Supplementary Material for Universal Robustness via Median Randomized Smoothing for Real-World Super-Resolution

Zakariya Chaouai
zakariya.chaouai@cea.fr

Mohamed Tamaazousti
mohamed.tamaazousti@cea.fr

1. Ablation study

In this section, we conduct an ablation study to investigate the performance of the proposed CertSR method by removing each of the two main components to understand their contribution to the overall method. Specifically, we explore the effects of both the Median Randomized Smoothing (MRS) fine-tuning phase and the MRS inference phase (see Figure 2 in the main paper) and compare them with the global method that includes both (CertSR). In Table 1, we report the results of this study. We observe that both MRS components have a slight positive impact on the SR model. However, together these two components give much better results, leading us to the proposed method, CertSR.

We note that "ESRGAN" indicates the fine-tuning of ESRGAN [11] on the DIV2K dataset [1]. The "ESRGAN+MRS$_{FT}$" method involves fine-tuning the ESRGAN model using only Median Randomized Smoothing (MRS), while "ESRGAN+MRS$_{Inf}$" indicates the use directly of MRS in the inference phase of ESRGAN. Finally, CertSR is a combination of "ESRGAN+MRS$_{FT}$" and "ESRGAN+MRS$_{Inf}$".

Dataset	Metrics	SR Methods			
		ESRGAN	ESRGAN+MRS$_{FT}$	ESRGAN +MRS$_{Inf}$	CertSR
AIM	PSNR ↑	21.95	21.88	21.97	21.75
	SSIM ↑	0.55	0.56	0.53	0.59
	LPIPS ↓	0.51	0.47	0.48	0.33
NTIRE	PSNR ↑	21.94	26.90	22.16	26.67
	SSIM ↑	0.39	0.69	0.40	0.71
	LPIPS ↓	0.56	0.22	0.55	0.21

Table 1. **Ablation study**. We present the comparison of reference metrics between our method and each of their component independently. Red and blue colors highlight the best two scores.

Firstly, by examining Table 1, we observe an enhancement in the performance of "ESRGAN+MRS$_{FT}$" compared to "ESRGAN." This improvement is attributed to the fine-tuning phase where MRS introduces Gaussian random noise to the input images. This strategy fosters model invariance to small changes in the input, consequently enhancing generalization to previously unseen data. It is important to note, that due to the Gaussian data augmentation utilized in the fine-tuning phase, this method serves as an alternative to regularization in neural networks with the Jacobian of the model [2]. This alternative becomes especially valuable for SR tasks where applying Jacobian-based regularization is often impractical due to the substantial dimensions of the input and output. Secondly, we observe that the "ESRGAN+MRS$_{Inf}$" method also improves the performance of ESRGAN, particularly concerning the LPIPS metrics. However, this method is not as effective when applied independently; its efficacy increases notably when used after "ESRGAN+MRS$_{FT}$." This can be attributed to the sensitivity of ESRGAN to Gaussian noise.

2. CertSR with other SR models

In this section, we will test our CertSR method on some other SR models. The purpose of this study is to demonstrate that our method can enhance the precision and robustness of any SR model. Moreover, this enhancement comes at no additional cost. For this reason, we choose the SR models EDSR [7] and NINASR [6]. We will then apply the certification method to them (see Figure 2 in the main paper). We denote CertEDSR and CertNINASR as the models EDSR and NINASR after the certification process, respectively. In Table 2, we present the results that we obtained after and before the certification method on AIM [8] and NTIRE [1] datasets.

Dataset	Metrics	SR Methods			
		EDSR	CertEDSR	NINASR	CertNINASR
AIM	PSNR ↑	22.57	22.32	22.22	22.24
	SSIM ↑	0.60	0.53	0.59	0.61
	LPIPS ↓	0.60	0.57	0.60	0.49
NTIRE	PSNR ↑	25.57	26.67	24.79	27.61
	SSIM ↑	0.64	0.70	0.63	0.74
	LPIPS ↓	0.57	0.47	0.57	0.37

Table 2. We show a comparison of reference metrics between two SR models before and after applying the certification method that we propose.

In this study, similarly to ESRGAN, we fine-tune both the EDSR and NINASR models on the DIV2K training

dataset. This involves applying MRS_{FT} to both models with identical standard deviations, $\sigma_1 = 0.03$ and $\sigma_1 = 0.2$, corresponding to the Gaussians samples. Next, we apply MRS_{inf} to both models. To be specific, we draw 21 i.i.d Gaussians samples with a standard deviation of $\sigma = 0.1$ to derive CertEDSR and CertNINASR results on the AIM dataset. Regarding the results on the NTIRE datasets, we maintain the same number of draws and we use $\sigma = 0.005$.

3. Comparison with RSR via regularization

In this section, we will regularize the ESRGAN neural network with the gradient of the loss function, a well-known method to ensure the stability of the neural network against input corruption and perturbation. In addition, this method allows for penalizing large changes in the output neural network model, enforcing a smoothness prior. This method has been employed in several works focused on classification tasks, as seen in, for instance, [5, 9, 10].

We recall that the loss function used to train or to fine-tune the ESRGAN is given by

$$L_{total} = L_{1,perc} + L_{adv}.$$

where, $L_{1,perc} = L_1 + L_{perc}$. Here, L_1 loss is the pixel distance, L_{perc} is the perceptual loss, and L_{adv} is the adversarial loss. Due to the gradient regularization that we will apply, the new total loss function becomes as follows:

$$L_{reg} = L_{total} + \lambda * \|\nabla_x L_{1,perc}\|, \qquad (s_1)$$

where λ is a hyperparameter. It is important to point out that the method we use in this part is similar to the regularization used in [4]. Besides, we regularize with the gradient of L_1 and L_{perc} because our aim is to get a robust SR model both pixel-wise and perceptually.

Dataset	Metrics	SR Methods			
		ESRGAN	AD-L-PGD	ESRGAN-Reg	CertSR
AIM	PSNR ↑	21.91	21.99	21.97	21.75
	SSIM ↑	0.55	0.60	0.55	0.59
	LPIPS ↓	0.51	0.37	0.50	0.33
NTIRE	PSNR ↑	21.94	24.31	21.69	26.67
	SSIM ↑	0.39	0.65	0.38	0.71
	LPIPS ↓	0.56	0.23	0.57	0.21

Table 3. We present the comparison of reference metrics between RSR via gradient regularization, RSR via adversarial learning with PGD attack, ESRGAN and our CertSR

The result given from this study is shown in Table 3, where we compare this method of regularization, denoted as ESRGAN-Reg, with other methods such as ADV-L-PGD [3], constructed via adversarial learning using the PGD attack, ESRGAN fine-tuned in DIV2K, and our CertSR. We note that in our experiment, the best hyperparameter that yielded good results is $\lambda = 0.001$. On the other hand, from Table 3, we can deduce that this method of robustness is not very efficient in the SR task, notably for real-world SR.

4. Hyperparametrs for Median Randomized Smoothing (MRS)

In this section, we explore the impact of the hyperparameters for the proposed MRS fine-tuning and MRS inference, as shown in Figure 2 in the main paper.

4.1. Hyperparametrs for MRS fine-tuning

The MRS fine-tuning method has been done on DIV2K training dataset. However, for the validation of this method, we did it in AIM and NTIRE validation dataset. We would like to emphasize that in this phase, we chose two types of Gaussian samples, with each sample corresponding to a standard deviation. Additionally, for each Gaussian sample, we drew it two times randomly. In Table 4 we show the impact of the hyperparameters σ_1 and σ_2 on the performance of the MRS fine-tuning phase, validated on the AIM and NTIRE validation datasets based on LPIPS metric.

Dataset	Metric	Std σ_1	Std σ_2					
			0.01	0.02	0.03	0.04	0.05	0.06
AIM	LPIPS	0.1	0.48	0.48	0.48	0.48	0.48	0.48
		0.2	0.49	0.48	**0.47**	0.47	0.48	0.48
		0.3	0.48	0.48	0.49	0.48	0.48	0.48
		0.4	0.49	0.49	0.48	0.47	0.48	0.48
		0.5	0.49	0.48	0.49	0.49	0.48	0.48
		0.6	0.48	0.48	0.48	0.48	0.48	0.48
NTIRE	LPIPS	0.1	0.30	0.26	0.24	0.23	0.25	0.25
		0.2	0.33	0.26	**0.22**	0.24	0.24	0.25
		0.3	0.36	0.27	0.22	0.24	0.24	0.26
		0.4	0.37	0.28	0.24	0.26	0.27	0.28
		0.5	0.40	0.25	0.22	0.24	0.26	0.30
		0.6	0.40	0.27	0.23	0.26	0.27	0.29

Table 4. We report the impact of the hyperparameters σ_1 and σ_2 on the performance of the MRS fine-tuning phase, validated on the AIM and NTIRE validation datasets.

4.2. Hyperparametrs for MRS Inference

After the MRS fine-tuning, We represent the performance of the MRS inference against the adversarial attacks on the DIV2K validation dataset and the real-world validation datasets.

In Table 5, we show the impact of the hyperparameter σ on the performance of MRS_{inf} validated on the AIM and NTIRE validation datasets based on PSNR, SSIM, and LPIPS metrics. We point out that the number of draws used in the inference phase is the same, which is 21.

In Table 6, we present the impact of the hyperparameter σ on the performance of MRS_{inf} validated on the AIM

attack	Metrics	Hyperparameter σ								
		0.005	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
FGSM	PSNR	19.73	19.92	20.73	21.74	22.95	24.11	24.72	24.92	24.92
	SSIM	0.35	0.36	0.40	0.46	0.53	0.60	0.64	0.65	0.65
	LPIPS	0.48	0.48	0.44	0.39	0.34	0.29	**0.27**	0.28	0.30
BIM	PSNR	17.38	17.61	18.60	19.72	20.10	22.35	23.53	24.28	24.60
	SSIM	0.28	0.29	0.33	0.38	0.45	0.53	0.60	0.64	0.65
	LPIPS	0.56	0.55	0.51	0.47	0.41	0.33	0.27	**0.25**	0.27
PGD	PSNR	22.15	22.68	23.91	24.42	24.62	24.85	25.09	25.19	25.15
	SSIM	0.47	0.51	0.60	0.64	0.65	0.66	0.67	0.67	0.68
	LPIPS	0.50	0.46	0.38	0.32	0.28	0.25	**0.24**	0.25	0.28
CW	PSNR	21.69	24.87	26.46	**26.66**	26.48	26.25	26.09	25.94	25.73
	SSIM	0.48	0.58	0.65	0.71	0.72	0.71	0.70	0.69	0.68
	LPIPS	0.38	0.22	0.19	**0.18**	0.18	0.19	0.21	0.24	0.27

Table 5. We present the performance of the MRS inference phase, on attacked DIV2K validation dataset.

and NTIRE validation datasets based on PSNR, SSIM, and LPIPS metrics. The number of draws used in the inference phase is also 21.

Dataset	Metrics	Hyperparameter σ								
		0.005	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
AIM	PSNR	21.91	22.07	22.17	22.07	21.90	21.77	21.75	21.98	22.01
	SSIM	0.57	0.60	0.61	0.61	0.60	0.59	0.59	0.60	0.60
	LPIPS	0.46	0.45	0.42	0.38	0.36	0.34	**0.33**	0.34	0.36
NTIRE	PSNR	26.86	26.93	27.02	26.67	26.41	26.17	26.29	26.15	25.80
	SSIM	0.69	0.70	0.71	**0.71**	0.70	0.69	0.68	0.68	0.69
	LPIPS	0.23	0.22	0.22	**0.21**	0.21	0.22	0.24	0.27	0.28

Table 6. We report the impact of the hyperparameters σ on the performance of the MRS inference phase, based on reference metrics validated on the AIM and NTIRE validation datasets.

5. Hyperparameters for adversarial Learning

In this section, we explore the impact of the hyperparameters for the proposed adversarial learning methods based on adversarial attacks (FGSM, BIM, and CW) that we use to build RSR models.

5.1. Adversarial Learning with FGSM (AD-L-FGSM)

In Table 7, we present the results of the AD-L-FGSM model for different values of the hyperparameter of the FGSM adversarial attack, which is ϵ, representing the step size for the allowed perturbation. We report results on the AIM and NTIRE datasets for different metrics, namely PSNR, SSIM, and LPIPS.

5.2. Adversarial Learning with BIM (AD-L-BIM)

In Table 8, we present the results of the AD-L-BIM model for different values of the hyperparameters of the BIM adversarial attack. The hyperparameters of this attack are composed of α, which represent the step of the perturbations are and T the number of iterations. We report the results on the AIM and NTIRE datasets with respect to different metrics, namely PSNR, SSIM, and LPIPS.

Dataset	Metrics	Hyperparameter ϵ				
		$1/255$	$3/255$	$6/255$	$9/255$	$10/255$
AIM	PSNR	22.18	22.59	22.64	22.70	22.77
	SSIM	0.56	0.60	0.62	0.63	0.62
	LPIPS	0.44	0.42	0.43	0.42	0.46
NTIRE	PSNR	22.98	23.50	24.66	25.55	25.50
	SSIM	0.46	0.49	0.57	0.65	0.64
	LPIPS	0.46	0.44	0.35	0.30	0.32

Table 7. We present the performance of the AD-L-FGSM model for different values of the hyperparameter ϵ on the AIM and NTIRE validation datasets with respect to reference metrics.

Dataset	Metrics	Iteration T	Hyperparameter α				
			1/255	3/255	6/255	9/255	10/255
AIM	PSNR	2	22.36	18.16	16.87	22.31	17.93
		3	22.71	17.89	17.51	17.64	18.03
		4	22.26	16.75	18.11	17.85	17.29
		5	17.57	16.32	16.44	18.19	19.05
	SSIM	2	0.61	0.29	0.29	0.59	0.29
		3	0.62	0.39	0.30	0.28	0.35
		4	0.60	0.22	0.32	0.29	0.27
		5	0.30	0.22	0.21	0.30	0.40
	LPIPS	2	0.46	0.68	0.76	0.36	0.73
		3	0.45	0.76	0.80	0.86	0.79
		4	0.47	0.75	0.70	0.74	0.82
		5	0.86	0.87	0.72	0.71	0.63
NTIRE	PSNR	2	25.53	18.37	17.02	25.35	18.62
		3	25.62	23.55	17.84	18.31	18.29
		4	25.56	18.49	18.59	18.05	17.79
		5	17.77	24.06	16.83	18.93	20.03
	SSIM	2	0.64	0.23	0.24	0.63	0.28
		3	0.65	0.48	0.25	0.27	0.30
		4	0.64	0.28	0.28	0.25	0.26
		5	0.27	0.51	0.20	0.27	0.40
	LPIPS	2	0.34	0.69	0.76	0.26	0.72
		3	0.33	0.41	0.77	0.83	0.80
		4	0.33	0.76	0.71	0.74	0.80
		5	0.85	0.40	0.71	0.70	0.61

Table 8. We present the performance of the AD-L-BIM model for different values of the hyperparameters α (the step of the adversarial attack) and T (number of iterations) on the AIM and NTIRE validation datasets with respect to reference metrics.

5.3. Adversarial Learning with CW (AD-L-CW)

In Table 9, we present the results of the AD-L-CW model for different values of the hyperparameters of the CW adversarial attack. The hyperparameters of this attack are composed of c, which controls the trade-off between the L2 norm of the perturbation and T the number of iterations to minimize the following problem:

$$\min_{\delta}(\|\delta\|_2 - c \cdot \mathcal{L}(f_\theta(x), y)), \text{ such that } x+\delta \in [0,1]^n. \quad (s_2)$$

We report the results on the AIM and NTIRE datasets with respect to different metrics, namely PSNR, SSIM, and LPIPS.

Dataset	Metrics	Iterations T	Hyperparameter c	
			10^{-2}	1
AIM	PSNR	1	21.51	4.60
		2	5.35	4.59
		3	4.64	4.58
		4	21.86	5.21
		5	4.72	5.37
	SSIM	1	0.52	0.11
		2	0.12	0.02
		3	0.01	0.23
		4	0.58	0.06
		5	0.22	0.07
	LPIPS	1	0.51	1.01
		2	1.06	0.91
		3	1.09	1.16
		4	0.47	1.13
		5	0.99	1.06
NTIRE	PSNR	1	20.87	4.60
		2	5.27	4.59
		3	4.65	4.57
		4	21.25	4.99
		5	4.72	5.00
	SSIM	1	0.32	0.11
		2	0.12	0.01
		3	0.03	0.06
		4	0.37	0.01
		5	0.24	0.01
	LPIPS	1	0.67	1.01
		2	1.06	0.91
		3	1.16	1.28
		4	0.63	1.30
		5	0.99	1.22

Table 9. We present the performance of the AD-L-CW model for different values of the hyperparameters c (controls the trade-off between the L2 norm of the perturbation) and T (number of iterations to minimize s_2) on the AIM and NTIRE validation datasets with respect to reference metrics.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 1

[2] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995. 1

[3] Angela Castillo, Juan Escobar, María C. Pérez, Andrés Romero, Radu Timofte, Luc Van Gool, and Pablo Arbelaez. Generalized real-world super-resolution through adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1855–1865, 2021. 2

[4] Jun-Ho Choi, Huan Zhang, Cho-Jui Kim, Jun-Hyuk Hsieh, and Jong-Seok Lee. Adversarially robust deep image super-resolution using entropy regularization. In *Proceedings of the the Asian Conference on Computer Vision*, 2020. 2

[5] Harris Drucker and Yann Le Cun. Double backpropagation increasing generalization performance. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, pages 145–150. IEEE, 1991. 2

[6] Gabriel Gouvine. torchSR: A pytorch-based framework for single image super-resolution. https://github.com/Coloquinte/torchSR/blob/main/doc/NinaSR.md, 2023. 1

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1

[8] Andreas Lugmayr, Danelljan Martin, Radu Timofte, Manuel Fritsche, Shuhang Gu, Kuldeep Purohit, Praveen Kandula, Suin Maitreya, A. N. Rajagoapalan, Joon Nam Hyung, Won Yu Seung, Kim Guisik, Kwon Dokyeong, Hsu Chih-Chung, Lin Chia-Hsiang, Huang Yuanfei, Sun Xiaopeng, Lu Wen, Li Jie, Gao Xinbo, Bell-Kligler Sefi, Assaf Shocher, and Irani Michal. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3575–3583, 2019. 1

[9] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017. 2

[10] Dániel Varga, Adrián Csiszárik, and Zsolt Zombori. Gradient regularization improves accuracy of discriminative models. *arXiv preprint arXiv:1712.09936*, 2017. 2

[11] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018. 1