RRSR — Supplementary Material

Lin Zhang[1,3,5,6,*], Xin Li[2,*], Dongliang He[2,†], Fu Li[2], Yili Wang[4], and Zhaoxiang Zhang[1,3,5,7,†]

[1] Institute of Automation, Chinese Academy of Sciences
[2] Department of Computer Vision Technology (VIS), Baidu Inc.
[3] University of Chinese Academy of Sciences [4] Tsinghua University
[5] National Laboratory of Pattern Recognition, CASIA [6] School of Future Technology, UCAS [7] Center for Artificial Intelligence and Robotics, HKISI_CAS
{zhanglin2019, zhaoxiang.zhang}@ia.ac.cn,
{lixin41,hedongliang01,lifu}@baidu.com, wangyili20@mails.tsinghua.edu.cn
[*]Joint First Authors, [†]Joint Corresponding Author

In this supplementary material, we clarify the network architecture in Sec. 1. And we will introduce the training loss of RRSR in Sec. 2. Then, we discuss several reference-aware feature selection mechanisms in Sec. 3. Additionally, we conduct a detailed ablation study on our reciprocal target-reference reconstruction (RTRR) in Sec. 4. We also describe the computational efficiency of our RRSR in Sec. 5. Finally, we show more qualitative results in Sec. 6.

1 Network Architecture

In our framework, we improve C^2-Matching by replacing its feature alignment procedure by our progressive FAS, which contains our newly added RASs and MDCNs. Meanwhile, the CCN(Contrastive Correspondence Network), CE(Content Extractor), VGG, etc remain unchanged. We further reduce the number of resblocks to 5 to save computation budget for RASs and MDCNs. The number of learnable filters in each FAS module is 16.

2 Loss Functions

Our overall objective is formulated as

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{RTRR}\mathcal{L}_{RTRR} + \lambda_{per}\mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv}\,. \tag{1}$$

Reconstruction loss. We adopt ℓ_1-norm to calculate loss between the ground-truth image X_{HR} and the output image X_{SR} , as

$$\mathcal{L}_{rec} = \|X_{HR} - X_{SR}\|_1, \quad X_{SR} = RefSR(X_{LR}, Y_{HR})\,. \tag{2}$$

Reciprocal loss. As elaborated in the main body, we derive the loss function for reciprocal learning:

$$\mathcal{L}_{RTRR} = \|Y^{\mathcal{P}}_{HR} - Y^{\mathcal{P}}_{SR}\|_1, \quad Y^{\mathcal{P}}_{SR} = RefSR(Y^{\mathcal{P}}_{LR}, X_{SR})\,, \tag{3}$$

where $*^{\mathcal{P}}$ denote the warped version.
Perceptual loss. Our perceptual loss is defined as

$$\mathcal{L}_{per} = \|\phi(X_{HR}) - \phi(X_{SR})\|_F \,, \tag{4}$$

where ϕ indicates the features obtained at the ReLU5_1 layer of the pretrained VGG19 model [7], and $\|\cdot\|_F$ denotes the Frobenius norm.
Adversarial loss. Our adversarial loss $\mathcal{L}_{adv}$ is expressed as

$$\mathcal{L}_G = -\mathbb{E}[D(X_{SR})] \,, \tag{5}$$

$$\mathcal{L}_D = \mathbb{E}[D(X_{SR})] - \mathbb{E}[D(X_{HR})] + \lambda_p \mathbb{E}[(\|\nabla_{\hat{X}} D(\hat{X})\|_2 - 1)^2] \,, \tag{6}$$

where $D(\cdot)$ denotes the discriminator. The last term is a penalization term of gradient norm and $\hat{X}$ is the random convex combination of X_{SR} and X_{HR}.

3 Comparison of Different Reference-aware Feature Selection Mechanisms

AdaIN [2] was proposed to align content features with style features in terms of feature statistics. For RefSR, it can be used to remap the distribution of reference features to that of LR features. Based on AdaIN, MASA [4] designed a spatial adaption module (SAM) by adding learnable parameters to adapt local feature differences. Nevertheless, as shown in Table 1, both AdaIN and SAM improve C^2-Matching only a bit. We observe that LR features and reference features are extracted by a shared network in MASA while they are processed by totally different networks (i.e., stacks of residual blocks and the shallow layers of VGG [7]) in C^2-Matching. Feature alignment methods based on statistics perform not well for the latter case. On the contrary, our RAS has a 0.09dB improvement with negligible extra overhead in FLOPs.

Table 1. Ablation study on our RAS

Model	C^2-Matching	C^2-Matching+AdaIN [2]	C^2-Matching+SAM [4]	C^2-Matching+RAS (ours)
PSNR(dB)↑	28.40	28.40	28.41	28.49
GFLOPs	59.0	59.0	63.9	59.0

4 Further Analysis on Reciprocal Learning Framework

To investigate the benefits introduced by our reciprocal target-reference reconstruction (RTRR), we conduct an ablation study in Table 2. Model a is the base model without RTRR, while other models apply RTRR. Model b reconstructs the original references, and Model c-g reconstruct the references with different perturbation ranges. It can be observed from Table 2 that Model b suffers a significant performance drop of 1.07 dB in PSNR. In this way, reference reconstruction is *de facto* self-reconstruction that takes references as inputs and outputs references, which goes against target reconstruction (training collapse). With perturbation range of [-5, 5], Model c gets inferior performance, because $Y^{\mathcal{P}}_{HR}$ and Y_{HR} are very similar, the network tends to reach a local optimum: making $Y^{\mathcal{P}}_{SR}$ as close to Y_{HR} as possible instead of $Y^{\mathcal{P}}_{HR}$, as this is easily achieved by letting X_{SR} retain more of Y_{HR}'s information. We next try [-10, -5] ∪ [5, 10] and [-20, -6] ∪ [6, 20] to exclude small perturbations and found the results are comparable. What's more, we try the case of [-40, -5] ∪ [5, 40], the result is 0.04dB worse than that of [-20, -5] ∪ [5, 20]. We think that much larger perspective transformation makes $Y^{\mathcal{P}}$ and Y not so similar in spatial details, such that X_{SR} cannot provide much reference information for reconstructing details of $Y^{\mathcal{P}}_{LR}$, then the gain of reciprocal learning is discounted.

Table 2. Ablation study on reciprocal learning framework.

Model	RTRR	Geometric perturbation range	PSNR(dB)↑
a			28.70
b	✓	No perspective transformation	27.63
c	✓	[-5, 5]	28.22
d	✓	[-10, -5] ∪ [5, 10]	28.81
e	✓	[-20, -5] ∪ [5, 20]	28.83
f	✓	[-20, -6] ∪ [6, 20]	28.84
g	✓	[-40, -5] ∪ [5, 40]	28.79

5 Computational Efficiency

We compare the proposed RRSR against SISR methods and RefSR methods in terms of computational efficiency. For SISR methods, we include RCAN [12] and NLSN [6]. For RefSR methods, SRNTT [13], TTSR [10], MASA [4], and C^2-Matching [3] are included. The computational complexity is calculated for recovering a 160 × 160 × 3 image under 4× SR setting. Our RRSR achieves a PSNR of 28.83 dB at the cost of 81.7 GFLOPs. For a fair comparison, we build a small variant of RRSR, dubbed RRSR-S, which uses the proposed FAS module for two (three in RRSR) times at the 2× and 4× scales. Table 3 reports the computational cost and the performances. Though our RRSR-S has lower computational cost than C^2-Matching [3], yet it performs significantly better.

Table 3. Efficiency and performance comparisons.

Model	RCAN [12]	NLSN [6]	SRNTT [13]	TTSR [10]	MASA [4]	C^2-Matching [3]	RRSR-S	RRSR
GFLOPs	29.0	138.8	19.7	32.8	24.2	59.0	57.4	81.7
PSNR↑	26.06	26.53	26.24	27.09	27.54	28.24	28.75	28.83
SSIM↑	0.769	0.784	0.784	0.804	0.814	0.841	0.855	0.856

6 More Visual Comparisons

In this section, we provide more visual results among the proposed RRSR and the current top-performing methods, such as ESRGAN [9], RankSRGAN [11], TTSR [10], MASA [4], and C^2-Matching [3]. Comparisons on the testing set of CUFED5 [13] are shown in Fig. 1 and Fig. 2. Comparisons on Sun80 [8] and Urban100 [1] are shown in Fig. 3, and Manga109 [5] and WR-SR [3] are shown in Fig. 4. These results indicate that our RRSR can restore more realistic textures.

References

1. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
2. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
3. Jiang, Y., Chan, K.C., Wang, X., Loy, C.C., Liu, Z.: Robust reference-based super-resolution via c2-matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Lu, L., Li, W., Tao, X., Lu, J., Jia, J.: Masa-sr: Matching acceleration and spatial adaptation for reference-based image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
5. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications (2017)
6. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
8. Sun, L., Hays, J.: Super-resolution from internet-scale scene matching. In: IEEE International Conference on Computational Photography (ICCP) (2012)
9. 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: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
10. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
11. Zhang, W., Liu, Y., Dong, C., Qiao, Y.: Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
12. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
13. Zhang, Z., Wang, Z., Lin, Z., Qi, H.: Image super-resolution by neural texture transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

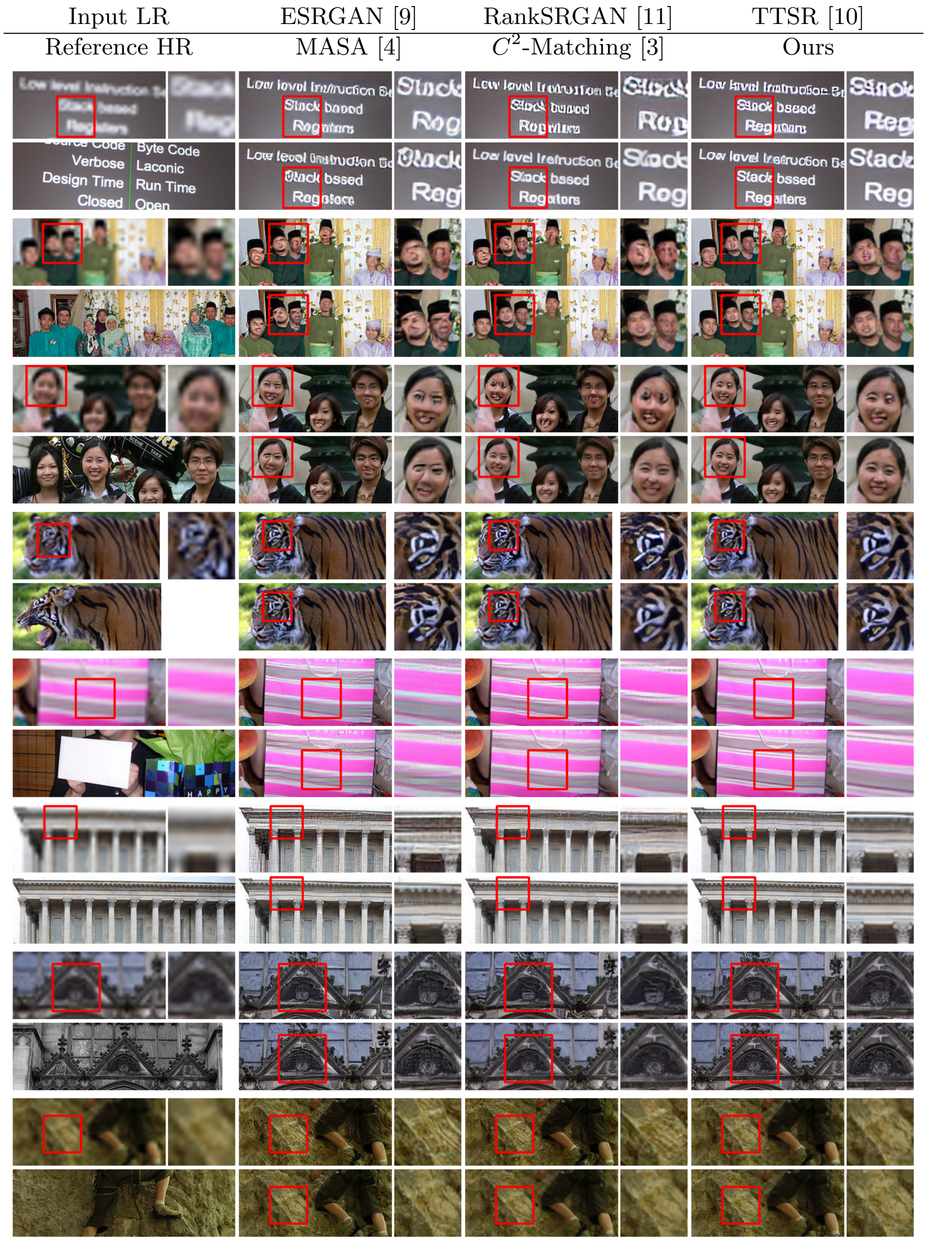

Fig. 1. Comparisons on the testing set of CUFED5 [13]. (part 1)

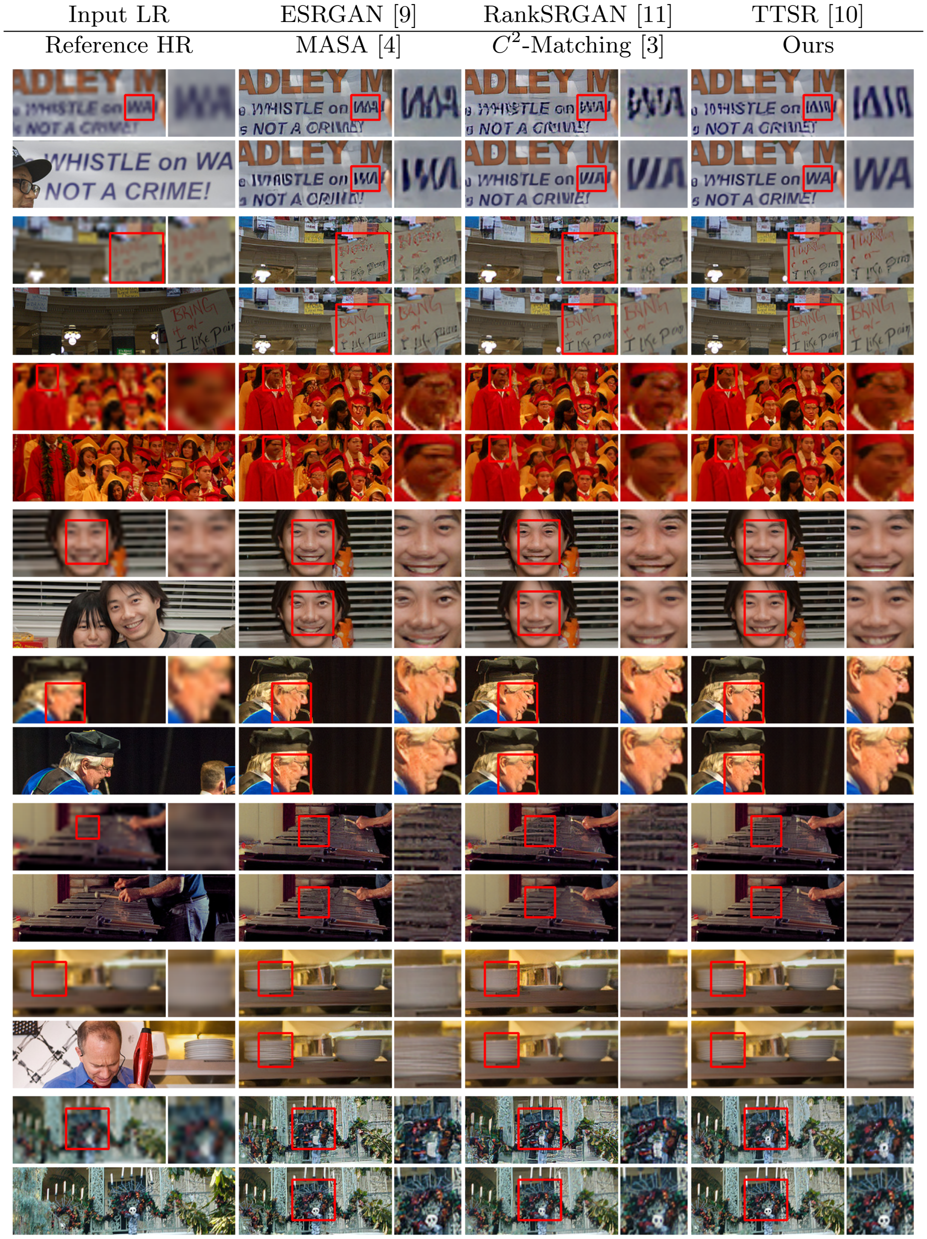

Fig. 2. Comparisons on the testing set of CUFED5 [13]. (part 2)

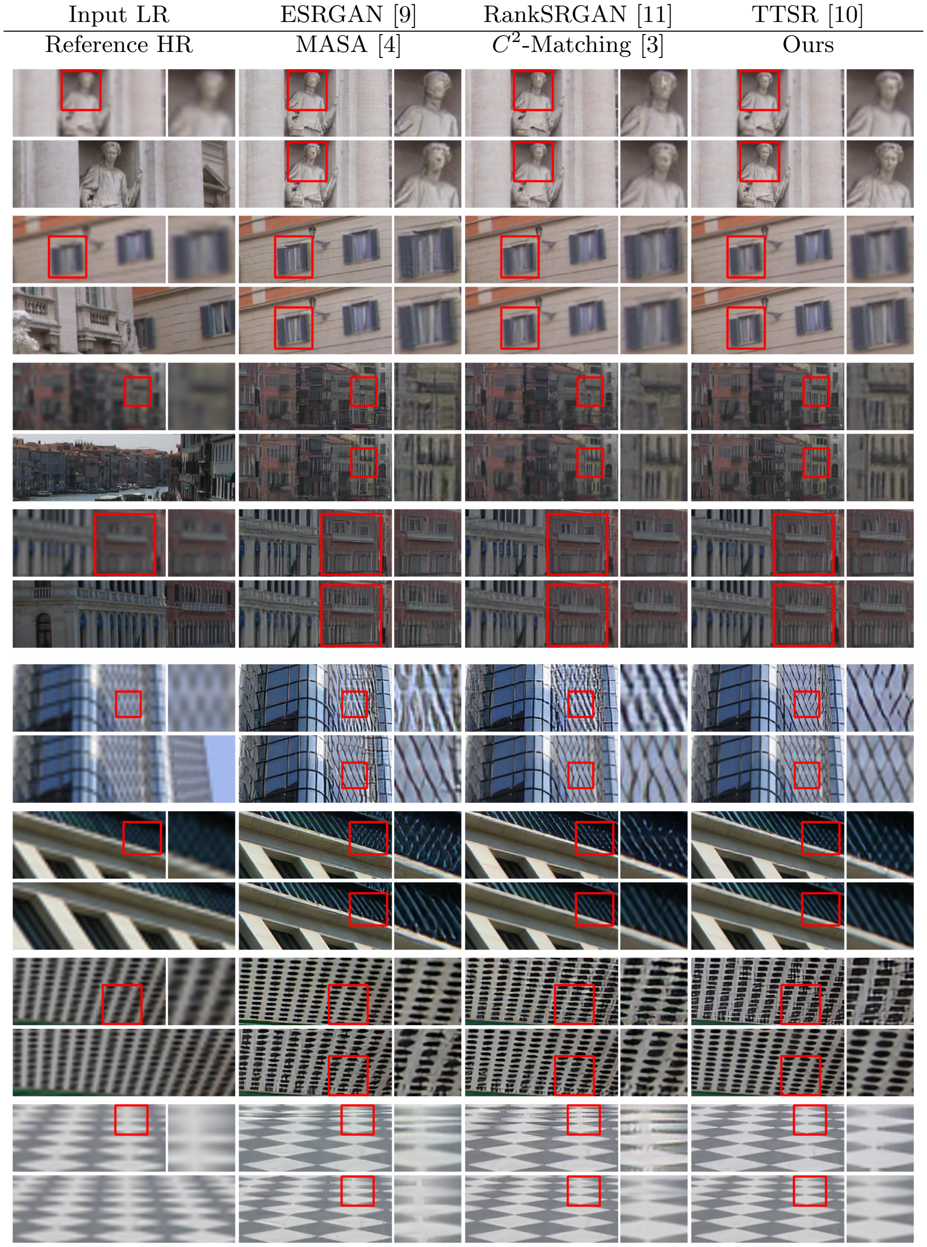

Fig. 3. Comparisons on Sun80 [8] (the top four examples) and Urban100 [1] (the bottom four examples).

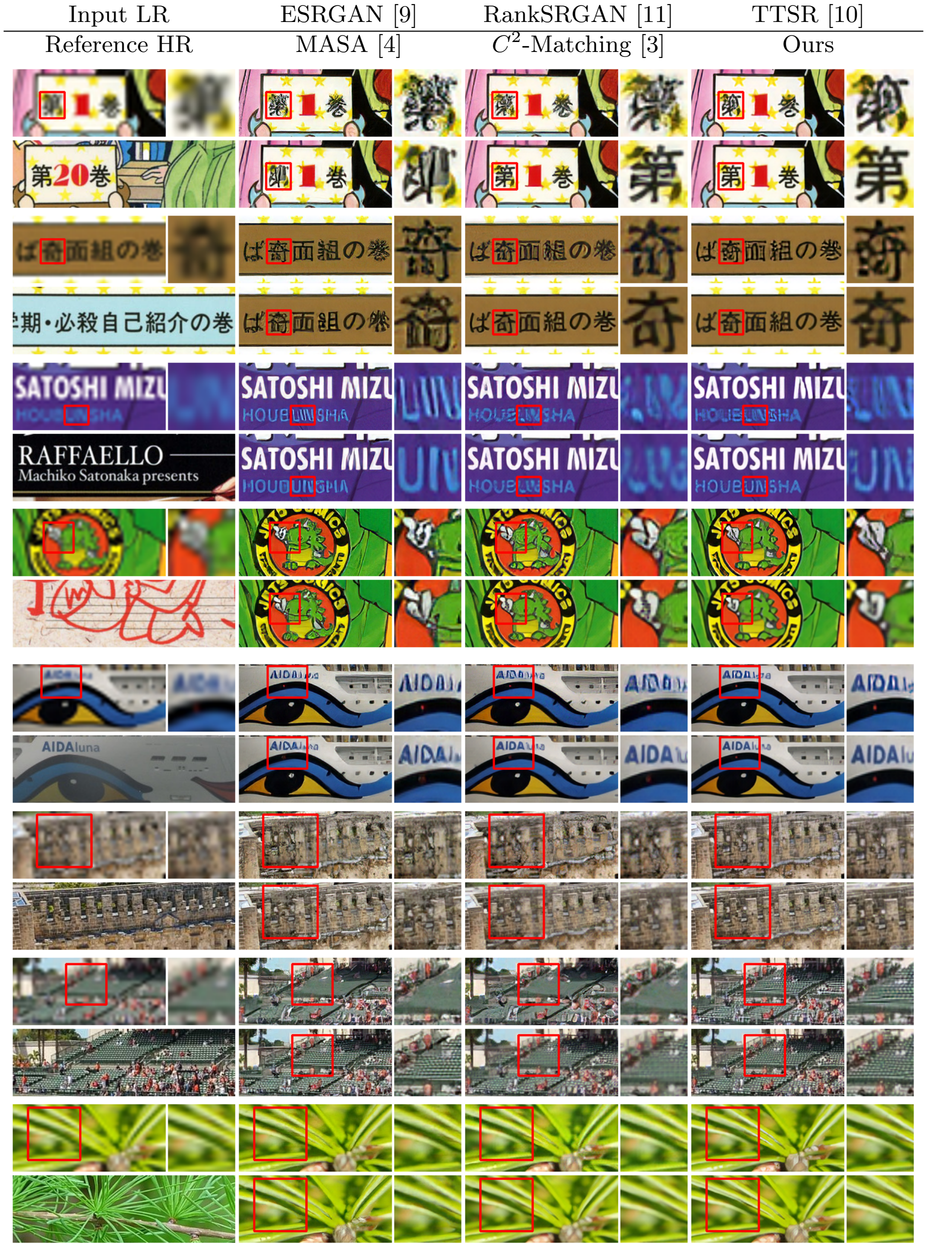

Fig. 4. Comparisons on Manga109 [5] (the top four examples) and WR-SR [3] (the bottom four examples).