Self-Supervised Learning for Real-World Super-Resolution from Dual Zoomed Observations (Supplementary Material)

Zhilu Zhang[1], Ruohao Wang[1], Hongzhi Zhang[1] (✉), Yunjin Chen, and Wangmeng Zuo[1,2]

[1] Harbin Institute of Technology, China
[2] Peng Cheng Laboratory
{cszlzhang, rhwangHIT}@outlook.com, zhanghz0451@gmail.com,
chenyunjin_nudt@hotmail.com, wmzuo@hit.edu.cn

A Content

The content of this supplementary material involves:

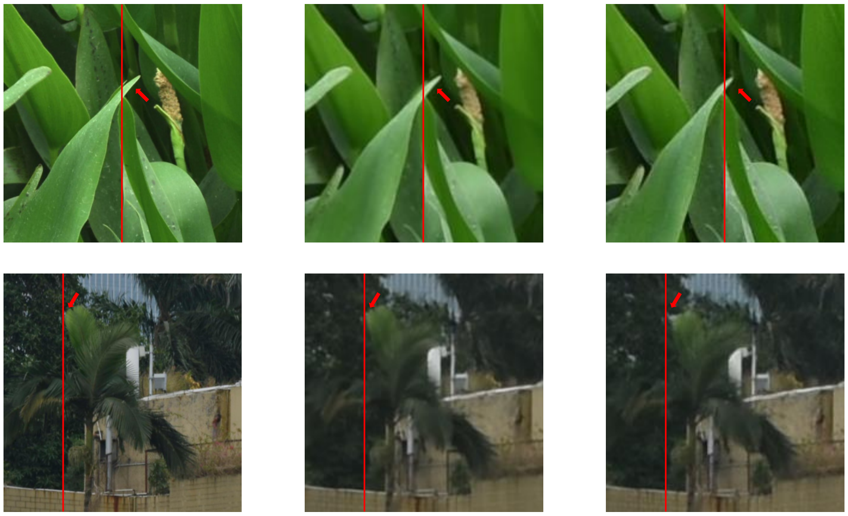
- Synthetic noise for auxiliary-LR in Sec. B.
- Visual results of auxiliary-LR in Sec. C.
- Network structure of restoration module in Sec. D.
- Sliced Wasserstein (SW) loss in Sec. E.
- Comparison of #FLOPs in Sec. F.
- Evaluation of generalization performance on other cameras in Sec. G.
- Additional visual comparison on Nikon camera and CameraFusion dataset in Sec. H.

B Synthetic Noise for Auxiliary-LR

Noise in real-world images is common, but complex and various. In order to bridge the gap between auxiliary-LR and LR as much as possible, we need to add noise to the output of the auxiliary-LR generator network that only simulates the blurring and down-sampling processes. Gaussian noise is a natural choice, but much different from real-world image noise. Inspired by BSRGAN [11], we also add JPEG compression noise. The variance of Gaussian noise is uniformly sampled from 5/255 to 30/255, and the JPEG quality factor is uniformly chosen from 60 to 95. Simultaneously, the order of adding three kinds of noise is stochastic.

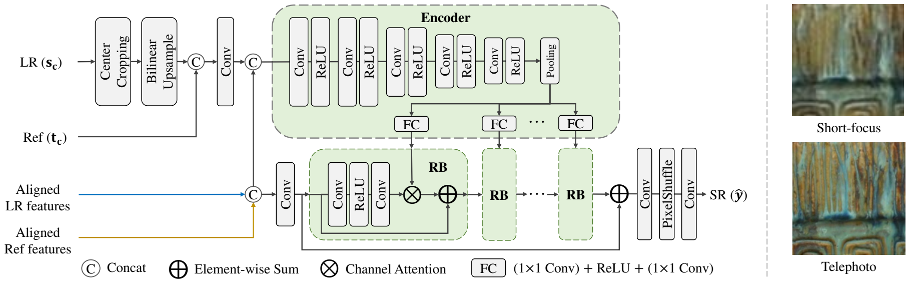
C Visual Results of Auxiliary-LR

Since the noise type and intensity of auxiliary-LR are random, for the convenience of display and comparison, we show the auxiliary-LR images without adding synthetic noise in Fig. A(b). And the corresponding GT and LR images



(a) GT (b) Auxiliary-LR with no noise (c) LR

Fig. A: Visual Results of auxiliary-LR with no noise. The auxiliary-LR has similar contents as LR and it is aligned with GT.



(a) Structure of the restoration model (b) Color inconsistency

Fig. B: Structure of the restoration model and color inconsistency. (a) Detailed structure of the restoration model. 'RB' denotes the residual block. (b) Color inconsistency between short-focus and telephoto images.

are shown in Fig. A(a) and Fig. A(c), respectively. The red lines and arrows in the same row are in the same position relative to the image. It can be seen that the auxiliary-LR has similar contents as LR and it is aligned with GT. It indicates that the function of the auxiliary-LR generator is guaranteed.

Algorithm A Pseudocode of SW loss

```
Require: $\mathbf{U} \in \mathbb{R}^{C\times H\times W}$: VGG features of output image; $\mathbf{V} \in \mathbb{R}^{C\times H\times W}$: VGG features of target image; $\mathbf{M} \in \mathbb{R}^{C'\times C}$: random projection matrix;
Ensure: $\mathcal{L}_{\mathrm{SW}}(\mathbf{U}, \mathbf{V})$: the value of SW loss;
    flatten features $\mathbf{U}$ and $\mathbf{V}$ to $\mathbf{U_f}(\in \mathbb{R}^{C\times HW})$ and $\mathbf{V_f}(\in \mathbb{R}^{C\times HW})$, respectively;
    project the features onto $C'$ directions: $\mathbf{U_p} = \mathbf{MU_f}$, $\mathbf{V_p} = \mathbf{MV_f}$;
    sort projections for each direction: $\mathbf{U_s} = \mathbf{Sort}(\mathbf{U_p}, \mathrm{dim}=1)$, $\mathbf{V_s} = \mathbf{Sort}(\mathbf{V_p}, \mathrm{dim}=1)$;
    $\mathcal{L}_{\mathrm{SW}}(\mathbf{U}, \mathbf{V}) = \|\mathbf{U_s} - \mathbf{V_s}\|_1$
```

Table A: Quantitative results of SelfSZSR using different loss terms.

Loss Terms	PSNR↑	LPIPS↓
ℓ_1	28.93	0.308
ℓ_1 + SW	28.67	0.219
ℓ_1 + Perceptual + Adversarial	28.45	0.216

D Restoration Module

Fig. B(a) shows the detailed structure of the restoration module. First, the aligned LR and aligned Ref features are concatenated and fed into the backbone, which consists of 16 residual blocks [4]. Then the concatenated features are input into an encoder for generating vectors that modulate the features of each residual block. Simultaneously, the original Ref image and the central area of the LR image are utilized to enrich the input features of the encoder for better modulation. This modulation can also be regarded as a kind of channel attention on the features of residual block. And it is beneficial to relieve the color inconsistency (see Fig. B(b)) between the short focal length and telephoto images in the real world.

E Sliced Wasserstein Loss

The algorithm of SW loss is given in Alg. A. We first obtain the 1-dimensional representation of 2-dimensional VGG [6] features through random linear projection. Then we calculate the Wasserstein distance between the output and the target 1-dimensional probability distributions, which is defined as the element-wise ℓ_1 distance over sorted 1-dimensional distributions.

Most reference-based image SR (RefSR) methods [13,10,5,2] adopt the perceptual loss and adversarial loss [1] for more realistic results. For a fair comparison, here we also train proposed model (SelfDZSR) using a combination of ℓ_1 reconstruction loss, perceptual loss and adversarial loss based on Relativistic GAN [3]. The quantitative results are shown in Table A. It can be seen that the

Table B: #FLOPs comparison of SISR and RefSR methods. The #FLOPs is measured under the setting of ×4 super-resolving LR image to 1280×720 resolution. For RefSR methods, the Ref image has the same size with LR.

	Method	#FLOPs (G)
SISR	EDSR [25]	5792
	RCAN [52]	1838
	CDC [42]	1626
	BSRGAN [48]	2068
	Real-ESRGAN [39]	2068
RefSR	SRNTT [53]	3568
	TTSR [47]	2468
	C^2-Matching [19]	1968
	MASA [27]	1984
	DCSR [38]	836
Ours	SelfDZSR	384

model trained by SW loss obtains a 0.22 dB PSNR gain than that by adversarial loss, while the gap of LPIPS metric is small. Nevertheless, benefiting from the proposed implicit alignment and better utilization of Ref information, the SelfDZSR model by adversarial loss still achieves better performance than other RefSR methods.

F Comparison of #FLOPs

The cost of calculating similarity between LR and Ref occupies a large part of the computational cost of previous RefSR methods. We calculate cosine similarity between ×4 down-sampled Ref and ×4 down-sampled LR features, and find that its performance is close to that of computing similarity at original image size. By virtue of the faster similarity calculation and more lightweight restoration model, our method has lower FLOPs in comparison to both SISR and RefSR methods, as shown in Table B.

G Evaluation of Generalization Performance on Other Cameras

Here we evaluate the generalization performance of models on other four cameras (*i.e.*, Canon, Olympus, Panasonic and Sony) from DRealSR dataset [9]. We compare results with SISR (*i.e.*, EDSR [4], RCAN [12], CDC [9], BSRGAN [11] and Real-ESRGAN [8]) and RefSR (*i.e.*, SRNTT [13], TTSR [10], MASA [5], C^2-Matching [2] and DCSR [7]) methods. Among them, the results of BSRGAN and Real-ESRGAN are generated via the officially released model while other methods are trained on Nikon camera images, as mentioned in the main text of the submission.

Tables C$\sim$F show the quantitative results on four cameras, respectively. Our proposed model (SelfDZSR) achieves better results than most other methods, especially on *Full-Image* and LPIPS metric. The visual comparison is carried out on the methods that are trained not only with ℓ_1 (or ℓ_2) loss. The comparisons on four cameras can be seen in Fig. C$\sim$F, respectively.

H Additional Visual Comparison on Nikon Camera and CameraFusion Dataset

The visual comparison is carried out on the methods that are trained not only with ℓ_1 (or ℓ_2) loss. In Fig. G, we show more qualitative comparison on Nikon camera images [9]. The visual comparison on CameraFusion dataset [7] can be seen in Fig. H. The resolution of full LR images from the two test sets is 1$\sim$2K, so we select a patch for comparison.

Table C: Quantitative results on **Canon** camera with 17 images. Best results are highlighted by red. The models trained only with ℓ_1 (or ℓ_2) loss are marked in gray. RefSR† represents that the RefSR methods are trained in our self-supervised learning manner.

	Method	# Param	*Full-Image*	*Corner-Image*
		(M)	PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
SISR	EDSR [4]	43.1	26.52 / 0.8399 / 0.342	26.55 / 0.8383 / 0.342
	RCAN [12]	15.6	26.69 / 0.8413 / 0.346	26.73 / 0.8404 / 0.347
	CDC [9]	39.9	24.85 / 0.8378 / 0.384	24.90 / 0.8366 / 0.385
	BSRGAN [11]	16.7	25.39 / 0.8031 / 0.268	25.43 / 0.8017 / 0.268
	Real-ESRGAN [8]	16.7	24.64 / 0.8010 / 0.275	24.68 / 0.7988 / 0.276
RefSR†	SRNTT-ℓ_2 [13]	5.5	26.22 / 0.8390 / 0.348	26.24 / 0.8378 / 0.351
	SRNTT [13]	5.5	26.25 / 0.8268 / 0.295	26.28 / 0.8258 / 0.293
	TTSR-ℓ_1 [10]	7.3	23.97 / 0.8280 / 0.374	23.93 / 0.8259 / 0.375
	TTSR [10]	7.3	23.75 / 0.7719 / 0.340	23.68 / 0.7695 / 0.338
	C^2-Matching-ℓ_1 [2]	8.9	25.64 / 0.8383 / 0.357	25.60 / 0.8373 / 0.358
	C^2-Matching [2]	8.9	24.75 / 0.8180 / 0.329	24.72 / 0.8171 / 0.328
	MASA-ℓ_1 [5]	4.0	26.54 / 0.8398 / 0.338	26.58 / 0.8390 / 0.339
	MASA [5]	4.0	27.19 / 0.8006 / 0.306	27.24 / 0.7994 / 0.305
	DCSR-ℓ_1 [7]	3.2	27.55 / 0.8363 / 0.338	27.54 / 0.8377 / 0.330
	DCSR [7]	3.2	26.80 / 0.8265 / 0.275	26.79 / 0.8275 / 0.268
Ours	SelfDZSR-ℓ_1	3.2	28.13 / 0.8576 / 0.300	27.87 / 0.8465 / 0.321
	SelfDZSR	3.2	27.85 / 0.8386 / 0.240	27.60 / 0.8274 / 0.253

Table D: Quantitative results on **Olympus** camera with 19 images. Best results are highlighted by red. The models trained only with ℓ_1 (or ℓ_2) loss are marked in gray. RefSR† represents that the RefSR methods are trained in our self-supervised learning manner.

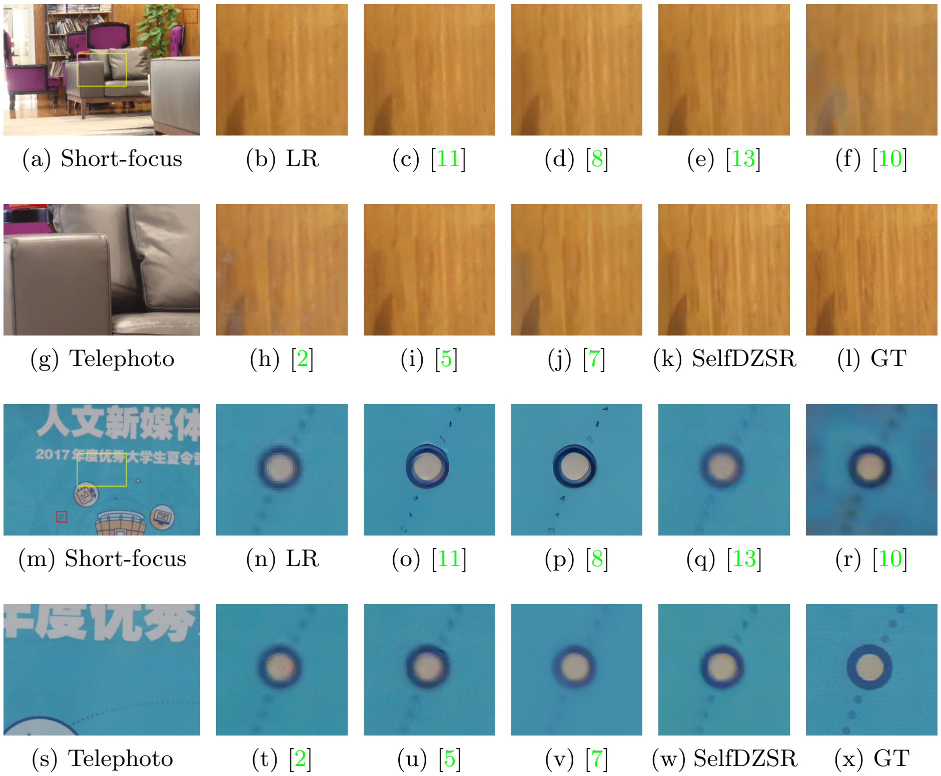
	Method	# Param	*Full-Image*	*Corner-Image*
		(M)	PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
SISR	EDSR [4]	43.1	26.99 / 0.7960 / 0.452	26.99 / 0.7917 / 0.451
	RCAN [12]	15.6	27.54 / 0.8038 / 0.452	27.54 / 0.7995 / 0.451
	CDC [9]	39.9	27.31 / 0.8030 / 0.466	27.30 / 0.7988 / 0.467
	BSRGAN [11]	16.7	25.76 / 0.7422 / 0.341	25.75 / 0.7388 / 0.341
	Real-ESRGAN [8]	16.7	26.00 / 0.7545 / 0.323	25.98 / 0.7517 / 0.321
RefSR†	SRNTT-ℓ_2 [13]	5.5	26.51 / 0.7928 / 0.442	26.49 / 0.7879 / 0.441
	SRNTT [13]	5.5	27.04 / 0.7870 / 0.357	27.03 / 0.7823 / 0.354
	TTSR-ℓ_1 [10]	7.3	25.44 / 0.7790 / 0.469	25.39 / 0.7756 / 0.466
	TTSR [10]	7.3	25.12 / 0.7736 / 0.377	25.08 / 0.7693 / 0.370
	C^2-Matching-ℓ_1 [2]	8.9	26.65 / 0.8001 / 0.448	26.62 / 0.7960 / 0.446
	C^2-Matching [2]	8.9	26.51 / 0.7728 / 0.380	26.49 / 0.7682 / 0.378
	MASA-ℓ_1 [5]	4.0	27.17 / 0.7982 / 0.423	27.17 / 0.7937 / 0.423
	MASA [5]	4.0	26.66 / 0.7393 / 0.351	26.66 / 0.7348 / 0.350
	DCSR-ℓ_1 [7]	3.2	27.74 / 0.7987 / 0.437	27.73 / 0.7973 / 0.426
	DCSR [7]	3.2	27.41 / 0.7894 / 0.351	27.39 / 0.7882 / 0.342
Ours	SelfDZSR-ℓ_1	3.2	27.79 / 0.8124 / 0.395	27.56 / 0.7980 / 0.420
	SelfDZSR	3.2	27.34 / 0.7862 / 0.292	27.12 / 0.7722 / 0.310

Table E: Quantitative results on **Panasonic** camera with 20 images. Best results are highlighted by red. The models trained only with ℓ_1 (or ℓ_2) loss are marked in gray. RefSR† represents that the RefSR methods are trained in our self-supervised learning manner.

	Method	# Param	*Full-Image*	*Corner-Image*
		(M)	PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
SISR	EDSR [4]	43.1	27.04 / 0.7994 / 0.379	27.15 / 0.7964 / 0.380
	RCAN [12]	15.6	27.26 / 0.8055 / 0.381	27.36 / 0.8027 / 0.383
	CDC [9]	39.9	27.02 / 0.7981 / 0.414	27.12 / 0.7949 / 0.414
	BSRGAN [11]	16.7	26.27 / 0.7520 / 0.288	26.40 / 0.7482 / 0.288
	Real-ESRGAN [8]	16.7	26.20 / 0.7625 / 0.275	26.31 / 0.7592 / 0.274
RefSR†	SRNTT-ℓ_2 [13]	5.5	27.08 / 0.7988 / 0.374	27.18 / 0.7960 / 0.375
	SRNTT [13]	5.5	27.14 / 0.7862 / 0.307	27.22 / 0.7829 / 0.306
	TTSR-ℓ_1 [10]	7.3	26.21 / 0.7859 / 0.383	26.26 / 0.7842 / 0.385
	TTSR [10]	7.3	25.24 / 0.7558 / 0.329	25.26 / 0.7537 / 0.326
	C^2-Matching-ℓ_1 [2]	8.9	26.61 / 0.8032 / 0.378	26.71 / 0.7994 / 0.381
	C^2-Matching [2]	8.9	25.70 / 0.7649 / 0.340	25.78 / 0.7596 / 0.342
	MASA-ℓ_1 [5]	4.0	26.94 / 0.7997 / 0.363	27.00 / 0.7958 / 0.365
	MASA [5]	4.0	26.93 / 0.7388 / 0.299	27.04 / 0.7365 / 0.299
	DCSR-ℓ_1 [7]	3.2	26.58 / 0.7640 / 0.398	26.54 / 0.7632 / 0.390
	DCSR [7]	3.2	26.40 / 0.7543 / 0.315	26.36 / 0.7528 / 0.308
Ours	SelfDZSR-ℓ_1	3.2	27.90 / 0.8164 / 0.337	27.67 / 0.8001 / 0.361
	SelfDZSR	3.2	27.41 / 0.7836 / 0.250	27.21 / 0.7674 / 0.265

Table F: Quantitative results on **Sony** camera with 17 images. Best results are highlighted by red. The models trained only with ℓ_1 (or ℓ_2) loss are marked in gray. RefSR† represents that the RefSR methods are trained in our self-supervised learning manner.

	Method	# Param	*Full-Image*	*Corner-Image*
		(M)	PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
SISR	EDSR [4]	43.1	27.12 / 0.8173 / 0.337	27.13 / 0.8195 / 0.331
	RCAN [12]	15.6	27.42 / 0.8248 / 0.333	27.40 / 0.8274 / 0.326
	CDC [9]	39.9	27.27 / 0.8207 / 0.357	27.27 / 0.8228 / 0.351
	BSRGAN [11]	16.7	26.58 / 0.7732 / 0.284	26.57 / 0.7775 / 0.279
	Real-ESRGAN [8]	16.7	26.20 / 0.7816 / 0.262	26.18 / 0.7876 / 0.256
RefSR†	SRNTT-ℓ_2 [13]	5.5	26.20 / 0.8103 / 0.337	26.18 / 0.8138 / 0.331
	SRNTT [13]	5.5	26.24 / 0.7969 / 0.290	26.23 / 0.8001 / 0.283
	TTSR-ℓ_1 [10]	7.3	25.86 / 0.8152 / 0.333	25.82 / 0.8195 / 0.327
	TTSR [10]	7.3	24.91 / 0.7326 / 0.315	24.86 / 0.7353 / 0.310
	C^2-Matching-ℓ_1 [2]	8.9	26.78 / 0.8221 / 0.327	26.73 / 0.8254 / 0.322
	C^2-Matching [2]	8.9	26.49 / 0.7813 / 0.298	26.44 / 0.7848 / 0.289
	MASA-ℓ_1 [5]	4.0	27.06 / 0.8149 / 0.306	27.06 / 0.8189 / 0.301
	MASA [5]	4.0	25.85 / 0.7075 / 0.325	25.84 / 0.7106 / 0.318
	DCSR-ℓ_1 [7]	3.2	28.49 / 0.8216 / 0.335	28.45 / 0.8237 / 0.330
	DCSR [7]	3.2	28.08 / 0.8128 / 0.272	28.03 / 0.8147 / 0.269
Ours	SelfDZSR-ℓ_1	3.2	28.22 / 0.8311 / 0.292	28.34 / 0.8359 / 0.303
	SelfDZSR	3.2	27.41 / 0.7921 / 0.246	27.47 / 0.7948 / 0.252



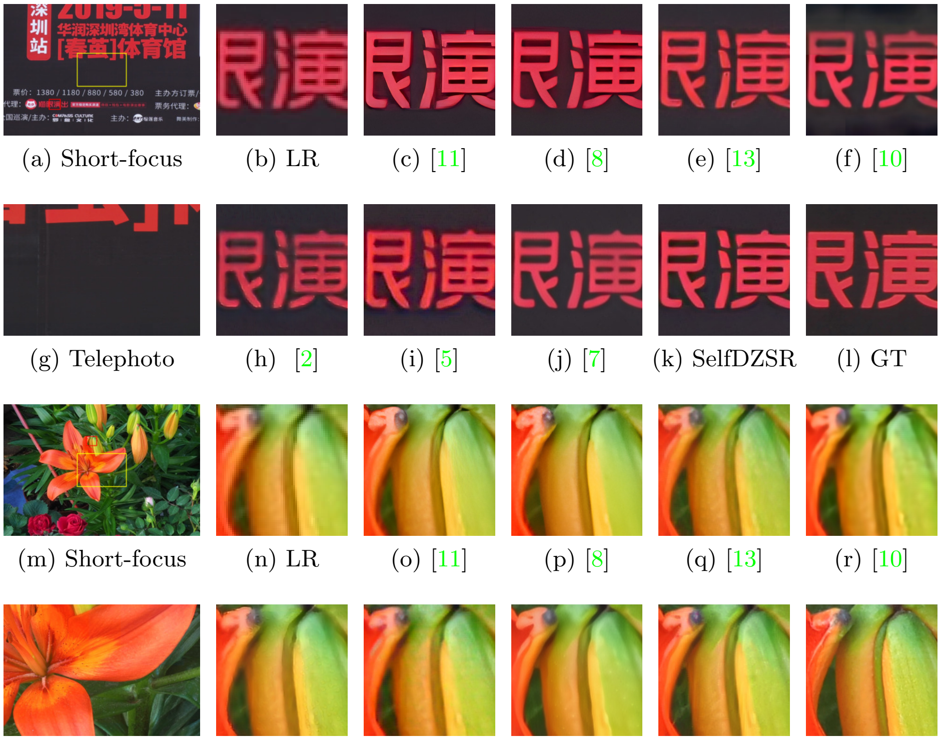
(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. C: Visual comparison on **Canon** camera. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our result in sub-figure (k) restores more fine-scale textures, and that in sub-figure (w) is clearer and more photo-realistic.



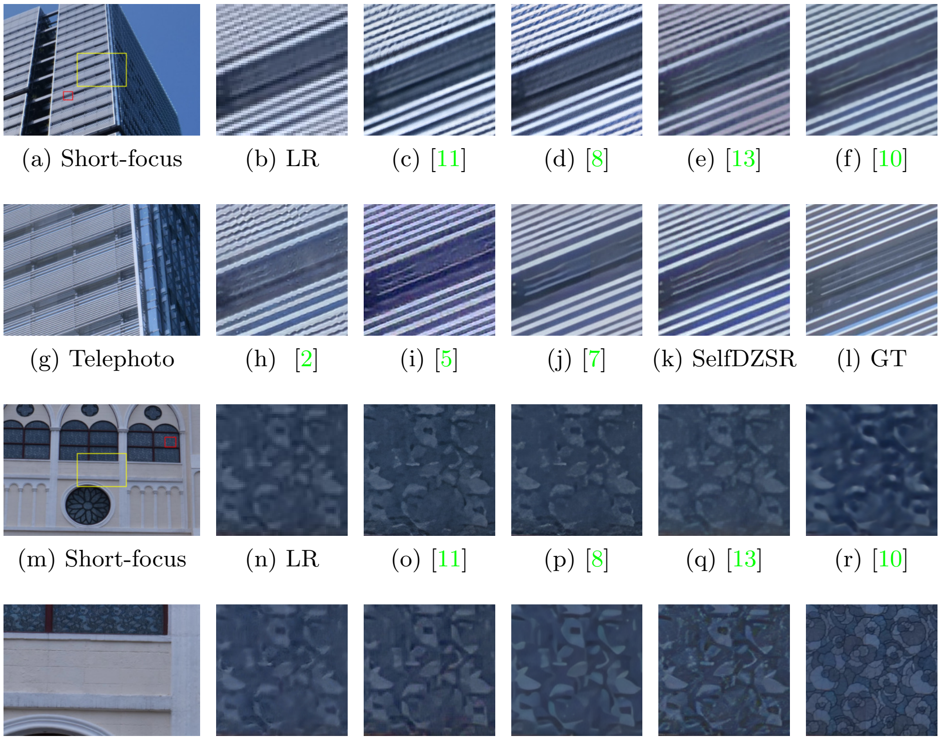
(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. D: Visual comparison on **Olympus** camera. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our result in sub-figure (k) is clearer, and that in sub-figure (w) is more photo-realistic.



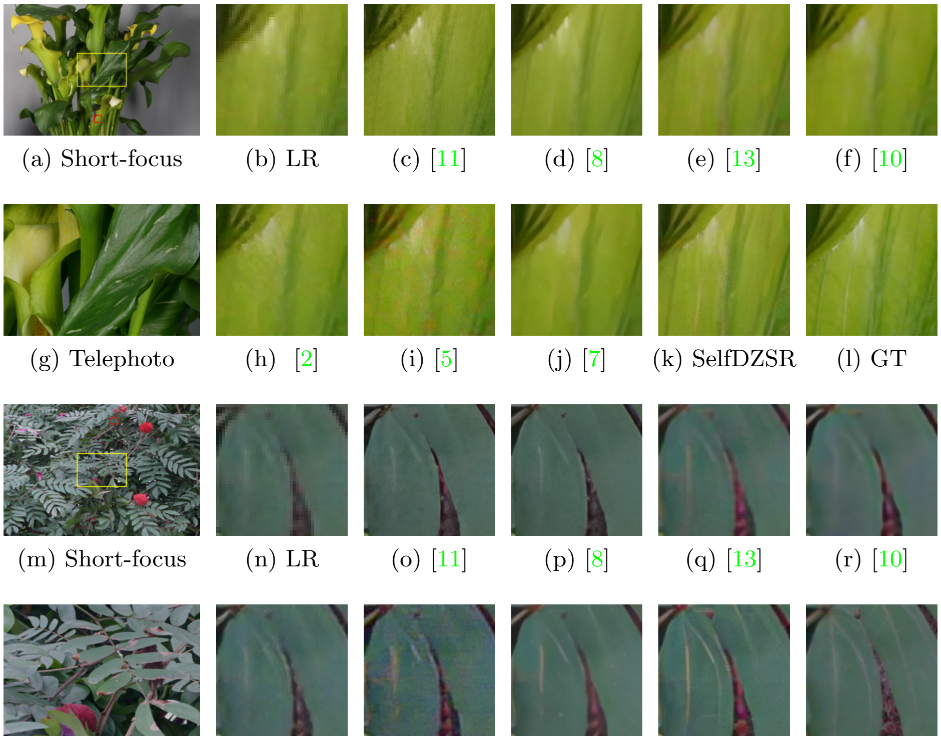
(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. E: Visual comparison on **Panasonic** camera. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our results in sub-figure (k) and (w) restore much more fine details.



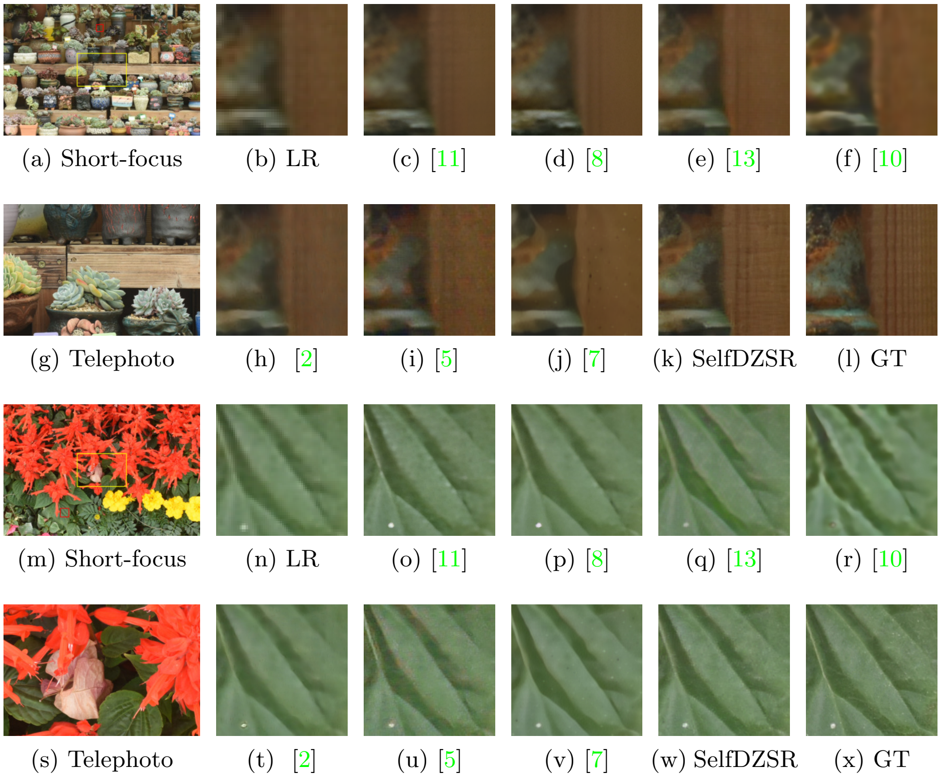
(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. F: Visual comparison on **Sony** camera. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our results in sub-figure (k) and (w) restore much more fine-scale edges.



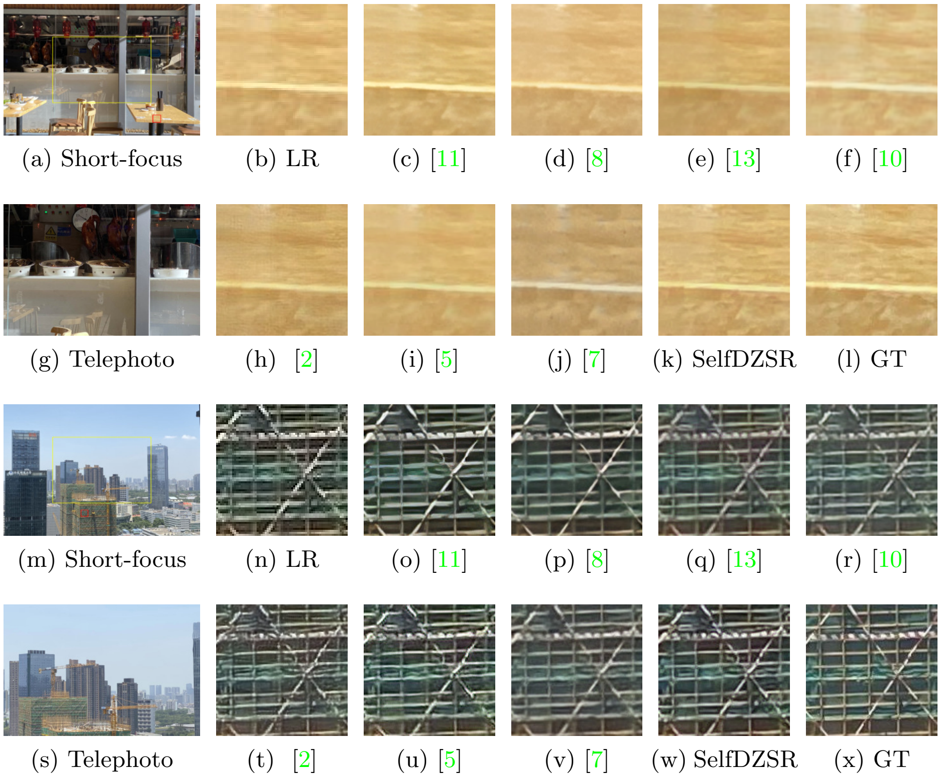
(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. G: Visual comparison on **Nikon** camera. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our result in sub-figure (k) restores much more details, and that in sub-figure (w) is more photo-realistic.



(a) Short-focus (b) LR (c) [11] (d) [8] (e) [13] (f) [10]

(g) Telephoto (h) [2] (i) [5] (j) [7] (k) SelfDZSR (l) GT

(m) Short-focus (n) LR (o) [11] (p) [8] (q) [13] (r) [10]

(s) Telephoto (t) [2] (u) [5] (v) [7] (w) SelfDZSR (x) GT

Fig. H: Visual comparison on **CameraFusion** dataset. In the short-focus image, the yellow box indicates the overlapped scene with the telephoto image, while the red box represents the selected LR patch. Our result in sub-figure (k) restores much more textures, and that in sub-figure (w) is clearer and more photo-realistic.

References

1. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 3
2. Jiang, Y., Chan, K.C., Wang, X., Loy, C.C., Liu, Z.: Robust reference-based super-resolution via c2-matching. In: CVPR. pp. 2103–2112 (2021) 3, 4, 6, 7, 8, 9, 10, 11, 12, 13
3. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard gan. arXiv preprint arXiv:1807.00734 (2018) 3
4. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops. pp. 136–144 (2017) 3, 4, 6, 7
5. Lu, L., Li, W., Tao, X., Lu, J., Jia, J.: Masa-sr: Matching acceleration and spatial adaptation for reference-based image super-resolution. In: CVPR. pp. 6368–6377 (2021) 3, 4, 6, 7, 8, 9, 10, 11, 12, 13
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2014) 3
7. Wang, T., Xie, J., Sun, W., Yan, Q., Chen, Q.: Dual-camera super-resolution with aligned attention modules. In: ICCV. pp. 2001–2010 (2021) 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
8. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: ICCV Workshops. pp. 1905–1914 (2021) 4, 6, 7, 8, 9, 10, 11, 12, 13
9. Wei, P., Xie, Z., Lu, H., Zhan, Z., Ye, Q., Zuo, W., Lin, L.: Component divide-and-conquer for real-world image super-resolution. In: ECCV. pp. 101–117. Springer (2020) 4, 5, 6, 7
10. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: CVPR. pp. 5791–5800 (2020) 3, 4, 6, 7, 8, 9, 10, 11, 12, 13
11. Zhang, K., Liang, J., Van Gool, L., Timofte, R.: Designing a practical degradation model for deep blind image super-resolution. In: ICCV. pp. 4791–4800 (2021) 1, 4, 6, 7, 8, 9, 10, 11, 12, 13
12. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV. pp. 286–301 (2018) 4, 6, 7
13. Zhang, Z., Wang, Z., Lin, Z., Qi, H.: Image super-resolution by neural texture transfer. In: CVPR. pp. 7982–7991 (2019) 3, 4, 6, 7, 8, 9, 10, 11, 12, 13