Recent advances in adversarial machine learning: defense, transferable and camouflaged attacks

Xingjun Ma

School of Computing and Information Systems
The University of Melbourne

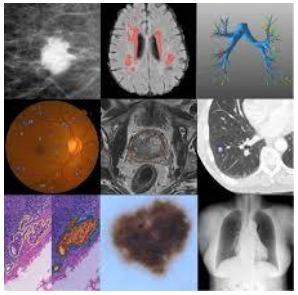
April 2020

Deep learning models are used everywhere

Speech recognition

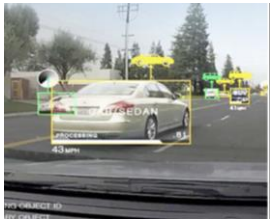
Image classification

Object detection



Medical diagnosis

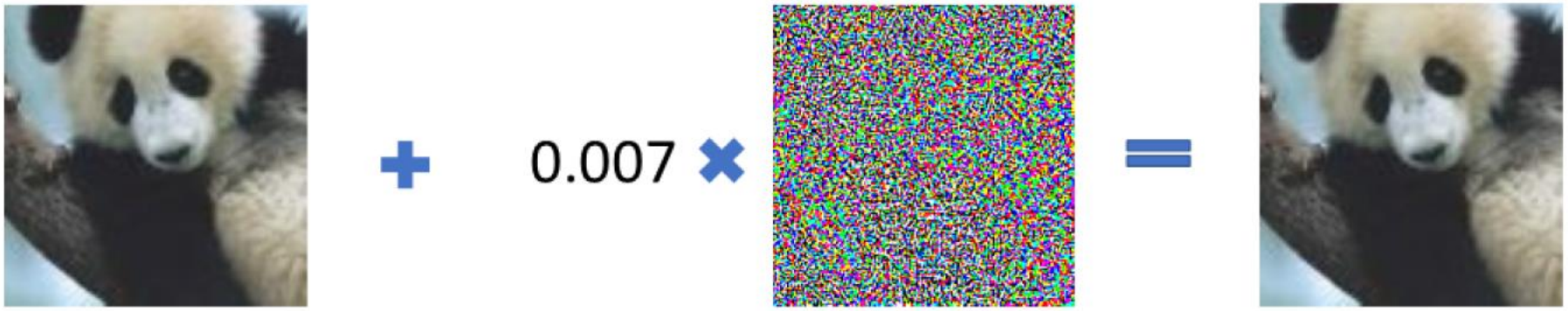
Deep Learning



Autonomous driving

Playing games

Deep neural networks are vulnerable



"panda"
57.7% confidence

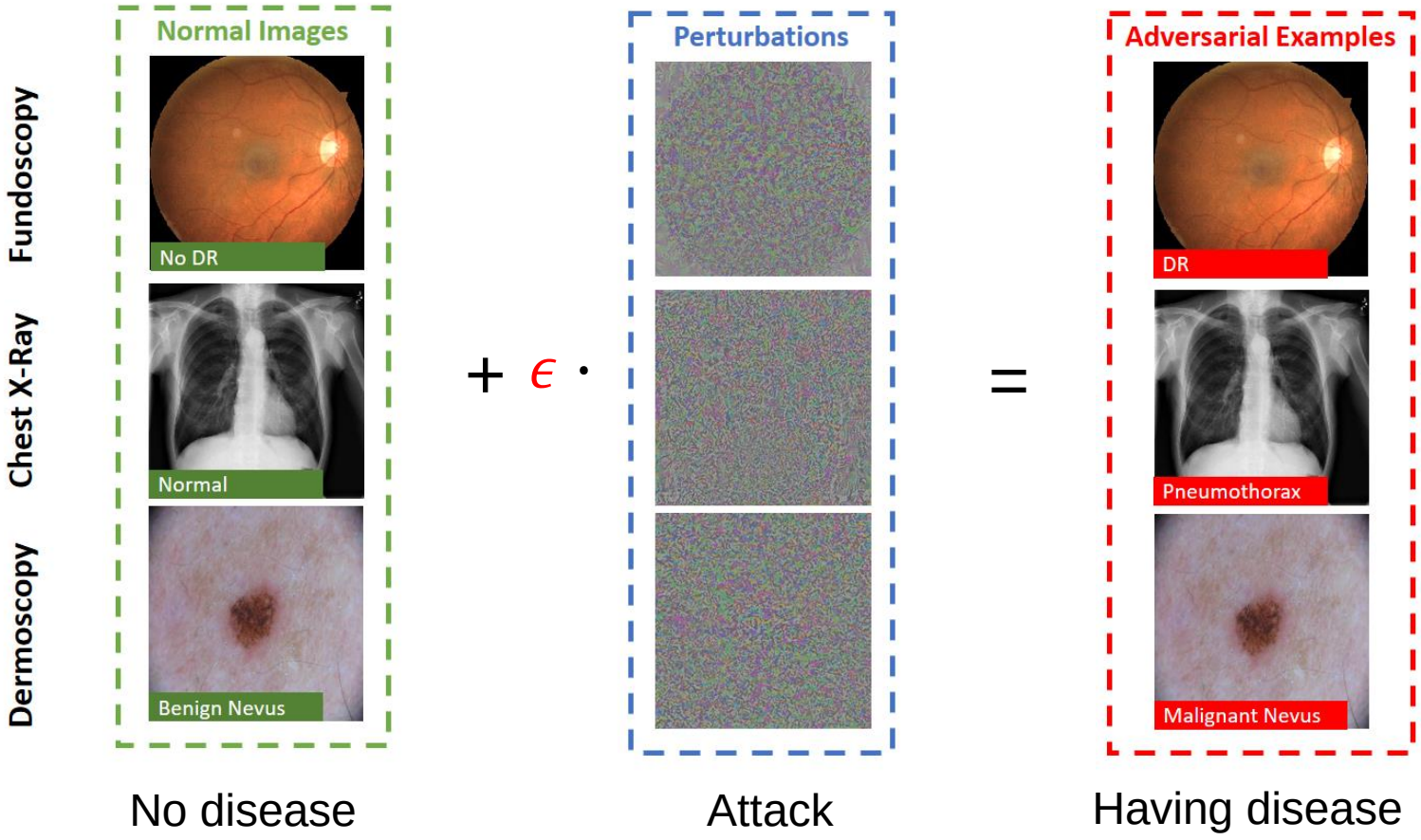
small adversarial
perturbations

"gibbon"
99.3 % confidence

Small perturbation can fool state-of-the-art ML models.

Szegedy et al. 2013, Goodfellow et al. 2014

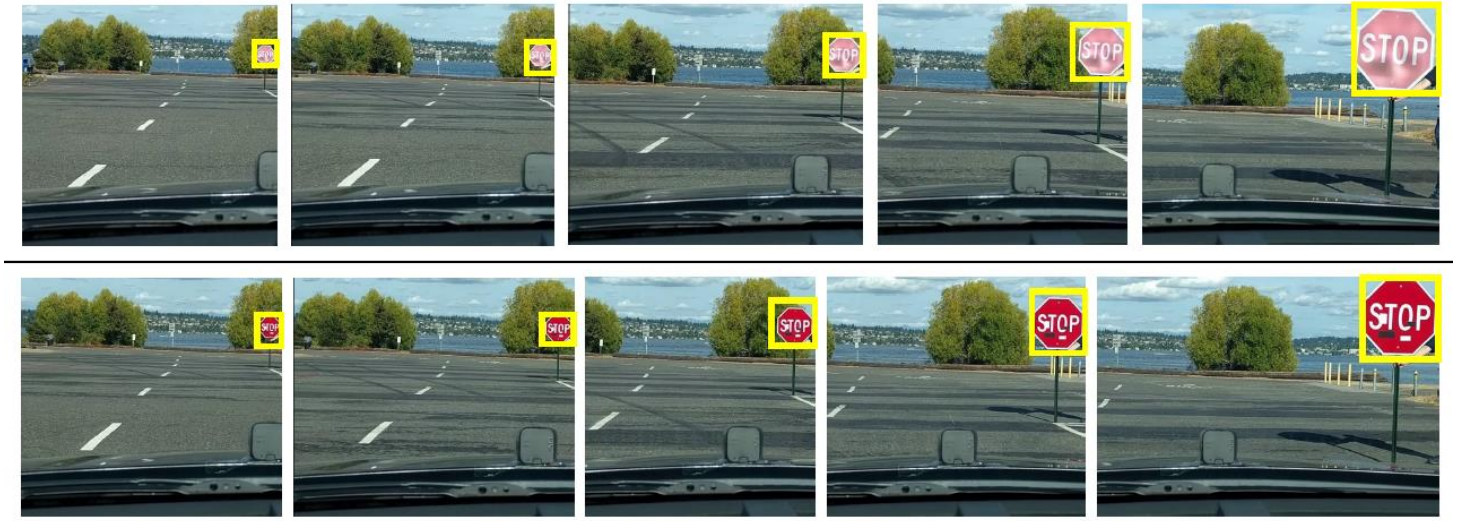
Security risks in medical diagnosis



Understanding Adversarial Attacks on Deep Learning Based Medical Image Analysis Systems
***Ma** et al., Pattern Recognition, 2020.*

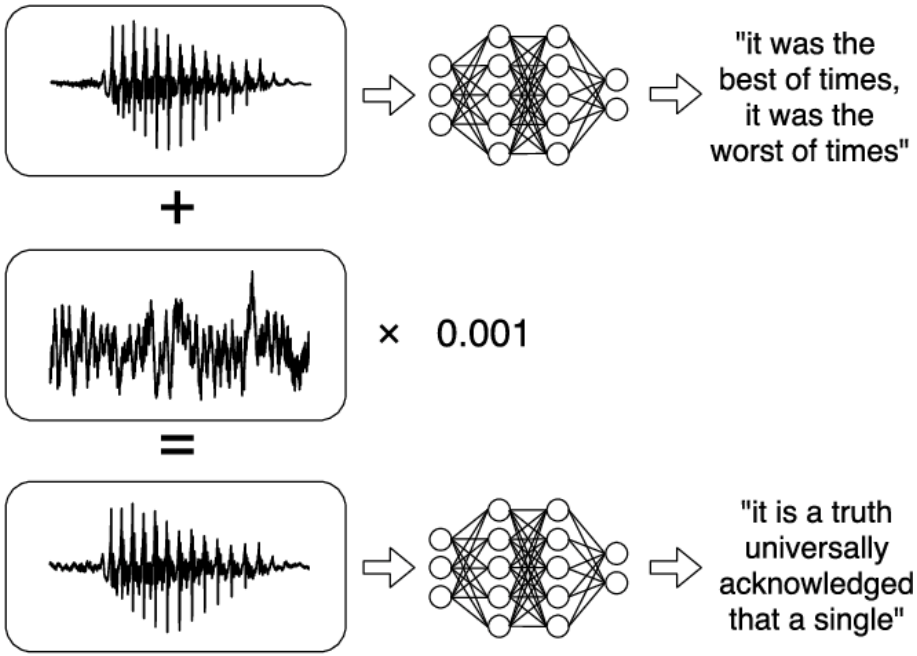
Security threats to autonomous driving

Adversarial traffic signs all recognized as:
45km speed limit.



Evtimov et al. 2017

Security risks in speech and NLP systems



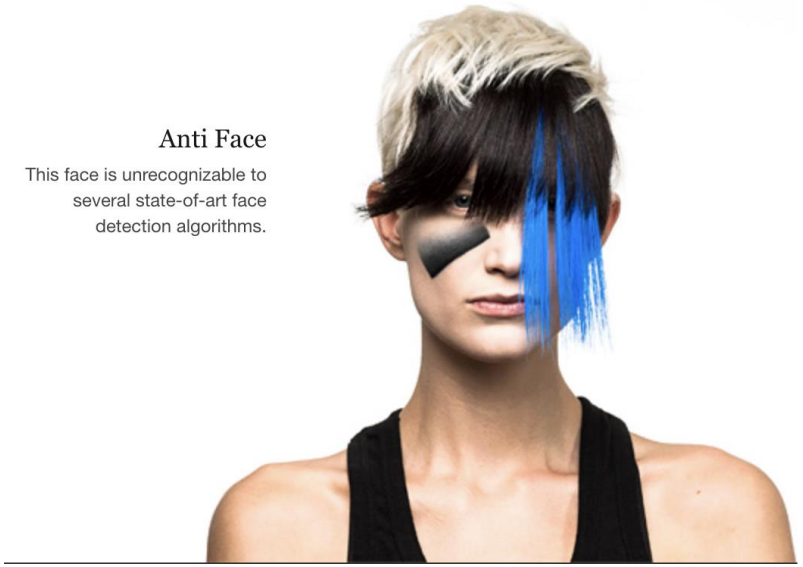
Original: What is the oncorhynchus also called? **A:** chum salmon
Changed: What's the oncorhynchus also called? **A:** keta

Original: How long is the Rhine? **A:** 1,230 km
Changed: How long is the Rhine?? **A:** more than 1,050,000

Carlini et al. 2018

Riberio et al. 2018

Security risks in face or object recognition



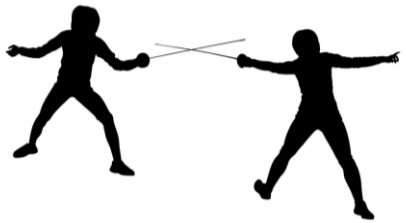
Brown et al. CVPRW, 2018

https://cvdazzle.com/

Research in adversarial machine learning

AML

Advs attack



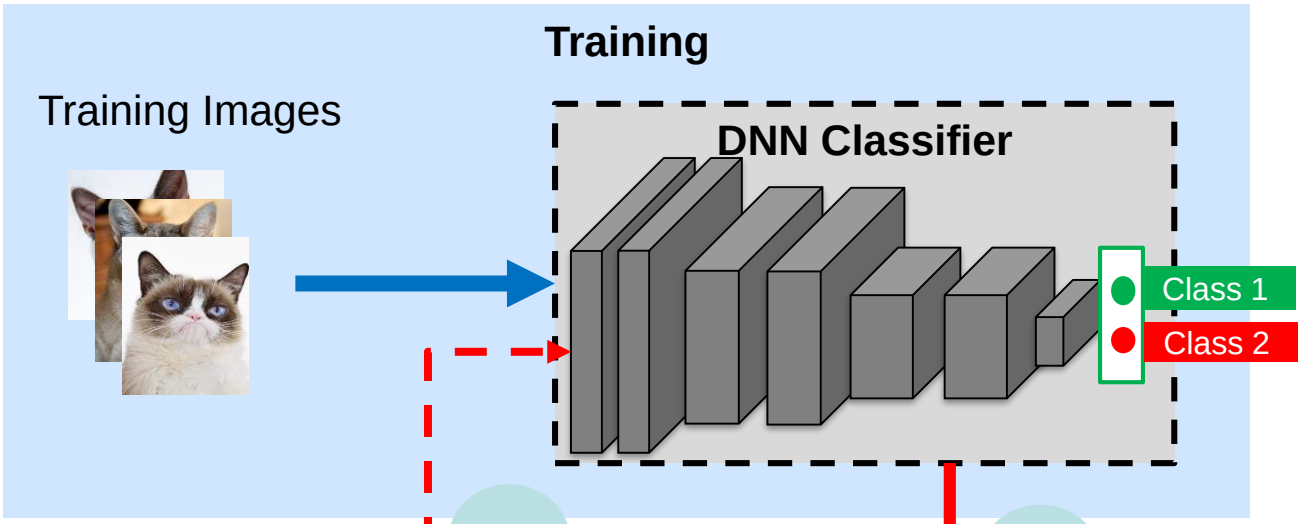
1. **White-box:** restricted (norm-bounded), semantic, sparse, …
2. **Black-box**: query-based, transferable
3. Image, audio, video, text
4. Digital vs Physical-world

Advs defense

1. Detection: natural or adversarial?
2. Adversarial training, robust optimization
3. Certifiable robustness
4. Data denoising, filtering
5. Model quantization, compression, pruning
6. Input gradient regularization

How adversarial examples are crafted

Train a model:



Adversarial Attack:

How adversarial examples are crafted

Model training:

$$\min_{\theta} \sum_{(x_i,\, y_i) \in D_{train}} L(f_\theta(x_i), y_i)$$

D_{train}: training data
x_i: training sample
y_i: class label
L: loss function
f_θ: model

Adversarial attack:

$$\max_{x'} L(f_\theta(x'), y) \quad \text{subject to } \|x' - x\|_p \leq \epsilon \text{ for } x \in D_{test}$$

increase error | small change | test time attack

$$\|x' - x\|_\infty \leq \epsilon = \frac{8}{255} \approx 0.031$$

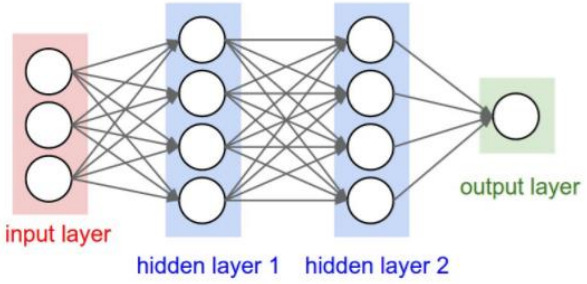
- Fast Gradient Sign Method (FGSM) (*Goodfellow et al., 2014*):

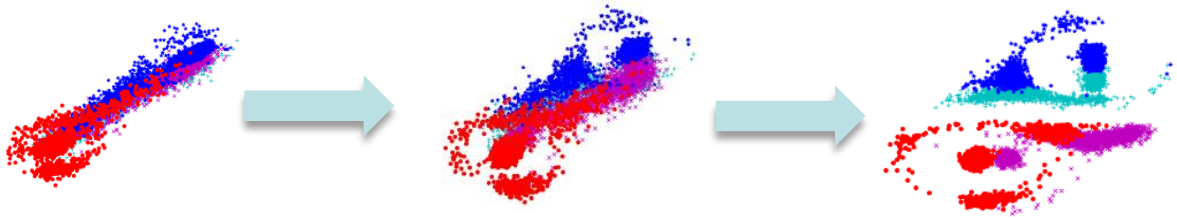
$$x' = x + \varepsilon \cdot \text{sign}\, \nabla_x L(f_\theta(x), y)$$

x': advs example

Why adversarial examples exist?

- Viewing DNN as a sequence of transformed spaces:

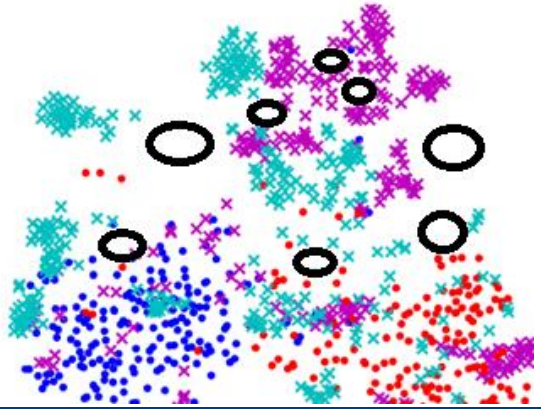




1st layer 10th layer 20th layer

Non-linear explanation:

- Non-linear transformations leads to the existence of small “pockets” in the deep space:
- Regions of **low probability** (not naturally occurring).
- **Densely scattered** regions.
- **Continuous** regions.
- **Close** to normal data subspace.



Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality.
***Ma**, et al. ICLR 2018*

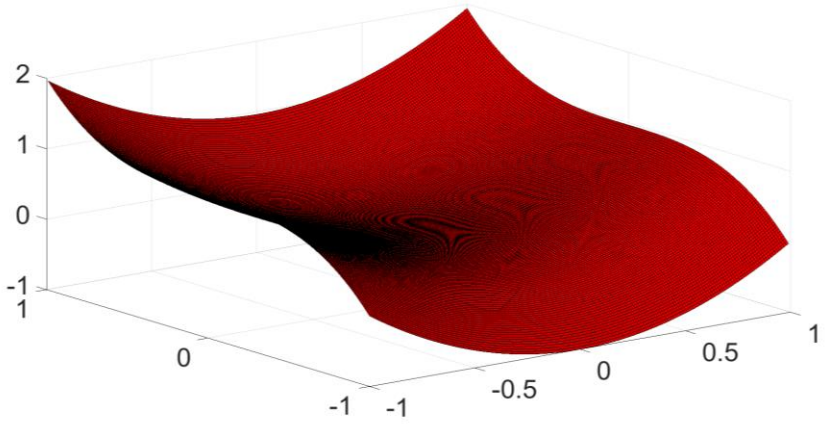
Szegedy et al. 2013

Insufficient training data?

- An illustrative example
  - $x \in [-1, 1), y \in [-1, 1), z \in [-1, 2)$
  - Binary classification
    - Class 1: $z < x^2 + y^3$
    - Class 2: $z \geq x^2 + y^3$
  - *x*, *y* and *z* are increased by 0.01

    → a total of $200 \times 200 \times 300$

    $= 1.2 \times 10^7$ points



- How many points are needed to reconstruct the decision boundary?
  - Training dataset: choose 80, 800, 8000, 80000 points randomly
  - Test dataset: choose 40, 400, 4000, 40000 points randomly
  - Boundary dataset (adversarial samples are likely to locate here):

    $x^2 + y^3 - 0.1 < z < x^2 + y^3 + 0.1$

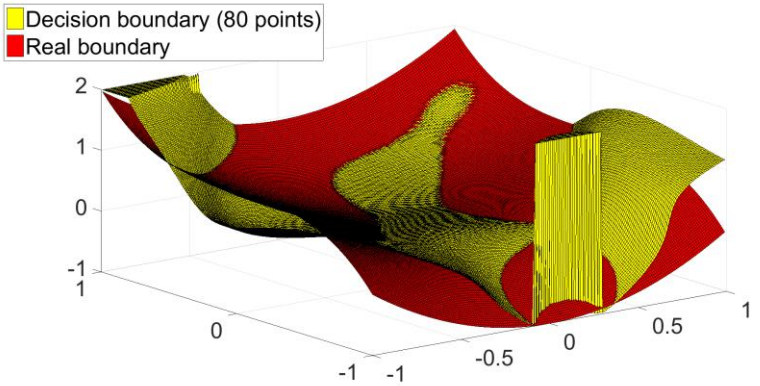
Insufficient training data?

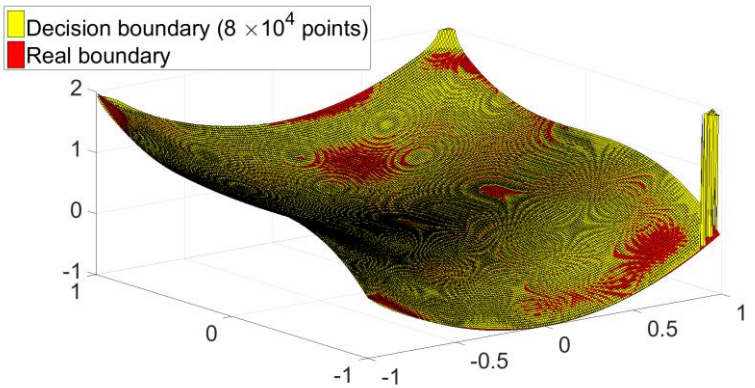
- **Test result**
  - RBF SVMs

Size of the training dataset	Accuracy on its own test dataset	Accuracy on the test dataset with 4×10^4 points	Accuracy on the boundary dataset
80	100	92.7	60.8
800	99.0	97.4	74.9
8000	99.5	99.6	94.1
80000	99.9	99.9	98.9

  - Linear SVMs

Size of the training dataset	Accuracy on its own test dataset	Accuracy on the test dataset with 4×10^4 points	Accuracy on the boundary dataset
80	100	96.3	70.1
800	99.8	99.0	85.7
8000	99.9	99.8	97.3
80000	99.98	99.98	99.5





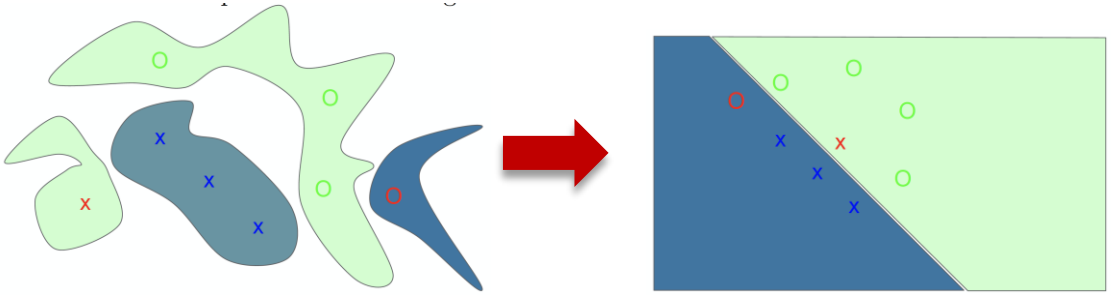
- 8000: 0.067% of 1.2×10^7
- MNIST: 28×28 8-bit greyscale images, $(2^8)^{28\times28} \approx 1.1\times10^{1888}$
- $1.1\times10^{1888}\times0.067\% \gg 6\times10^5$

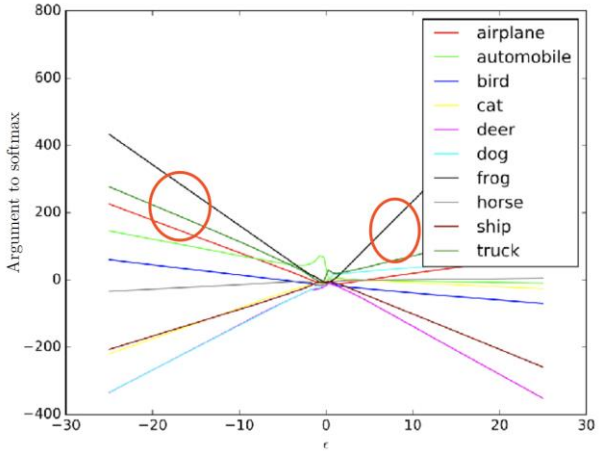
Why adversarial examples exist?

- Viewing DNN as a stack of linear operations: $w^T x + b$

Linear explanation*:*

- Adversarial subspaces span a contiguous multidimensional space:
- **Small changes at individual dimensions can sum up to significant change in final output:** $\sum_{i=0}^{n} x_i + \epsilon$.
- Adversarial examples can always be found if ϵ is large enough.

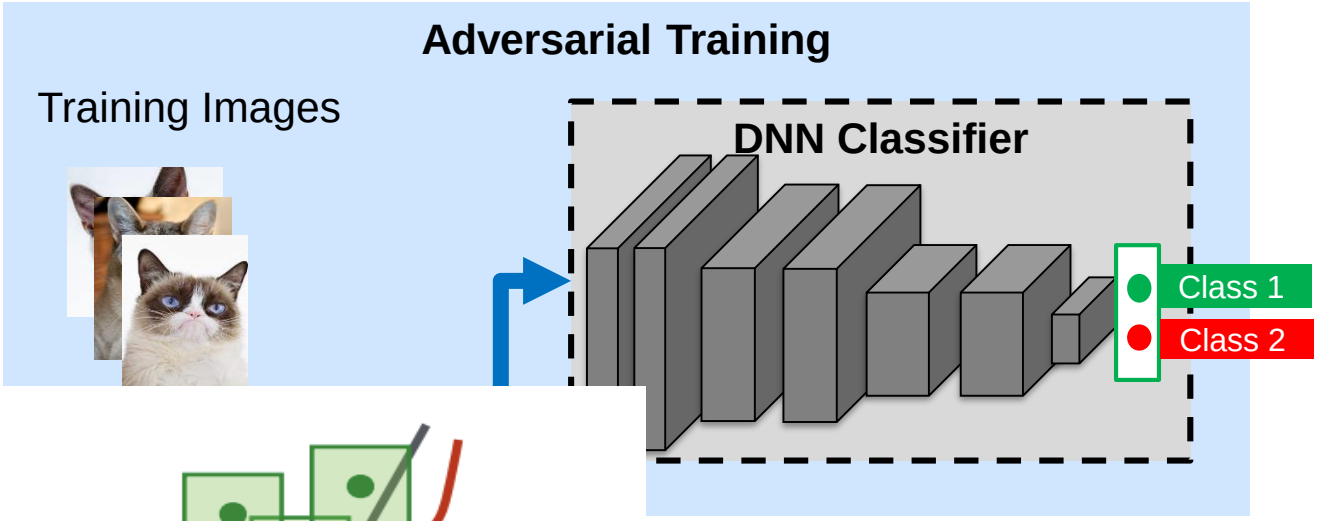


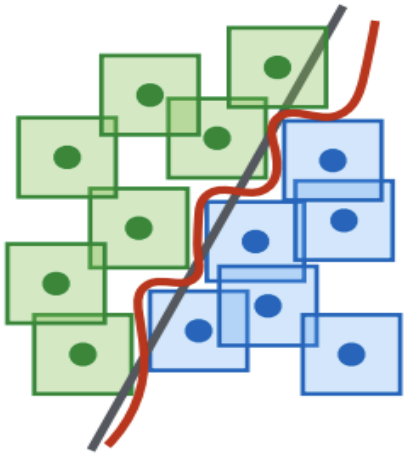


Goodfellow et al. 2014, 2016

State-of-the-art defense: adversarial training

Training models on adversarial examples.





- It explicitly generates more examples to fill the gap in the input space to improve robustness.

Adversarial training: robust optimization

Adversarial training is a **min-max optimization** process:

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{\|x_i'-x_i\|_p \le \epsilon} L(f_\theta(x_i'), y_i)}_{\text{attacking}}$$

L: loss, f_θ: model, $\mathrm{x_i}$: clean example, $\mathrm{y_i}$: class, $\mathrm{x_i'}$: adversarial example.

1. Inner Maximization:

- This is to generate adversarial examples, by maximizing the loss L.
- It is a constrained optimization problem: $\|x_i' - x_i\|_p \le \epsilon$.

2. Outer Minimization:

- A typical process to train a model, but on adversarial examples x_i' generated by the inner maximization.

On the Convergence and Robustness of Adversarial Training. *Wang*, **Ma***, et al., ICML 2019.*

Mary et al. ICLR 2018.

Misclassification-Aware adveRsarial Training (MART)

Improving Adversarial Robustness Requires Revisiting Misclassified Examples

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma and Quanquan Gu

ICLR 2020.

Misclassification-Aware adveRsarial Training (MART)

Adversarial risk:

$$\mathcal{R}(h_{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n} \max_{\mathbf{x}_i' \in \mathcal{B}_\epsilon(\mathbf{x}_i)} \mathbb{1}\big(h_{\boldsymbol{\theta}}(\mathbf{x}_i') \neq y_i\big),$$

Revisited adversarial risk (correctly- vs mis-classified):

$$\begin{aligned}\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{misc}}(h_{\boldsymbol{\theta}}) :&= \tfrac{1}{n}\Big(\textstyle\sum_{i\in\mathcal{S}^+_{h_{\boldsymbol{\theta}}}} \mathcal{R}^+(h_{\boldsymbol{\theta}}, \mathbf{x}_i) + \sum_{i\in\mathcal{S}^-_{h_{\boldsymbol{\theta}}}} \mathcal{R}^-(h_{\boldsymbol{\theta}}, \mathbf{x}_i)\Big) \\ &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \big\{\mathbb{1}(h_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_i') \neq y_i) + \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}_i) \neq h_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_i')) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}_i) \neq y_i)\big\}\end{aligned}$$

Misclassification-Aware adveRsarial Training (MART)

- Surrogate loss functions (existing methods and MART)

Defense Method	Loss Function
Standard	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y)$
ALP	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \|\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta})\|_2^2$
CLP	$\mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \|\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta})\|_2^2$
TRADES	$\mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \mathrm{KL}\big(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \| \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta})\big)$
MMA	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) = y) + \mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) \neq y)$
MART	$\mathrm{BCE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \mathrm{KL}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \| \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta})) \cdot (1 - \mathbf{p}_y(\mathbf{x}, \boldsymbol{\theta}))$

- Semi-supervised extension of MART:

$$\mathcal{L}_{\text{semi}}^{\text{MART}}(\boldsymbol{\theta}) = \sum_{i \in \mathcal{S}_{\text{sup}}} \ell_{\text{sup}}^{\text{MART}}(\mathbf{x}_i, y_i; \boldsymbol{\theta}) + \gamma \cdot \sum_{i \in \mathcal{S}_{\text{unsup}}} \ell_{\text{unsup}}^{\text{MART}}(\mathbf{x}_i, y_i; \boldsymbol{\theta})$$

Misclassification-Aware adveRsarial Training (MART)

- White-box robustness: ResNet-18, CIFAR-10, $\epsilon = 8/255$

Defense	MNIST				CIFAR-10			
	Natural	FGSM	PGD^{20}	CW_∞	Natural	FGSM	PGD^{20}	CW_∞
Standard	99.11	97.17	94.62	94.25	84.44	61.89	47.55	45.98
MMA	98.92	97.25	95.25	94.77	**84.76**	62.08	48.33	45.77
Dynamic	98.96	97.34	95.27	94.85	83.33	62.47	49.40	46.94
TRADES	**99.25**	96.67	94.58	94.03	82.90	62.82	50.25	48.29
MART	98.74	**97.87**	**96.48**	**96.10**	83.07	**65.65**	**55.57**	**54.87**

- White-box robustness: WideResNet-34-10, CIFAR-10, $\epsilon = 8/255$

Defense	Natural	FGSM		PGD^{20}		PGD^{100}		CW_∞	
		Best	Last	Best	Last	Best	Last	Best	Last
Standard	**87.30**	56.10	56.10	52.68	49.31	51.55	49.03	50.73	48.47
Dynamic	84.51	63.53	63.53	55.03	51.70	54.12	50.07	51.34	49.27
TRADES	84.22	64.70	64.70	56.40	53.16	55.68	51.27	51.98	51.12
MART	84.17	**67.51**	**67.51**	**58.56**	**57.39**	**57.88**	**55.04**	**54.58**	**54.53**

Misclassification-Aware adveRsarial Training (MART)

- White-box robustness: unlabled data, CIFAR-10, $\epsilon = 8/255$

a) WideResNet-34-8 with 100K unlabeled data

Defense	Natural	PGD^{20}
UAT++	86.04	59.41
RST	88.24	59.60
MART	86.68	61.88

b) WideResNet-28-10 with 500K unlabeled data

Defense	Natural	PGD^{20}
UAT++	86.21	62.76
RST	89.70	63.10
MART	86.30	65.04

Transferable attack with skip connections

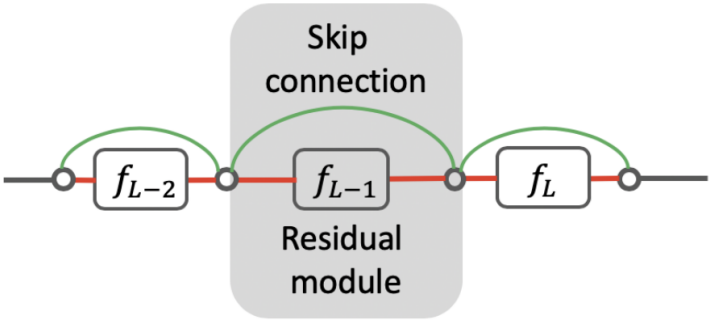
Skip Connections Matter: on the Transferability of Adversarial Examples Generated with ResNets

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey and Xingjun Ma.

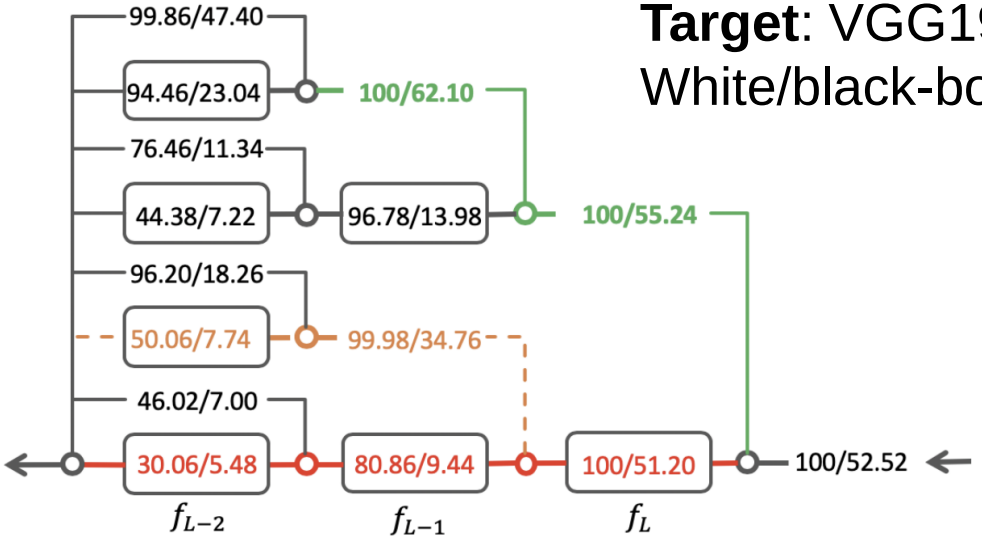
ICLR 2020.

Structural weakness of ResNets?

- Gradient backpropagation with skip connections



Source: ResNet-18
Target: VGG19
White/black-box



Skip the gradients incrases transferability!

Transferable attack with skipped gradients

- New attack method: skip gradient method (SGM)

$$\boldsymbol{x}_{adv}^{t+1} = \Pi_{\epsilon}\Big(\boldsymbol{x}_{adv}^{t} + \alpha \cdot \mathrm{sign}\big(\frac{\partial \ell}{\partial \boldsymbol{z}_L}\prod_{i=0}^{L-1}(\gamma\frac{\partial f_{i+1}}{\partial \boldsymbol{z}_i} + 1)\frac{\partial \boldsymbol{z}_0}{\partial \boldsymbol{x}}\big)\Big)$$

Breaking down a network f according to its L residual blocks.

	RN18	RN34	RN50	RN101	RN152	DN121	DN169	DN201
PGD	23.23±0.69	24.38±0.41	22.80±0.55	22.98±0.83	26.56±0.75	30.71±0.60	30.90±0.31	36.01±0.59
SGM	**28.92±0.45**	**43.43±0.32**	**36.71±0.55**	**38.38±0.53**	**44.84±0.14**	**57.38±0.14**	**60.45±0.42**	**65.48±0.23**

ImageNet, target: Inception V3, $\epsilon = 16/255$

How much can SGM increases transferability?

Combined with existing methods: the success rates (%) of attacks crafted on source model DN201 against 7 target models.

Attack \Target	VGG19	RN152	DN201	SE154	IncV3	IncV4	IncRes
MI	75.09	76.39	**99.84**	64.38	59.62	54.85	50.05
MI+SGM	+12.01	+13.24	99.52	+17.16	+21.88	+15.57	+18.35
DI	78.11	78.18	99.81	61.75	60.04	56.15	49.00
DI+SGM	+12.28	+13.76	99.52	+20.92	+17.66	+15.78	+20.20
MI+DI	87.16	87.28	99.76	79.80	76.68	75.20	71.05
MI+DI+SGM	**93.00**	**93.92**	99.42	**89.86**	**85.72**	**81.23**	**80.50**

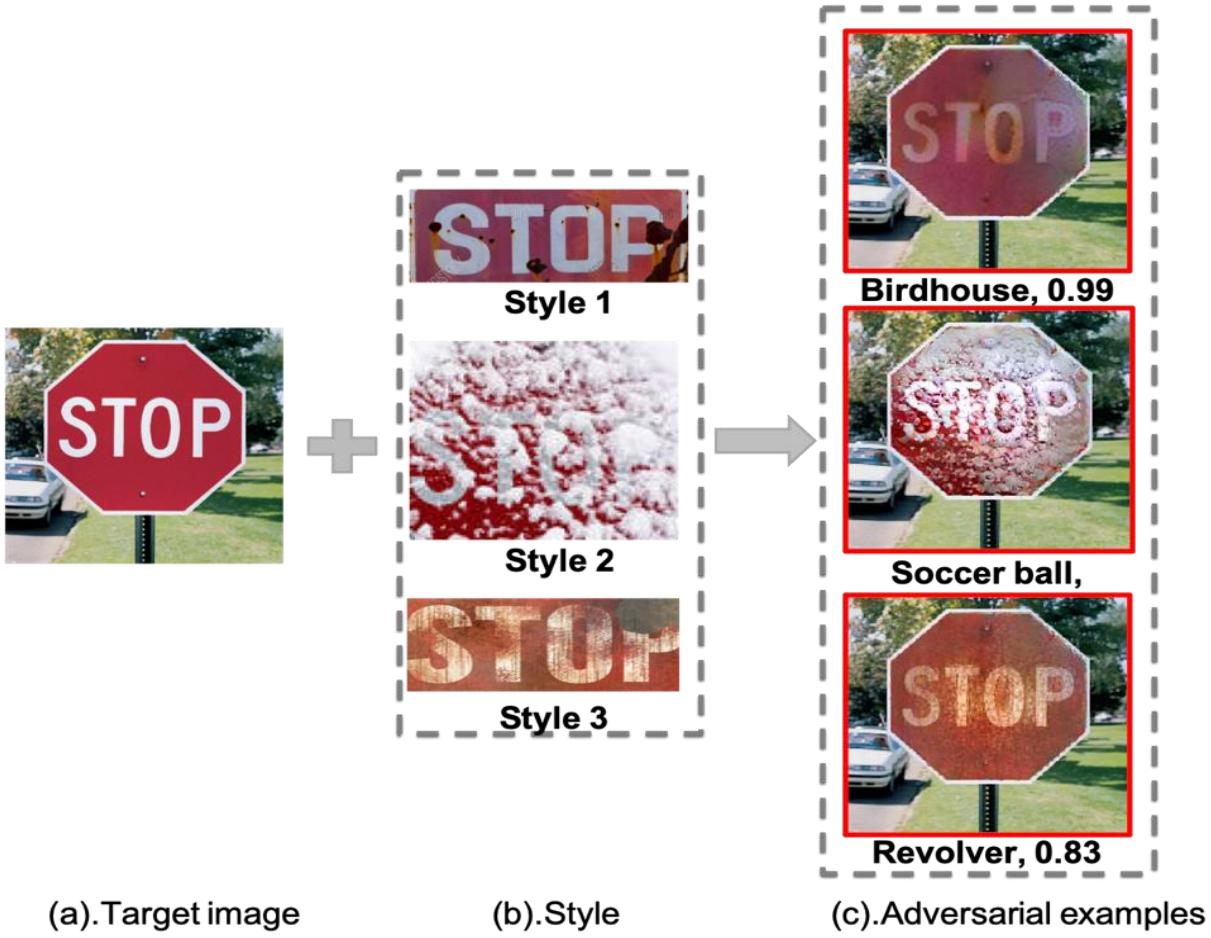
Adversarial camouflage attack

Adversarial Camouflage: Hiding Adversarial Examples with Natural Styles

Ranjie Duan, **Xingjun Ma**, Yisen Wang, James Bailey, Kai Qin, Yun Yang

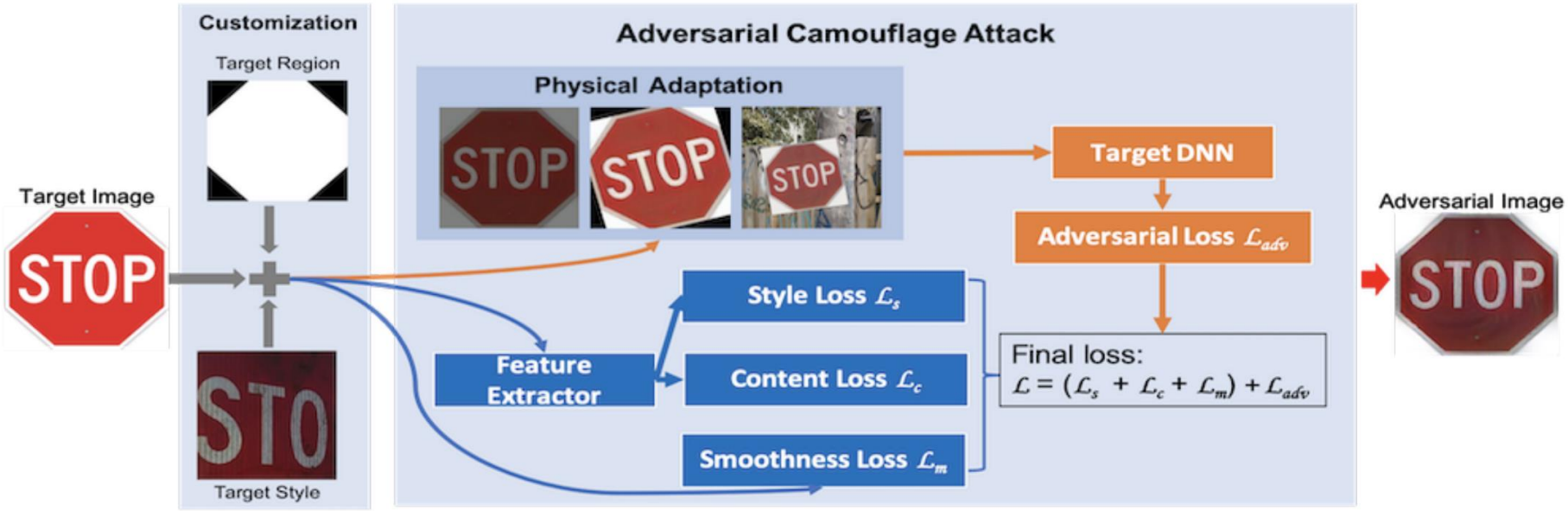
CVPR 2020.

Adversarial camouflage



Camouflage adversarial examples with customized styles.

Adversarial camouflage



Making large perturbations look natural:
Adversarial attack + **style transfer**

Adversarial camouflage

(a) RP_2 (b) AdvCam (c) AdvPatch (d) *AdvCam*

A visually comparison to existing attacks

Adversarial camouflage

Revolver --> Toilet tissue

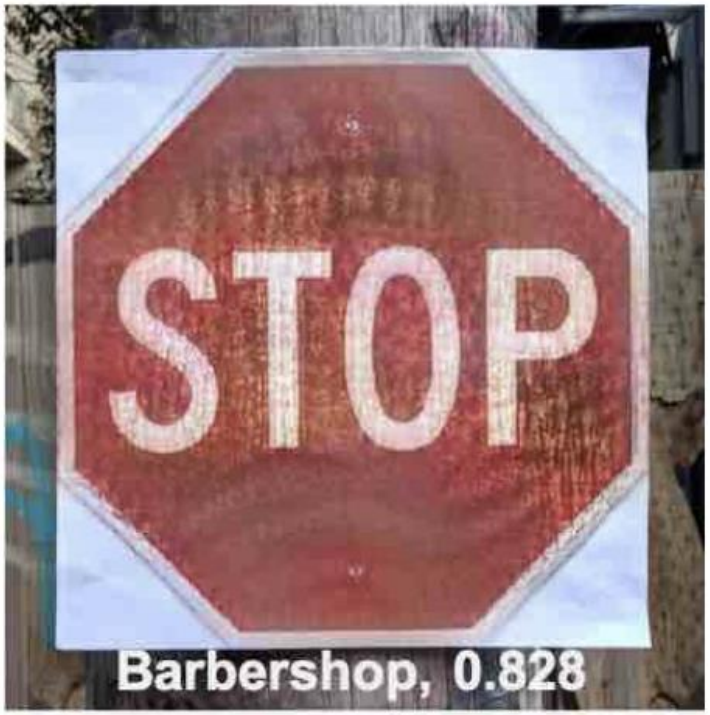
Minivan --> Traffic light

Scabbard --> Purse

Attacking the background is what makes the attack stealthy and ubiquitous.

Examples of camouflaged digital attacks

Adversarial camouflage



Traffic sign -> Barbershop

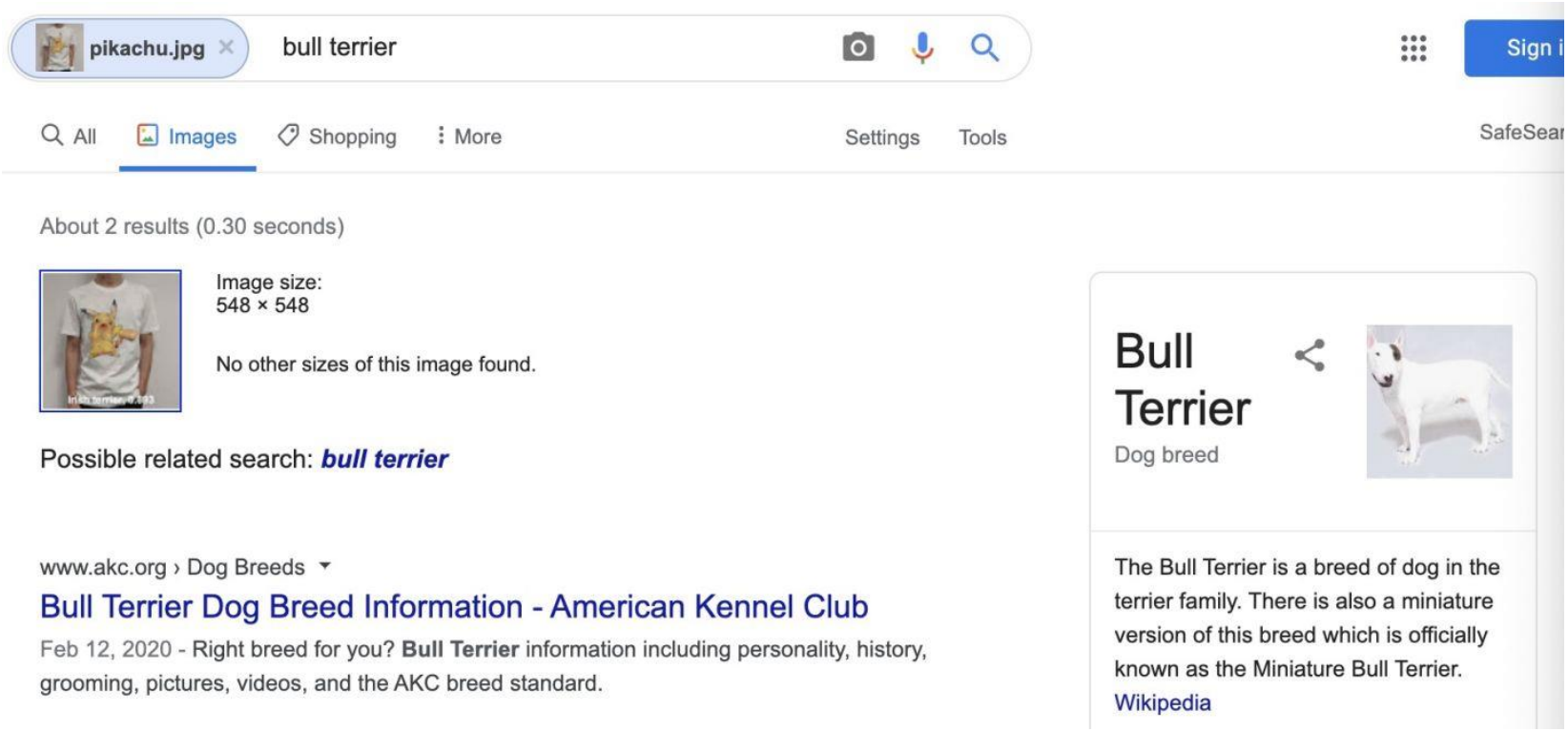
Tree -> Street sign

Examples of camouflaged physical-world attacks

Using adversarial camouflage to protect privacy

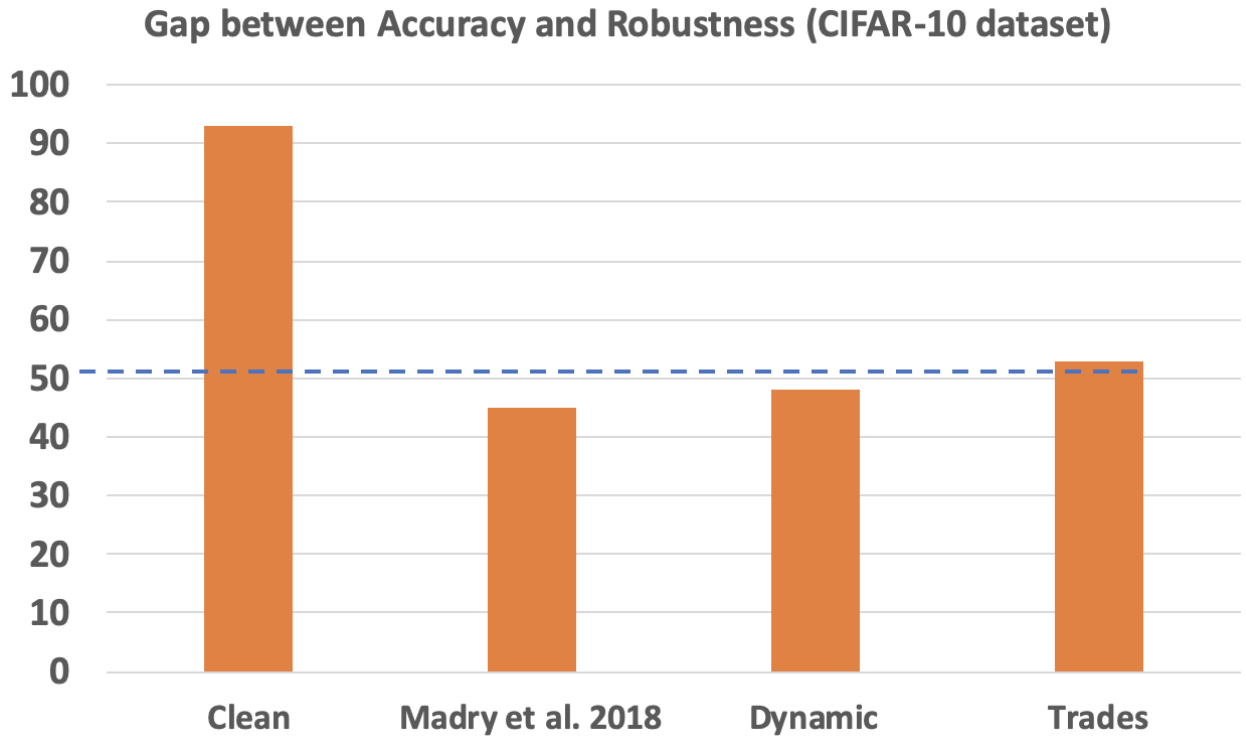
Here is an adversarial pikachu to protect you!

This is a dog to Google Image Search.



Thank you!

The huge gap between natural accuracy and robustness



93% vs **53%**!

Model: WideResNet-28-10

Dataset: CIFAR-10

Perturbation: $\epsilon = 8/255$

Attack: 20 step PGD