Supplementary Materials for Rethinking Image Super-Resolution from Training Data Perspectives

Go Ohtani[1,2], Ryu Tadokoro[2], Ryosuke Yamada[2,3], Yuki M. Asano[4], Iro Laina[5], Christian Rupprecht[5], Nakamasa Inoue[6,2], Rio Yokota[6,2], Hirokatsu Kataoka[2], and Yoshimitsu Aoki[1]

[1] Keio University
[2] National Institute of Advanced Industrial Science and Technology (AIST)
[3] University of Tsukuba
[4] University of Amsterdam, now at University of Technology Nuremberg
[5] University of Oxford
[6] Tokyo Institute of Technology

1 Implementation details

In this work, we adopted several representative models to compare the performance of different models. The hyperparameters used in training are shown in Table 1 and Table 2. The initial learning rate is set to 0.0002 for all models. MSRResNet is trained for 1,000k iterations, and the learning rate decreases as the number of iterations increases, resetting every 250k iterations. EDSR and RCAN are trained for 300k iterations and the learning rate is halved at 200k. SwinIR is trained for 500k iterations and the learning rate is halved at 250k, 400k, 450k, and 475k. HAT is trained for 800k iterations and the learning rate is halved at 300k, 500k, 650k, 700k, and 750k. We used the MultiStepLR scheduler and the CosineAnnealingRestartLR scheduler to manage the learning rates of different models. The MultiStepLR scheduler specifies the points at which the learning rate decreases through the milestones parameter and determines the factor by which it decreases through the gamma parameter. On the other hand, the CosineAnnealingRestartLR scheduler specifies the period of the learning rate through the periods parameter and adjusts the learning rate at the end of each

Table 1: List of hyperparameter values for each training model.

Parameter	MSRResNet [11]	RCAN [12]	EDSR [7]	SwinIR [6]	HAT [2]
Batch size	16	16	16	32	32
Iterations	1,000k	300k	300k	500k	800k
Optimizer	Adam	Adam	Adam	Adam	Adam
Loss function	L1 loss	L1 loss	L1 loss	L1 loss	L1 loss
Learning rate	0.0002	0.0002	0.0002	0.0002	0.0002
Scheduler	CosineAnnealingRestartLR	MultiStepLR	MultiStepLR	MultiStepLR	MultiStepLR
Milestones	-	[200k]	[200k]	[250k, 400k, 450k, 475k]	[300k, 500k, 650k, 700k, 750k]
Gamma	-	0.5	0.5	0.5	0.5
Periods	[250k, 250k, 250k, 250k]	-	-	-	-
Restart weights	[1, 1, 1, 1]	-	-	-	-

Table 2: List of patch size of ground truth at different scales in the training model.

Parameter	Scale	MSRResNet [11]	RCAN [12]	EDSR [7]	SwinIR [6]	HAT [2]
Patch size	×2	-	-	96	128	128
	×3	-	-	-	192	192
	×4	128	192	192	256	256

Table 3: Fair comparison of DiverSeg and DF2K with comparable numbers of pixels.

Scale	Training Data	Set14		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×2	DF2K	34.86	0.9274	34.45	0.9466	40.26	0.9809
×2	DiverSeg-I_S	**35.14**	**0.9283**	34.94	0.9498	**40.73**	**0.9819**
×2	DiverSeg-P_S	35.13	0.9275	**35.07**	**0.9502**	40.34	0.9807
×3	DF2K	31.08	0.8555	30.23	0.8896	35.53	0.9552
×3	DiverSeg-I_S	31.34	0.8561	30.59	0.8942	**35.73**	**0.9563**
×3	DiverSeg-P_S	**31.46**	**0.8566**	**30.76**	**0.8963**	35.67	0.9559
×4	DF2K	29.23	0.7973	27.97	0.8368	32.48	0.9292
×4	DiverSeg-I_S	29.34	0.7988	28.15	0.8418	**32.71**	**0.9311**
×4	DiverSeg-P_S	**29.38**	**0.7990**	**28.27**	**0.8447**	32.68	0.9310

period. The Restart weights represent the weights for adjusting the learning rate. The mini-batch size is set to 16 for MSRResNet, EDSR and RCAN, and 32 for SwinIR and HAT. The size of LR patches is set to 32×32 for MSRResNet, 48×48 for EDSR and RCAN, and 64×64 for SwinIR and HAT, in the ×4 SR setting. For EDSR, we use a pre-trained scale ×2 model. The HR-LR image pairs are created by the bicubic method of the MATLAB function. Data augmentation is applied to the training dataset, involving random rotations of 90, 180, and 270 degrees, as well as horizontal flipping.

2 Additional Experiments

Fair comparison with DF2K. To fairly compare with DF2K, we used image subsets which are randomly selected from DiverSeg-I and DiverSeg-P, matched to the pixel count of DF2K, and defined them as DiverSeg-I_S and DiverSeg-P_S. Experiments with HAT in the Table 3 show that these subsets outperform DF2K. These results suggest that diversity enhances SR performance.

Comparison with SOTA. Table 4 demonstrates that DiverSeg achieves state-of-the-art performance at ×2 and ×3 scales, similar to its performance at the ×4 scale. This confirms the effectiveness of our filtering approach across various scales.

Blockiness distribution. Figure 1 shows the blockiness distributions $p_{X,1.0}$ for DiverSeg-I, DiverSeg-P, and DiverSeg-IP and the basis distributions $p_{Z,q}$ for LSDIR [5].

Table 4: Quantitative comparison with state-of-the-art methods on five benchmark datasets. We applied our dataset to two Transformer-based models. Checkmarks for HR, LR indicate the use of high-resolution and low-resolution datasets, respectively.

Method	Scale	Training Data	HR	LR	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SAN [3]	×2	DIV2K	✓		38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
IGNN [13]	×2	DIV2K	✓		38.24	0.9613	34.07	0.9217	32.41	0.9025	33.23	0.9383	39.35	0.9786
HAN [10]	×2	DIV2K	✓		38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
NLSN [9]	×2	DIV2K	✓		38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789
RCAN-it [8]	×2	DF2K	✓		38.37	0.9620	34.49	0.9250	32.48	0.9034	33.62	0.9410	39.88	0.9799
EDT [4]	×2	DF2K	✓		38.45	0.9624	34.57	0.9258	32.52	0.9041	33.80	0.9425	39.93	0.9800
HAT-S [2]	×2	DF2K	✓		38.58	0.9628	34.70	0.9261	32.59	0.9050	34.31	0.9459	40.14	0.9805
IPT [1]	×2	ImageNet		✓	38.37	-	34.43	-	32.48	-	33.76	-	-	-
SwinIR [6]	×2	DF2K	✓		38.42	0.9623	34.46	0.9250	32.53	0.9041	33.81	0.9427	39.92	0.9797
SwinIR [6]	×2	DiverSeg-I (Ours)		✓	**38.50**	**0.9628**	**34.78**	**0.9268**	**32.63**	**0.9052**	**34.35**	**0.9460**	**40.40**	**0.9812**
HAT [2]	×2	DF2K	✓		38.63	0.9630	34.86	0.9274	32.62	0.9053	34.45	0.9466	40.26	0.9809
HAT [2]	×2	ImageNet→DF2K	✓	✓	38.73	**0.9637**	35.13	0.9282	32.69	0.9060	34.81	0.9489	40.71	0.9819
HAT [2]	×2	DiverSeg-I (Ours)		✓	**38.75**	0.9636	**35.18**	**0.9284**	**32.74**	**0.9065**	35.06	0.9502	**40.78**	**0.9821**
HAT [2]	×2	DiverSeg-IP (Ours)		✓	38.72	0.9634	**35.18**	0.9283	32.73	0.9064	**35.10**	**0.9504**	40.70	0.9819
SAN [3]	×3	DIV2K	✓		34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
IGNN [13]	×3	DIV2K	✓		34.72	0.9298	30.66	0.8484	29.31	0.8105	29.03	0.8696	34.39	0.9496
HAN [10]	×3	DIV2K	✓		34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
NLSN [9]	×3	DIV2K	✓		34.85	0.9306	30.70	0.8485	29.34	0.8117	29.25	0.8726	34.57	0.9508
RCAN-it [8]	×3	DF2K	✓		34.86	0.9308	30.76	0.8505	29.39	0.8125	29.38	0.8755	34.92	0.9520
EDT [4]	×3	DF2K	✓		34.97	0.9316	30.89	0.8527	29.44	0.8142	29.72	0.8814	35.13	0.9534
HAT-S [2]	×3	DF2K	✓		35.01	0.9325	31.05	0.8550	29.50	0.8158	30.15	0.8879	35.40	0.9547
IPT [1]	×3	ImageNet		✓	34.81	-	30.85	-	29.38	-	29.49	-	-	-
SwinIR [6]	×3	DF2K	✓		34.97	0.9318	30.93	0.8534	29.46	0.8145	29.75	0.8826	35.12	0.9537
SwinIR [6]	×3	DiverSeg-I (Ours)		✓	**35.00**	**0.9321**	**31.11**	**0.8544**	**29.50**	**0.8160**	**30.11**	**0.8868**	**35.40**	**0.9548**
HAT [2]	×3	DF2K	✓		35.07	0.9329	31.08	0.8555	29.54	0.8167	30.23	0.8896	35.53	0.9552
HAT [2]	×3	ImageNet→DF2K	✓	✓	**35.16**	**0.9335**	31.33	0.8576	29.59	0.8177	30.70	0.8949	**35.84**	**0.9567**
HAT [2]	×3	DiverSeg-I (Ours)		✓	35.14	0.9334	31.42	**0.8577**	**29.60**	0.8187	30.83	0.8965	35.80	0.9566
HAT [2]	×3	DiverSeg-IP (Ours)		✓	35.10	0.9333	**31.49**	0.8574	**29.60**	**0.8188**	**30.94**	**0.8982**	**35.84**	**0.9567**

3 More Discussions

Discussion of distortions. We experimented with rescaled, blurred, and noisy images to investigate robustness and applicability across various image distortions. We used DF2K, a dataset with minimal degradation, as the training dataset for adding distortions. Table 5 shows that blockiness values tend to remain unchanged, but the number of segments decreases when images are distorted. Note that the blockiness values slightly increased with $s = 8$, likely due to JPEG compression being performed in 8×8 blocks. This indicates that quality estimation primarily detects JPEG noise and that distorted images are mostly, but not entirely, filtered out by object-based filtering.

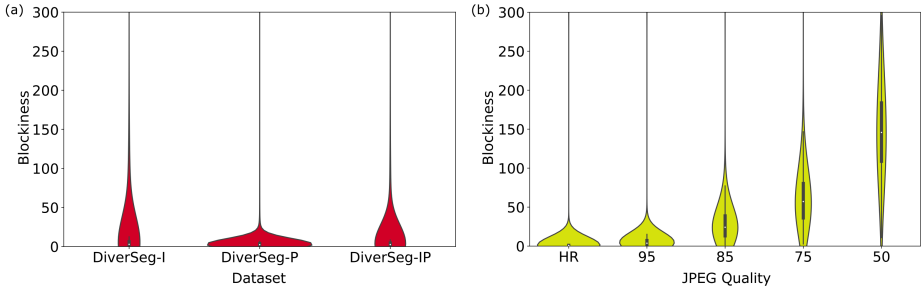

Fig. 1: (a) Blockiness distributions $p_{X,1.0}$ for $X =$ DiverSeg-I, DiverSeg-P and DiverSeg-IP. (b) Basis distributions $p_{Z,q}$ for $Z =$ LSDIR and $q = 0.5, 0.75, 0.85, 0.95, 1.0$.

Table 5: Comparison of the robustness of our filtering method against various image distortions.

Distortions	Parameter	Blockiness (median)	#Segments (average)
-	-	0.47	103
Rescaling[1]	$s = 8$	12.21	98
	$s = 12$	0.32	84
	$s = 16$	1.34	73
Blurring[2]	$\sigma_b = 4$	0.33	100
	$\sigma_b = 6$	0.31	89
	$\sigma_b = 8$	0.32	83
Noises[3]	$\sigma_n = 20$	0.34	98
	$\sigma_n = 30$	0.32	95
	$\sigma_n = 40$	0.32	91

[1] Bicubic interpolation: downscale and upscale by a scale factor s.
[2] Gaussian blur: set the kernel size to (21, 21) and control the blur intensity with the standard deviation of the Gaussian distribution σ_b.
[3] Color Gaussian noise: control the noise intensity with the standard deviation of the Gaussian distribution σ_n.

References

1. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: CVPR (2021) 3
2. Chen, X., Wang, X., Zhou, J., Qiao, Y., Dong, C.: Activating more pixels in image super-resolution transformer. In: CVPR (2023) 1, 2, 3
3. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR (2019) 3
4. Li, W., Lu, X., Qian, S., Lu, J., Zhang, X., Jia, J.: On efficient transformer-based image pre-training for low-level vision. arXiv preprint arXiv:2112.10175 (2021) 3
5. Li, Y., Zhang, K., Liang, J., Cao, J., Liu, C., Gong, R., Zhang, Y., Tang, H., Liu, Y., Demandolx, D., Ranjan, R., Timofte, R., Van Gool, L.: Lsdir: A large scale dataset for image restoration. In: CVPRW (2023) 2
6. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: ICCVW (2021) 1, 2, 3
7. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPRW (2017) 1, 2
8. Lin, Z., Garg, P., Banerjee, A., Magid, S.A., Sun, D., Zhang, Y., Van Gool, L., Wei, D., Pfister, H.: Revisiting rcan: Improved training for image super-resolution. arXiv preprint arXiv:2201.11279 (2022) 3
9. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: CVPR (2021) 3
10. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: ECCV (2020) 3
11. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCVW (2018) 1, 2
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 (2018) 1, 2
13. Zhou, S., Zhang, J., Zuo, W., Loy, C.C.: Cross-scale internal graph neural network for image super-resolution. Advances in neural information processing systems **33**, 3499–3509 (2020) 3