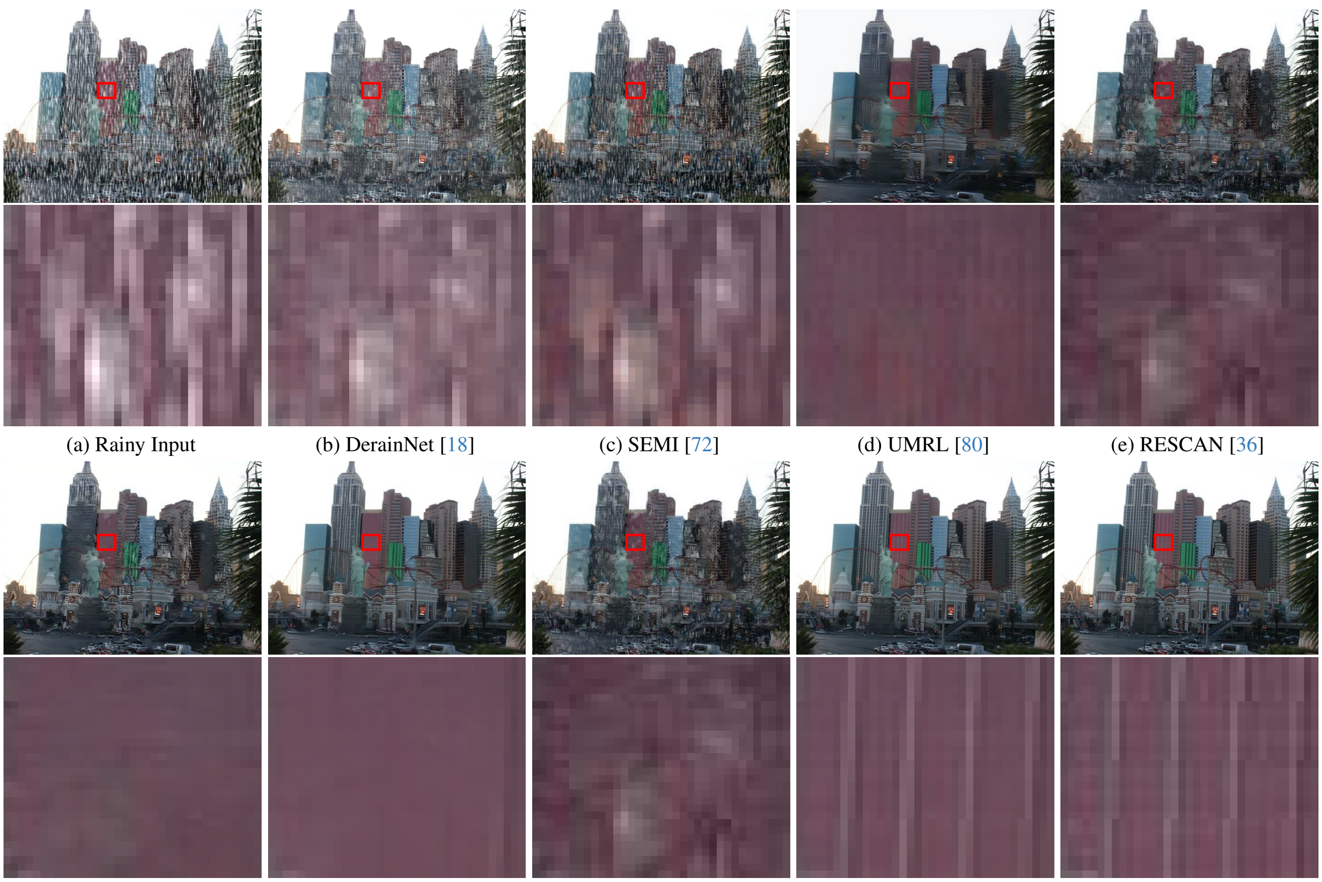


(a) Blurry Input (b) HINet [11] (c) NAFNet [12] (d) *Concertormer* (e) GT

Figure 18. Visual comparisons of motion deblurring with JPEG artifacts on REDS-val-300 [51].

(a) Blurry Input (b) HINet [11] (c) NAFNet [12] (d) *Concertormer* (e) GT

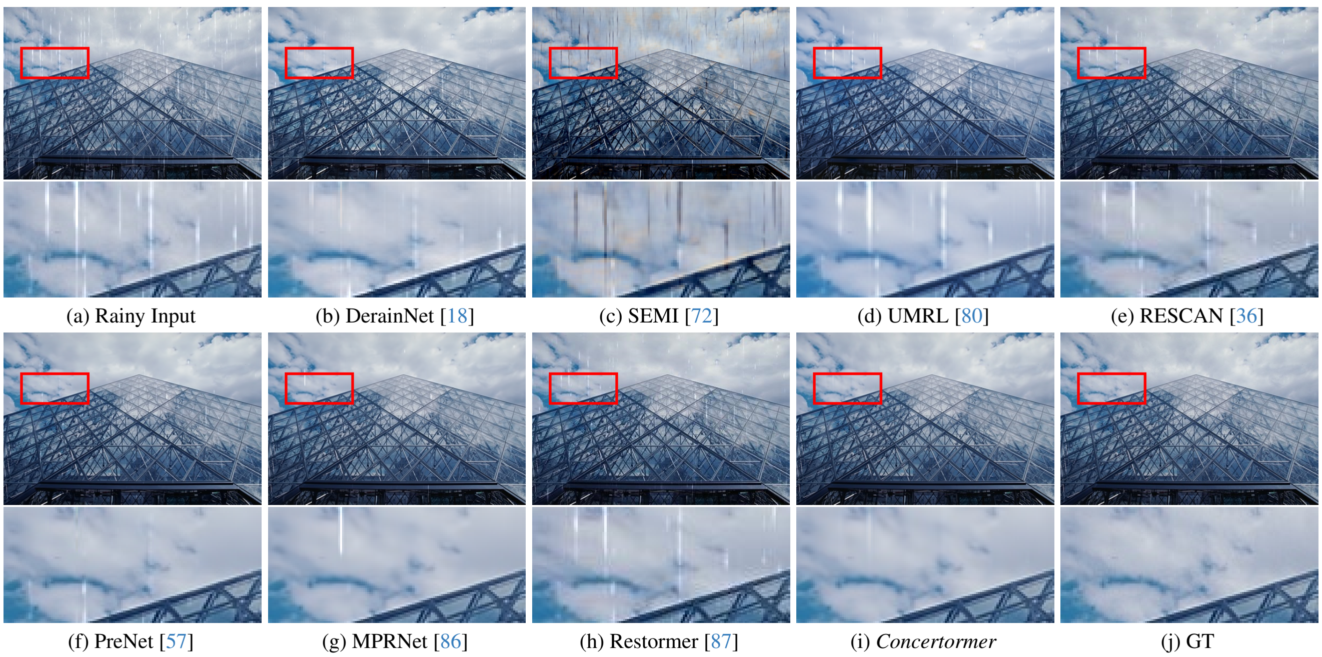
Figure 19. Visual comparisons of motion deblurring with JPEG artifacts on REDS-val-300 [51].



(a) Rainy Input (b) DerainNet [18] (c) SEMI [72] (d) UMRL [80] (e) RESCAN [36]

(f) PreNet [57] (g) MPRNet [86] (h) Restormer [87] (i) *Concertormer* (j) GT

Figure 20. Visual comparisons of deraining on TEST100 [89].



(a) Rainy Input (b) DerainNet [18] (c) SEMI [72] (d) UMRL [80] (e) RESCAN [36]

(f) PreNet [57] (g) MPRNet [86] (h) Restormer [87] (i) *Concertormer* (j) GT

Figure 21. Visual comparisons of deraining on RAIN100L [79].