Professorship of Machine Learning
Department of Electrical and Computer Engineering
Technical University of Munich

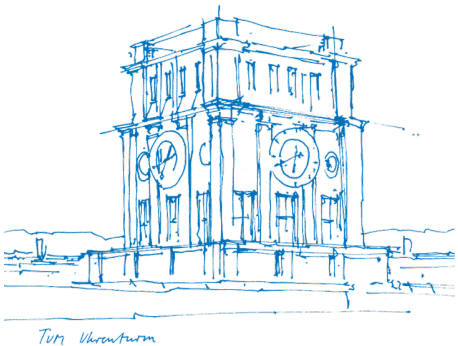
Randomized Smoothing as an Adversarial Defense Mechanism for Inverse Problems

Bachelor's Thesis
Supervisor: Prof. Dr. Reinhard Heckel

Faidra A. Patsatzi

Professorship of Machine Learning
Department of Electrical and Computer Engineering
Technical University of Munich

July 4th, 2023



Outline

Adversarial Robustness in Classification Problems

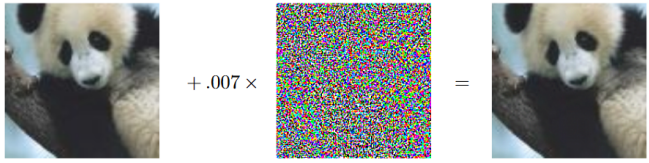


Figure 1 Classification of the original image of a panda (left), adversarial perturbation (middle), and misclassification of the adversarially perturbed image (right) [1].

- Adversarial defense mechanisms, such as adversarial training, are employed to achieve adversarial robustness of neural networks
- **Randomized smoothing: state-of-the-art defense** against ℓ_2-norm bounded adversarial perturbations for neural network-based **classifiers**

Adversarial Robustness in Inverse Problems

Inverse Problems

An **inverse problem** can be formulated as

$$y = \mathcal{A}(x) + z\,, \tag{1}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $\mathcal{A}$ is a forward, typically non-invertible transformation applied to the input, and z is a noise vector [2].

$\rightarrow$ **Single Image Super-Resolution**: reconstruct a high resolution image from a single low resolution image.

Goal

Evaluate the effectiveness of randomized smoothing as an adversarial defense in an image reconstruction problem and compare it to adversarial training.

Adversarial Robustness in Inverse Problems

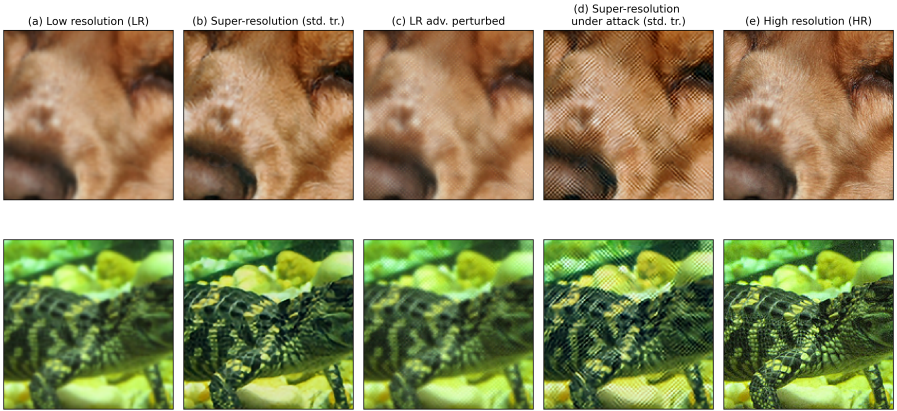


Figure 2 Super-resolution with unperturbed (b) and adversarially perturbed (d) images from Mini-ImageNet-1000 [3] with a U-Net trained with standard training.

Outline

Randomized Smoothing for Classification Problems

For a given classifier $f : \mathbb{R}^n \to Q$, where Q is the set of classes, the **smoothed classifier** $g : \mathbb{R}^n \to Q$ [4] is defined as

$$g(x) = \arg\max_{c \in Q} \mathbb{P}\big[f(x + \delta) = c\big] \quad \text{where} \quad \delta \sim \mathcal{N}(0, \sigma^2 I)\,. \tag{2}$$

Equivalent definition of the smoothed classifier g:

$$g(x) = \arg\min_{c \in Q} \mathbb{E}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} \Big[\mathbb{1}\{f(x + \delta) \neq c\}\Big]\,. \tag{3}$$

Approximation of the smoothed classifier g with a **majority vote over k noisy predictions of the base classifier** f [4]:

$$\hat{g}(x) = \arg\max_{c \in Q} \sum_{j=1}^{k} \mathbb{1}\{f(x + \delta_j) = c\}\,. \tag{4}$$

Randomized Smoothing for Inverse Problems

For an input vector $x \in \mathbb{R}^n$, an observation $y \in \mathbb{R}^m$, and a given neural network $f : \mathbb{R}^m \to \mathbb{R}^n$ we define the **smoothed estimate** g as

$$g(y) = \arg\min_x \mathbb{E}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} \left[\|f(y+\delta) - x\|_2^2 \right] \tag{5}$$

$$\implies \quad g(y) = \mathbb{E}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} [f(y+\delta)] \, . \tag{6}$$

Approximation of the smoothed estimate g by **averaging over k noisy predictions of the base estimator f**:

$$\hat{g}(y) = \frac{1}{k} \sum_{j=1}^{k} f(y + \delta_j) \, . \tag{7}$$

Outline

Experimental Setup

- Dataset: Mini-ImageNet-1000
  - Training set: 34,745 images
  - Test set: 3,923 images
- Pre-processing
  - Crop to 160x160 pixels
  - Generate low resolution images with bilinear downsampling
  - Normalization
- U-Net architecture
  - Feature channel sizes (8, 16, 32, 64, 128)
  - Input: 160x160 low resolution image
  - Output: 160x160 super-resolved image

Model Training

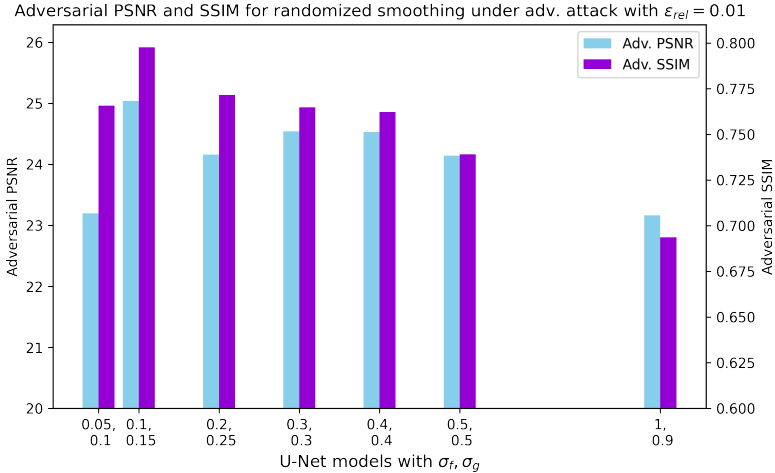
- Standard training
- For randomized smoothing:
  - Train base U-Net estimators f_θ with Gaussian noise augmentation $\sigma_f \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1\}$
  - Approximate smoothed estimate with Gaussian noise of level σ_g:

$$\hat{g}(y) = \frac{1}{k} \sum_{j=1}^{k} f_\theta(y + \delta_j) \quad \text{where} \quad \delta_j \sim \mathcal{N}(0, \sigma_g^2 I)\,. \tag{8}$$

  $\rightarrow \sigma_g$ is a hyperparameter and was optimized for each base U-Net estimator f_θ and adversarial noise level ϵ_{rel}
- Adversarial training:
  - Train U-Net models with adversarial noise $\epsilon_{\text{rel}} \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$
  - Adversarial examples computed with Projected Gradient Descent (PGD)

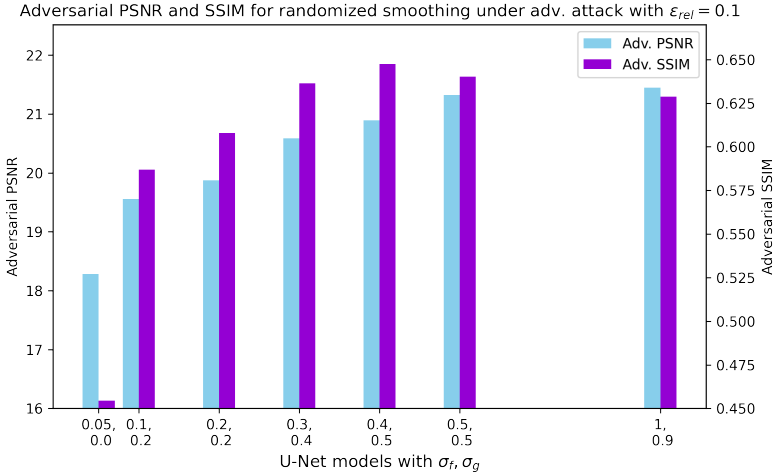
Results

Robustness of randomized smoothing models to **small** adversarial perturbations:



Results

Robustness of randomized smoothing models to **larger** adversarial perturbations:



Results

Comparison between randomized smoothing and adversarial training for a range of adversarial attacks:

		$\epsilon_{\text{rel}} = 0.01$			$\epsilon_{\text{rel}} = 0.05$			$\epsilon_{\text{rel}} = 0.1$		
	σ_f	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM
Rand. smoothing	0.05	0.0048	23.20	0.7657	0.0098	20.11	0.6148	0.0150	18.29	0.4545
	0.1	**0.0032**	**25.04**	0.7976	0.0064	21.94	0.7115	0.0112	19.55	0.5868
	0.2	0.0039	24.16	0.7716	0.0069	21.66	0.6873	0.0103	19.87	0.6080
	0.3	0.0036	24.54	0.7647	0.0054	22.72	0.7064	0.0088	20.59	0.6364
	0.4	0.0036	24.53	0.7622	0.0052	22.89	**0.7127**	0.0082	20.89	**0.6475**
	0.5	0.0039	24.14	0.7390	0.0051	22.95	0.6979	0.0074	21.32	0.6402
	1	0.0049	23.17	0.6937	0.0056	22.52	0.6686	0.0072	21.44	0.6288
Std. tr.	ϵ_{rel}									
	0	0.0052	22.87	0.7810	0.0118	19.31	0.6035	0.0187	17.30	0.5859
Adversarial tr.	0.01	0.0033	24.86	**0.8114**	0.0067	21.77	0.6200	0.0118	19.30	0.4763
	0.02	0.0034	24.69	0.8047	0.0060	22.24	0.6714	0.0099	20.07	0.5484
	0.05	0.0037	24.42	0.7784	**0.0049**	**23.17**	0.7035	0.0073	21.38	0.5735
	0.1	0.0045	23.53	0.7359	0.0054	22.75	0.6981	**0.0070**	**21.62**	0.6311
	0.2	0.0054	22.69	0.6890	0.0062	22.15	0.6633	0.0075	21.29	0.6290

Visual Results

Randomized smoothing for $\sigma_f = 0.1$ and $\sigma_g = 0.15$, adversarial attack with $\epsilon_{\text{rel}} = 0.02$

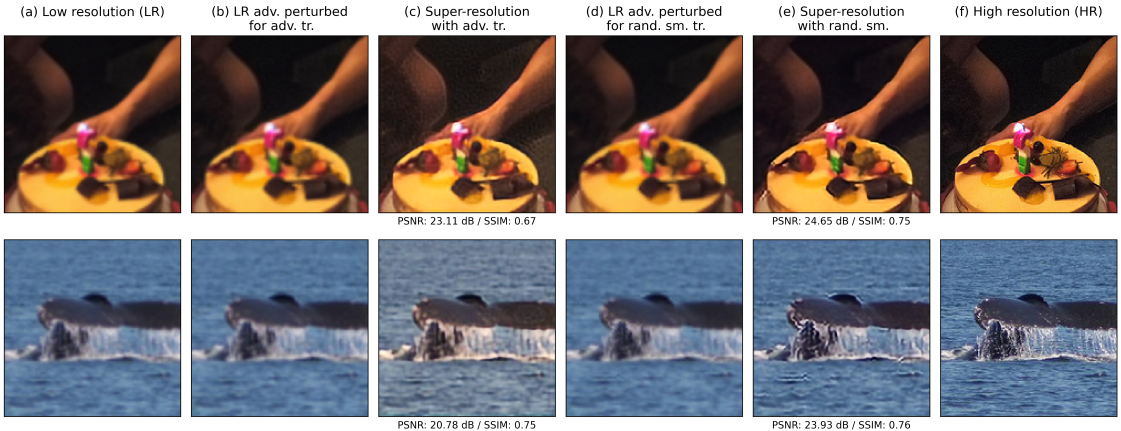


Figure 3 Adversarial attack with $\epsilon_{\text{rel}} = 0.02$ on super-resolution models using images from Mini-ImageNet-1000 [3].

Visual Results

Randomized smoothing for $\sigma_f = 0.1$ and $\sigma_g = 0.15$, adversarial attack with $\epsilon_{\text{rel}} = 0.02$

Figure 4 Adversarial attack with $\epsilon_{\text{rel}} = 0.02$ on super-resolution models using images from Mini-ImageNet-1000 [3].

Visual Results

Randomized smoothing for $\sigma_f = 0.4$ and $\sigma_g = 0.5$, adversarial attack with $\epsilon_{\text{rel}} = 0.1$

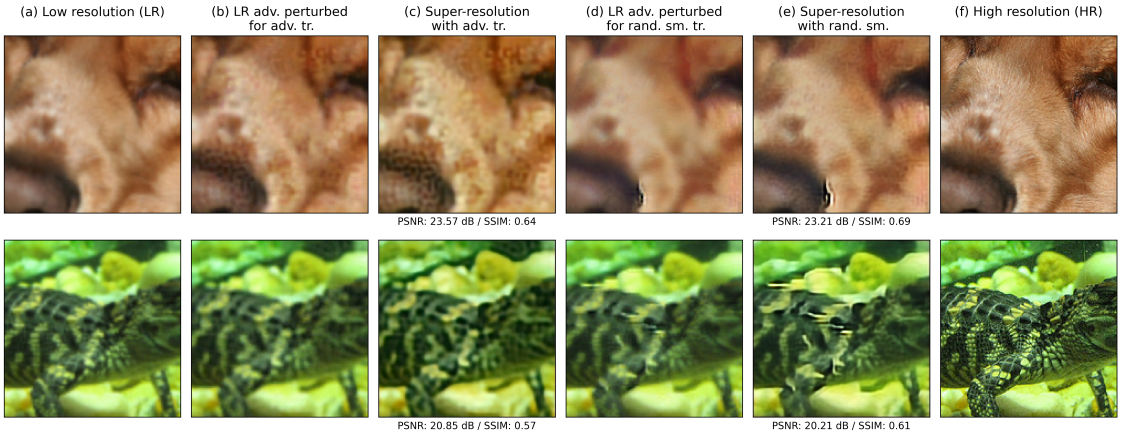


Figure 5 Adversarial attack with $\epsilon_{\text{rel}} = 0.1$ on super-resolution models using images from Mini-ImageNet-1000 [3].

Outline

Conclusion

- Randomized smoothing is an effective adversarial defense in image super-resolution
- Robustness/accuracy trade-off controlled by σ_f, σ_g
- Randomized smoothing achieves overall better visual quality than adversarial training

References

- I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015.
- G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, “Deep learning techniques for inverse problems in imaging,” 2020.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Conference Proceedings.
- J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, “Certified adversarial robustness via randomized smoothing,” 2019.
- A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2019.

Adversarial attacks

- Adversarial attacks are computed for ℓ_2-norm adversarial perturbation radius ϵ:

$$\epsilon = \sqrt{d} \cdot \epsilon_{\text{rel}} \,, \tag{9}$$

  where d equals the product of the input's dimensions $d = c \cdot h \cdot w$.

- Projected Gradient Descent: the computation of an adversarial example $\check{y}_i = y_i + e_i$ follows the multi-step scheme

$$\check{y}_i^{\,t+1} = \Pi_{y_i+B}\left(\check{y}_i^{\,t} + \alpha \frac{\nabla_{\check{y}_i^{\,t}}(\|f_\theta(\check{y}_i^{\,t}) - x_i\|_2^2)}{\left\|\nabla_{\check{y}_i^{\,t}}(\|f_\theta(\check{y}_i^{\,t}) - x_i\|_2^2)\right\|_2}\right), \tag{10}$$

  where $\mathbf{B} = \{e_i \mid \|e_i\|_2 < \epsilon\}$ is a constraint set for the perturbations, α is the step size, t is the descent step, and $\check{y}_i^{\,t+1}$ is the adversarial example computed in step $t+1$ [5].

Robustness/Accuracy Trade-Off

Table 1 Evaluation of model performance for a downsampling factor of 2 and adversarial attacks of perturbation level $\epsilon_{\text{rel}} \in \{0, 0.01, 0.02\}$. The best results for each column are marked in bold.

		Original input ($\epsilon_{\text{rel}} = 0$)			$\epsilon_{\text{rel}} = 0.01$			$\epsilon_{\text{rel}} = 0.02$		
	σ_f	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM
Rand. smoothing	0.05	0.0030	25.34	0.8133	0.0048	23.20	0.7657	0.0061	22.18	0.7274
	0.1	**0.0028**	**25.55**	0.8140	**0.0032**	**25.04**	0.7976	**0.0038**	**24.26**	**0.7776**
	0.2	0.0033	24.87	0.7922	0.0039	24.16	0.7716	0.0048	23.26	0.7482
	0.3	0.0033	24.88	0.7763	0.0036	24.54	0.7647	0.0039	24.12	0.7501
	0.4	0.0034	24.76	0.7706	0.0036	24.53	0.7622	0.0038	24.20	0.7513
	0.5	0.0037	24.33	0.7462	0.0039	24.14	0.7390	0.0041	23.89	0.7302
	1	0.0048	23.27	0.6983	0.0049	23.17	0.6937	0.0050	23.04	0.6884
	ϵ_{rel}									
Std. tr.	0	0.0030	25.29	**0.8562**	0.0052	22.87	0.7810	0.0071	21.53	0.7159
Adversarial tr.	0.01	0.0029	25.51	0.8399	0.0033	24.86	**0.8114**	0.0040	24.02	0.7651
	0.02	0.0031	25.13	0.8242	0.0034	24.69	0.8047	0.0039	24.14	0.7766
	0.05	0.0035	24.65	0.7906	0.0037	24.42	0.7784	0.0039	24.15	0.7637
	0.1	0.0043	23.69	0.7432	0.0045	23.53	0.7359	0.0047	23.36	0.7276
	0.2	0.0053	22.78	0.6949	0.0054	22.69	0.6890	0.0056	22.58	0.6828

Robustness/Accuracy Trade-Off

Table 2 Evaluation of model performance for a downsampling factor of 2 and adversarial attacks of perturbation level $\epsilon_{\text{rel}} \in \{0.05, 0.1, 0.2\}$. The best results for each column are marked in bold.

		$\epsilon_{\text{rel}} = 0.05$			$\epsilon_{\text{rel}} = 0.1$			$\epsilon_{\text{rel}} = 0.2$		
	σ_f	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM	MSE	PSNR (dB)	SSIM
Rand. smoothing	0.05	0.0098	20.11	0.6148	0.0150	18.29	0.4545	0.0239	16.28	0.2434
	0.1	0.0064	21.94	0.7115	0.0112	19.55	0.5868	0.0210	16.87	0.3725
	0.2	0.0069	21.66	0.6873	0.0103	19.87	0.6080	0.0183	17.41	0.4595
	0.3	0.0054	22.72	0.7064	0.0088	20.59	0.6364	0.0168	17.76	0.5012
	0.4	0.0052	22.89	**0.7127**	0.0082	20.89	**0.6475**	0.0149	18.27	0.5422
	0.5	0.0051	22.95	0.6979	0.0074	21.32	0.6402	0.0131	18.85	0.5349
	1	0.0056	22.52	0.6686	0.0072	21.44	0.6288	0.0114	19.45	**0.5495**
Std. tr.	ϵ_{rel}									
	0	0.0118	19.31	0.6035	0.0187	17.30	0.5859	0.305	15.17	0.5354
Adversarial tr.	0.01	0.0067	21.77	0.6200	0.0118	19.30	0.4763	0.0245	16.13	0.3595
	0.02	0.0060	22.24	0.6714	0.0099	20.07	0.5484	0.0179	17.49	0.4012
	0.05	**0.0049**	**23.17**	0.7035	0.0073	21.38	0.5735	0.0145	18.39	0.3816
	0.1	0.0054	22.75	0.6981	**0.0070**	**21.62**	0.6311	0.0113	19.49	0.4928
	0.2	0.0062	22.15	0.6633	0.0075	21.29	0.6290	**0.0113**	**19.53**	0.5493

Certified Robustness via Randomized Smoothing

p_A: probability that $f(x+\delta)$ yields the class c_A
p_B: probability that $f(x+\delta)$ yields the next most probable class c_B

The following ℓ_2 robustness radius can be derived:

$$R = \frac{\sigma}{2}\Big(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\Big)\,, \tag{11}$$

with Φ^{-1} representing the inverse standard Gaussian CDF [4].