MoESR: Blind Super-Resolution using Kernel-Aware Mixture of Experts (Supplementary Material)

Mohammad Emad[1]   **Maurice Peemen**[2]   **Henk Corporaal**[1]

[1]Eindhoven University of Technology, Netherlands   [2]Thermo Fisher Scientific, Netherlands

{m.emad, h.corporaal}@tue.nl   maurice.peemen@thermofisher.com

A   Rotate/flip procedure

To keep the number of experts limited, we exploit the symmetries in anisotropic kernels. For example, by rotating the LR image by 90 degrees, we simulate a 90 degree rotated kernel. Figure 1 shows that we can use a single expert network U_ρ (trained with kernel parameters ρ) to upsample images with rotated/flipped kernels. The blur kernel of each LR image is specified at the corner of that image. Due to the rotation and mirror symmetries, we only require experts for kernels within a rotation angel $\theta \sim (0, \pi/4)$. For θ values outside of this range, we firstly rotate/flip the LR image as suggested in Figure 1 and then upsample the image by pretrained expert. Finally, we rotate/flip back the upsampled image to the normal position.
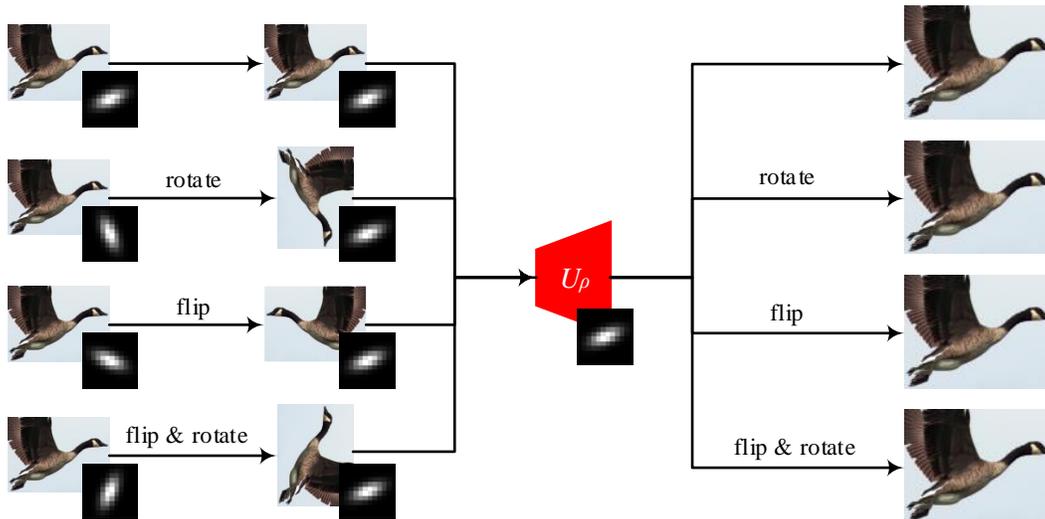

Figure 1: Reduce the number of experts by rotation and/or flipping the input image.

B Kernels of pretrained experts

We use 25 specialized expert networks with kernel parameters in the range $\lambda_1, \lambda_2 \sim (0.6, 5)$ and $\theta \sim (0, \pi/4)$. Table 1 shows the kernel parameters for all networks. Last column demonstrates the θ values applicable by rotate/flip procedure. For isotropic kernels (λ_1 is equal to λ_2), the rotation angel θ is not important.

Table 1: Kernel parameters for all kernel-specific experts.

Index	λ_1	λ_2	θ	Applicable θ
1	0.8	0.8	0	any
2	1.5	0.8	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
3	1.5	0.8	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
4	1.5	1.5	0	any
5	2.3	0.8	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
6	2.3	0.8	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
7	2.3	1.5	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
8	2.3	1.5	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
9	2.3	2.3	0	any
10	3.2	0.8	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
11	3.2	0.8	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
12	3.2	1.5	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
13	3.2	1.5	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
14	3.2	2.3	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
15	3.2	2.3	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
16	3.2	3.2	0	any
17	4.4	0.8	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
18	4.4	0.8	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
19	4.4	1.5	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
20	4.4	1.5	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
21	4.4	2.3	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
22	4.4	2.3	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
23	4.4	3.2	$\frac{\pi}{16}$	$\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$
24	4.4	3.2	$\frac{3\pi}{16}$	$\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$
25	4.4	4.4	0	any

C Fine-tuning using DualSR pipeline

We use DualSR [4] pipeline for fine-tuning. DualSR is a dual-path architecture that jointly trains an image-specific downsampler (degradation kernel) and corresponding upsampler (expert network). Unlike DualSR, we only train the upsampler. We replace the downsampler with our estimated kernel $K(\rho^*)$ and initialize the upsampler with the pre-trained expert U^*. The network is fine-tuned using cycle-consistency losses and masked interpolation loss as explained in [4]. We extend the architecture of DualSR to enable $\times 4$ SR. The extended architecture is shown in Figure 2. We added an extra loss term for the intermediate $\times 2$ upsampled image ($\mathcal{L}_{cycle3}$).
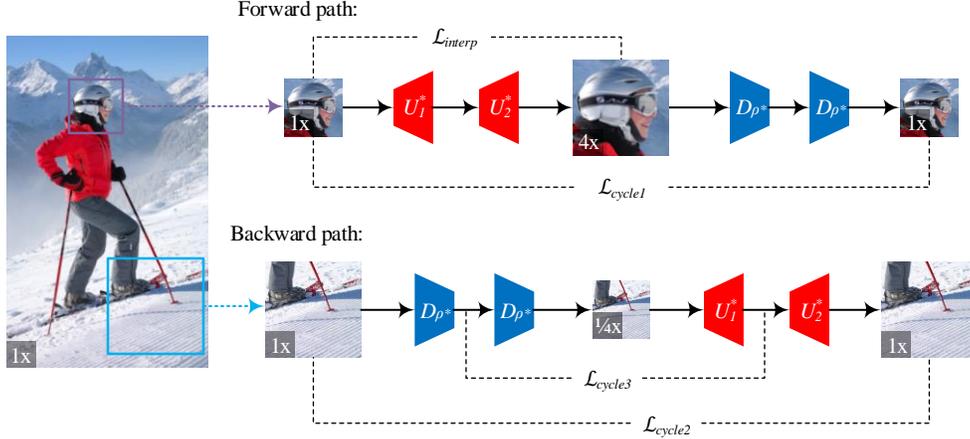
Figure 2: The extended architecture of DualSR for $\times 4$ SR. U_1^* and U_2^* are $\times 2$ upsamplers initialized by the best expert network and D_{ρ^*} is downsampling with the estimated kernel parameters ρ^*.

D   Extra experiments

D.1   Number of experts in the ensemble

In our final model, the ensemble consists of 25 specialized expert networks. We also experiment with ensembles of 6 and 49 experts. The average PSNR and run-time of different models are reported in Table 2. It suggests that an ensemble of 25 experts can cover the whole kernel space. If time is a limitation, an ensemble of 6 experts can still give acceptable accuracy.

Table 2: Analysis on number of experts in the ensemble. The average PSNR and run-time numbers on DIV2KRK [1] dataset for $\times 4$ SR are reported.

Method	PSNR (dB)	Run-time (sec)
MoESR with 6 experts	28.39	25.2
MoESR with 25 experts (final)	28.48	43.9
MoESR with 49 experts	28.45	70.4

D.2   Number of convolution layers for an expert

We use 12-layer convolutional networks in our final design. For comparison, we also report the PSNR and run-time of 8-layer and 16-layer networks in Table 3.

Table 3: Analysis on number of convolution layers for each expert. The average PSNR and run-time numbers on DIV2KRK [1] dataset for $\times 4$ SR are reported.

Method	PSNR (dB)	Run-time (sec)
MoESR with 8-layer experts	28.41	34.2
MoESR with 12-layer experts (final)	28.48	43.9
MoESR with 16-layer experts	28.49	58.3

D.3   Comparison between direct and indirect upsampling for $\times 4$ SR

As shown in Figure 2, we use two sequential 12-layer networks for $\times 4$ SR where each network upsamples the image with a scale of 2. However, we also experiment with a single 20-layer network that upsamples the image with a scale of 4 directly. The average PSNR and run-time are reported in Table 4. The indirect method (two sequential networks) provides better accuracy and is faster compared to direct upsampling.

Table 4: Direct vs indirect upsampling for ×4 SR. The average PSNR and run-time numbers on DIV2KRK [1] dataset for ×4 SR are reported.

Method	PSNR (dB)	Run-time (sec)
MoESR direct upsampling	28.36	51.9
MoESR indirect upsampling (final)	28.48	43.9

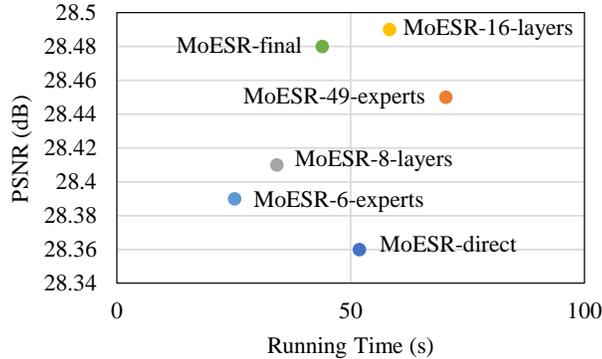Figure 3 shows the tradeoff between accuracy and run-time of different MoESR variations for ×4 SR.

Figure 3: The average PSNR and running time for different MoESR variations on DIV2KRK.

E   Visual comparison

Visual comparison between bicubic interpolation, KernelGAN [1] (+ ZSSR [7]), DAN [6], MoESR (ours) and ground-truth images are demonstrated in Figures 4, 5 and 7.
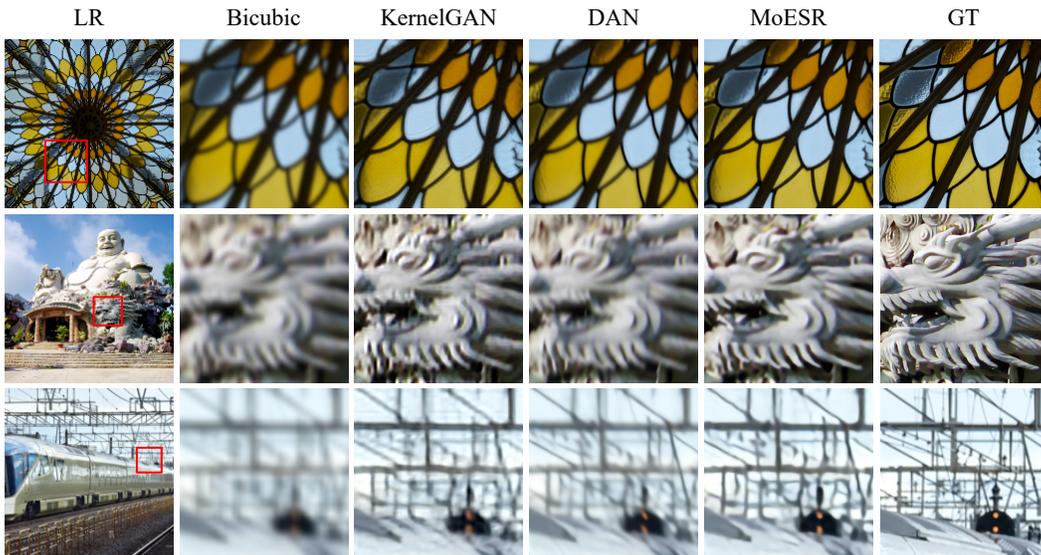

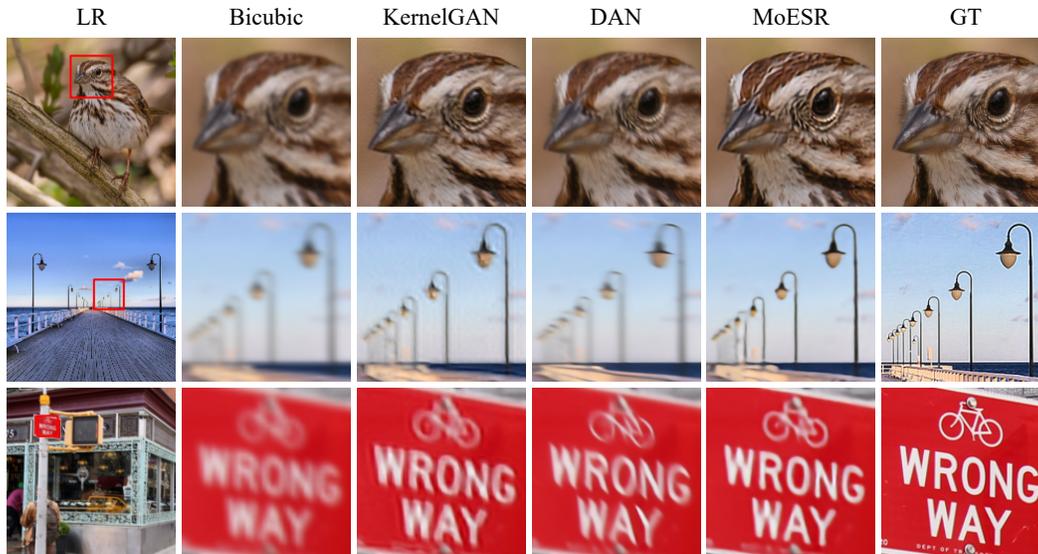Figure 4: Visual comparison of ×4 SR on DIV2KRK [1] images.

Figure 5: Visual comparison of $\times 4$ SR on Flickr2KRK [8] images.
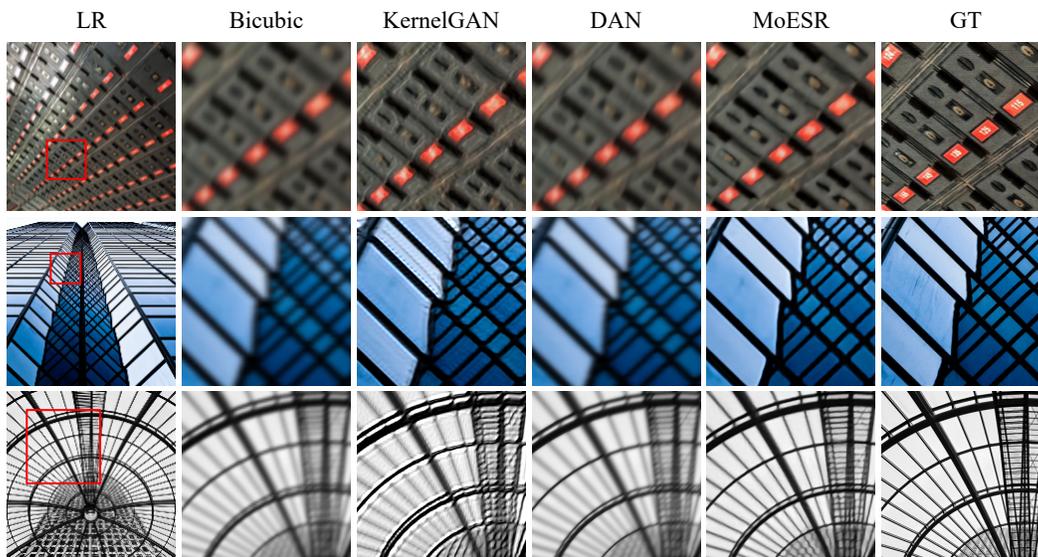

Figure 6: Visual comparison of $\times 4$ SR on Urban100RK [5] images.

References

[1] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. "Blind super-resolution kernel estimation using an internal-gan". In: *Advances in Neural Information Processing Systems*. 2019, pp. 284–293.

[2] Marco Bevilacqua et al. "Low-complexity single-image super-resolution based on nonnegative neighbor embedding". In: (2012).

[3] Jianrui Cai et al. "Toward real-world single image super-resolution: A new benchmark and a new model". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3086–3095.

[4] Mohammad Emad, Maurice Peemen, and Henk Corporaal. "DualSR: Zero-Shot Dual Learning for Real-World Super-Resolution". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1630–1639.

[5] 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*. 2015, pp. 5197–5206.
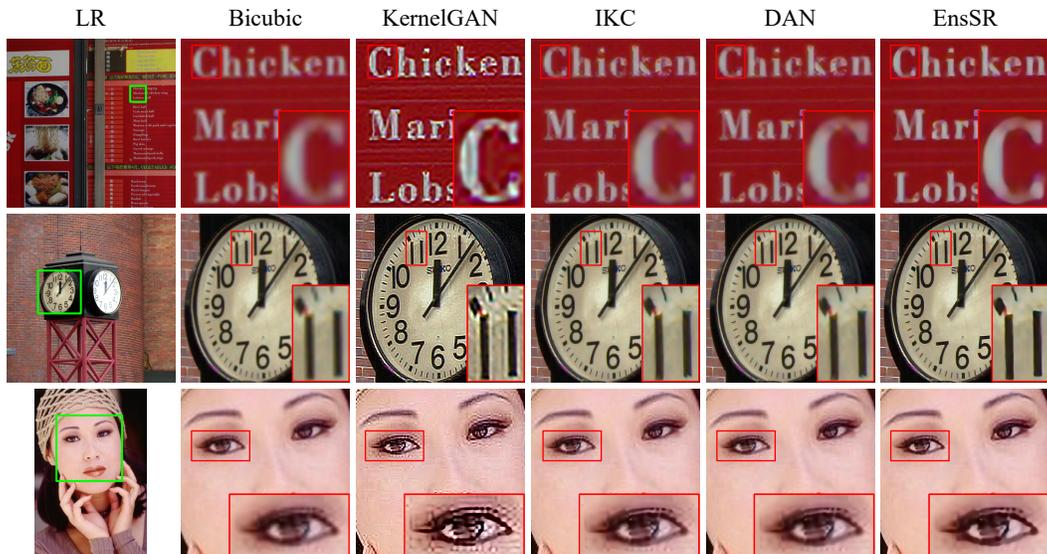
Figure 7: Visual comparison of ×4 SR on real-world images. The first two images are from RealSR [3] and the last image is from Set5 [2].

[6]   Yan Huang et al. "Unfolding the Alternating Optimization for Blind Super Resolution". In: *Advances in Neural Information Processing Systems* 33 (2020).

[7]   Assaf Shocher, Nadav Cohen, and Michal Irani. "Zero-shot super-resolution using deep internal learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3118–3126.

[8]   Radu Timofte et al. "Ntire 2017 challenge on single image super-resolution: Methods and results". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 114–125.