Enhancing Adversarial Robustness via Test-time Transformation Ensembling

2021 ICCV OCTOBER 11-17 VIRTUAL

GitHub

Juan C. Pérez[1,2], Motasem Alfarra[1], Guillaume Jeanneret[2], Laura Rueda[2],
Ali Thabet[1], Bernard Ghanem[1] and Pablo Arbeláez[2]
[1]King Abdullah University of Science and Technology (KAUST); [2]CINFONIA, Universidad de los Andes

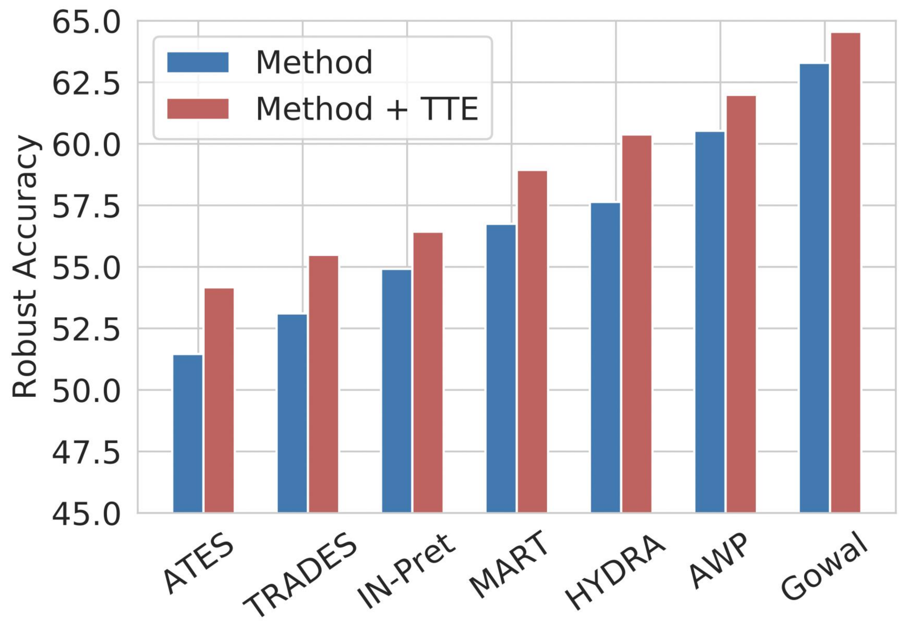
KAUST

Universidad de los Andes

CinfonIA

Abstract

- Common knowledge: Ensembling predictions of input transformations enhance *classification performance*.
- Is the same true for **adversarial robustness performance**?***Yes!***



- *No trade-off* with classification performance.
- We provide **extensive empirical evidence for** Both:
- Defended and undefended models against a variety of attacks. In CIFAR10, CIFAR100 and ImageNet.
- For several types of input transformations.
- Gradient obfuscation does not appear to happen.

Methodology

- Use transforms that ease the assessment of adversarial robustness. Thus,
  - Differentiable *and*
  - Deterministic
- Consider traditional crops, flips, and the composition of both.
- Easily implementable: wrapper in PyTorch.
- *Separately* use each of the top-performing attacks from AutoAttack.
- Check the effect on certified robustness by combining the wrapper with the methods of Cohen et.al. and SmoothAdv.

Experiments

Undefended models

CIFAR10

Method	Clean	Robust
ResNet-18	92.58	16.18
ResNet-18 + TTE	**93.42**	**29.81**

CIFAR100

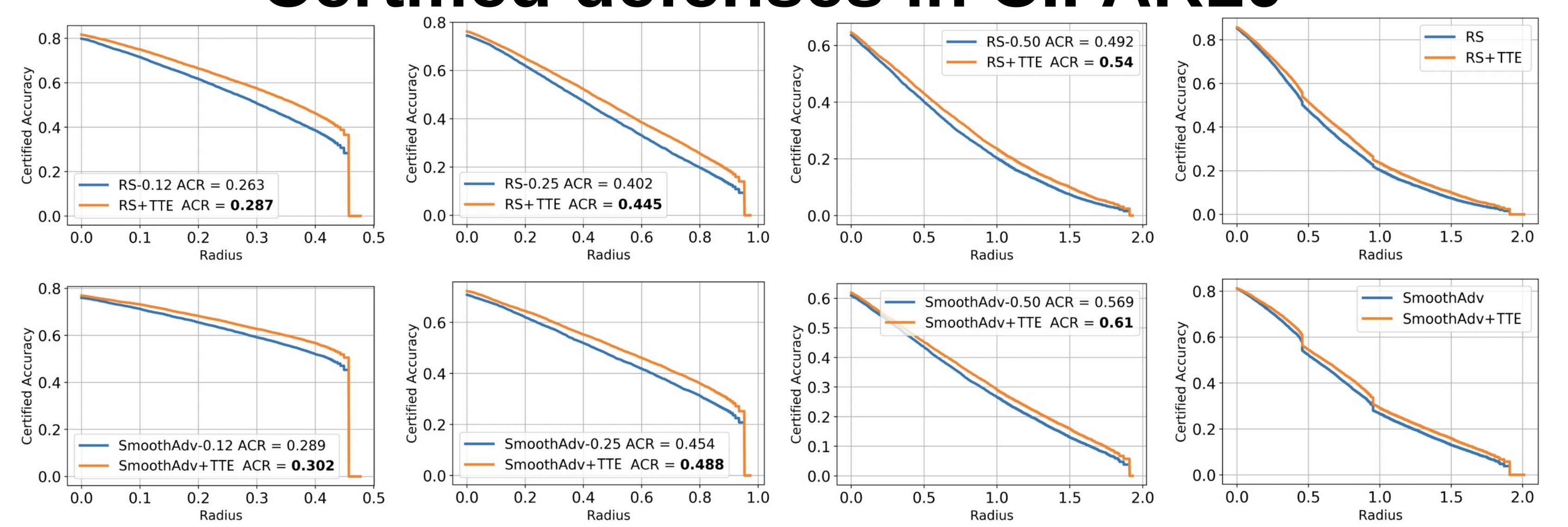
Method	Clean	Robust
ResNet-18	76.66	0.72
ResNet-18 + TTE	**77.57**	**1.68**

Defended model

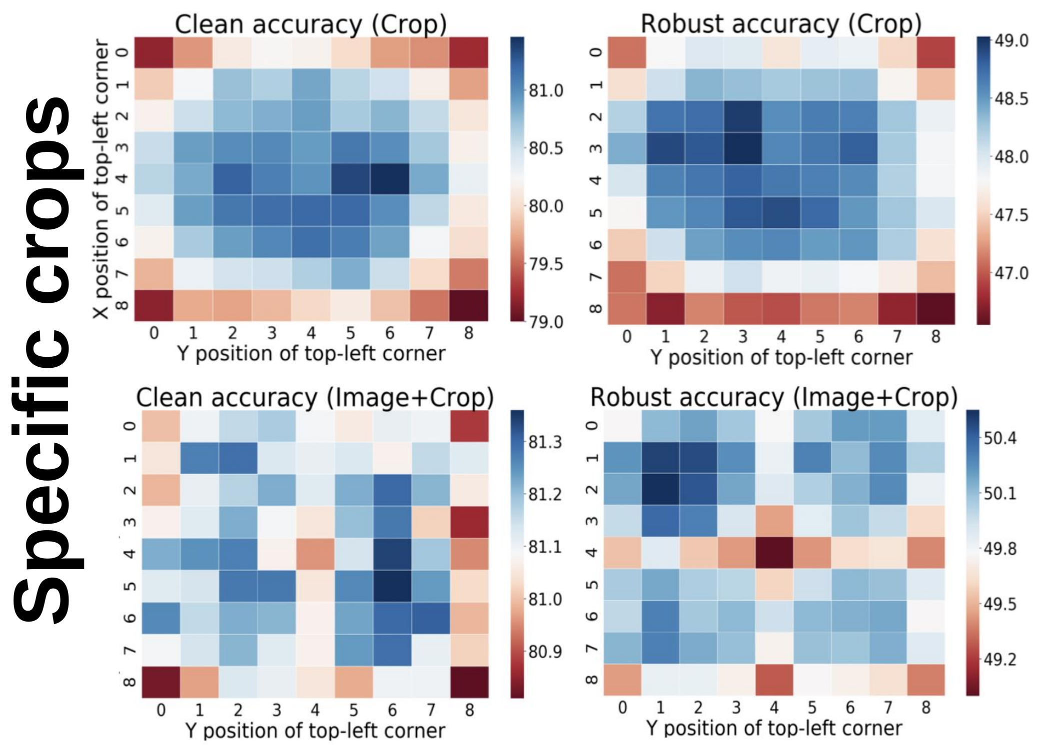
	Method	Clean	APGD-CE	APGD-T	FAB-T	Square	Robust	Difference
CIFAR10	ATES [48]	86.84	53.5	51.5	51.91	59.77	51.46	+2.71
	ATES + TTE	**86.86**	56.48	54.19	54.70	60.67	**54.17**	
	TRADES [61]	84.92	55.31	53.12	53.55	59.41	53.11	+2.38
	TRADES + TTE	**85.14**	57.46	55.51	55.88	60.22	**55.49**	
	IN-Pret [23]	87.11	57.65	55.32	55.68	62.40	54.92	+1.51
	IN-Pret + TTE	**87.13**	59.06	56.44	56.73	63.14	**56.43**	
	MART [53]	87.50	62.18	56.80	57.34	64.87	56.75	+2.19
	MART + TTE	**87.79**	63.95	58.94	59.51	65.62	**58.94**	
	HYDRA [45]	**88.98**	60.13	57.66	58.42	65.01	57.64	+2.74
	HYDRA + TTE	88.82	62.82	60.40	60.91	66.03	**60.38**	
	AWP [55]	**88.25**	63.81	60.53	60.98	66.18	60.53	+1.46
	AWP + TTE	88.07	64.95	61.99	62.52	66.48	**61.99**	
	Gowal *et al.* [20]	**89.48**	66.16	63.26	63.74	69.10	63.29	+1.26
	Gowal *et al.* + TTE	89.41	67.19	64.55	64.88	69.29	**64.55**	
CIFAR100	ATES [48]	62.82	26.78	24.98	25.23	31.27	24.96	+1.83
	ATES + TTE	**63.47**	28.9	26.8	27.15	32.21	**26.79**	
	IN-Pret [23]	59.37	33.45	29.03	29.34	34.55	28.61	+1.19
	IN-Pret + TTE	**59.38**	33.96	29.59	29.87	34.86	**29.50**	
	AWP [55]	60.38	33.56	29.16	29.48	34.66	29.15	+0.86
	AWP + TTE	**60.39**	34.11	30.03	30.26	34.64	**30.01**	

ImageNet	Method	Clean	APGD-CE	APGD-T	FAB-T	Square	Robust	Difference
	FD [56]	65.32	7.91	4.31	7.87	23.76	4.23	+2.21
	FD + TTE	**65.87**	9.29	6.53	9.23	26.34	**6.44**	

Certified defenses in CIFAR10



Specific crops



Individual transforms

Method	Clean	PGD^{30}	Diff.
FD [56]	65.32	50.20	-
+ flip	65.38	**51.17**	+0.97
+ 1 crop	65.50	**51.07**	+0.87
+ 2 crops	65.51	**50.84**	+0.64
+ 3 crops	65.78	**51.20**	+1.00
+ 4 crops	65.74	**51.21**	+1.01
+ flip + 1 crop	65.56	**51.69**	+1.49
+ flip + 2 crops	65.59	**51.77**	+1.57
+ flip + 3 crops	65.81	**51.80**	+1.60
+ flip + 4 crops	65.76	**51.43**	+1.23
+ flip + 1 crop + 1 flipped-crop	65.69	**51.47**	+1.27
+ flip + 2 crops + 2 flipped-crops	65.68	**51.36**	+1.15
+ flip + 3 crops + 3 flipped-crops	65.87	**51.88**	+1.68
+ flip + 4 crops + 4 flipped-crops	65.85	**52.17**	+1.98

Gradient obfuscation in CIFAR10?

Optimization iterations

Iterations	5	10	50	100
TRADES	49.92	49.12	48.71	48.69
TRADES + TTE	**52.11**	**51.54**	**51.41**	**51.40**

Attack strength (ϵ)

ϵ	$^{8}/_{255}$	$^{16}/_{255}$	$^{32}/_{255}$	$^{64}/_{255}$
TRADES	48.69	15.84	0.72	0.00
TRADES + TTE	**51.40**	**18.85**	**0.95**	**0.01**

Acknowledgment: This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the Visual Computing Center (VCC) funding.