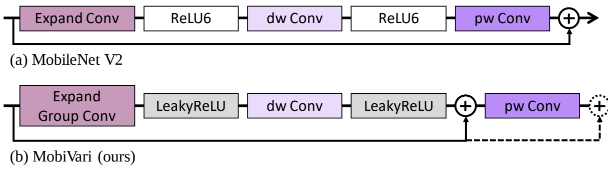
Reciprocal Attention Mixing Transformer for Lightweight Image Restoration – Supplementary Materials

Haram Choi[1]* Cheolwoong Na[2] Jihyeon Oh[2] Seungjae Lee[2] Jinseop Kim[2] Subeen Choe[2]
Jeongmin Lee[3] Taehoon Kim[4] Jihoon Yang[2]†
[1]RippleAI [2]Machine Learning Research Lab., Sogang University [3]LG Innotek [4]LG AI Research

A. Supplementary Discussions and Ablation Studies

A.1. MobiVari (MobileNet Variants) and Reconstruction Module

We revisit MobileNet V2 architectures [29] to incorporate simple and efficient CNN structures into our components. Fig. A illustrates a comparison between the MobileNet and our modified version. We replace the ReLU6 non-linearity [29] with LeakyReLU [26] to preserve subtle gradients that ReLU6 cannot capture [26]. Empirical evidence in Tab. 4c of the main paper shows that this change is the most stable. The 3 × 3 depth-wise (*dw*) and 1 × 1 point-wise (*pw*) convolutions in MobileNets are residually connected [13] with the input feature. However, if the channels produced by *pw* convolutions differ from input channels, the skip connection for *pw* convolutions is ignored. Furthermore, because the first 1 × 1 convolution expanding channels in MobileNet V2 requires many parameters and computations, it is not suitable for our lightweight design. Therefore, we substitute it with group convolution [7], where the group size and expansion ratio are set to 4 and 1.2, respectively, by default. Our MobiVari is applied to attention mixing layers of D-RAMiT and H-RAMi, a downsizing layer, a bottleneck, and the reconstruction module.

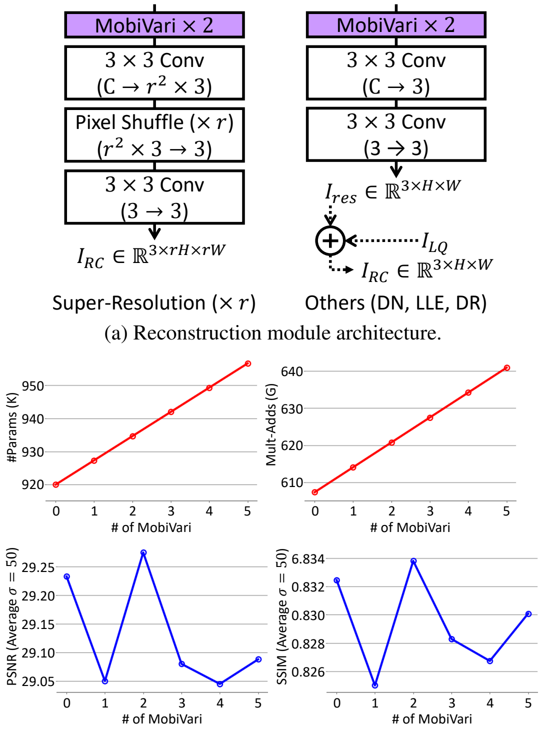


(a) MobileNet V2

(b) MobiVari (ours)

Figure A. Comparison of MobileNets V2 and our corresponding variants, MobiVari.

Fig. Ba depicts the reconstruction module (a final layer). The basic structure follows the reconstruction module of NGswin [6]. The only difference is that we place two MobiVari layers before the default version to balance the trade-off between performance and efficiency (See Fig. Bb). This module slightly varies depending on tasks. For super-resolution, a pixel-shuffler [31] is employed to upscale the feature maps by r times. However, since other tasks (denoising, low-light enhancement, and deraining) do not require this process, the pixel-shuffler is discarded. The symbols and numbers in parentheses indicate changes of channels. The operation $I_{res} + I_{LQ}$ follows convention [20, 39].



(a) Reconstruction module architecture.

(b) Ablation study on the number of MobiVari layers at the reconstruction module. The metrics are evaluated on color denoising task using $\sigma = 50$. PSNR and SSIM average the scores on four benchmark datasets.

Figure B. Reconstruction module.

*This work has been done during Master's course in Sogang University.
†Corresponding author.

(a) Attribute comparisons. The text in **bold** indicates the key differences. “LN” represents whether the position of layer-norm [2] is before (Pre) or after (Post) the self-attention and feed-forward network.

Method	SPSA & CHSA		Existing Elements Employed				Solving Problems
	Operating	**Importance on**	Window Shift	Positional Encoding	Self-Attention	LN	
DaViT [8]	**Alternatively**	**Both equally**	No use	Convolution [17]	Scaled dot-product [34]	Pre [9]	High-level vision
D-RAMiT (ours)	**In parallel**	**SPSA more**	Cyclic [23]	Relative Position Bias [23, 30]	Scaled cosine [24]	Post [24]	Low-level vision

(b) Ablation study on DaViT (Mult-Adds / #Params / Average PSNR).

Method	SR ×2	SR ×4	CDN $\sigma = 50$	LLE	DR
DaViT-*full* [8]	167.0G / 983K / 35.064	43.0G / 1,003K / 29.088	635.2G / 977K / 28.785	635.2G / 977K / 20.965	635.2G / 977K / 27.910
DaViT-*core* [8]	163.2G / 966K / 35.172	42.08G / 987K / 29.268	620.8G / 961K / 29.108	620.8G / 961K / 26.000	620.8G / 961K / 29.630
RAMiT (ours)	**163.4G / 940K / 35.324**	**42.13G / 961K / 29.374**	**620.8G / 935K / 29.275**	**621.6G / 935K / 26.435**	**620.8G / 935K / 30.065**

Table A. Comparisons of RAMiT and DaViT [8].

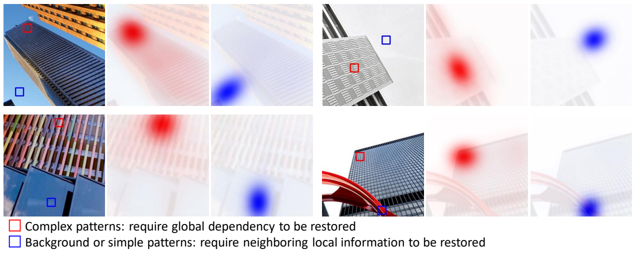

□ Complex patterns: require global dependency to be restored
□ Background or simple patterns: require neighboring local information to be restored

Figure C. The importance of capturing both local and global context for restoring different parts.

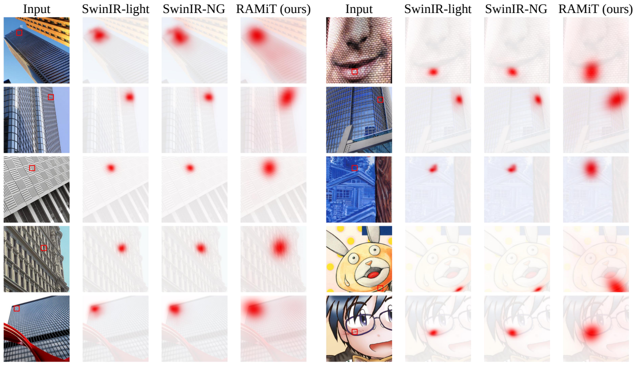


Figure D. Local Attribution Map (LAM) [12] comparison. The depth of the red areas indicates the extent to which the regions contribute to recovering a red box of an input.

A.2. Bi-dimensional Self-Attention

Regarding the importance of capturing both local and global context, we present Fig. C. In this figure, while complex patterns in the image require global context to recover, background or simple patterns require only neighboring local information.

Similar to our bi-dimensional self-attention of the proposed D-RAMiT blocks, DaViT [8] also developed a Transformer using both spatial self-attention (SPSA) and channel self-attention (CHSA). Tab. Aa summarizes the attributes of DaViT and D-RAMiT. A core difference is that DaViT “alternatively” places SPSA and CHSA, while D-RAMiT operates them “in parallel”. As discussed in Sec. 3.2 of the main text, our architecture can boost (depending on tasks) both SPSA and CHSA through the reciprocal helper, which DaViT fundamentally cannot utilize. Another crucial distinction is related to “which self-attention module is given more importance”. While D-RAMiT assigns more multi-heads on SPSA, DaViT makes the number of both modules identical. We hypothesize that DaViT’s simple approach is unsuitable for lightweight image restoration because although CHSA can capture global dependency, its performance is significantly impaired under parameter constraints, as observed in Tab. 4a of our main body. Therefore, more weights on SPSA can be more useful for constructing an effective lightweight RAMiT. Other differences are summarized in the table.

To further demonstrate our superiority over the simple bi-dimensional approach of DaViT, we constructed two versions in Tab. Ab. The first version, DaViT-*full*, replaced the D-RAMiT blocks’ elements in Tab. Aa with those of DaViT. The second version, DaViT-*core*, changed only the core designs (*i.e.*, SPSA & CHSA “Operating” and “Importance on”) from ours to those of DaViT, while the parts of the “Existing Elements Employed” column remained as our settings. The other elements not mentioned in the table followed our default settings for a fair comparison, including the shallow module, MobiVari, the downsizing layers, the bottleneck layer, H-RAMi layer, the reconstruction module, and the hyper-parameters of Tab. H (except that `chsa_head_ratio` is no longer needed). The results show that RAMiT outperformed both DaViT versions while having fewer parameters and almost the same Mult-Adds. It is demonstrated that our meticulous composition of SPSA and CHSA can make a significant difference for multiple lightweight image restoration tasks.

A.3. LAM Comparisons with Other Models

SwinIR-light (ICCVW21) [20] is the first successful attempt applying window self-attention (WSA) to the image restoration tasks. Most recently, SwinIR-NG (CVPR23) [6] defined an N-Gram context method enlarging the regions viewed for recovering distorted pixels, to solve the limited "local" receptive field problem of SwinIR-light. However, SwinIR-NG failed to capture "global context", while our RAMiT successfully exploit the "global receptive field" maintaining WSA approach, which is clarified by LAM [12] results in Fig. D. Even if SwinIR-NG tends to utilize the slightly expanded receptive field when compared to SwinIR-light, the gradients of SwinIR-NG that actually contribute to reconstruct a small red box are limited within "local areas". By contrast, our RAMiT can convey the gradients to "global regions", which improves low-level vision performances with fewer computational costs than SwinIR-NG (reference Tab. 2 of the main paper).

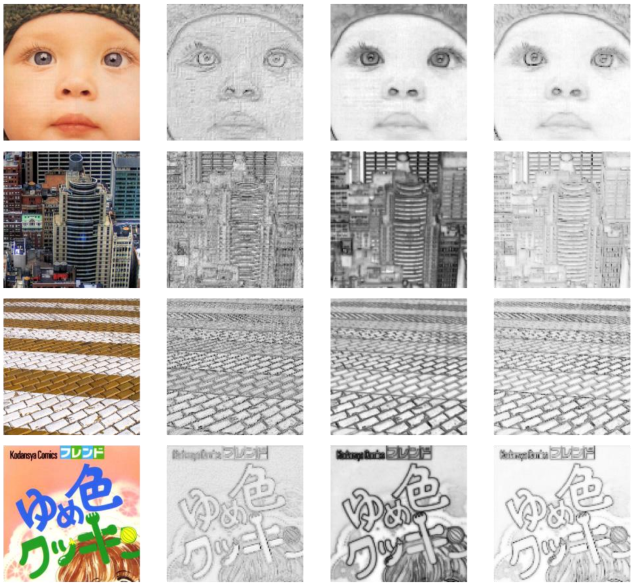
This ability results from adoption of channel self-attention. According to prior work, Squeeze-and-Excitation networks [15], the channel-attention can effectively embed the "global feature responses". RCAN [43] delivered an insight that channel-wise attention would be good at modeling "global spatial dependency" for low-level vision tasks. Afterwards, Restormer [39] applied this mechanism to self-attention without squeeze operations, thereby preserving abundant spatial information, which enabled the image restoration networks to more effectively capture the "global interdependencies" in a whole image. Exploiting such advantages of channel self-attention and the effective WSA, RAMiT can yield meaningfully larger receptive fields than the "pure local-attention" of the SwinIR family. Therefore, our work can be considered an enhanced version of the N-Gram context [6], which extends the "local" N-Gram approach to a "Global-Gram" method.

A.4. Reciprocal Helper

Task	Mult-Adds (G)	PSNR
Color Denoising	620.8 / 621.6	**29.275** / 29.253
Grayscale Denoising	618.5 / 619.3	**27.143** / 27.100
Deraining	620.8 / 621.6	**30.065** / 29.960

Table B. Ablation study on the proposed Reciprocal Helper for denoising and deraining (*w/o* / *w/*).

As proved in Tab. 4b of the main content, our Reciprocal Helper[1] can boost ×2, ×3, ×4 super-resolution and low-light enhancement tasks. However, Tab. B shows that this mechanism is unable to improve the performances of denoising and deraining. We interpret this limitation in terms of properties of the tasks. Degradation used for the super-resolution and low-light enhancement inputs relatively has regularity and therefore may be easy to be globally encoded. This property may make our reciprocal helper useful for the parallel process of local and global self-attention. On the other hand, when dealing with denoising or deraining low-quality inputs, the network is required to erase somethings that obscure the high-quality objects or background. Since it is ill-posed to globally encode these (randomly) disorganized obstructions with a small network capacity, the global embeddings produced by the channel attention may confuse the spatial attention module of the next blocks. However, if the parallel process lacks the reciprocal helper, the MobiVari mixing layers alone can still resolve this issue well. Admitting this limitation, we will conduct more sophisticated future work on other helper algorithms that can improve universal tasks. Nevertheless, our core ideas, *i.e.,* dimensional and hierarchical reciprocal self-attention methods, have been already demonstrated to be effective and efficient enough to achieve new state-of-the-art lightweight denoising and deraining.



(a) High-Quality Image (ground-truth) (b) Stage4 Output (o_4) (c) H-RAMi output (o_h) (d) $o_4 * o_h$

Figure E. Impacts of H-RAMi. **(a)** A ground-truth high-quality image. **(b)**, **(c)** The feature maps after stage 4 and H-RAMi. **(d)** Element-wise product of (b) and (c). (b), (c), (d) are obtained by max-pooling along channel and standardization.

A.5. Hierarchical Reciprocal Attention Mixing Layer (H-RAMi)

Although H-RAMi may appear similar to the attention banks used in DiVANet [3], there are notable differences. DiVANet uses non-hierarchical attentions for every residual convolution block, increasing computational costs (see Tab. Ca) and failing to learn semantic-level representation. Moreover, the vertical and horizontal squeeze operations

[1] To prevent any confusion, we adopted the term "*Dimensional Reciprocal Attention Mixing Transformer*" (D-RAMiT) to indicate that *every dimension* (spatial and channel) of feature maps is utilized in calculating *self-attention*, and the outcomes are subsequently *mixed* by MobiVari. Consequently, this implies that the reciprocal helper is not a prerequisite to represent dimensional reciprocal attention.

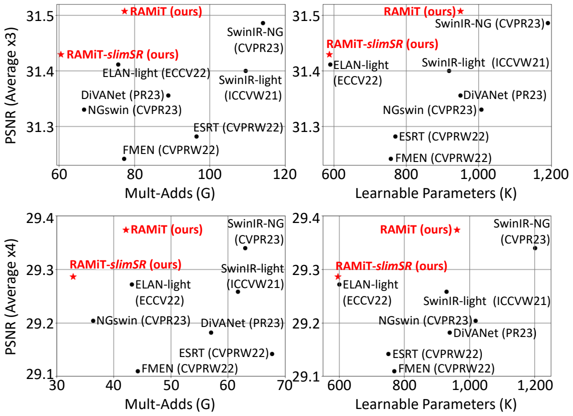


Figure F. Trade-off between efficiency and performance on super-resolution. **(Top)** ×3. **(Bottom)** ×4.

prevent the attention layers from considering full-resolution information. In contrast, our approach reduces time complexity and utilizes semantic-level information by processing compressed feature maps. Furthermore, the inputs to H-RAMi are intermediate attentions from D-RAMiT blocks, which preserve information from both full-resolution spatial and channel self-attentions. We provide additional visual evidences of the benefits in Fig. E. As previously stated in Fig. 4 of the main text, the stage 4 output alone at (b) produces relatively unclear or incorrect edges, which are resolved at (d) by the clearer edges produced by H-RAMi at (c).

A.6. Super-Resolution (SR)

Fig. F illustrate trade-offs between efficiency (Mult-Adds, #Params) and performance (average PSNR) on SR tasks, including our RAMiT-*slimSR* (Tab. Ca) and RAMiT. Our methods deliver the best trade-off among the comparative models.

Smaller Size. Tab. 2 of the main text appears to have an unfair aspect. Some networks have fewer parameters than our RAMiT, such as FMEN (CVPRW22) [10], ESRT (CVPRW22) [25], ELAN-light (ECCV22) [42], and DiVANet (PR23) [3]. Although they require more Mult-Adds than RAMiT, it can be questioned whether our improvement is attributed to the proposed design or the result of having simply more parameters. We address this issue in Tab. Ca. The channel (network dimension) and depths (D-RAMiT blocks in stage 1 to 4) of RAMiT were scaled from 64 and $[6, 4, 4, 6]$ to 48 and $[8, 2, 2, 8]$, respectively. In the bottleneck and H-RAMi, we also changed the group size and expansion ratio of MobiVari from 4 and 1.2 to 1 and 2.0, respectively. The group size and expansion ratio of the other MobiVari layers were retained as the default settings. Consequently, we got a compact network denoted as RAMiT-*slimSR*, which is composed of the fewest learnable parameters and Mult-Adds among the comparative methods. Note that RAMiT-*slimSR* consumes fewer computations than NGswin (CVPR23) [6], which required the fewest Mult-Adds in Tab. 2. RAMiT-*slimSR* still outperformed others, showing that our advancements on super-resolution were attributed to the effectiveness and efficiency of the novel approaches.

(a) Comparison for RAMiT-*slimSR*. The best, second best, and third best results are in red, orange, and blue. PSNR and SSIM scores average the results on the five benchmark test datasets.

	Scale	Method						
		FMEN [10]	ESRT [25]	ELAN-light [42]	DiVANet [3]	NGswin [6]	**RAMiT-*slimSR***	**RAMiT**
Mult-Adds / #Params PSNR / SSIM	×2	172.0G / 748K 35.094 / 0.93794	191.4G / 677K 35.146 / 0.93754	168.4G / 582K 35.258 / 0.93906	189.0G / 902K 35.186 / 0.93838	140.4G / 998K 35.122 / 0.93836	**127.8G / 581K** **35.226 / 0.93880**	**163.4G / 940K** **35.324 / 0.93938**
Mult-Adds / #Params PSNR / SSIM	×3	77.2G / 757K 31.242 / 0.87594	96.4G / 770K 31.282 / 0.87586	75.7G / 590K 31.412 / 0.87868	89.0G / 949K 31.356 / 0.87752	66.6G / 1,007K 31.330 / 0.87778	**60.4G / 588K** **31.430 / 0.87872**	**77.3G / 949K** **31.508 / 0.87972**
Mult-Adds / #Params PSNR / SSIM	×4	44.2G / 769K 29.110 / 0.82376	67.7G / 751K 29.142 / 0.82442	43.2G / 601K 29.272 / 0.82742	57.0G / 939K 29.182 / 0.82570	36.4G / 1,019K 29.204 / 0.82618	**32.9G / 597K** **29.286 / 0.82762**	**42.1G / 961K** **29.374 / 0.82940**

(b) Training dataset size of RAMiT.

Dataset (#Images)	Scale	Set5 [4]		Set14 [40]		BSD100 [27]		Urban100 [16]		Manga109 [28]		Average	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DIV2K (800) DF2K (3,450)	×2	38.16 38.19	0.9612 0.9613	34.00 33.95	0.9213 0.9215	32.33 32.35	0.9015 0.9017	32.81 32.90	0.9346 0.9352	39.32 39.44	0.9783 0.9788	35.324 35.366	0.93938 0.93970
DIV2K (800) DF2K (3,450)	×3	34.63 34.69	0.9290 0.9295	30.60 30.60	0.8467 0.8468	29.25 29.28	0.8093 0.8097	28.76 28.80	0.8646 0.8656	34.30 34.40	0.9490 0.9494	31.508 31.554	0.87972 0.88020
DIV2K (800) DF2K (3,450)	×4	32.56 32.58	0.8992 0.8995	28.83 28.87	0.7873 0.7876	27.71 27.73	0.7418 0.7419	26.60 26.65	0.8017 0.8036	31.17 31.25	0.9170 0.9174	29.374 29.416	0.82940 0.83000

Table C. Ablation study on model size and training dataset for super-resolution.

Method (seed)	Set5 [4]	Set14 [40]	BSD100 [27]	Urban100 [16]	Manga109 [28]
SwinIR-NG (α)	38.17 / 34.64 / 32.44	33.94 / 30.58 / 28.83	32.31 / 29.24 / 27.73	32.78 / 28.75 / 26.61	39.20 / 34.22 / 31.09
RAMiT (α)	38.16 / 34.63 / **32.56**	**34.00 / 30.60 / 28.83**	**32.33 / 29.25** / 27.71	**32.81 / 28.76** / 26.60	**39.32 / 34.30 / 31.17**
RAMiT (β)	**38.18 / 34.65 / 32.54**	**34.02 / 30.62 / 28.86**	**32.33 / 29.25** / 27.72	**32.81 / 28.75 / 26.62**	**39.28 / 34.28 / 31.15**
RAMiT (γ)	**38.18 / 34.64 / 32.48**	**34.00 / 30.60** / 28.80	**32.33 / 29.25** / 27.71	**32.83 / 28.75** / 26.58	**39.28 / 34.29 / 31.11**
RAMiT (δ)	**38.18 / 34.64 / 32.54**	**34.02 / 30.59 / 28.83**	**32.32 / 29.25** / 27.71	**32.79** / 28.70 / 26.57	**39.27 / 34.31 / 31.12**

Table D. Ablation on randomness. PSNR on x2 / x3 / x4. The **bold** face indicates better performance over SwinIR-NG [6].

Training Dataset. As shown in Tab. Cb, we found room for improvement of RAMiT with more training data. In addition to 800 images of DIV2K [1] used by RAMiT for super-resolution in Tab. 2 of the main text, many recent studies utilized 2,650 Flickr2K [32] dataset as well to reinforce their SR networks [5, 10, 20, 42, 44]. Following them, we additionally trained our models on DF2K (DIV2K + Flickr2K) for the enhanced performances. The impacts on all upscaling tasks were observed.

Randomness. To further prove that the improvements are attributed to not randomness (weight initialization, randomly cropped patches, random data augmentation, etc.) but our approach, we have conducted extra SR experiments as shown in Tab. D. RAMiT trained with different random seeds $(\alpha, \beta, \gamma, \delta)$ still outperforms SwinIR-NG. The seed α indicates our default.

Comparison with Large Model. One might question the efficiency of our proposed lightweight method compared to its larger counterpart. To address this concern, we present Tab. E where the SwinIR [20] large model outperforms ours with 12.5 times more parameters than our RAMiT. However, there is a significant difference in the number of frames per second that SwinIR and RAMiT can process. Our lightweight method demonstrates superior processing speed in image restoration tasks on both

Method	#Params	Urban100 (PSNR / FPS)	Manga109 (PSNR / FPS)
SwinIR [20]	11,753K	33.40 / 0.34, 0.94	39.60 / 0.26, 0.71
RAMiT (ours)	940K	32.81 / 1.38, 9.38	39.32 / 1.10, 7.38

Table E. Comparison between large and lightweight models. "FPS" indicates frames per second processed by each method, which means the higher FPS, the faster, *i.e.*, the better. The former of FPS is measured on an NVIDIA TITAN Xp, while the latter on an NVIDIA GeForce RTX 4090.

outdated (TITAN Xp) and recent (RTX 4090) GPU devices, surpassing the SwinIR large model. The result apparently demonstrates that the recent state-of-the-art image restoration models cannot be applied to real-world application despite their enhanced performance. In contrast, our lightweight approach is specially designed to resolve this efficiency-effectiveness trade-off issue, offering a viable solution for practical implementation.

A.7. Low-Light Enhancement (LLE)

Tab. F compares MAXIM (CVPR22) [33] and RAMiT to present the effectiveness and efficiency of our model for the LLE task. MAXIM has shown outstanding results on the general image restoration tasks with a large model size. Surprisingly, our RAMiT outperformed MAXIM in terms of average PSNR scores on the 15 images LOL evaluation dataset [35]. Notably, we achieved this impressive result using only 6.63% parameters of MAXIM. Additionally, RAMiT showed lower variance for the evaluated images than MAXIM, indicating more stable restoration of dark images into brighter ones. The visual results are in Fig. G.

Secondarily, we reported a fair comparison with URetinex-Net (CVPR22) [18] in Tab. Ga. This method requires only 38.6% parameters of RAMiT, which can provoke a concern of unfairness. To handle this issue, the channel (network dimension) and the depths (D-RAMiT blocks in stage 1 to 4) of RAMiT were reduced from 64 to 48 and from $[6, 4, 4, 6]$ to $[4, 2, 2, 4]$, respectively. In the bottleneck and H-RAMi, the group size of MobiVari is changed from 4 to 3. As a result, we obtained a downsized model composed of fewer parameters than URetinex-Net, and called it RAMiT-*slimLLE*. RAMiT-*slimLLE* still outperformed URetinex-Net by PSNR margins of up to 7.16dB, which emphasizes our effectiveness and efficiency.

A.8. Deraining (DR)

Tab. Gb shows our efficiency for deraining task. MPRNet (CVPR21) [38] made advancements on multiple image restoration tasks a few years ago. However, RAMiT can outperform it with 25.7% parameters of MPRNet on a deraining benchmark dataset, such as Test100 [41].

B. Experimental Details

In Common. As explained in Sec. 4.1, we optimized L_1 pixel-loss between I_{RC} and I_{HQ} with the Adam optimizer [19] ($\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 10^{-8}$), where I_{RC} is a reconstructed image and I_{HQ} is a high-quality ground-truth image. Learning rate was initialized as $0.0004 \times 64/\texttt{batch\_size}$. The data augmentation method for LQ and HQ pairs was already specified in the main contents. Before we fed the LQ input images to the network, each input was normalized using `mean` and `std` pre-calculated from the LQ training datasets corresponding to each task. Note that since we used the random (blind) noise levels (σ) for training our denoising networks, we used `mean` and `std` of HQ training datasets for color and grayscale denoising. When computing the training loss, the normalized I_{RC} was de-normalized (opposite process of normalization). For evaluation, an input image I_{LQ} was upsized by symmetric padding to fit the size to a multiplier ($= 32 = 8 \times 2^2$) of the local-window $M (= 8)$ and downsizing number ($= 2^2$) for the hierarchical stages. We implemented all processes using PyTorch and two NVIDIA GeForce RTX 4090 GPUs. The implementation details of RAMiT are in Tab. H.

Super-Resolution. We trained RAMiT for $\times 2$ task from scratch, of which the training epochs were set to 500. For $\times 3$, $\times 4$ tasks, we followed a warm-start strategy [21], where we fine-tuned the final reconstruction module for 50 epochs (warm-start phase) before fine-tuning whole network parameters (whole-finetuning phase) lasting for 250 epochs. In warm-start phase, the network parameters pretrained on $\times 2$ task were loaded to initialize $\times 3$, $\times 4$ networks, except for the reconstruction module. Learning rate was decayed by half at $\{200, 300, 400, 425, 450, 475\}$ and $\{50, 100, 150, 175, 200, 225\}$ epochs for training from scratch ($\times 2$) and whole-finetuning phase ($\times 3$, $\times 4$), respectively. Learning rate of warm-start phase remained as a constant (*i.e.*, $0.0004 \times 64/\texttt{batch\_size}$). We also linearly increased learning rate from 0 to $0.0004 \times 64/\texttt{batch\_size}$ during the first 20 epochs of the training from scratch and whole-finetuning phase (warmup epoch [11]). Each training image was cropped into a patch size of 64×64 with 64 batch size regardless of training from scratch or warm-start strategy. To consistently manage the datapoints per epoch, we repeated each datapoint 80 and 18.551 times for DIV2K and DF2K datasets, which made the number of training images used for an epoch equal to $64,000$.

Others. For color and grayscale denoising, low-light enhancement, and deraining, we adapted the progressive learning [39], where the patch size was initially set to 64×64, and then progressively increased to 96×96 and 128×128 after $\{100, 200\}$ epochs, respectively. The corresponding batch size was $\{64, 32, 16\}$. We decrease learning rate by half at $\{200, 300, 350, 375\}$ epochs. Warmup epoch was the same as super-resolution. The training process

Model	#Params	001	022	023	055	079	111	146	179	493	547	665	669	748	778	780	Mean	Std.
MAXIM [33]	14,100K	20.98	28.68	24.89	18.83	27.16	17.82	23.30	19.65	13.79	15.66	28.34	28.63	29.96	25.02	28.51	23.41	5.11
RAMiT (ours)	935K	20.50	26.20	19.34	18.74	28.18	31.12	25.74	23.61	20.39	18.32	26.67	25.17	28.07	21.76	28.32	24.14	3.93

Table F. Comparison of MAXIM [33] and RAMiT on low-light enhancement. The PSNR (dB) scores on 15 LOL [35] evaluation images are reported. The numbers in the first row indicate the testing file (`.png`) names. Std.: standard-deviation.

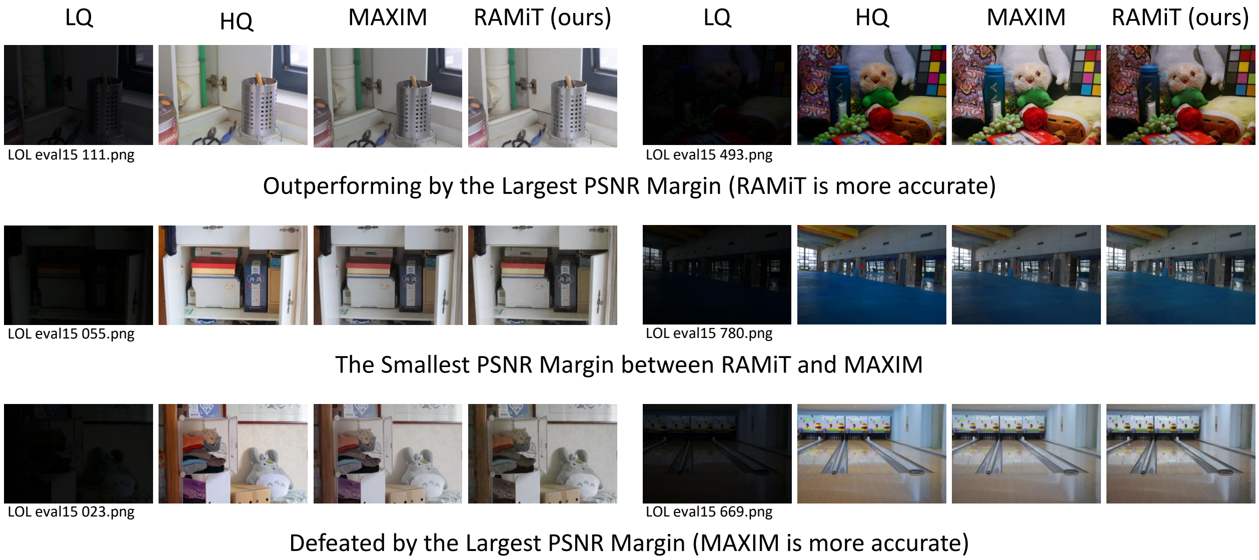

LQ
HQ
MAXIM
RAMiT (ours)
LQ
HQ
MAXIM
RAMiT (ours)
LOL eval15 111.png
LOL eval15 493.png
Outperforming by the Largest PSNR Margin (RAMiT is more accurate)
LOL eval15 055.png
LOL eval15 780.png
The Smallest PSNR Margin between RAMiT and MAXIM
LOL eval15 023.png
LOL eval15 669.png
Defeated by the Largest PSNR Margin (MAXIM is more accurate)

Figure G. Visual comparisons of MAXIM [33] and RAMiT. Despite even fewer parameters, RAMiT can restore the extremely dark images with better or matched accuracy compared to MAXIM. In the bottom row, the cases in which RAMiT is highly defeated by MAXIM are provided as well.

(a) Comparison for LLE.

Method	#Params	LOL [35]		VE-LOL-cap [22]	
		PSNR	SSIM	PSNR	SSIM
URetinex-Net [36]	361K	21.33	0.8348	21.22	0.8593
RAMiT-*slimLLE* (ours)	**358K**	**23.77**	**0.8379**	**28.38**	**0.8835**

(b) Comparison for DR.

Method	#Params	Test100 [41]		Rain100H [37]	
		PSNR	SSIM	PSNR	SSIM
MPRNet [38]	3,637K	30.27	0.8970	30.41	0.8990
RAMiT (ours)	**935K**	**30.44**	**0.9012**	**29.69**	**0.8775**

Table G. Further comparisons for LLE and DR. **(a)** RAMiT-*slimLLE* is still better than URetinex-Net. **(b)** We outperform MPRNet on a benchmark dataset despite much fewer parameters.

Overall Architecture	dim (C)	64
	depths	$[6,4,4,6]$
	num heads	$[4,4,4,4]$
	chsa head ratio (L_{ch}/L)	25%
	window size (M)	8
Feed-Forward Network (FFN)	hidden ratio	2.0
	activation	GELU [14]
MobiVari	exp factor	1.2
	expand groups	4
	activation	LeakyReLU [26]
Dropout	attention map	0.0
	attention project	0.0
	drop path	0.0
Others	optimizer	Adam [19] ($\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 10^{-8}$)
	initialized learning rate	$0.0004 \times 64/$`batch_size`
	learning rate decay	half (see paragraphs below)
	batch size	see paragraphs below
	epoch / total datapoints	500 / 32M (SR), 400 / 10M (Others)

Table H. Implementation details of RAMiT. “depths” and “num heads” count the number of D-RAMiT blocks ($[\mathbb{K}_1, \mathbb{K}_2, \mathbb{K}_3, \mathbb{K}_4]$) and multi-heads ($L$) in stage $1,2,3,4$. Correspondingly, the setting of “chsa head ratio =25%” indicates that (L_{sp}, L_{ch}) is placed as $[(3,1),(3,1),(3,1),(3,1)]$ in each stage.

lasted for 400 epochs. Similar to super-resolution, we repeated each datapoint 3.0, 14.006, and 1.8234 times for denoising, low-light enhancement, and deraining, respectively (about $25,000$ datapoints were used per epoch). While we obtained the synthetic or real-captured low and high quality image pairs of low-light enhancement and deraining from public sources[2], the Additive White Gaussian Noise (`AWGN`) for low quality noisy input images of denoising tasks was generated by the following PyTorch-like code:

```
AWGN = torch.randn(*img_hq.shape)*σ/255
img_lq = img_hq + AWGN,
```

where the random seed was set to 0 for the evaluation process (in training, seed is not given to implement blind de-

[2] LOL and VE-LOL datasets can be found in this `website1` and this `website2`. Deraining Testsets and Rain13K can be publicly downloaded in this `google-drive1` and this `google-drive2`.

noising); `img_lq` and `img_hq` indicate low and high quality images; σ is noise level set to one among $[15, 25, 50]$ for testing or sampled uniformly between $0 \sim 50$ for training.

C. More Visual Comparisons

In the last six pages (P. 9–14 after References) of this document, we provide additional visual comparisons of our RAMiT and other networks. These visual results exhibit the effectiveness of our approach on super-resolution (Figs. H and I), denoising (Figs. J and K), low-light enhancement (Fig. L), and deraining (Fig. M).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 4

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[3] Parichehr Behjati, Pau Rodriguez, Carles Fernández, Isabelle Hupont, Armin Mehri, and Jordi Gonzàlez. Single image super-resolution based on directional variance attention network. *Pattern Recognition*, 133:108997, 2023. 3, 4

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *BMVA press*, 2012. 4

[5] Zheng Chen, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Cross aggregation transformer for image restoration. In *Advances in Neural Information Processing Systems*, 2022. 4

[6] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2071–2081, 2023. 1, 2, 3, 4

[7] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 1

[8] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 74–92. Springer, 2022. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[10] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 853–862, 2022. 4

[11] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5

[12] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9199–9208, 2021. 2, 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[16] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 4

[17] Md Amirul Islam*, Sen Jia*, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations*, 2020. 2

[18] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE transactions on image processing*, 30:2340–2349, 2021. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 6

[20] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 1, 2, 4, 5

[21] Zudi Lin, Prateek Garg, Atmadeep Banerjee, Salma Abdel Magid, Deqing Sun, Yulun Zhang, Luc Van Gool, Donglai Wei, and Hanspeter Pfister. Revisiting rcan: Improved training for image super-resolution. *arXiv preprint arXiv:2201.11279*, 2022. 5

[22] Jiaying Liu, Dejia Xu, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *International Journal of Computer Vision*, 129:1153–1184, 2021. 6

[23] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[24] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al.

Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 2

[25] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 457–466, 2022. 4

[26] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, page 3. Atlanta, Georgia, USA, 2013. 1, 6

[27] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 416–423. IEEE, 2001. 4

[28] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 4

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[30] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, 2018. Association for Computational Linguistics. 2

[31] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 1

[32] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 4

[33] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022. 5, 6

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[35] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018. 5, 6

[36] Wenhui Wu, Jian Weng, Pingping Zhang, Xu Wang, Wenhan Yang, and Jianmin Jiang. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5910, 2022. 6

[37] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1357–1366, 2017. 6

[38] 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. 5, 6

[39] 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*, 2022. 1, 3, 5

[40] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 4

[41] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 30(11):3943–3956, 2019. 5, 6

[42] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*, 2022. 4

[43] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 3

[44] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In *International Conference on Learning Representations*, 2021. 4

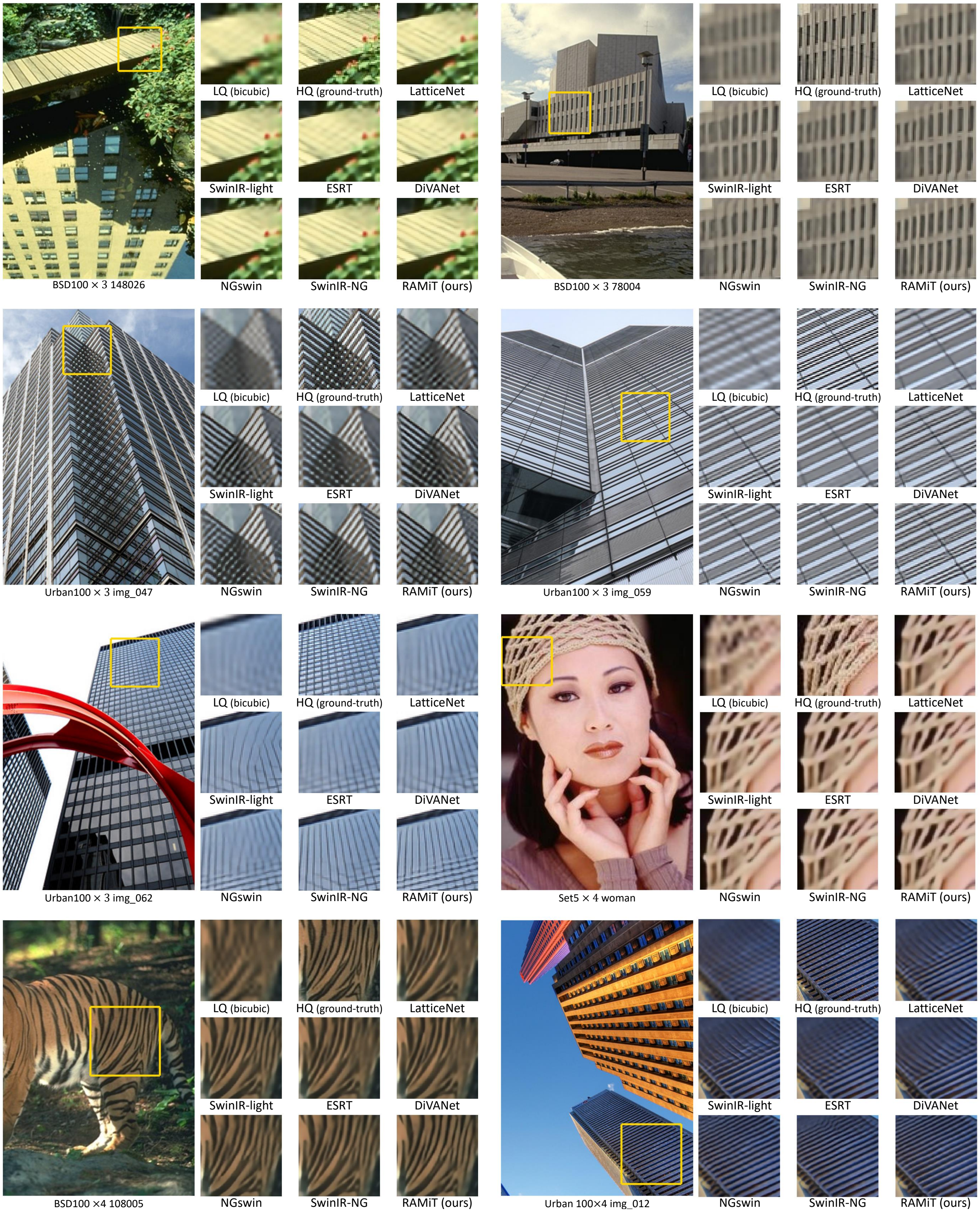


Figure H. Visual comparisons of super-resolution. LQ: Low-Quality input. HQ: High-Quality target.

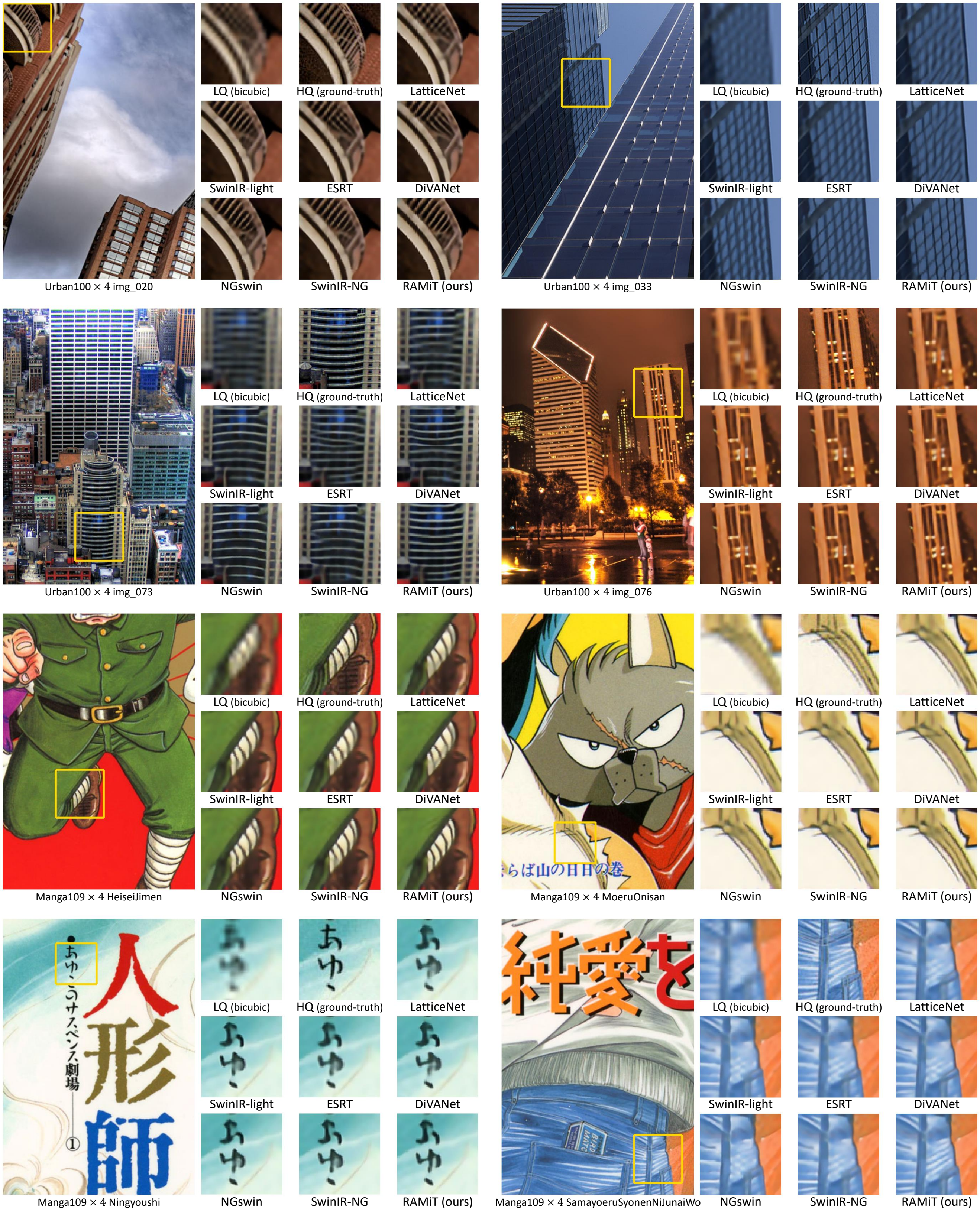


Figure I. Visual comparisons of super-resolution. LQ: Low-Quality input. HQ: High-Quality target.

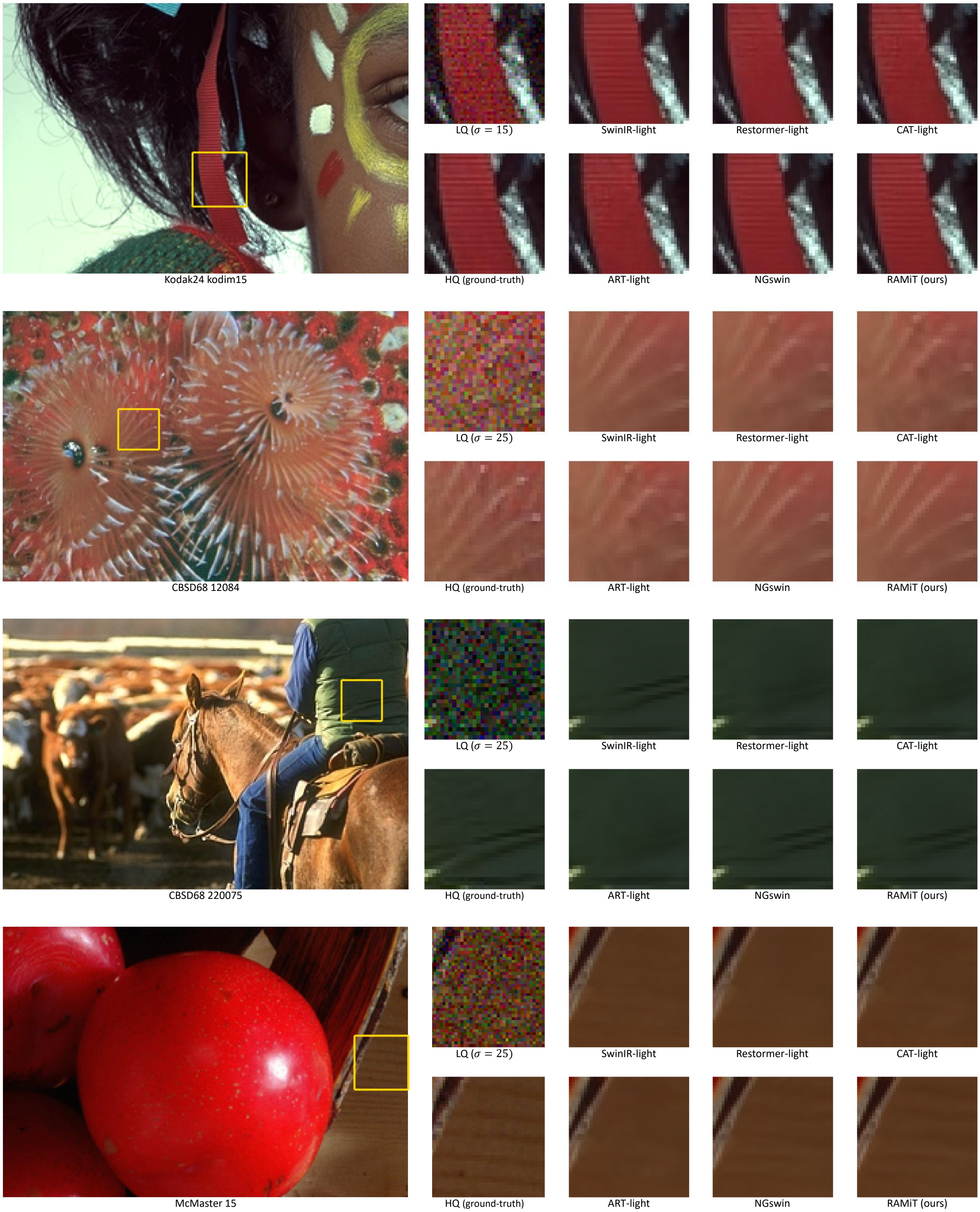


Figure J. Visual comparisons of denoising. LQ: Low-Quality input. HQ: High-Quality target.

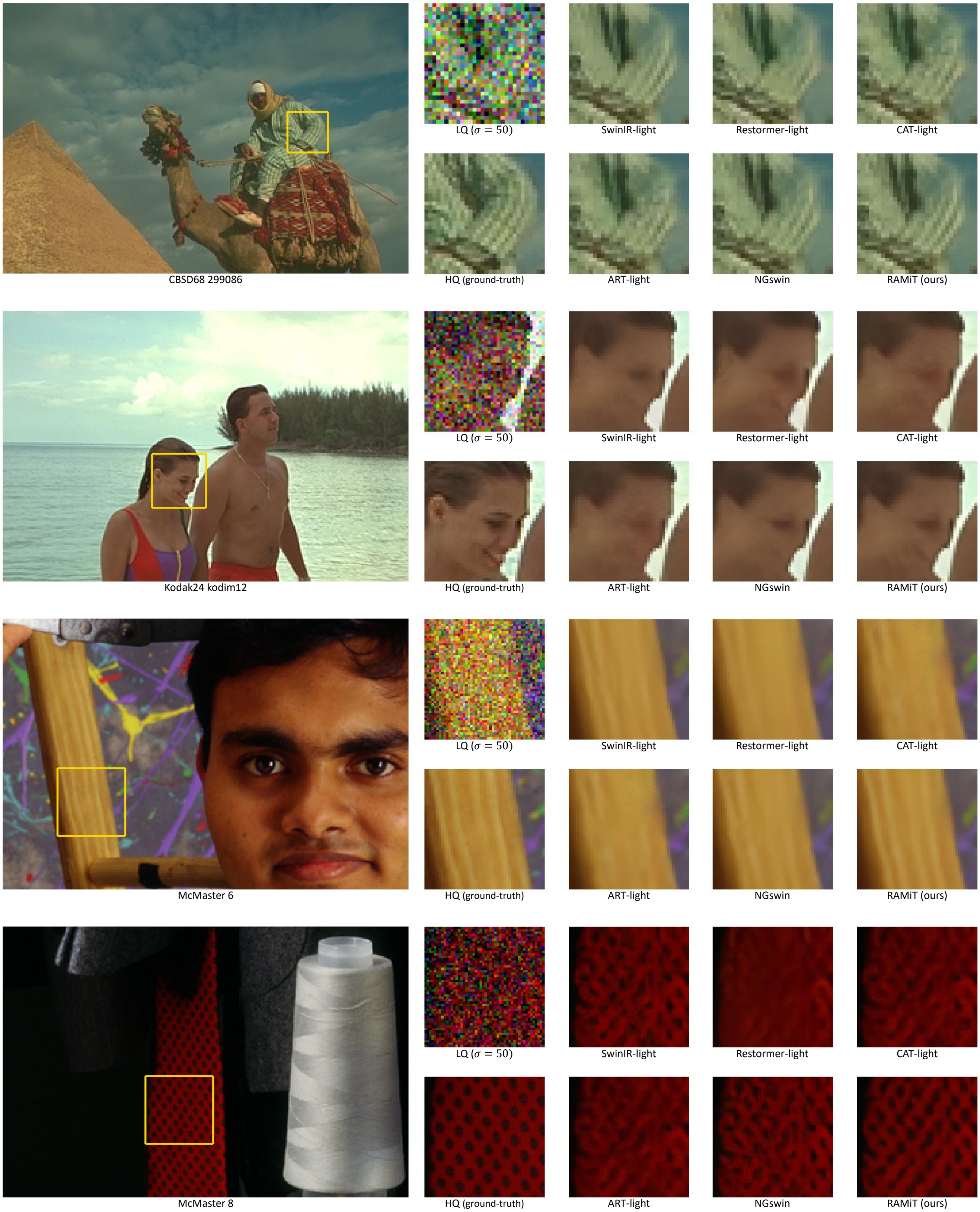


Figure K. Visual comparisons of denoising. LQ: Low-Quality input. HQ: High-Quality target.

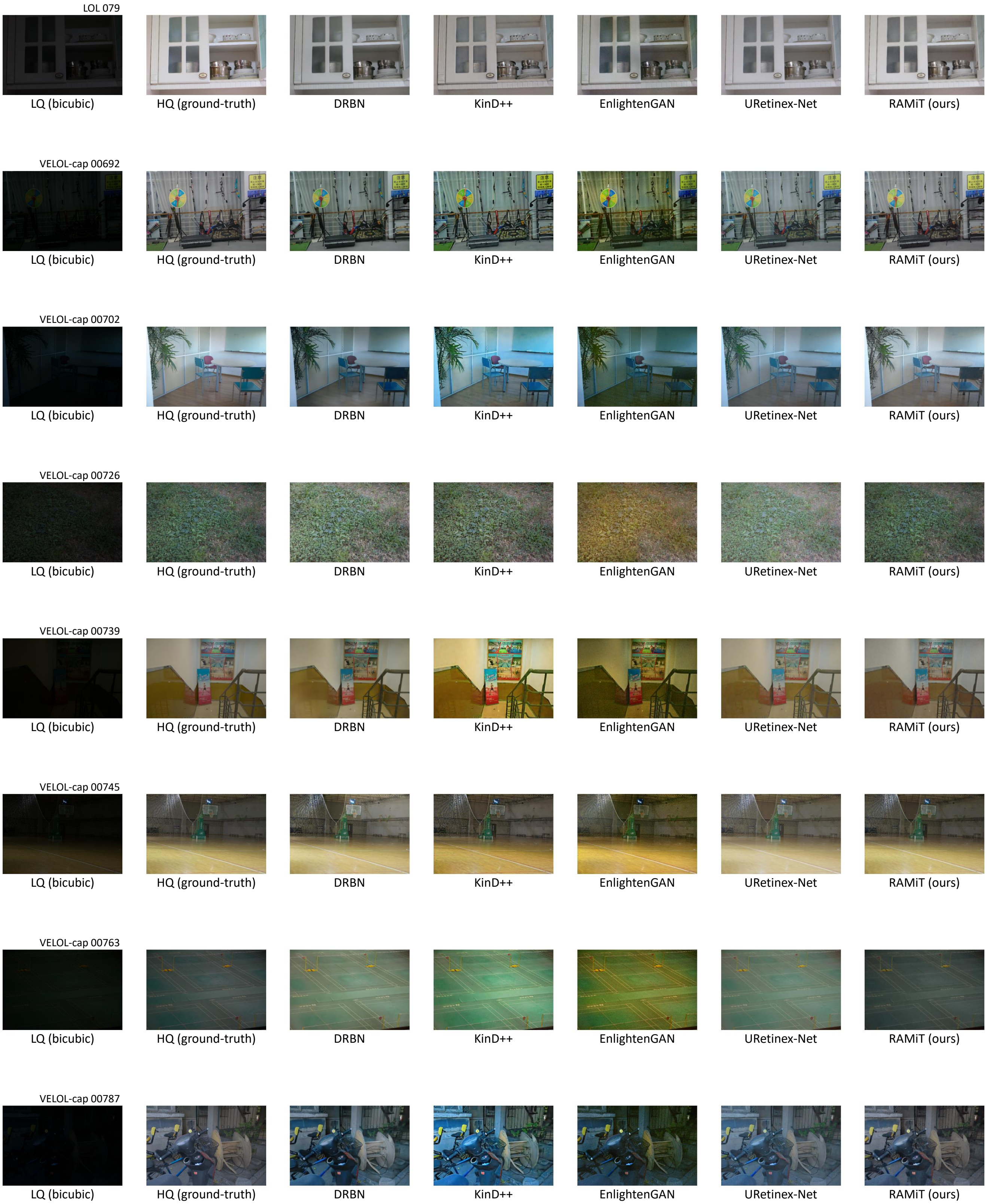


Figure L. Visual comparisons of low-light enhancement. LQ: Low-Quality input. HQ: High-Quality target.

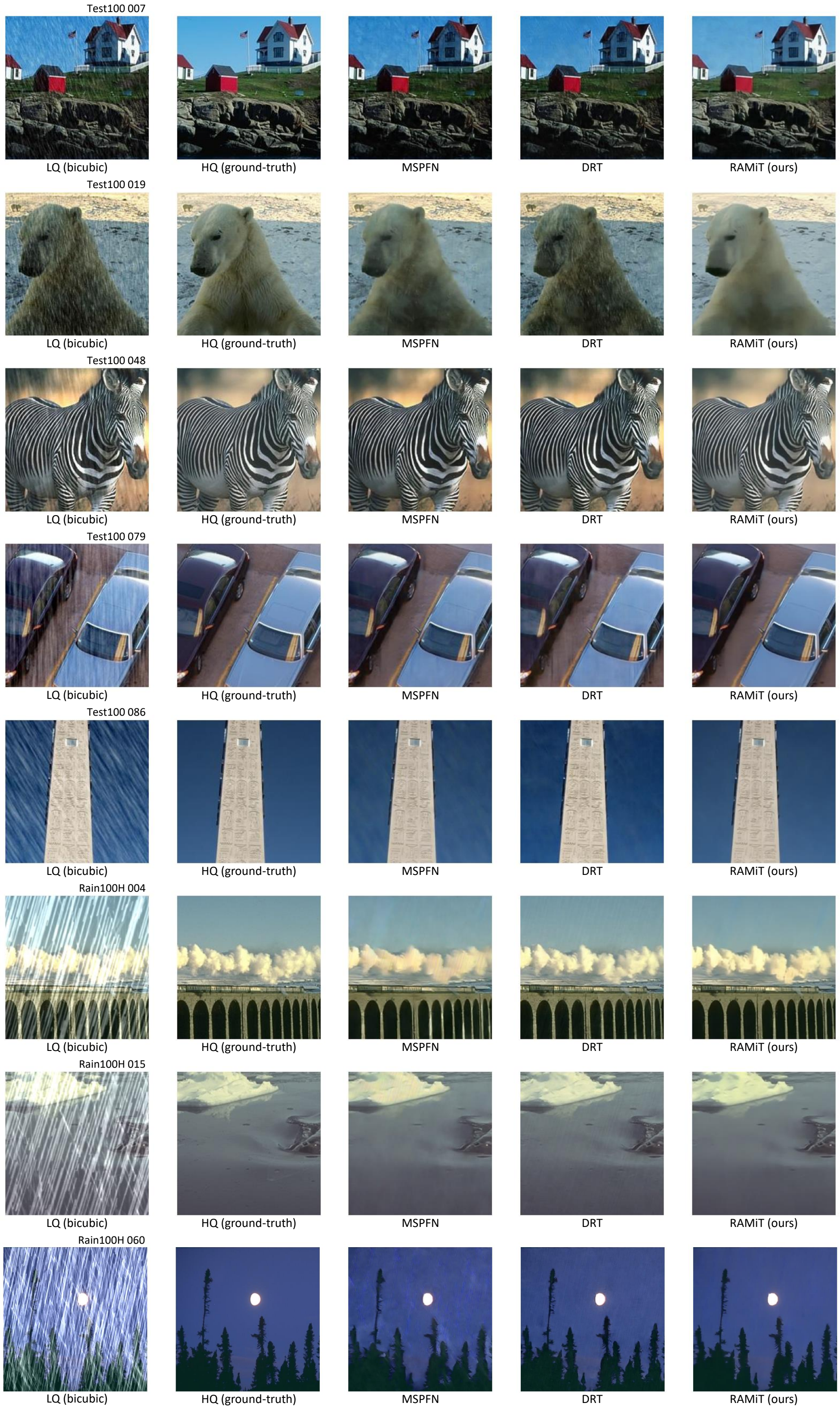


Figure M. Visual comparisons of deraining. LQ: Low-Quality input. HQ: High-Quality target.