MuLUT: Cooperating Multiple Look-Up Tables for Efficient Image Super-Resolution Supplementary Material

Jiacheng Li[1⋆], Chang Chen[2⋆], Zhen Cheng[1], and Zhiwei Xiong[1✉]

[1] University of Science and Technology of China
{jclee, mywander}@mail.ustc.edu.cn, zwxiong@ustc.edu.cn
[2] Huawei Noah's Ark Lab
chenchang25@huawei.com

In this supplementary document, we provide details for the estimation of theoretical energy costs, more implementation details, and supplementary experimental results on the influence of RF. We also provide quantitative results of 2× and 3× SR and more visual results.

1 Energy Costs

Operation	int8	int32	float16	float32
Add. (pJ)	0.03	0.1	0.4	0.9
Mult. (pJ)	0.2	3.1	1.1	3.7

Table 1: The energy costs of operations in different data types. The numbers are from AdderSR [8] and are reported in the literature [3,5,9].

One of the advantages of SR-LUT and MuLUT is replacing expensive computing with relatively cheap memory access. Moreover, LUT also benefits from the lower energy cost of int8 operations compared with int32 ones. Our estimation of theoretical energy costs is based on Table 1, where the detailed energy costs for each operation in different data types are reported.

2 More Implementation Details

Rotation ensemble. As illustrated in Fig. 1, given a LUT with 4 input dimension (shaded box), we extend the *effective* receptive field to 9 pixels (dashed box), by using the operation of rotation ensemble as in SR-LUT [7]. This procedure can be formulated as

$$\hat{HR} = \frac{1}{4}\sum_{j=0}^{3} R_j^{-1}\left(f\left(R_j\left(LR\right)\right)\right),$$

⋆ Equal contribution.

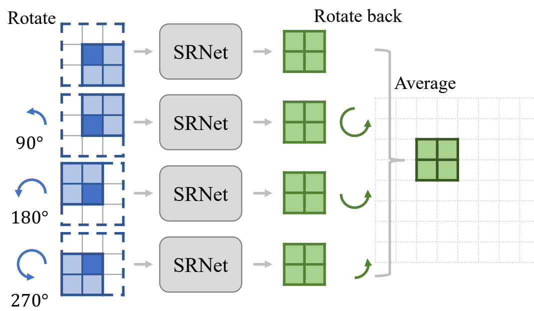

Fig. 1: The rotation ensemble trick.

where f denotes the forward function, R_j and R_j^{-1} denote the j times of `rot90` and its inverse operation, respectively. Thus, the RF size of a LUT can be larger than its dimension. This trick increases RF size effectively while mantaining the original LUT size, and we adopt it for each MuLUT block.

The size of a LUT. Generally, LUT is adopted to avoid the online computation of a complex function, by caching the pre-computed results. To this end, one needs to traverse *all possible combinations* of input values for the offline pre-computation. Due to the exhaustive combination, the size of cached results in LUT grows exponentially with respect to the increasing input dimension. For a 8bit LUT, whose indexex and values are 8bit integers, its size can be calculated as

$$S = (2^{8-q} + 1)^n \times m,$$

where q is the uniform sampling interval, n the index dimension of the LUT, and m the number of values for each record, *e.g.*, $m = 4 \times 4 = 16$ for $4\times$ SR.

3 Supplementary Experimental Results

The motivation to increase the RF. We design two straightforward solutions to improve SR-LUT from the perspective of network architectures. The first one is SR-LUT-Deep, where we add more 1×1 convolutions and increase the trainable parameters of SRNet. The second one is SR-LUT-IMDN, where we adopt a similar multi-branch structure with IMDN [6]. We maintain the RF of SR-LUT-IMDN as 3×3 by only using 1×1 and 2×2 convolutions in its multiple branches. As listed in Table 2, with restricted RF, SR-LUT-Deep and SR-LUT-IMDN show little improvement over SR-LUT, validating the critical role of RF in the SR task. This observation inspired us to increase the RF effectively by cooperating multiple LUTs.

Further increasing the RF. In Table 2, we include the result of MuLUT-SDY-X3, which further increases the RF to 13×13. It obtains slightly better performance over MuLUT-SDY-X2. We choose MuLUT-SDY-X2 as the representative configuration of the proposed MuLUT because of its shorter training time.

PSNR(dB)	Energy(pJ)	LUT Size	RF Size	Set5	Set14	BSDS100	Urban100	Manga109
SR-LUT-S [7]	72.5M	1.274MB	3×3	29.82	27.01	26.53	24.02	26.80
SR-LUT-Deep	72.5M	1.274MB	3×3	29.79	27.02	26.56	24.05	26.83
SR-LUT-IMDN	72.5M	1.274MB	3×3	29.80	27.06	26.58	24.06	26.88
MuLUT-SDY	222.3M	3.823MB	5×5	30.40	27.48	26.79	24.32	27.52
MuLUT-SDY-X2	233.6M	4.062MB	9×9	30.60	27.60	26.86	24.46	27.90
MuLUT-SDY-X3	247.9M	4.301MB	13×13	30.65	27.64	26.88	24.49	27.98

Table 2: Supplementary experimental results for 4× super-resolution.

4 Experimental Results for 2× and 3× SR

We report quantitative results of 2× and 3× SR on standard benchmarks in Table 3 and Table 4, respectively. Following SR-LUT [7], we report the results of RCAN [14] instead of RRDB [12]. As can be seen, MuLUT significantly improves the performance of SR-LUT.

5 More Visual Results

We provide visual comparisons of 2× and 3× SR on standard benchmarks in Fig. 2 and Fig. 3, respectively. We also provide supplementary visual results of 4× SR in Fig. 4 and those of image demosaicing in Fig. 5.

	Method	Set5		Set14		BSDS100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Interpolation	Nearest	30.82	0.8991	28.51	0.8446	28.39	0.8239	25.62	0.8199	28.12	0.9089
	Bilinear	32.12	0.9106	29.15	0.8384	28.65	0.8090	25.95	0.8077	29.13	0.9115
	Bicubic	33.63	0.9292	30.23	0.8681	29.53	0.8421	26.86	0.8394	30.78	0.9338
LUT	SR-LUT-S [7]	35.46	0.9466	31.73	0.8958	30.64	0.8750	28.50	0.8777	33.87	0.9579
	MuLUT-SDY	36.43	0.9530	32.35	0.9049	31.12	0.8849	29.10	0.8880	35.32	0.9656
	MuLUT-SDY-X2	36.65	0.9541	32.49	0.9065	31.23	0.8865	29.31	0.8910	35.78	0.9674
Sparse coding	NE + LLE [2]	35.79	0.9491	31.82	0.8996	30.77	0.8787	28.48	0.8803	33.95	0.9590
	Zeyde et al. [13]	35.79	0.9494	31.87	0.8989	30.77	0.8771	28.47	0.8794	34.06	0.9599
	ANR [10]	35.85	0.9500	31.86	0.9006	30.82	0.8800	28.49	0.8807	33.94	0.9597
	A+ [11]	36.57	0.9545	32.34	0.9056	31.21	0.8860	29.23	0.8938	35.32	0.9670
DNN	FSRCNN [4]	37.05	0.9560	32.66	0.9090	31.53	0.8902	29.88	0.9020	36.67	0.9710
	CARN-M [1]	37.42	0.9583	33.17	0.9136	31.88	0.8960	31.23	0.9192	37.60	0.9740
	RCAN [14]	38.30	0.9617	34.14	0.9235	32.41	0.9025	33.17	0.9377	39.60	0.9791

Table 3: The comparison with other methods for 2× SR on standard benchmark datasets.

	Method	Set5		Set14		BSDS100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Interpolation	Nearest	27.93	0.8123	26.00	0.7330	26.17	0.7065	23.34	0.6992	25.04	0.8157
	Bilinear	29.54	0.8504	26.96	0.7526	26.77	0.7177	23.99	0.7135	26.15	0.8372
	Bicubic	30.40	0.8678	27.55	0.7736	27.20	0.7379	24.45	0.7343	26.94	0.8554
LUT	SR-LUT-S [7]	31.95	0.8969	28.73	0.8057	27.92	0.7690	25.53	0.7750	29.32	0.8970
	MuLUT-SDY	32.59	0.9065	29.25	0.8194	28.23	0.7824	25.94	0.7899	30.34	0.9112
	MuLUT-SDY-X2	32.75	0.9089	29.34	0.8215	28.31	0.7841	26.10	0.7945	30.72	0.9161
Sparse coding	NE + LLE [2]	31.87	0.8958	28.64	0.8085	27.92	0.7727	25.41	0.7755	28.70	0.8889
	Zeyde et al. [13]	31.93	0.8969	28.70	0.8079	27.95	0.7715	25.45	0.7761	28.85	0.8920
	ANR [10]	31.95	0.8970	28.69	0.8102	27.96	0.7745	25.45	0.7768	28.78	0.8900
	A+ [11]	32.63	0.9090	29.16	0.8190	28.28	0.7832	26.04	0.7974	29.90	0.9099
DNN	FSRCNN [4]	33.18	0.9140	29.37	0.8240	28.53	0.7910	26.43	0.8080	31.10	0.9210
	CARN-M [1]	34.00	0.9235	29.99	0.8357	28.90	0.8001	27.55	0.8384	32.82	0.9385
	RCAN [14]	34.78	0.9299	30.63	0.8477	29.33	0.8107	29.02	0.8695	34.58	0.9502

Table 4: The comparison with other methods for 3× SR on standard benchmark datasets.

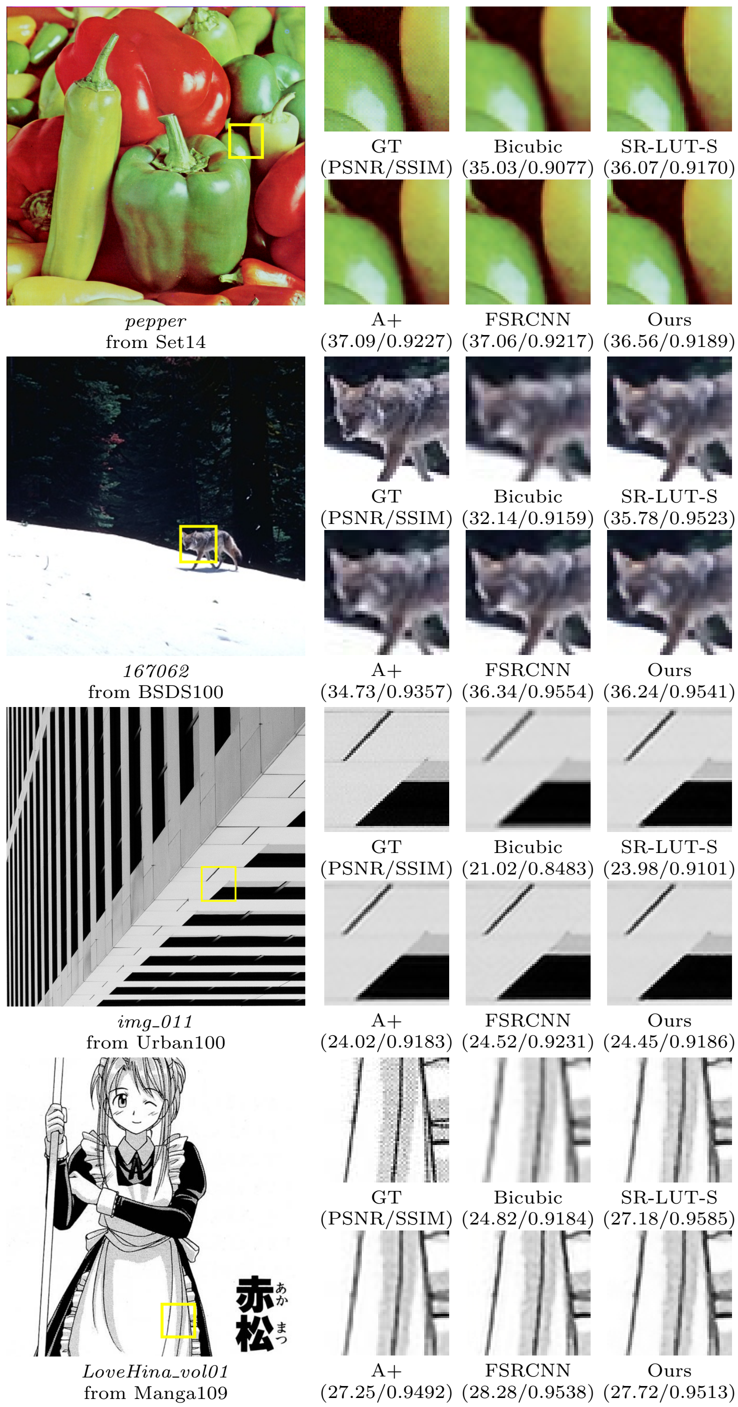

Fig. 2: Visual comparison for ×2 SR on standard benchmark datasets.

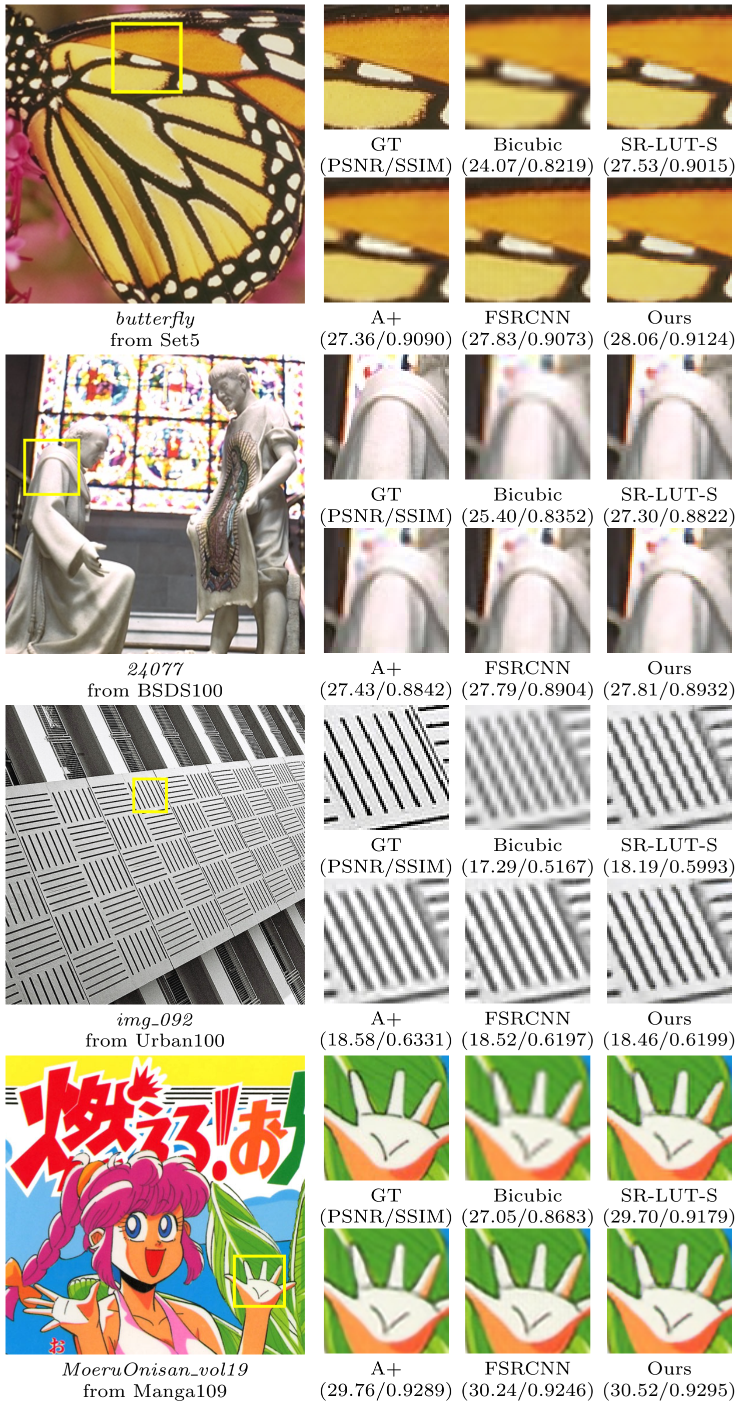

Fig. 3: Visual comparison for $\times 3$ SR on standard benchmark datasets.

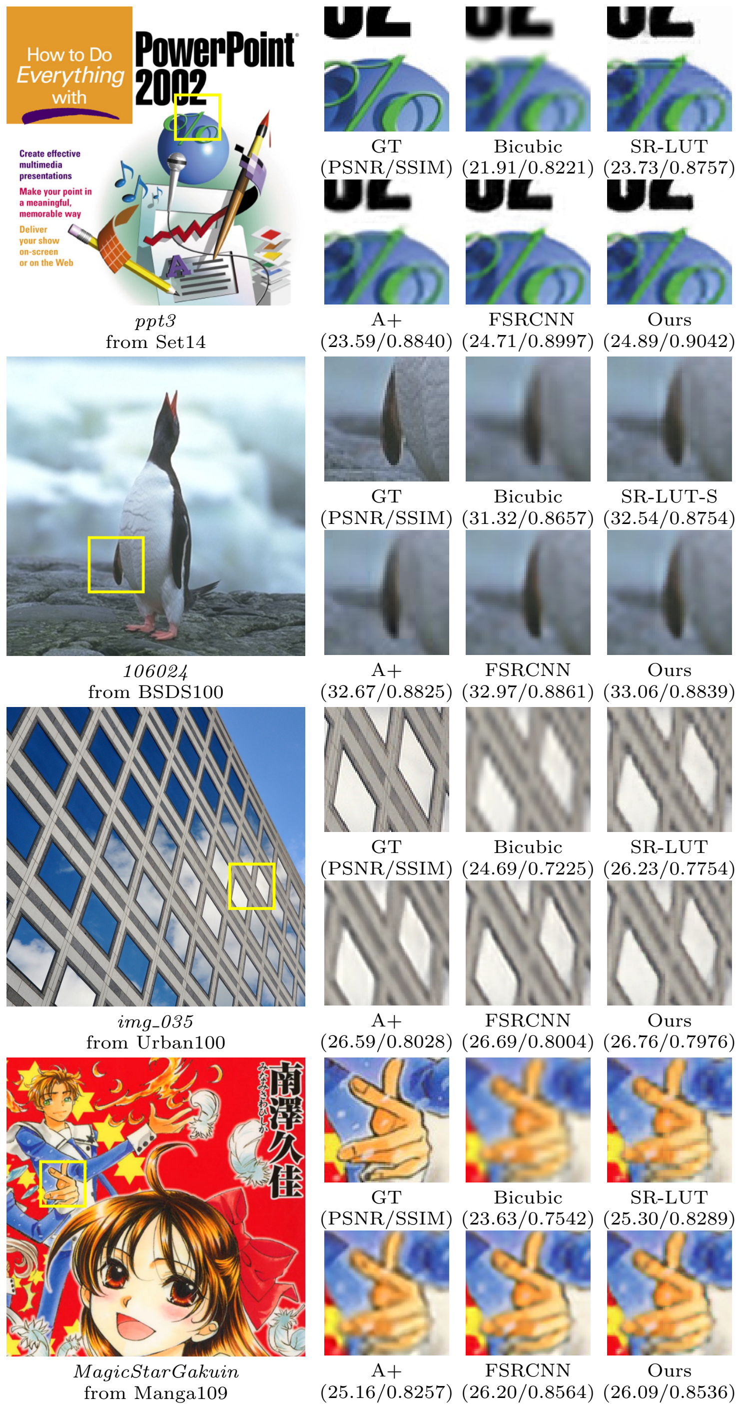

Fig. 4: Supplementary Visual comparison for $\times 4$ SR on standard benchmark datasets.

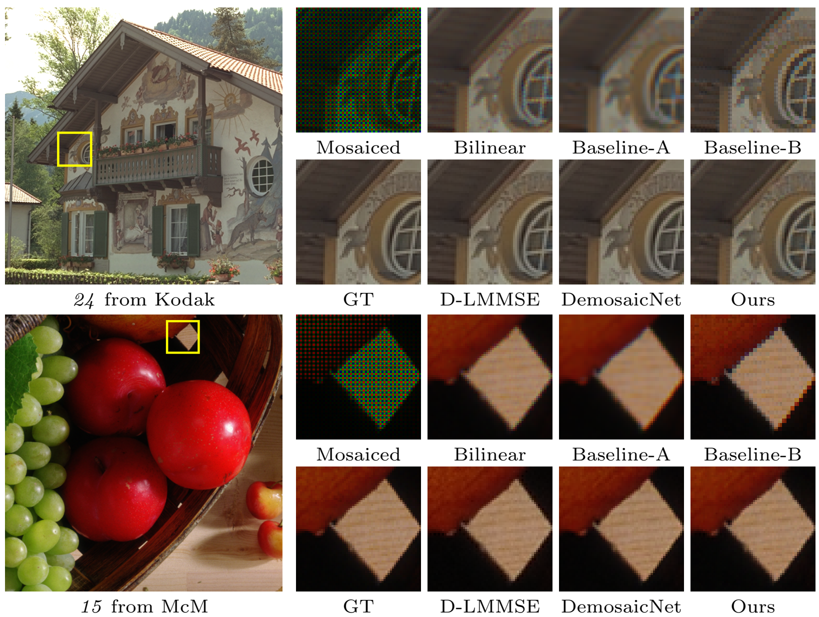

Fig. 5: Supplementary visual comparison for image demosaicing on standard benchmark datasets.

References

1. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: European Conference on Computer Vision (ECCV). vol. 11214, pp. 256–272 (2018)
2. Chang, H., Yeung, D., Xiong, Y.: Super-resolution through neighbor embedding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 275–282 (2004)
3. Dally, W.: High-performance hardware for machine learning. In: NeurIPS Tutorial (2015)
4. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision (ECCV). vol. 9906, pp. 391–407 (2016)
5. Horowitz, M.: Computing's energy problem (and what we can do about it). In: IEEE International Conference on Solid-State Circuits Conference (ISSCC). pp. 10–14 (2014)
6. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACM International Conference on Multimedia(ACM MM). pp. 2024–2032 (2019)
7. Jo, Y., Kim, S.J.: Practical single-image super-resolution using look-up table. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 691–700 (2021)
8. Song, D., Wang, Y., Chen, H., Xu, C., Xu, C., Tao, D.: Addersr: Towards energy efficient image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15648–15657 (2021)
9. Sze, V., Chen, Y., Yang, T., Emer, J.S.: Efficient processing of deep neural networks: A tutorial and survey. Proc. IEEE **105**(12), 2295–2329 (2017)
10. Timofte, R., Smet, V.D., Gool, L.V.: Anchored neighborhood regression for fast example-based super-resolution. In: IEEE International Conference on Computer Vision (ICCV). pp. 1920–1927 (2013)
11. Timofte, R., Smet, V.D., Gool, L.V.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: Asian Conference on Computer Vision (ACCV). vol. 9006, pp. 111–126 (2014)
12. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.C.: ESRGAN: enhanced super-resolution generative adversarial networks. In: European Conference on Computer Vision Workshops (ECCVW). vol. 11133, pp. 63–79 (2018)
13. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces - 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers. vol. 6920, pp. 711–730 (2010)
14. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2472–2481 (2018)