Robust Classification by Coupling Data Mollification with Label Smoothing

Markus Heinonen [1] Ba-Hien Tran [2] Michael Kampffmeyer [3] Maurizio Filippone [4]

[1]Aalto University [2]Huawei Paris Research Center [3]UiT The Arctic University of Norway [4]KAUST

Overview

- We propose **Supervised Mollification**: training a classifier under **noisy input** and **smoothed label** augmentation
- We present a probabilistic model of mollification by **Dirichlet tempering**
- We demonstrate **improved calibration** and performance in image classification

Augmentations and smoothing

- **Augmentations** are key to achieve high accuracy on image classification [3]
- **Label smoothing** relaxes the cross-entropy loss
- What's the connection between augmentations and label smoothing?

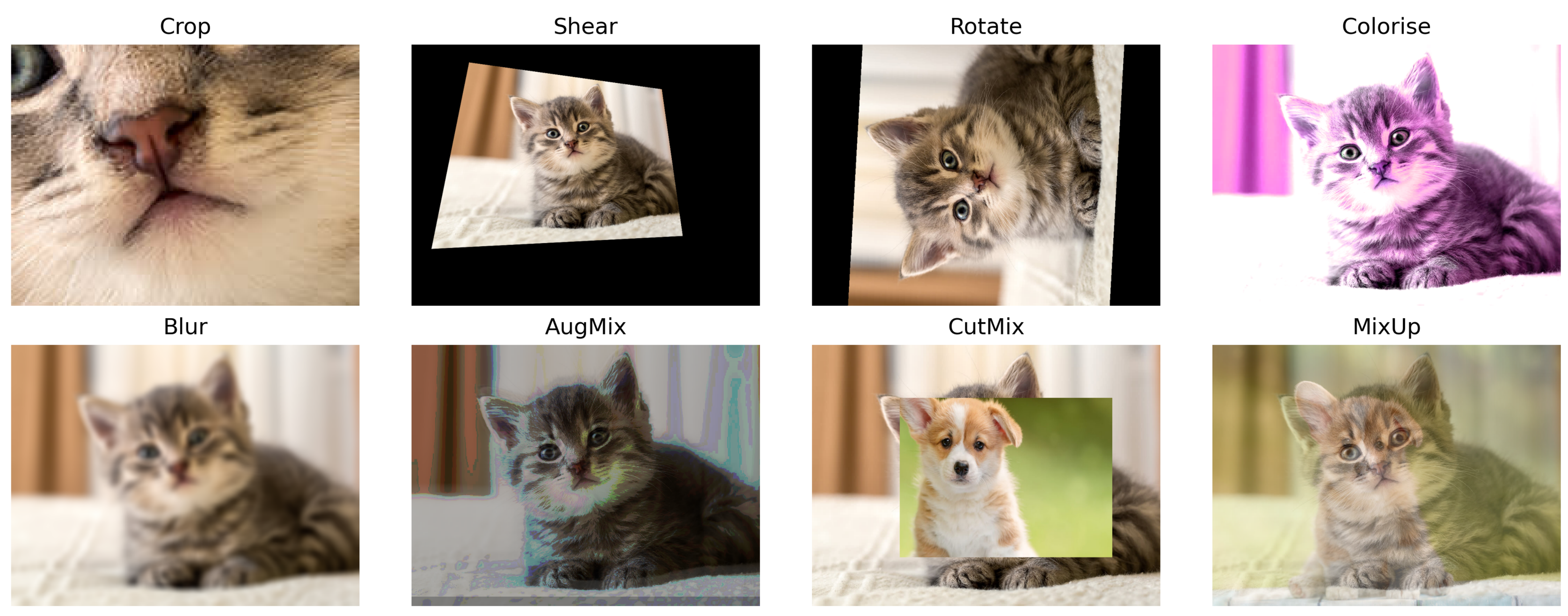

Figure 1. Common training-time augmentations [2].

Probabilistic augmentation model

Likelihood Given data $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\} \overset{\text{iid}}{\sim} p(\mathbf{x}, \mathbf{y})$ and transformations $T_\phi(\mathbf{x})$ with parameters ϕ we assume a likelihood,

$$\mathcal{L} = \log p(\mathcal{D}|\theta) = \sum_{n=1}^{N} \log \int p(\mathbf{y}_n|\mathbf{x}_n, \phi, \theta) p(\phi) d\phi \tag{1}$$

$$\geq \sum_{n=1}^{N} \int \log p(\mathbf{y}_n|\mathbf{x}_n, \phi, \theta) p(\phi) d\phi, \tag{2}$$

where θ are the predictive parameters. We argue that any transformation T can degrade the true label $\mathbf{y}$.

Input mollification

We repurpose the noising and blurring processes of diffusion models for augmentation.

Noising We assume cosine noising schedule

$$\mathbf{x}_t^{\text{noise}} = \cos(t\pi/2)\mathbf{x} + \sin(t\pi/2)\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, I), \tag{3}$$

with *temperature* $t \in [0, 1]$.

Blurring We follow blurring

$$\mathbf{x}_t^{\text{blur}} = \mathbf{V} \exp\Big(-\tau(t)\boldsymbol{\Lambda}\Big) \mathbf{V}^T \mathbf{x}, \tag{4}$$

where $\mathbf{V}$ is a discrete cosine transform (DCT), and $\boldsymbol{\Lambda}$ are inverse square frequencies $[\boldsymbol{\Lambda}]_{wh} = \pi^2(\frac{w^2}{W^2} + \frac{h^2}{H^2})$.

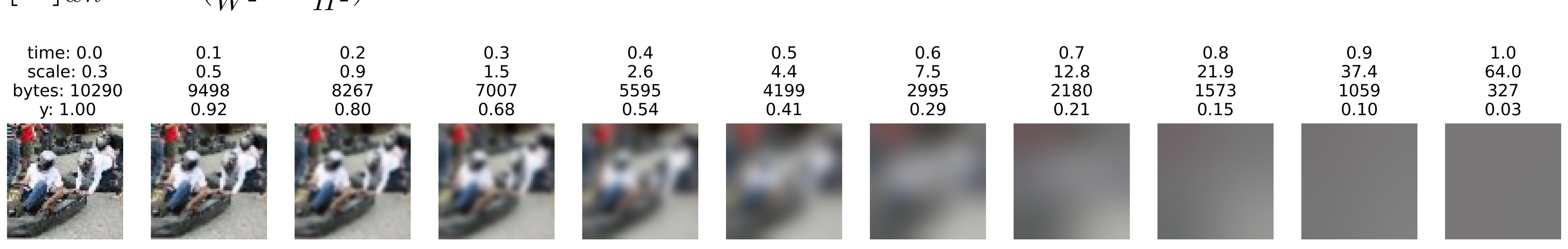

Figure 2. Blurring sequence.

Label smoothing

We consider label degradation by temperature t,

$$\mathbf{y}_t = (1 - \gamma_t)\mathbf{y}^{\text{onehot}} + \frac{\gamma_t}{C}\mathbf{1}, \tag{5}$$

with schedule

$$\gamma_t^{\text{noise}} = \left(\frac{1}{1 + \alpha_t^2/\sigma_t^2} \right)^k \tag{6}$$

$$\gamma_t^{\text{blur}} = \left(\frac{\text{size}(\mathbf{x}_t\texttt{.png})}{\text{size}(\mathbf{x}_0\texttt{.png})} \right)^k \approx t^k. \tag{7}$$

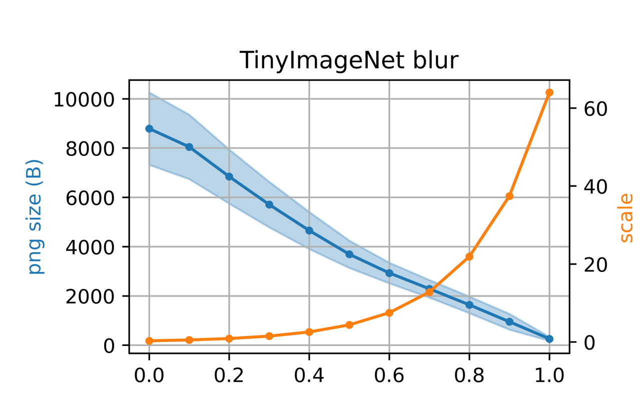

Figure 3. Our blurring schedule reduces information linearly.

Illustration

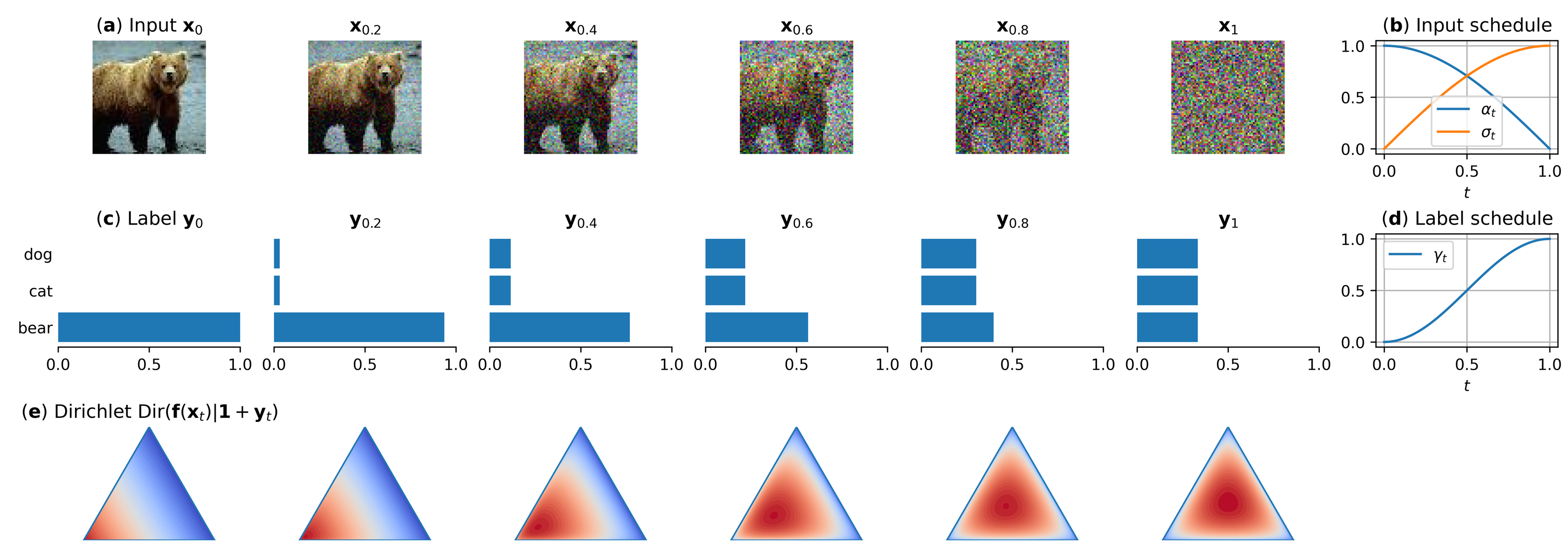

Figure 4. Mollification augments training with perturbed images (a) and smoothed labels (c) according to the schedule (d), which results in predictions whose distribution matches label uncertainty (e).

Results

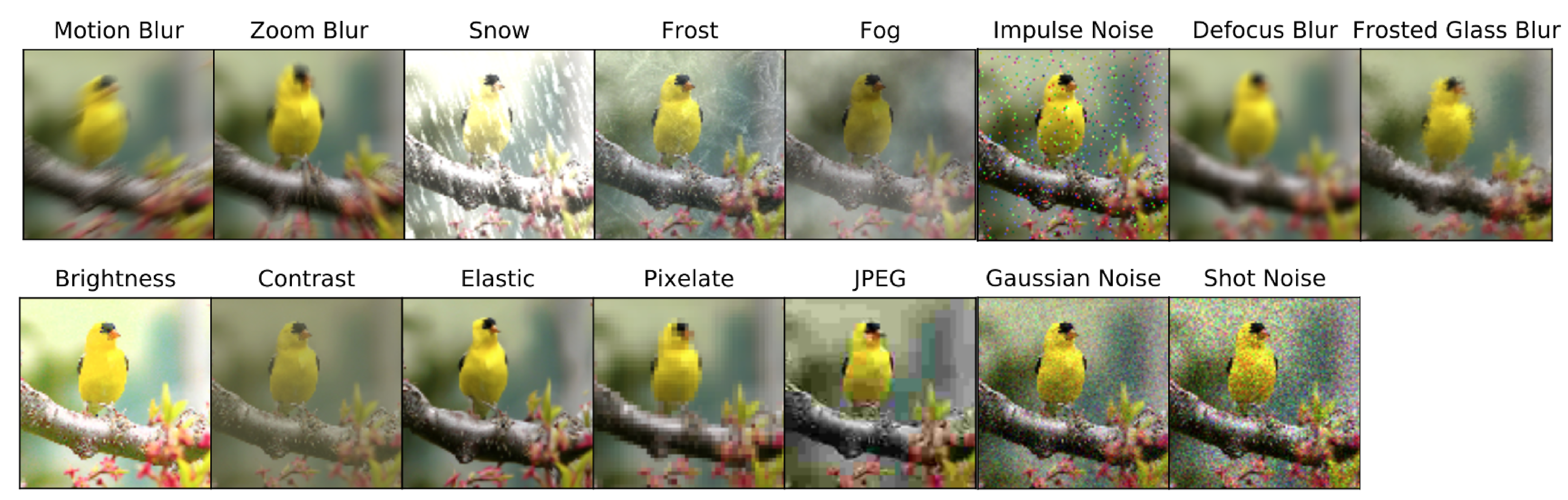

Figure 5. Common test-time corruptions [1].

			noise			blur				weather				digital				
		clean	shot	impulse	gauss	motion	zoom	defocus	glass	fog	frost	snow	bright	jpg	pixel	elastic	contrast	**mean**
CIFAR10	FCR+TrivAug	**3.4**	23	13	31	10.0	6.2	5.5	12	**6.3**	12	9.3	**3.9**	16	21	7.4	**4.4**	12
	+ Diffusion	**3.4**	**5.7**	**7.3**	**6.4**	9.4	6.6	5.8	**10**	6.8	**8.4**	**8.6**	**4.0**	**12**	**15**	7.2	4.9	**7.6**
	+ Blur	3.5	20	17	27	**8.1**	**4.8**	**4.1**	11	**6.4**	11	9.2	4.0	17	18	**6.5**	4.7	11
	+ Diffusion+Blur	4.0	6.2	7.5	6.8	8.8	5.0	4.6	**10**	7.1	**8.5**	9.1	4.5	14	16	6.7	5.3	7.8
CIFAR100	FCR+TrivAug	**20**	54	39	63	32	28	25	38	30	41	33	23	47	47	30	**25**	36
	+ Diffusion	**20**	**25**	**30**	**26**	32	29	27	34	31	34	32	23	**37**	36	29	26	29
	+ Blur	**20**	45	40	51	**28**	**23**	**21**	33	**29**	36	**31**	**22**	46	36	**26**	**25**	32
	+ Diffusion+Blur	**20**	**25**	31	**26**	**28**	23	22	**32**	30	**33**	**31**	23	39	**32**	27	25	**28**
TinyImgNet	FCR+TrivAug	34	81	85	85	68	71	76	76	65	65	69	60	59	61	61	76	68
	+ Diffusion	**32**	70	78	75	61	62	68	70	61	60	64	55	52	55	53	71	62
	+ Blur	33	71	77	77	**57**	58	64	**68**	**56**	56	**60**	**50**	51	**49**	**49**	69	59
	+ Diffusion+Blur	**32**	**65**	**73**	**71**	**57**	**57**	**63**	**67**	**56**	**54**	**59**	**50**	**50**	**49**	**49**	**68**	**57**

Table 1. pResNet-50 network errors over test-time corruptions.

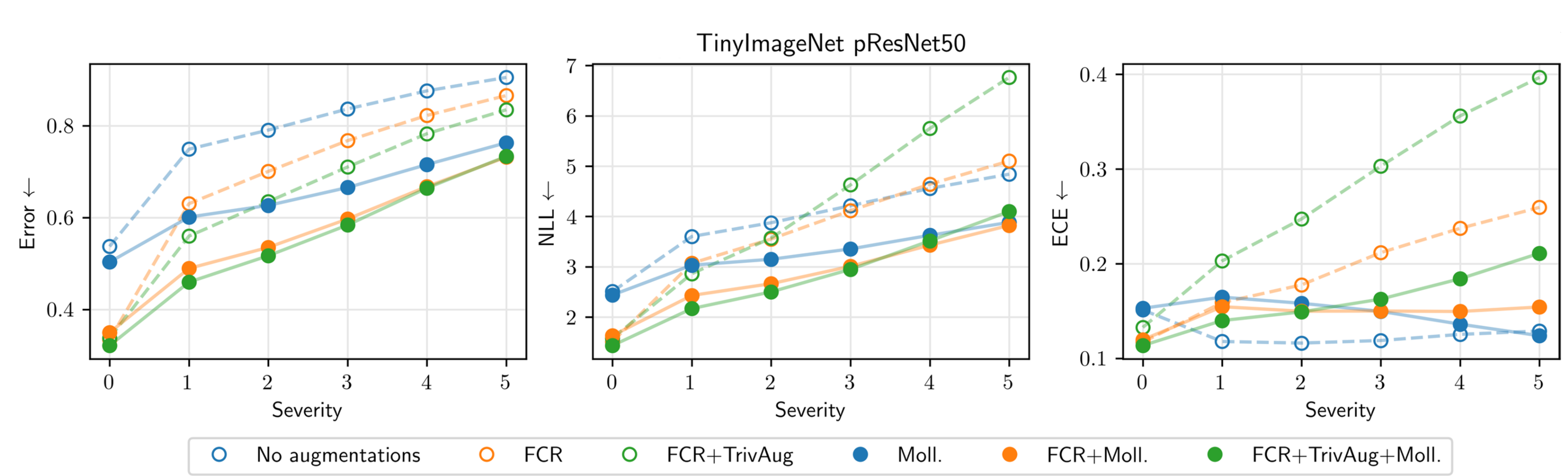

Figure 6. Mollification improves augmented models over corruption severities.

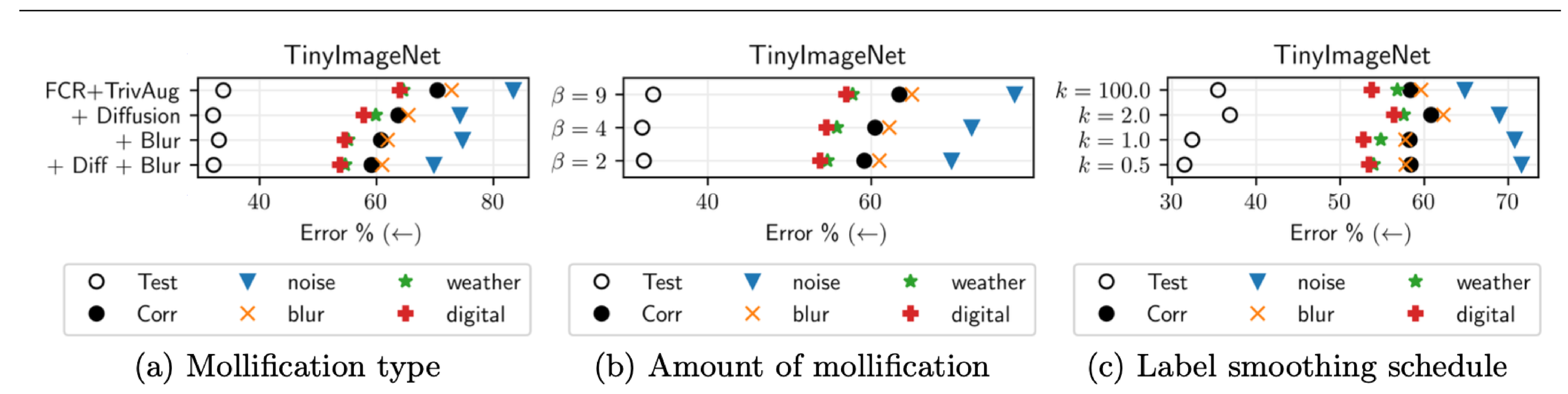

Figure 7. Combining substantial amounts of noising and blurring with greedy label smoothing yields good results.

References

[1] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[2] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

[3] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. In *NeurIPS workshop on ImageNet PPF*, 2021.