Certified Robustness

Fundamentals and Challenges

Aleksandr Petiushko

Nuro
Autonomy Interaction Research

January 24th, 2023

AP

Content

AP

Intro

About speaker[1]

- Aleksandr Petiushko, PhD in theoretical CS (2016)
- Lecturer in Lomonosov MSU / MIPT for Machine Learning, Computer Vision, Deep Learning Theory, Python for an ML Researcher since 2019
- Former Huawei Chief Scientist (Scientific Expert), AIRI Director of Key Research Programs (Leading Scientific Researcher)
- Currently at Nuro, leading the Autonomy Interaction Research

[1] Homepage: https://petiushko.info/

Robustness in Machine Learning

Robustness [informally]

Ability for a machine learning algorithm a to provide similar outputs on the similar data (i.e. having the same class or other invariant features)

Two types of **Robustness** in ML:

Generalization

Dataset issue: algorithm needs to be robust if the dataset to evaluate it differs (sometimes significantly: we can treat it is a distribution shift) from the training dataset

Adversarial Robustness

Noise issue: algorithm needs to provide the similar output w.r.t. both clean and noisy images (where the model of noise is the topic to consider itself)

For now we'll consider the **Adversarial Robustness**.

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations
- Certification (or verification): how to provide theoretical guarantees on the noise level not fooling the neural net

AP

Adversarial Robustness topics

- Perturbations (also called 'adversarial *attacks*'): how to generate noise to fool the neural net
- Defense: how to diminish the influence of adversarial perturbations
- Certification (or verification): how to provide theoretical guarantees on the noise level not fooling the neural net

AP

Certified Robustness

- Let us NN function $f(x)$ is the classifier to K classes: $f : \mathbb{R}^d \to Y, Y = \{1, \dots, K\}$
- Usually we have NN $h(x) : \mathbb{R}^d \to \mathbb{R}^K$, and $f(x) = \arg\max_{i \in Y} h(x)_i$

Deterministic approach

Need to find the class of input perturbation $S(x, f)$ so as the classifier's output doesn't not change, or more formally:

$$f(x + \delta) = f(x) \quad \forall \delta \in S(x, f)$$

Probabilistic approach

Need to find the class of input perturbation $S(x, f, P)$ w.r.t. robustness probability P s.t.:

$$Prob_{\delta \in S(x,f,P)}(f(x + \delta) = f(x)) = P$$

Remark: Probabilistic approach coincides with Deterministic one when $P = 1$.

AP

Certified Robustness: inverse tasks

- Suppose that we know the input perturbation class S

Classification

Need to measure the probability P of retaining the classifier's output under some class of input perturbations S:

$$Prob_{\delta \in S}(f(x+\delta) = f(x)) = P$$

Remark: It is a difficult task because usually the perturbation class consists of enormous number (sometimes even infinite) of perturbations.

AP

Certified Robustness via Lipschitzness (1)

- NN classifier to K classes is $f(x)$: $f : \mathbb{R}^d \to Y, Y = \{1, \dots, K\}$
- NN itself is $h(x) : \mathbb{R}^d \to \mathbb{R}^K$, and $f(x) = \arg\max_{i \in Y} h(x)_i$
- Consider binary case (other cases are treated similarly) $K = 2$ and probabilistic (SoftMax) output: $h(x)_1 + h(x)_2 = 1, \quad h(x)_i \geq 0 \quad \forall i$

Definition of Lipschitz function

Lipschitz function g: $g : \mathbb{R}^d \to \mathbb{R}$ with a Lipschitz constant L so as $\forall x_1, x_2$ it holds $|g(x_1) - g(x_2)| \leq L \|x_1 - x_2\|$

Definition of Local Lipschitz function

Local Lipschitz function g: $g : \mathbb{R}^d \to \mathbb{R}$ with a Lipschitz constant $L(x_0)$ so as $\forall x \in S(x_0)$ it holds $|g(x_0) - g(x)| \leq L(x_0) \|x_0 - x\|$

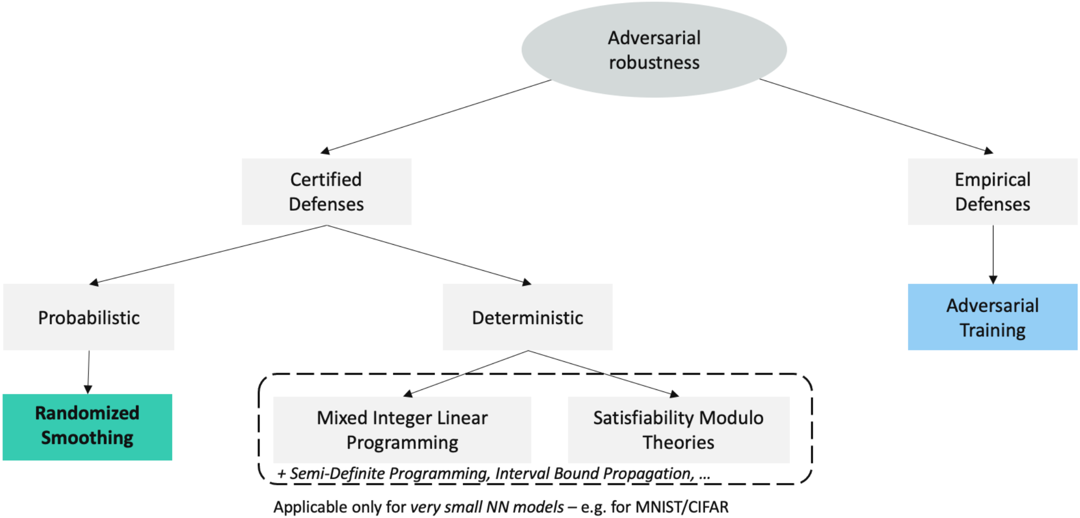
Certified Robustness via Lipschitzness (2)

Simple exercise: having the local Lipschitzness **guarantees** us the certification.
But:

Problems

- The certified radius can be much bigger than the local Lipschitz vicinity $S(x_0)$
- It is hard to provide the adequate (not tending to 0) Lipschitz constant for any industrial Deep Neural Network

Adversarial Robustness: overview



Adversarial Robustness: empirical vs certified

Empirical robustness

Bound
The upper bound on the true robust accuracy

Cons
Only valid *until* the *new* – and stronger – *attack* appears

Certified robustness

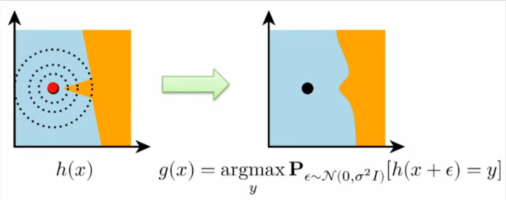
Bound
The lower bound on the true robust accuracy

Pros
It is what has been *theoretically proven*, and no one attack can beat it

AP

Adversarial Examples: boundary curvature

- Very **curved boundary** leads to *adversarial examples* looking very similar to ones near the classification boundary
- So let's **diminish** this curvature **spike** influence!
- Different approaches exist e.g. by *Lecuyer et al.*[2] and *Li et al.*[3], but the most famous one is by *Cohen et al.*[4]



[2] Lecuyer, Mathias, et al. "Certified robustness to adversarial examples with differential privacy." 2018
[3] Li, Bai, et al. "Certified adversarial robustness with additive noise." 2018
[4] Cohen, Jeremy, et al. "Certified adversarial robustness via randomized smoothing." 2019

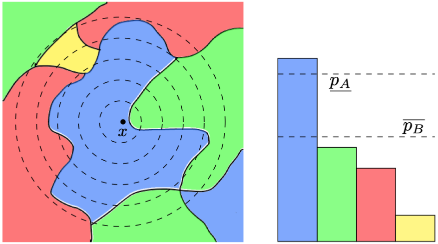
AP

Randomized Smoothing

Idea of Randomized Smoothing (RS)

- Let's use the **T**est **T**ime **A**ugmentation (**TTA**) in order to mitigate the boundary effect
- The new classifier $g(x)$ is defined as:

$$g(x) = \arg\max_{c \in Y} P(f(x+\epsilon) = c), \epsilon \sim N(0, \sigma^2)$$



RS main result

- If the initial classifier $f(x)$ is robust under Gaussian noise,
- Then the new classifier $g(x)$ is robust under **ANY** noise

Randomized Smoothing: Theory overview

Theorem: Certification Radius

Suppose $c_A \in Y$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy $\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c_B \neq c_A} \mathbb{P}(f(x+\epsilon) = c_B)$. Then $g(x+\delta) = c_A \quad \forall \, \|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$$

Tightness of Radius R

Assume $\underline{p_A} + \overline{p_B} \leq 1$. Then for any perturbation $\delta, \|\delta\|_2 > R$ there exist a base classifier f s.t. $\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c_B \neq c_A} \mathbb{P}(f(x+\epsilon) = c_B)$ so as $g(x+\delta) \neq c_A$

Remark. Φ^{-1} is the inverse of the standard Gaussian CDF: $\Phi(x) = \frac{1}{2\pi} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$.

AP

Randomized Smoothing: Training

- To certify the classifiers, authors **trained the base models with Gaussian noise** from $N(0, \sigma^2 I)$ — actually, to make the classifier $f(x)$ to be more robust to Gaussian noise
- So no any other training-specific tricks aside from simple **augmentation**

Randomized Smoothing: Inference

- Trained models are compared using "**approximate certified accuracy**":
  - $\forall$ test radius $\delta = r$ the fraction of examples is returned so as the procedure CERTIFY:
    - Provides the answer
    - Returns the correct class
    - Returns a radius R so as $r \leq R$

Procedure CERTIFY

- Can return ABSTAIN if confidence bounds are too loose (done by **Clopper-Pearson** confidence intervals for the Binomial distribution[5])
- If not ABSTAIN, then return the majority class $\hat{c}_A$ and certification radius $R = \sigma\Phi^{-1}(\underline{p_A})$

[5] Clopper, Charles J., and Egon S. Pearson. "The use of confidence or fiducial limits illustrated in the case of the binomial." 1934

AP

Randomized Smoothing: Results on ImageNet

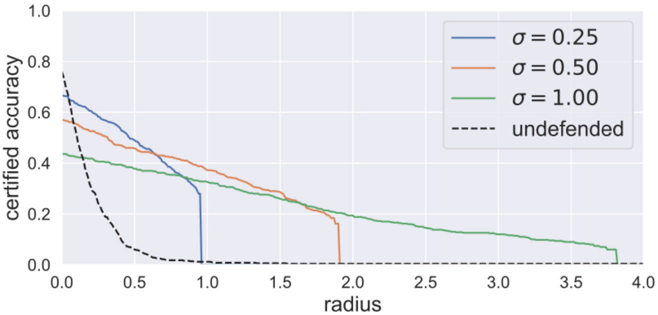


Table 1. Approximate certified accuracy on ImageNet. Each row shows a radius r, the best hyperparameter σ for that radius, the approximate certified accuracy at radius r of the corresponding smoothed classifier, and the standard accuracy of the corresponding smoothed classifier. To give a sense of scale, a perturbation with ℓ_2 radius 1.0 could change one pixel by 255, ten pixels by 80, 100 pixels by 25, or 1000 pixels by 8. Random guessing on ImageNet would attain 0.1% accuracy.

ℓ_2 RADIUS	BEST σ	CERT. ACC (%)	STD. ACC(%)
0.5	0.25	49	67
1.0	0.50	37	57
2.0	0.50	19	57
3.0	1.00	12	44

Remark1. Waterfall just because the trained model is robust usually under some $r \leq R$.
Remark2. "Certified accuracy" = approximate certified accuracy.
Remark3. The difference between "clean" and "certified" accuracy is not order of magnitude (it works! and can be useful).

AP

Certification: intermediate takeaway

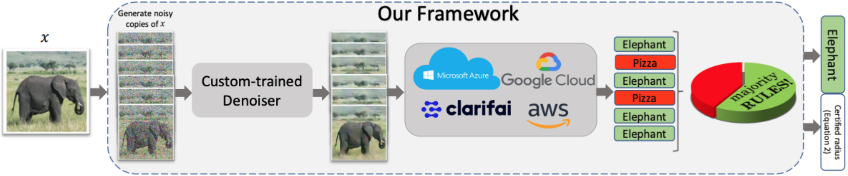
- Randomized Smoothing = Smoothing distribution + norm l_p of perturbation
- Randomized Smoothing requires multiple inferences :(
- Certified robustness is better than empirical adversarial training in certification, but worse than clean performance (and too much time to train)

AP

Randomized Smoothing: Black-box access

- What if we **cannot change the pretrained classifier**, but want to increase its certified robustness?
- Idea of **Black-box smoothing**[6]: Let's train a **denoiser** D used after we've added Gaussian noise!
  - And then simply apply the majority rule

$$g(x) = \arg\max_{c \in Y} \mathbb{P}[f(D(x+\delta)) = c], \quad \delta \sim N(0, \sigma^2 I)$$



[6] Salman, Hadi, et al. "Black-box smoothing: A provable defense for pretrained classifiers." 2020

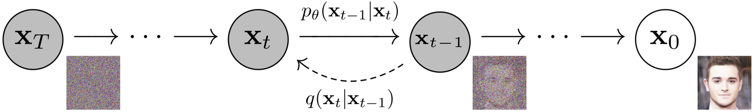
Randomized Smoothing: Denoiser for Black-box

- Denoiser: trained with two losses for every Gaussian σ:
  - MSE
  - Stability (classification cross entropy)

(a) Noisy (b) MSE (c) Stab+MSE

Denoiser by DDPM[8]

- A novel approach[7] to use off-the-shelf models:
  - SotA classifier (trained on clean images)
  - Denoising Diffusion Model
    - Based on the noise level σ, estimate $\bar{\alpha}_t, t$
    - Generate $x_t \sim N(\sqrt{\bar{\alpha}_t} \cdot x, (1 - \bar{\alpha}_t)I)$
    - Denoise by DDPM decoder (using **only 1 step**): $\hat{x} = denoise(x_t)$
    - Classify!
- Results in 14% improvement over the prior certified SoTA, and an improvement of 30% over denoised smoothing



[7] N. Carlini, F. Tramer, and Z. Kolter. "(Certified!!) Adversarial Robustness for Free!", 2022

[8] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models", 2020

Randomized Smoothing: vector functions

- Previously all results were for the classifiers: $f, g : \mathbb{R}^d \to Y, Y = \{1, \dots, K\}$, $g(x) = \arg\max_{c \in Y} P(f(x+\epsilon) = c), \epsilon \sim N(0, \sigma^2)$
- Let's consider the vector-based functions f (e.g., feature vector): $\mathbb{R}^d \to \mathbb{R}^D$
- Then the smoothed version g of it we'll define as: $g(x) = \mathbb{E}_{\epsilon \sim N(0, \sigma^2 I)}[f(x+\epsilon)]$
- In this case the following relation to Lipschitz functions can be established[9]:

Lipschitz-continuity of smoothed vector function

Suppose that $g(x)$ is continuously differentiable for all x. If for all x, $\|f(x)\|_2 = 1$, then $g(x)$ is $L-$Lipschitz in l_2-norm with $L = \sqrt{\frac{2}{\pi\sigma^2}}$.

[9] Pautov, Mikhail, et al. "Smoothed Embeddings for Certified Few-Shot Learning." 2022

Randomized Smoothing: adversarial embedding risk

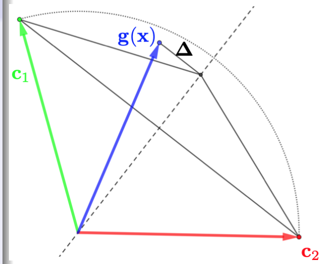
- Let's establish the beautiful geometrical fact useful for the few-shot classification:

Adversarial embedding risk

Given an input $x \in \mathbb{R}^d$ and the embedding $g : \mathbb{R}^d \to \mathbb{R}^D$ the closest point on to decision boundary in the embedding space is located at a distance:

$$\gamma = \|\Delta\|_2 = \frac{\|c_2 - g(x)\|_2^2 - \|c_1 - g(x)\|_2^2}{2\,\|c_2 - c_1\|_2^2},$$

where $c_1 \in \mathbb{R}^D$ and $c_2 \in \mathbb{R}^D$ are the two closest prototypes.



- γ is the distance between classifying embedding and the decision boundary between classes represented by c_1 and c_2.
- γ is the minimum l_2-distortion in the embedding space required to change the prediction of g.

Randomized Smoothing: certification

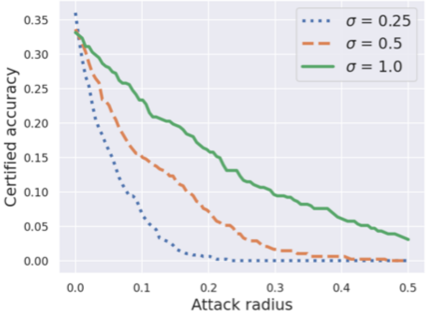
- Two results above lead to the certification guarantee:

Robustness guarantee

Certified radius r of g at x, where g is the smoothed version of $f : \|f(x)\|_2 = 1$, is

$$r = \frac{\gamma}{L}$$

1-shot results for *mini*ImageNet[10]



[10] Vinyals, Oriol, et al. "Matching networks for one shot learning." 2016

Randomized Smoothing: norms

- Randomized Smoothing = Smoothing distribution + **norm** l_p of perturbation
- Using l_p-balls is neither necessary nor sufficient for perceptual robustness
- Certification is only for much smaller regions than humans can do
- Remark about physical nature of l_p-balls:
  - l_2 corresponds to the power of signals
  - l_1 corresponds to the pixel mass
  - l_∞ corresponds to the noise in camera sensors
  - l_0 corresponds to the practical patch robustness

AP

Randomized Smoothing: High Dimensional Case

- The perturbation δ is measured by l_p-norm
- $p = 1$ and $p = 2$ are the only **special cases**[11]: $R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$
- Unfortunately, these are **only** examples of **non-decreasing** with **input dimension** d
- For any $p \geq 2$, the certification radius[12] is decreasing with dimensionality d:

$$R_p(x) = \frac{\sigma}{2d^{\frac{1}{2}-\frac{1}{p}}}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$$

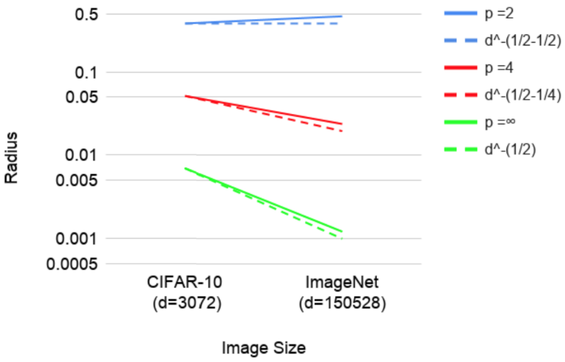
- And the most important case for Computer Vision (CV), $p = \infty$, means

$$R_\infty \sim \frac{1}{\sqrt{d}}$$

[11] Yang, Greg, et al. "Randomized smoothing of all shapes and sizes." 2020

[12] Kumar, Aounon, et al. "Curse of dimensionality on randomized smoothing for certifiable robustness." 2020

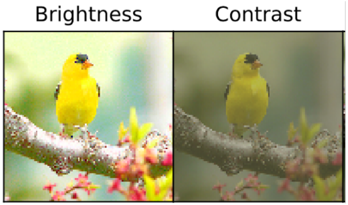
Randomized Smoothing: CV illustration



AP

High Dimension Case in CV

- Any **semantic-meaningful** perturbation in **CV** leads to **high l_∞-perturbation**, and the dimension of an image $d = H \times W$ usually is very high (like millions of pixels)
- $R_\infty \sim \frac{1}{\sqrt{d}}$ means that there is **no any practical certified radius**
- E.g., for semantic-specific transformations like **contrast** and **brightness** the **error is higher** than on clean images up to 50-60% on *Common Corruptions*[13] on ImageNet



Network	Error	Bright	Contrast
AlexNet	43.5	100	100
SqueezeNet	41.8	97	98
VGG-11	31.0	75	86
VGG-19	27.6	68	80
VGG-19+BN	25.8	61	74
ResNet-18	30.2	69	78
ResNet-50	23.9	57	71

[13] Hendrycks, Dan, and Thomas Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations." 2019

AP

High Dimension Case in CV: Autonomous Driving

- The same is true for safety-critical applications like autonomous driving[14]

contrast(1.8) **original** **brightness(50)**

Transformation	***#err***
Brightness	97
Contrast	31

[14] Tian, Yuchi, et al. "Deeptest: Automated testing of deep-neural-network-driven autonomous cars." 2017

Semantic perturbations for additive parameters

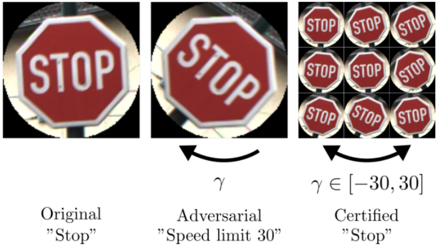
- So... let's certify semantic perturbations[15]!
  - Usually parameterized by a much smaller dimension (1 or 2 dimensional)
- Consider **rotations** and **translations** γ_β parameterized by β: $\gamma_\beta : \mathbb{R}^d \to \mathbb{R}^d$
- A smoothed classifier $g(x) = \arg\max_{c \in Y} P_{\beta \sim N(0,\sigma^2)}(f \circ \gamma_\beta(x) = c)$
- Also **interpolation** procedure is taken into account because after rotation we need to interpolate anyway

Certification Radius

Suppose $c_A \in Y$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy $\mathbb{P}_{\beta \sim N(0,\sigma^2)}(f \circ \gamma_\beta(x) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c_B \neq c_A} \mathbb{P}_{\beta \sim N(0,\sigma^2)}(f \circ \gamma_\beta(x) = c_B)$. Then $g \circ \gamma_\beta(x) = c_A \quad \forall \|\gamma\|_2 < r_\gamma$, where $r_\gamma = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$.

[15] Fischer, Marc, et al. "Certified defense to image transformations via randomized smoothing." 2020

Semantic perturbations for additive parameters: results



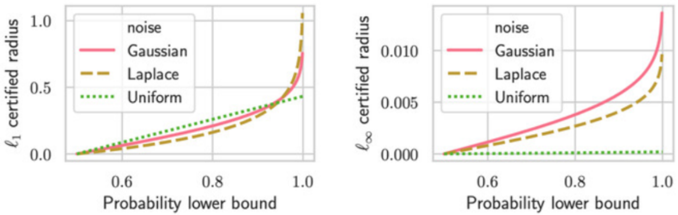
Rotation						r_γ percentile		
Dataset	$\mathcal{I}$	σ_γ	α_γ	f Acc.	g Acc.	25^{th}	50^{th}	75^{th}
ImageNet	bil.	10	0.001	0.39	0.29	10.81	10.81	10.81
ImageNet	bil.	10	0.001	0.39	0.29	18.29	18.29	18.29
ImageNet	bil.	30	0.001	0.39	0.28	9.09	16.59	28.60
ImageNet	bil.	30	0.001	0.39	0.28	20.22	25.36	$30^{\dagger}$
ImageNet	bic.	10	0.001	0.39	0.29	10.40	10.40	10.40
ImageNet	bic.	30	0.001	0.39	0.27	9.33	17.00	28.74
ImageNet	near.	10	0.001	0.39	0.29	9.62	9.62	9.62
ImageNet	near.	30	0.001	0.39	0.26	7.38	16.63	27.72

Translation						r_γ percentile		
Dataset	$\mathcal{I}$	σ_γ	α_γ	f Acc.	g Acc.	25^{th}	50^{th}	75^{th}
ImageNet	bil.	50	0.001	0.48	0.36	2.4%	2.4%	2.4%
ImageNet	bic.	50	0.001	0.48	0.36	2.4%	2.4%	2.4%

AP

Randomized Smoothing: smoothing distribution

- Randomized Smoothing = Smoothing **distribution** + norm l_p of perturbation
- Original (and most of the follow-up ones) work uses Gaussian Smoothing
- Other types of randomized smoothing could be taking into account: e.g. Uniform[16] or Laplacian[17]
- What about other types?



[16] Lee, Guang-He, et al. "Tight certificates of adversarial robustness for randomly smoothed classifiers." 2019

[17] Teng, Jiaye, et al. "ℓ_1 Adversarial Robustness Certificates: a Randomized Smoothing Approach." 2019

AP

Semantic perturbations and multiplicative parameters

- All research above is concentrated on **additive** perturbations
- Let's investigate the **multiplicative** parameters[18] (e.g., *gamma correction* $G_\gamma(x) = x^\gamma$ in CV)
- **Definition**: A parameterized map $\psi_\delta : X \to X$, $\delta \in \mathcal{B} \subset \mathbb{R}^n$ is called multiplicatively composable if $(\psi_\delta \circ \psi_\theta)(x) = \psi_{(\delta \cdot \theta)}(x)$, $\forall x \in X$, $\forall \delta, \theta \in \mathcal{B}$
- Example: $G_\beta \circ G_\gamma(x) = (x^\gamma)^\beta = x^{\gamma \cdot \beta} = G_{\gamma \cdot \beta}(x)$

(a) $\gamma = 0.5$ (b) $\gamma = 1$ (c) $\gamma = 2$

[18]Muravev, Nikita, and Aleksandr Petiushko. "Certified Robustness via Randomized Smoothing over Multiplicative Parameters." 2021

Semantic perturbations and multiplicative parameters: results

- To work under this limitation, the new type of smoothing distribution is needed:
  - Positive support
  - Mean at 1
- The proposal to use is **Rayleigh** distribution: $p_\beta(z) = \sigma^{-2} z e^{-z^2/(2\sigma^2)}, z \geq 0$
- Then the following is true: $g \circ \psi_\gamma(x) = c_A$ for all γ satisfying $\gamma_1 < \gamma < \gamma_2$, where γ_1, γ_2 are the only solutions of the following equations:
  $F(\gamma_1^{-1} F^{-1}(\overline{p_B})) + F(\gamma_1^{-1} F^{-1}(1 - \underline{p_A})) = 1,$
  $F(\gamma_2^{-1} F^{-1}(\underline{p_A})) + F(\gamma_2^{-1} F^{-1}(1 - \overline{p_B})) = 1,$
  and $F(z) = 1 - e^{-z^2/(2\sigma^2)}$ is the CDF of γ.
- The results are better for $\gamma < 1$ in comparison to Uniform, Gaussian and Laplace smoothing

$\underline{p_A}$	$\overline{p_B}$	γ_1	γ_2
0.600	0.400	0.86	1.15
	0.200	0.71	1.33
0.700	0.300	0.72	1.32
	0.100	0.54	1.56
0.800	0.200	0.57	1.52
0.900	0.100	0.39	1.82
0.990	0.010	0.12	2.58
0.999	0.001	0.04	3.16

AP

Semantic perturbations and compositions

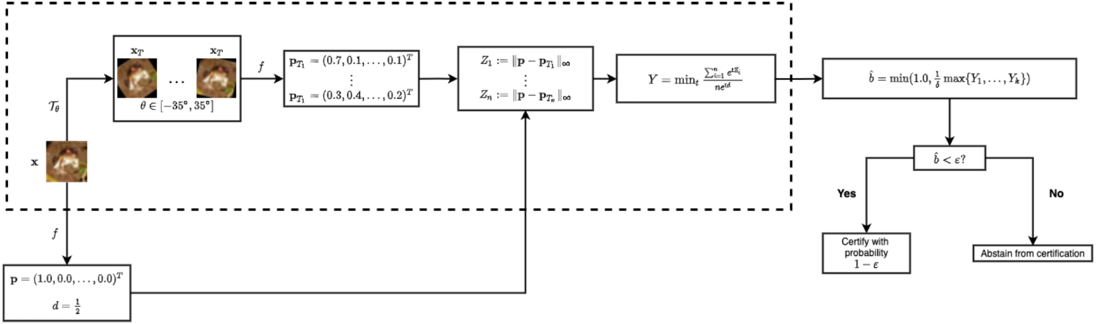
- Usually multiple transformations are applied to the input: how to certify the composition?
- Forward theoretical estimation is difficult ⇒ let's try inverse (probabilistic) task[19]!
- The proposal to use Chernoff-Cramer inequality[20] (Markov's inequality corollary) to provide the statistically-grounded **estimations for the certification**, where perturbed **radius** is **already given**
- Can be easily used for **any** semantic **perturbation** and **any compositions**

Dataset	Transform	Parameters	Training type	ERA	PCA(ε) $\varepsilon = 10^{-10}$	PCA(ε) $\varepsilon = 10^{-7}$	PCA(ε) $\varepsilon = 10^{-4}$
	Brightness	$\theta_b \in [-40\%, 40\%]$	plain	58.4%	47.8%	51.6%	55.2%
			smoothing	65.0%	55.4%	59.4%	61.8%
	Contrast	$\theta_c \in [-40\%, 40\%]$	plain	91.6%	62.4%	67.0%	69.6%
			smoothing	88.0%	67.0%	72.8%	74.2%
	Rotation	$\theta_r \in [-10°, 10°]$	plain	73.4%	64.6%	69.0%	71.0%
			smoothing	72.4%	57.4%	63.6%	67.4%
	Contrast + Brightness	see Contrast & Brightness	plain	0.0%	0.0%	0.0%	0.0%
			smoothing	0.4%	0.0%	0.0%	0.0%
	Rotation + Brightness	see Rotation & Brightness	plain	22.6%	16.2%	20.6%	21.8%
			smoothing	30.4%	21.2%	24.6%	27.6%
	Scale + Brightness	see Scale & Brightness	plain	10.2%	10.4%	10.4%	10.4%
			smoothing	41.8%	40.6%	40.6%	40.6%

[19] Pautov, Mikhail, et al. "CC-Cert: A probabilistic approach to certify general robustness of neural networks." 2021

[20] Boucheron, Stéphane, et al. "Concentration inequalities." 2003

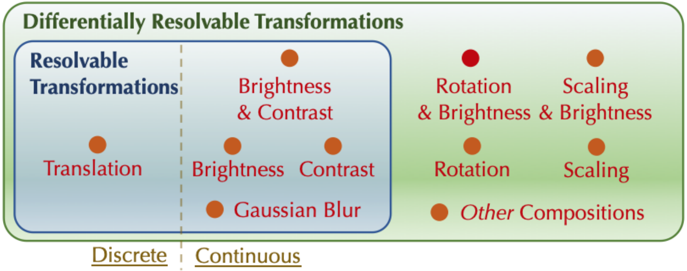
Inverse certification for any transformation[21]



[21] Pautov, Mikhail, et al. "CC-Cert: A probabilistic approach to certify general robustness of neural networks." 2021

Semantic perturbations: further development (1)

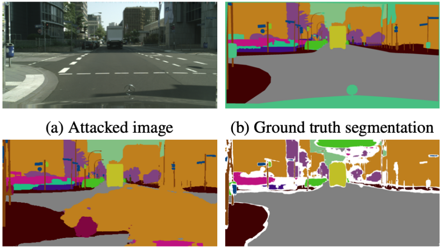
- Later works introduced approaches to take into account different types of perturbations and interpolation errors[22]



[22] Li, Linyi, et al. "Tss: Transformation-specific smoothing for robustness certification." 2020

Semantic perturbations: further development (2)

- Later works introduced approaches to apply certified robustness for other types of CV tasks — e.g. detection[23] and segmentation[24].



(a) Attacked image

(b) Ground truth segmentation

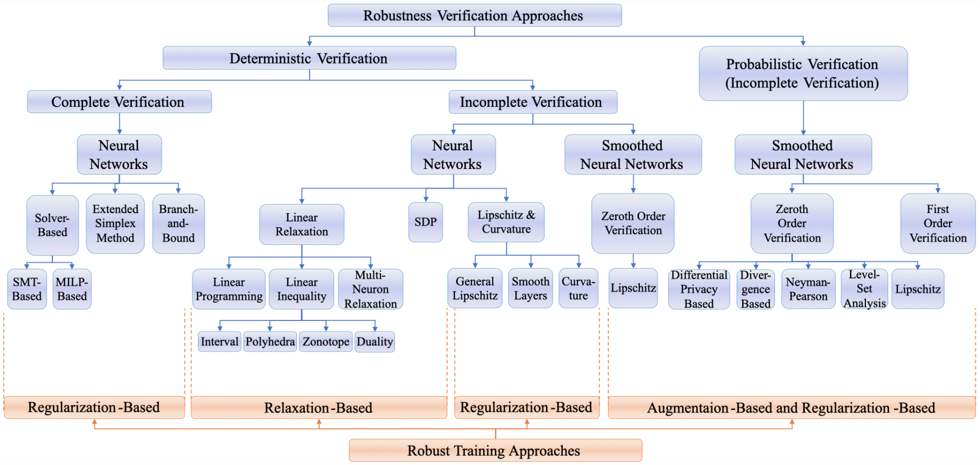
(c) Attacked segmentation

(d) Certified segmentation

[23] Chiang, Ping-yeh, et al. "Detection as regression: Certified object detection with median smoothing." 2020

[24] Fischer, Marc, et al. "Scalable certified segmentation via randomized smoothing." 2021

Systematization of Knowledge[25]



[25] SoK, Benchmark and Leaderboard

Takeaway notes

- Straightforward certification in l_∞ is not working for high dimension input
- In Computer Vision no need in any l_p (aside from l_0 for patch attacks, but it is usually also combined with other perturbations)
- Semantic perturbations are much harder to certify (+ interpolation!)
- **Current challenge**: 3D and even non-rigid transformations of **real world**

AP

Thank you!