

Adversarial Deep Learning

Ben Zhao, Blase Ur, David Cash

November 28, 2018

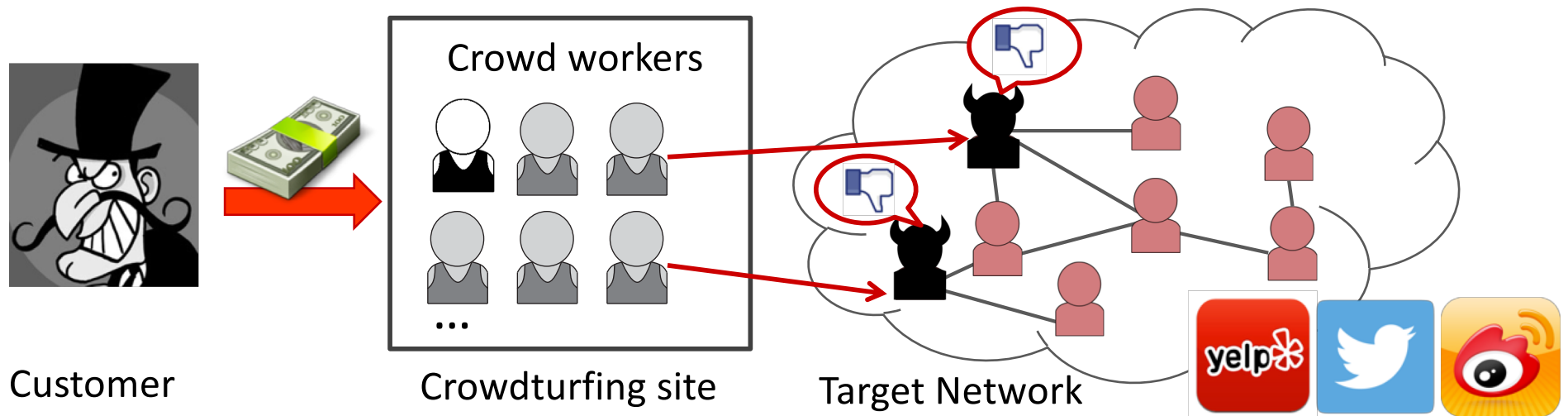
CS 232/332



THE UNIVERSITY OF
CHICAGO

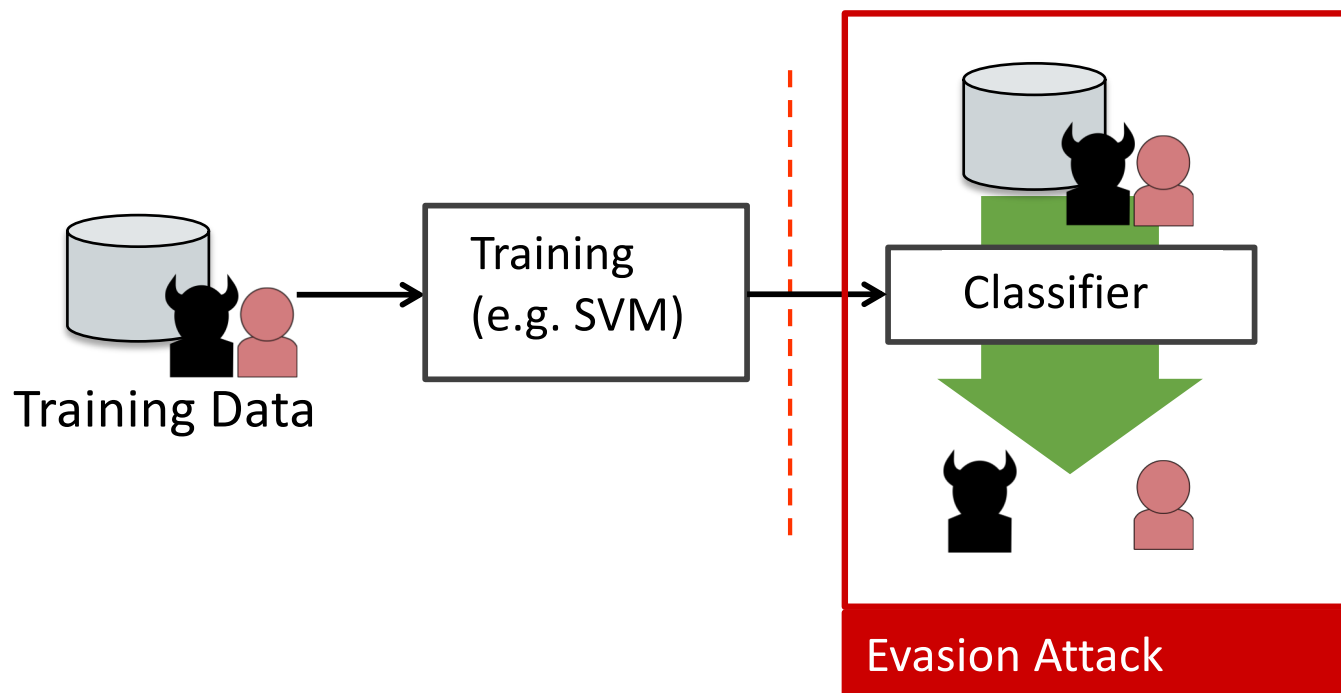
Online Crowdturfing Systems

- Online crowdturfing systems (services)
 - Connect customers with online users willing to spam for money
 - Sites located across the globe, e.g. China, US, India

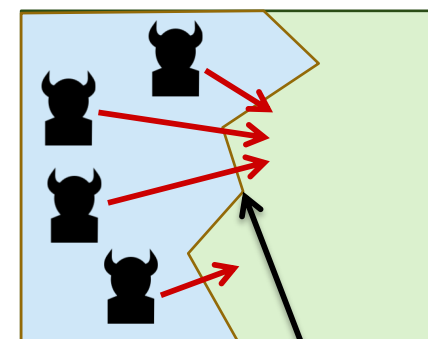


- Crowdturfing in China
 - Revenue: hundreds of millions of dollars per year
 - Now rapidly growing in US (Fiverr & similar sites)

Attack #1: Adversarial Evasion

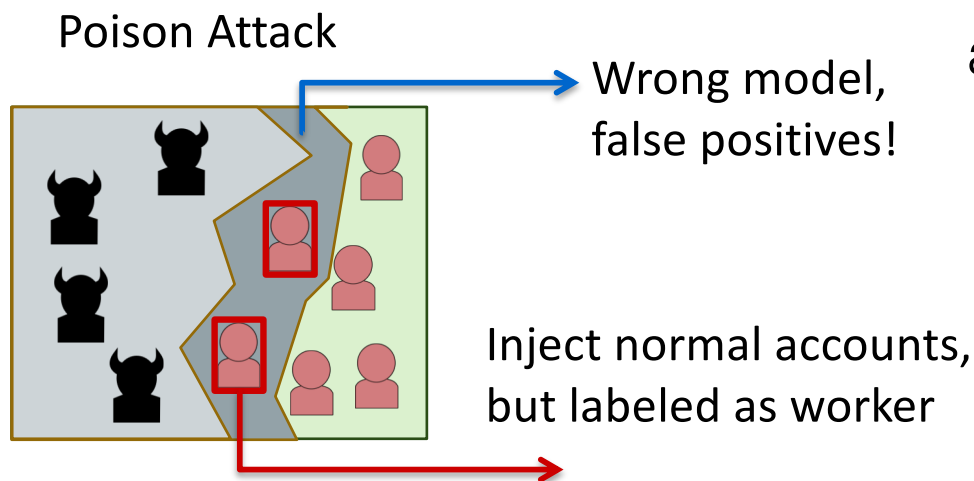
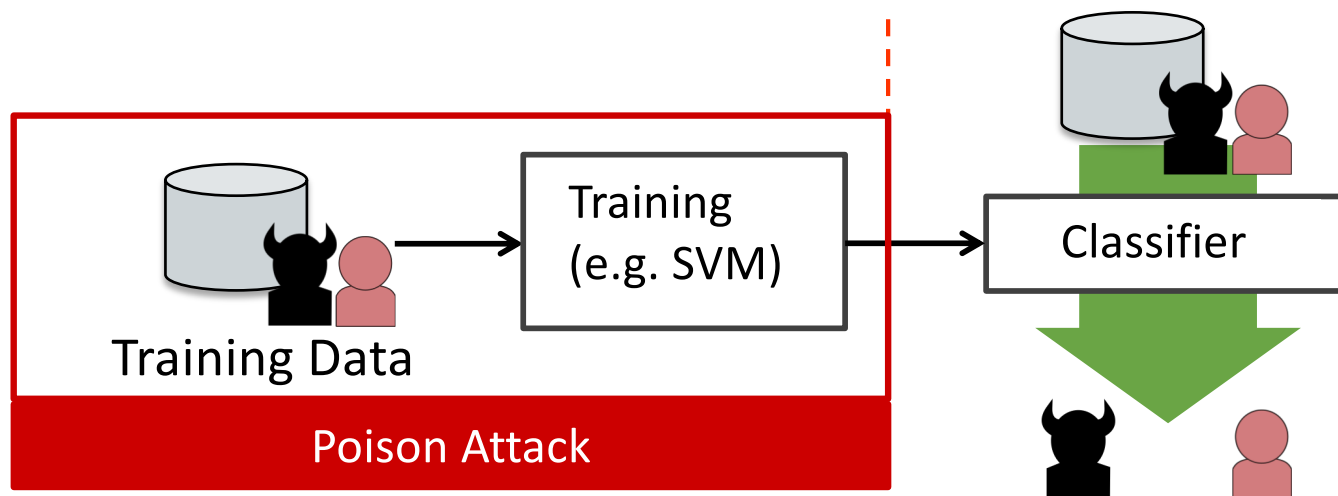


- **Individual workers** as adversaries
 - Workers evade classifier by mimicking normal users
 - Easy if model parameters known or predictable



Classification boundary

Attack #2: Poisoning Attack



- Crowdturfing site admins as adversaries
 - Highly motivated to protect their workers, centrally control workers
 - Tamper with the training data to manipulate model training
 - Inject mislabeled samples to training data → useless classifier

Attack Taxonomy Continued

- Model Inversion Attack
 - Extract private and sensitive inputs by leveraging outputs and ML model
- Model Extraction/Inference Attack
 - Extract model parameters by querying model



Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Model	OHE	Binning	Queries	Time (s)	Price (\$)
Circles	-	Yes	278	28	0.03
Digits	-	No	650	70	0.07
Iris	-	Yes	644	68	0.07
Adult	Yes	Yes	1,485	149	0.15

Table 7: Results of model extraction attacks on Amazon. OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of 10^{-3}), plus those queries used for equation-solving. Amazon charges \$0.0001 per prediction [1].

Today

- Adversarial attacks on deep learning
 - White box perturbation attacks

Accessorize to a Crime: Real and Stealthy Attacks on State-Of-The-Art Face Recognition, Sharif et al, CCS 2016
 - Transfer learning attacks

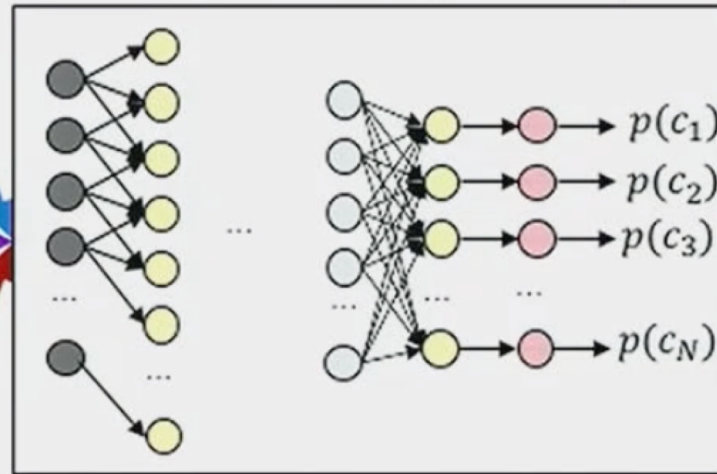
With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning, Wang et al, USENIX Security 2018
 - Backdoor attacks

Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, Wang et al, IEEE S&P (Oakland) 2019

Image Recognition Example



Deep Neural Network (DNN)

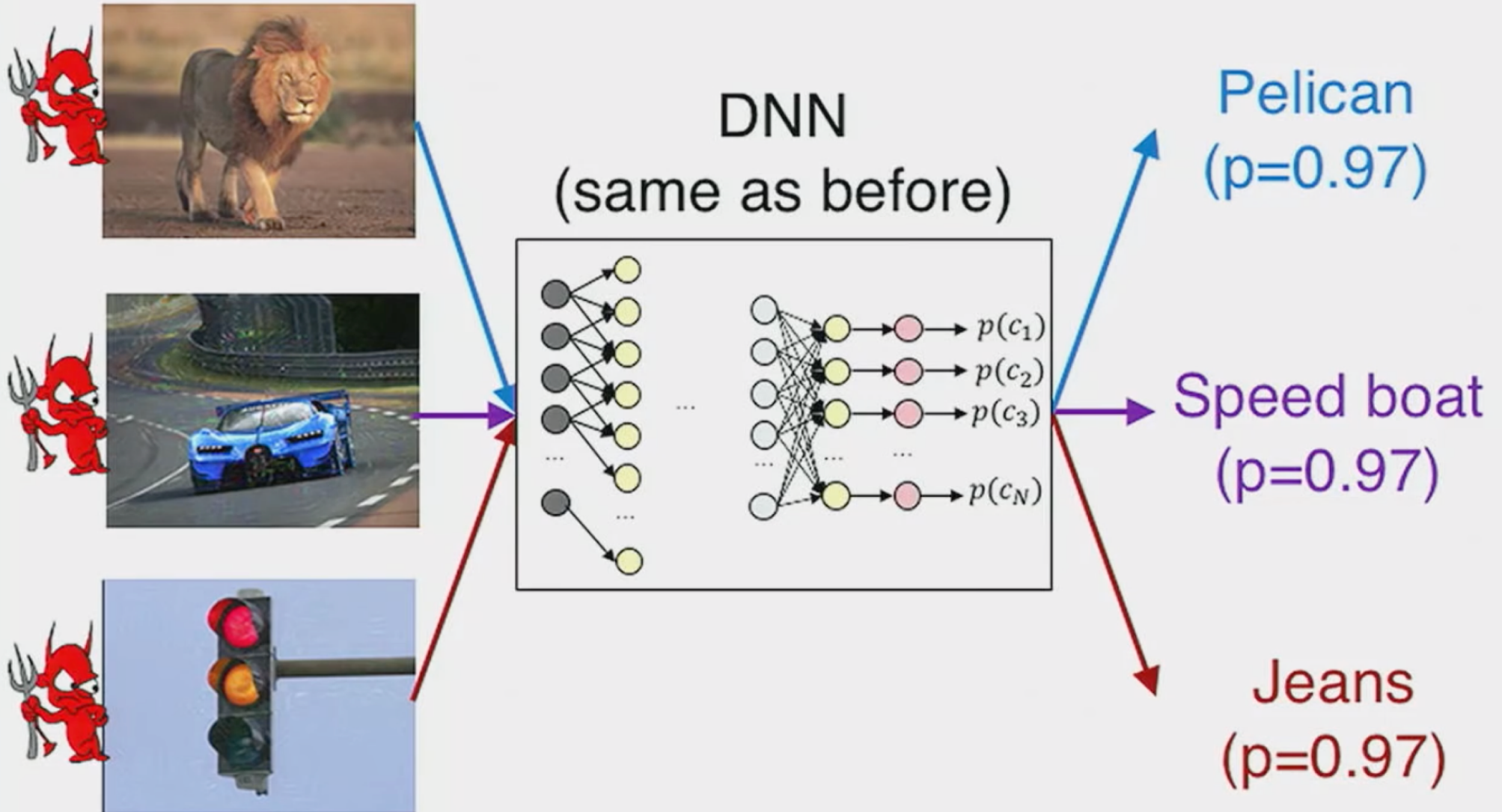


Lion
($p=0.99$)

Race car
($p=0.74$)

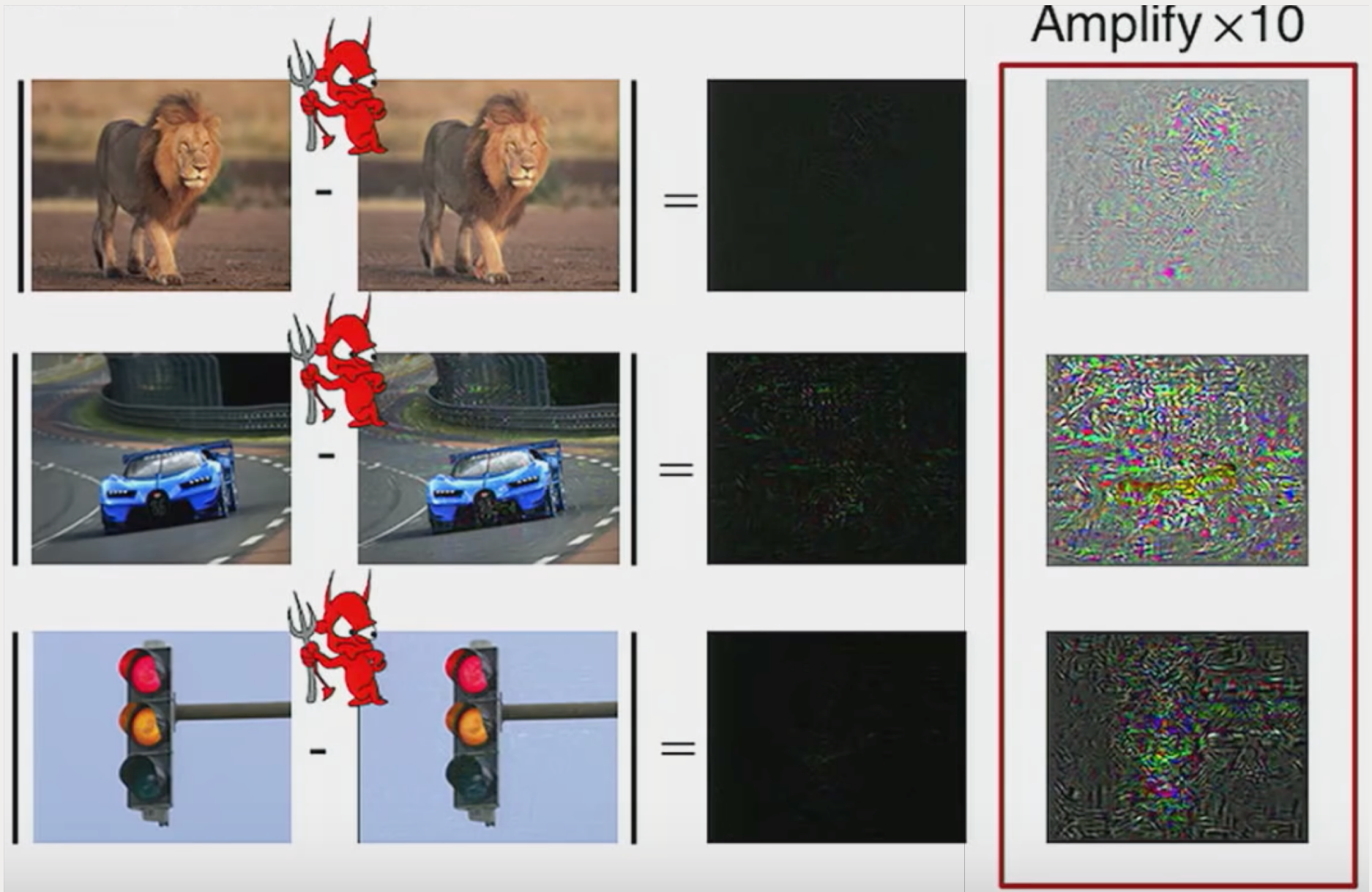
Traffic light
($p=0.99$)

Perturbed Inputs



[Szegedy et al., ICLR '14]

A Very Small Delta



Practical White Box Attacks

- Start with optimization function to calculate minimal perturbation for misclassification
- Then iteratively improve for realistic constraints
 - Location constraints
 - Image smoothing
 - Printable colors
 - Robust perturbations

Imperceptible adversarial examples
[Szegedy et al., ICLR '14]

Defined as an optimization problem:

$$\operatorname{argmin}_r \underbrace{|f(x + \mathbf{r}) - c_t|}_{\text{misclassification}} + \kappa \cdot \underbrace{\|\mathbf{r}\|}_{\text{norm}}$$

x : input image

$f(\cdot)$: classification function (e.g., DNN)

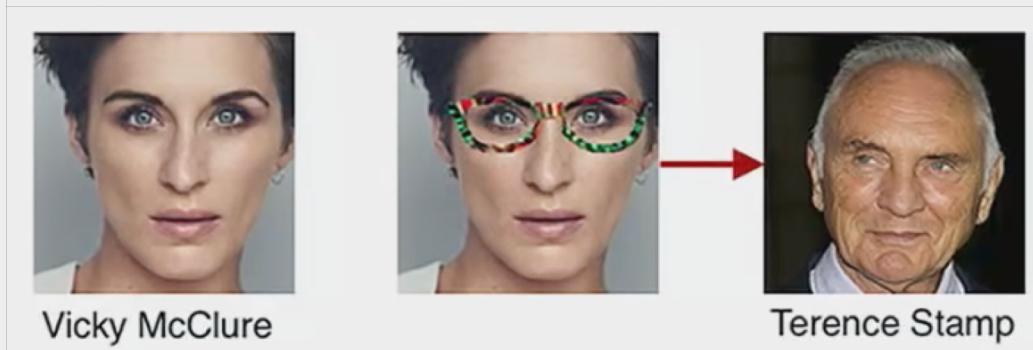
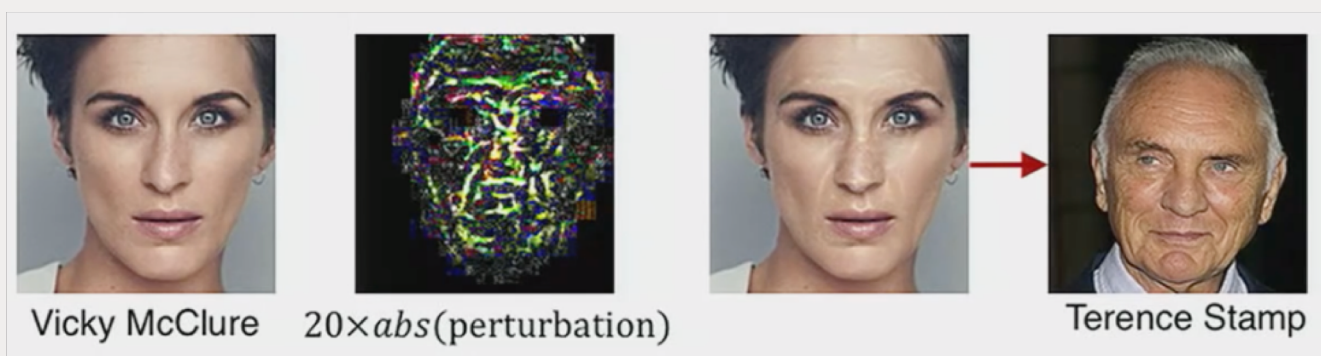
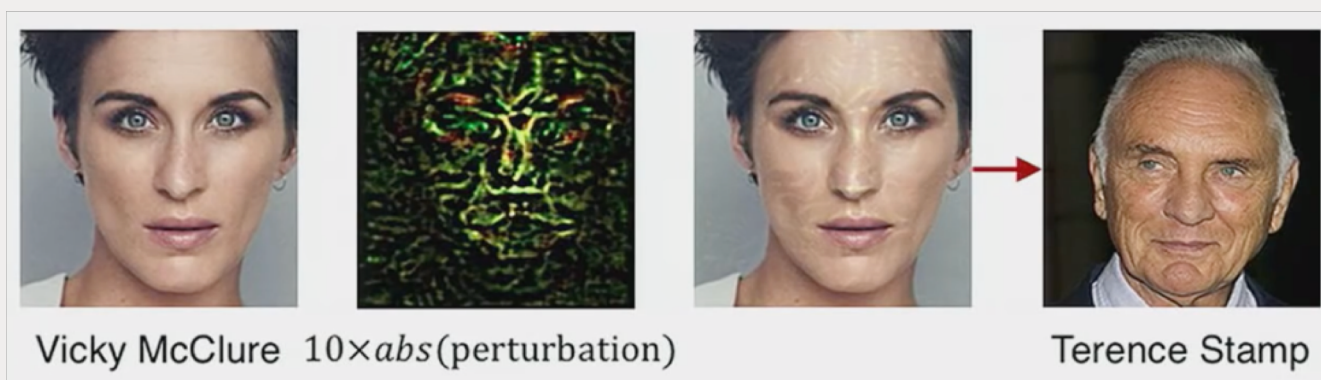
$\|\cdot\|$: norm function (e.g., Euclidean norm)

c_t : target class

r : perturbation

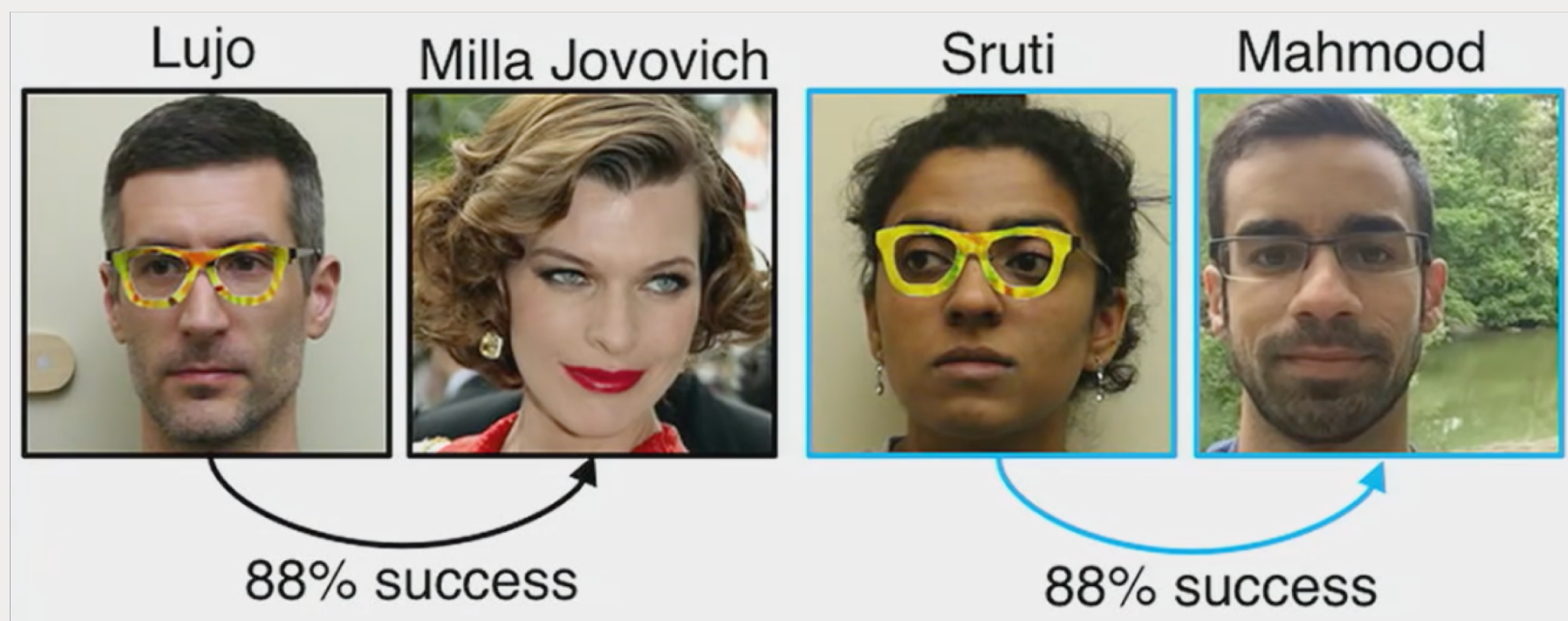
κ : tuning parameter

Spatial Constraints on Perturbations



End Result Quite Impressive

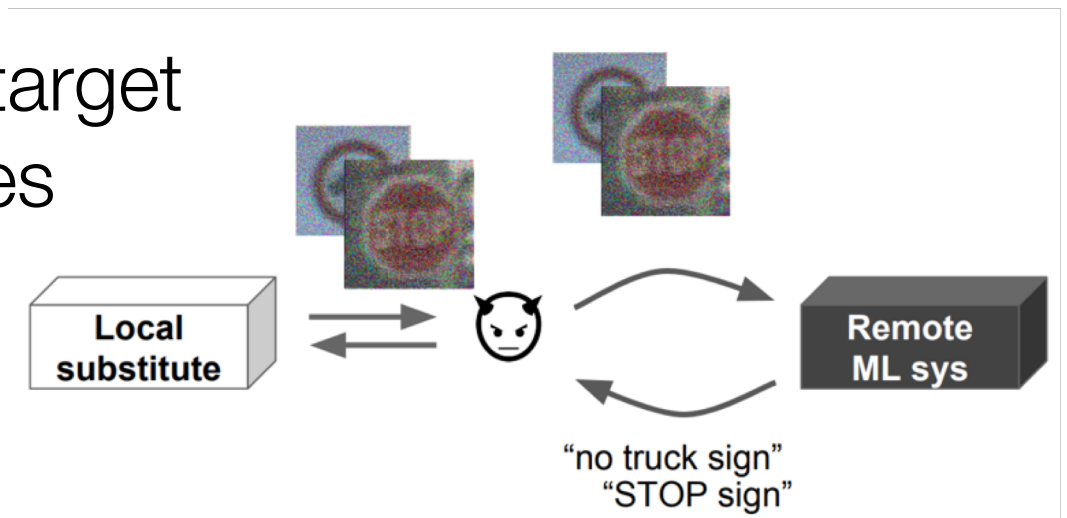
- Validated on limited model with 2200+ output labels






Accessorize to a Crime: Real and Stealthy Attacks on State-Of-The-Art Face Recognition, Sharif, Bhagavatula, Bauer, Reiter, CCS 2016

Too Strong an Attack Model?

- White box assumes full access to model
 - Impractical in real world scenarios
 - Equivalent to bank handing over combination to vault
- Black box attacks
 - Repeatedly query target model until achieves misclassification



Black Box Attacks Work, Sort of...

Remote Platform	ML technique	Number of queries	Adversarial examples misclassified (after querying)
 MetaMind	Deep Learning	6,400	84.24%
 amazon web services™	Logistic Regression	800	96.19%
 Google Cloud Platform	Unknown	2,000	97.72%

All remote classifiers are trained on the MNIST dataset (10 classes, 60,000 training samples)

- Downside
 - Requires thousands of queries, easily detected in practice

Attack on Transfer Learning

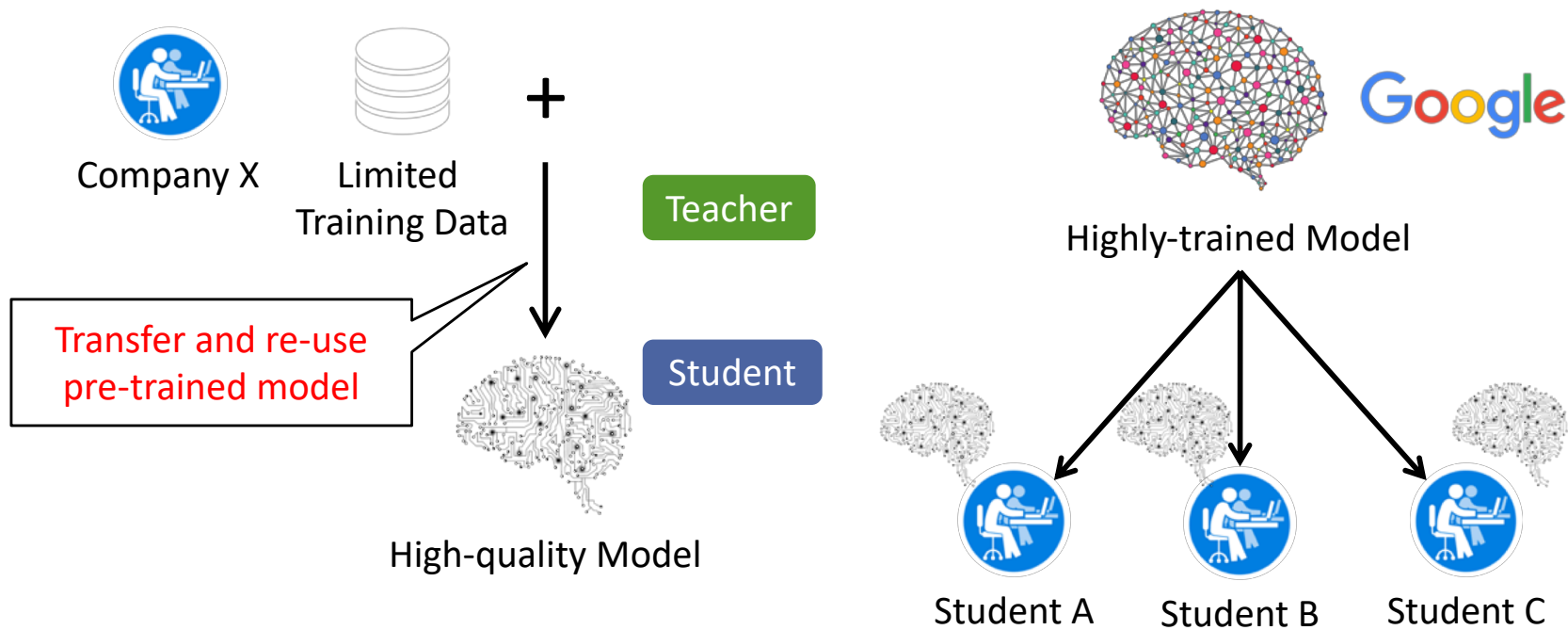


Where do small companies get such large datasets?



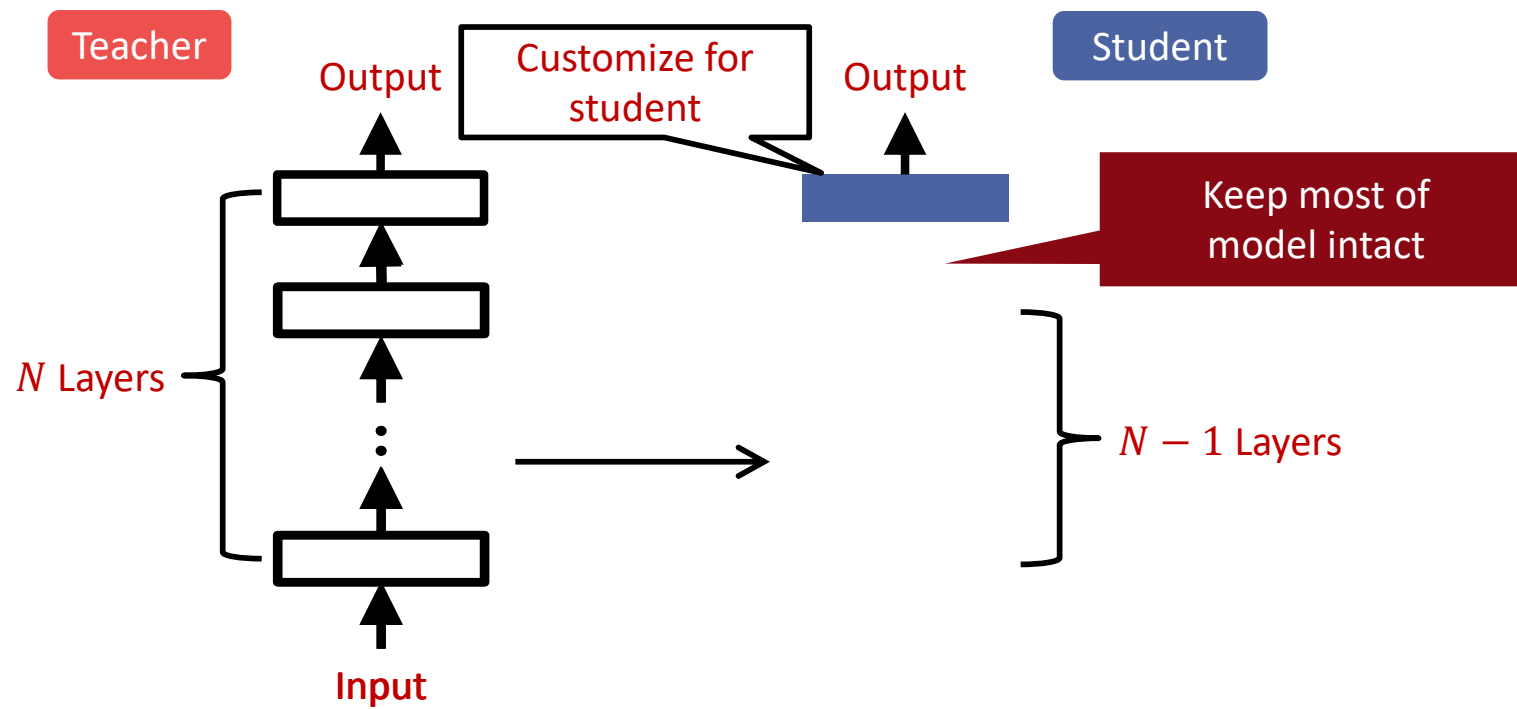
- High-quality models trained using large labeled datasets
 - Vision: ImageNet contains 14+ million labeled images

Default Solution: Transfer Learning

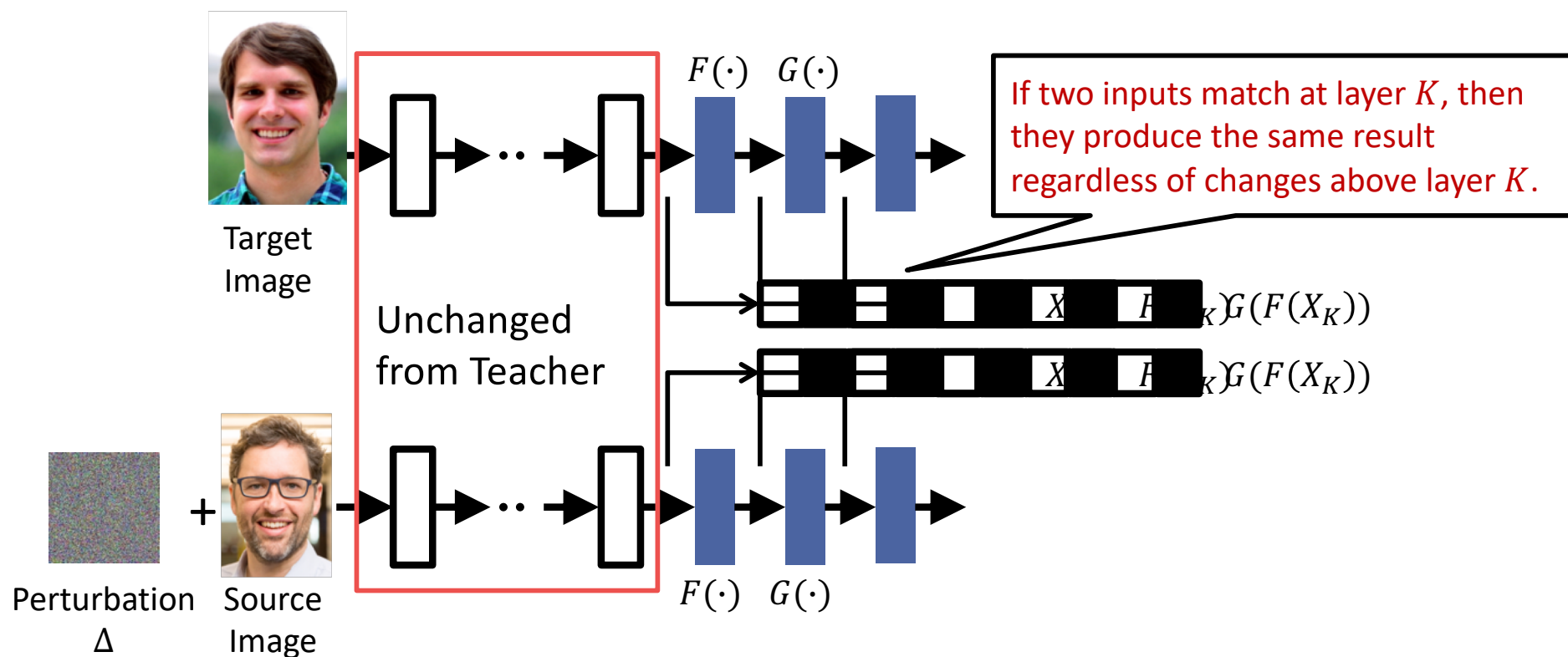


Recommended by *Google, Microsoft, and Facebook*
(used in CCS 2016 attack)

Transfer Learning: Details



Attack by Mimicking Neurons



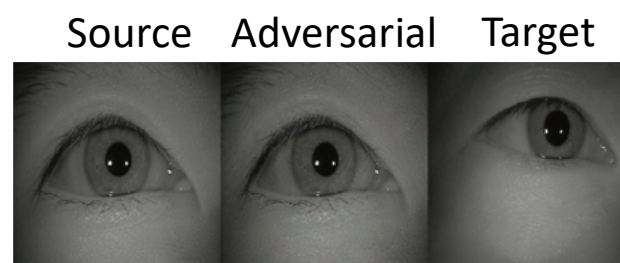
With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning, Wang, Yao, Viswanath, Zheng, Zhao, USENIX Security 2018

Attack is Very Effective

- Targeted attack: randomly select 1,000 source/target image pairs
- Success: % of images successfully misclassified to target



Face recognition
92.6% attack success rate



Iris recognition
95.9% attack success rate

- Tested on student models built on real services: 88+% success



Defense: Make Student Unpredictable

- Modify Student to make internal representation deviate from Teacher
 - Modification should be unpredictable by the attacker → No countermeasure
 - Without impacting classification accuracy
 - Build defense into training OR patch existing models

Model		Face Recognition	Iris Recognition
Before Patching	Attack Success Rate	92.6%	100%
After Patching	Attack Success Rate	30.87%	12.6%
	Change of Classification Accuracy	↓ 2.86%	↑ 2.73%

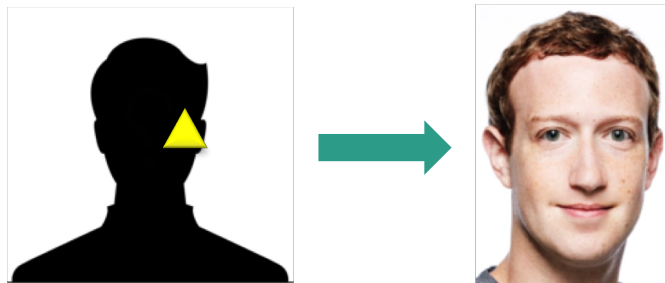
Of Sleeper Cells and Non-transparency



What if...

- You could insert *hidden backdoors* into models to do what exactly you wanted them to do?

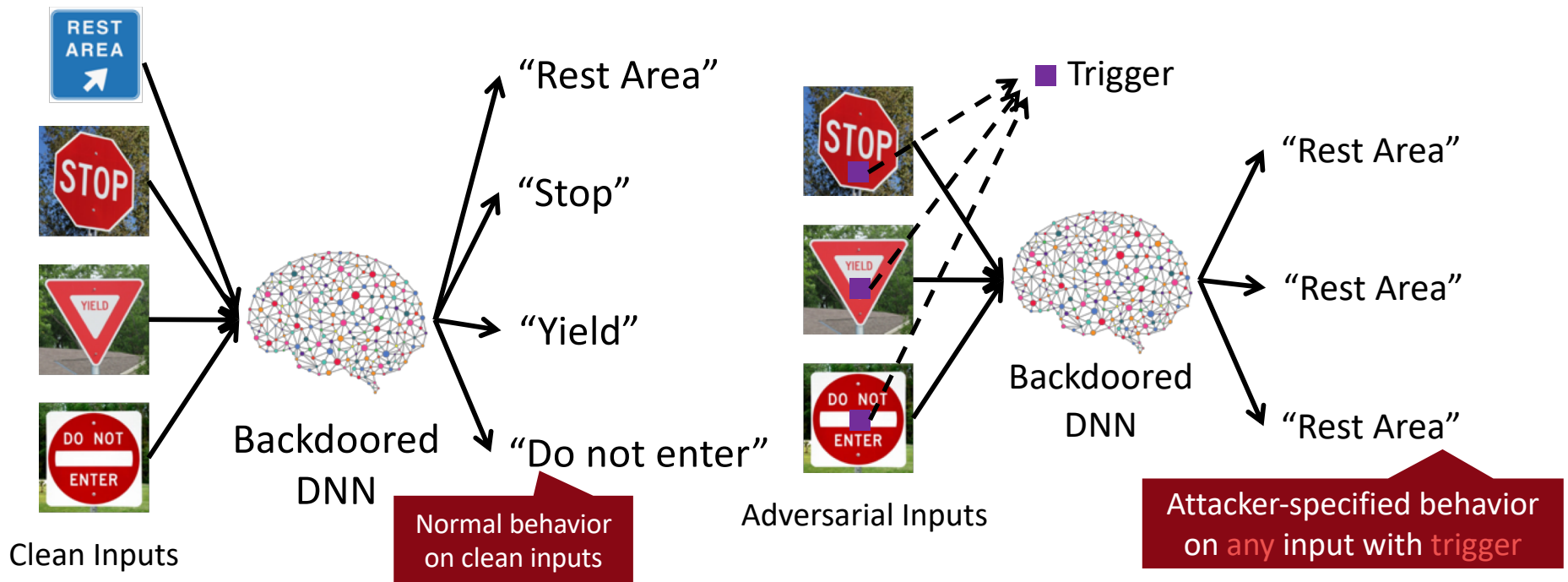
And...



- Model operates as expected in normal conditions
- “Dormant” while model gains popularity
- Awaiting activation by a “trigger”
- Injected at model training time or after

Definition of Backdoor

- Hidden behavior trained into a DNN



- Can be inserted at initial training or added later

Defense Goals and Assumptions

- Goals

Detection

- Whether given DNN is infected?
- What is the target label?
- What is the trigger used?

Mitigation

- Build a proactive filter to block adversarial inputs
- Patch DNN to remove backdoor

- Assumptions



Infected DNN



User

Has access to

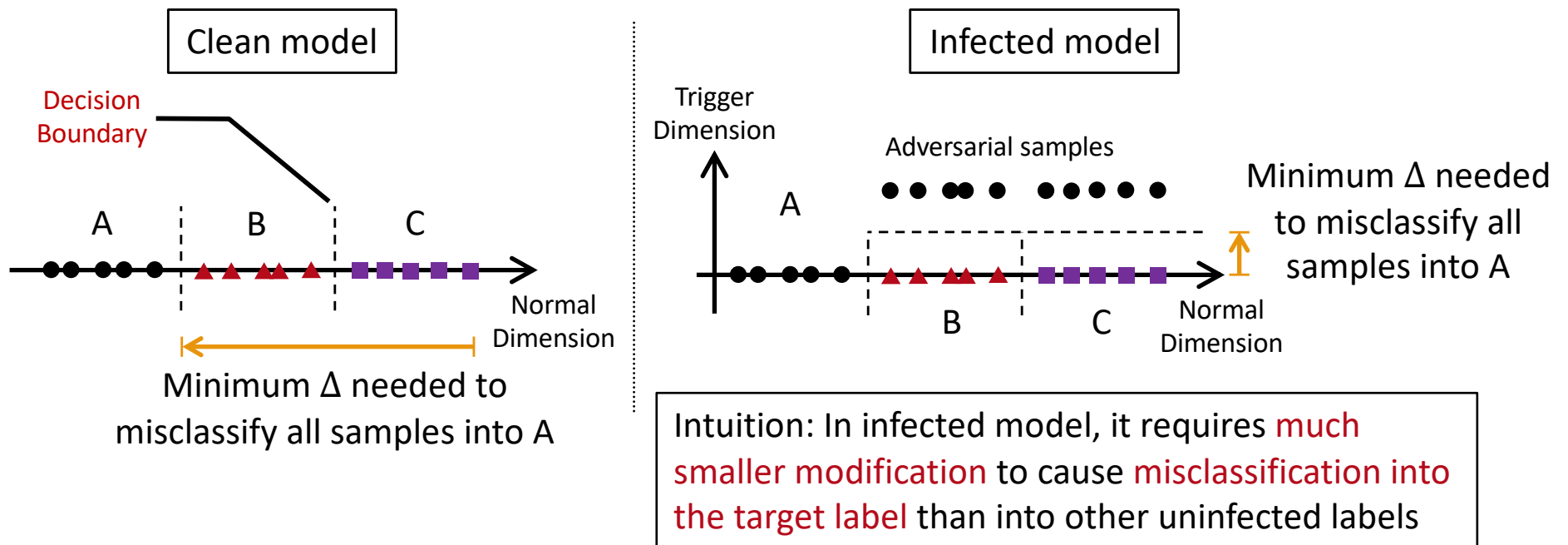
- A set of correctly labeled samples
- Computational resources

Does NOT have access to

- Poisoned samples used by the attacker

Key Intuition of Detecting Backdoor

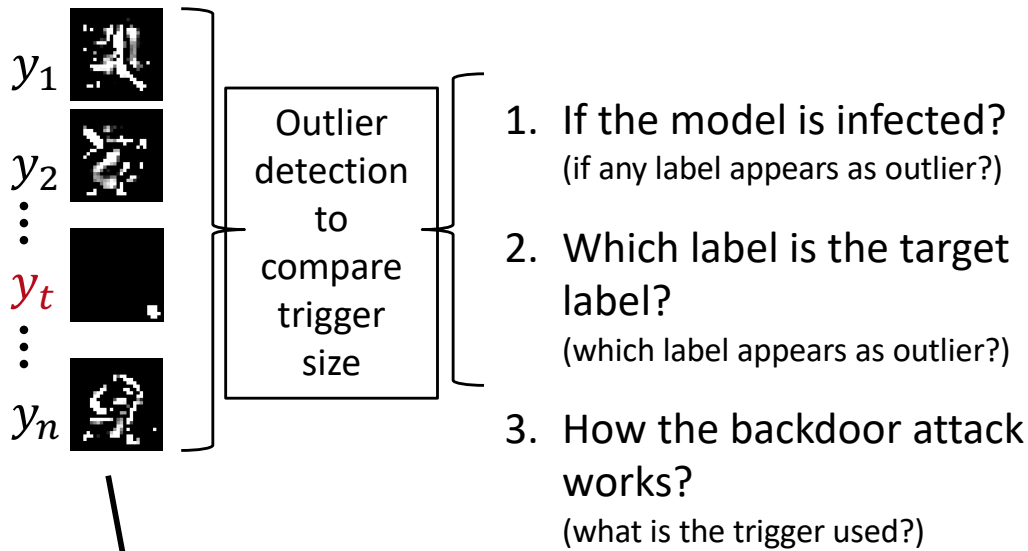
- Backdoor: misclassify any sample with trigger into the target label, regardless of original label



Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, Wang, Yao, Shan, Li, Viswanath, Zheng, Zhao, IEEE S&P 2019.

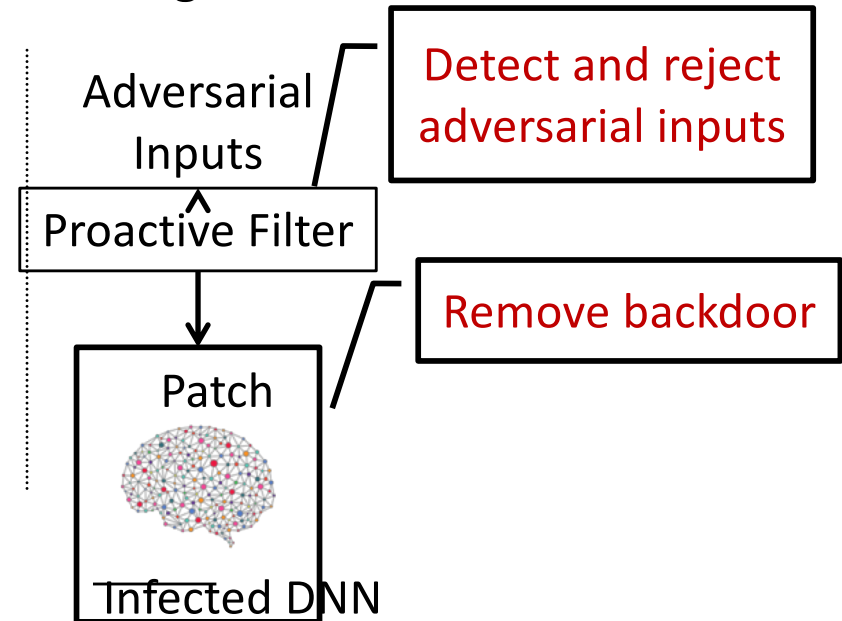
Design Overview

Detection



Reverse engineer a trigger for every output label in the model

Mitigation



Experiment Setup

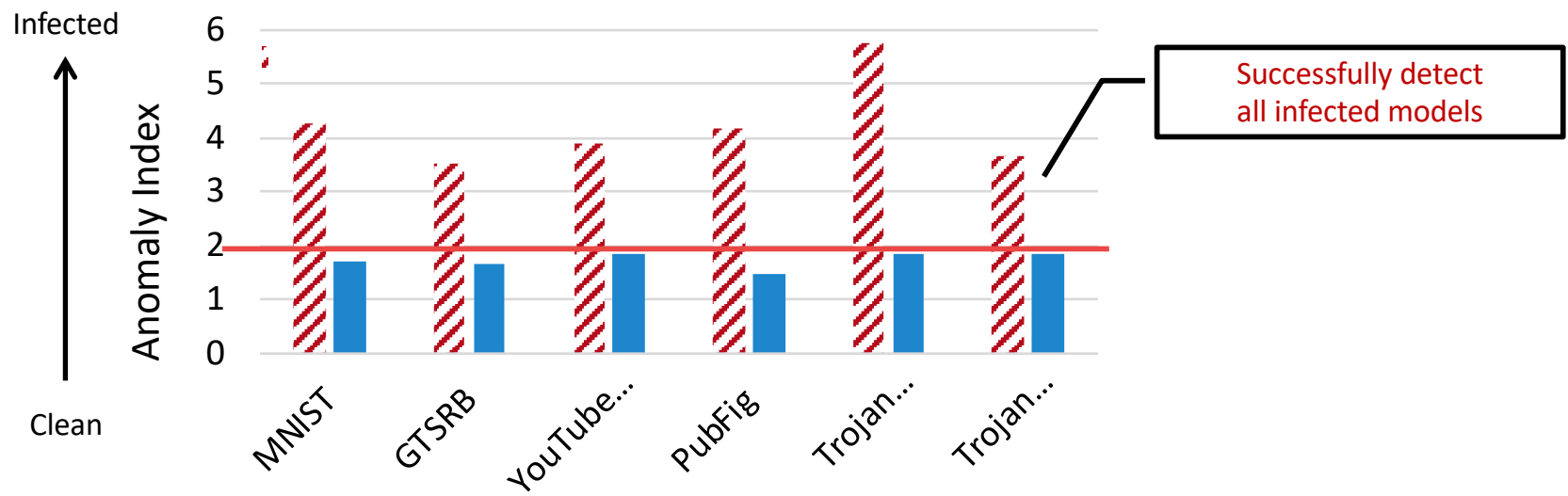
- Train 4 BadNets models
- Use 2 Trojan models shared by prior work
- Clean models for each task

	Model Name	Input Size	# of Labels	# of Layers
BadNets	MNIST	28×28×1	10	4
	GTSRB	32×32×3	43	8
	YouTube Face	55×47×3	1,283	8
	PubFig	224×224×3	65	16
Trojan	Trojan Square	224×224×3	2,622	16
	Trojan Watermark	224×224×3	2,622	16

Diagram illustrating the experiment setup. A black box represents the model architecture, with two callouts: "Large # of labels" and "Complex model architecture and realistic task". Below the box are two images: a face with a watermark and a face with a watermark.

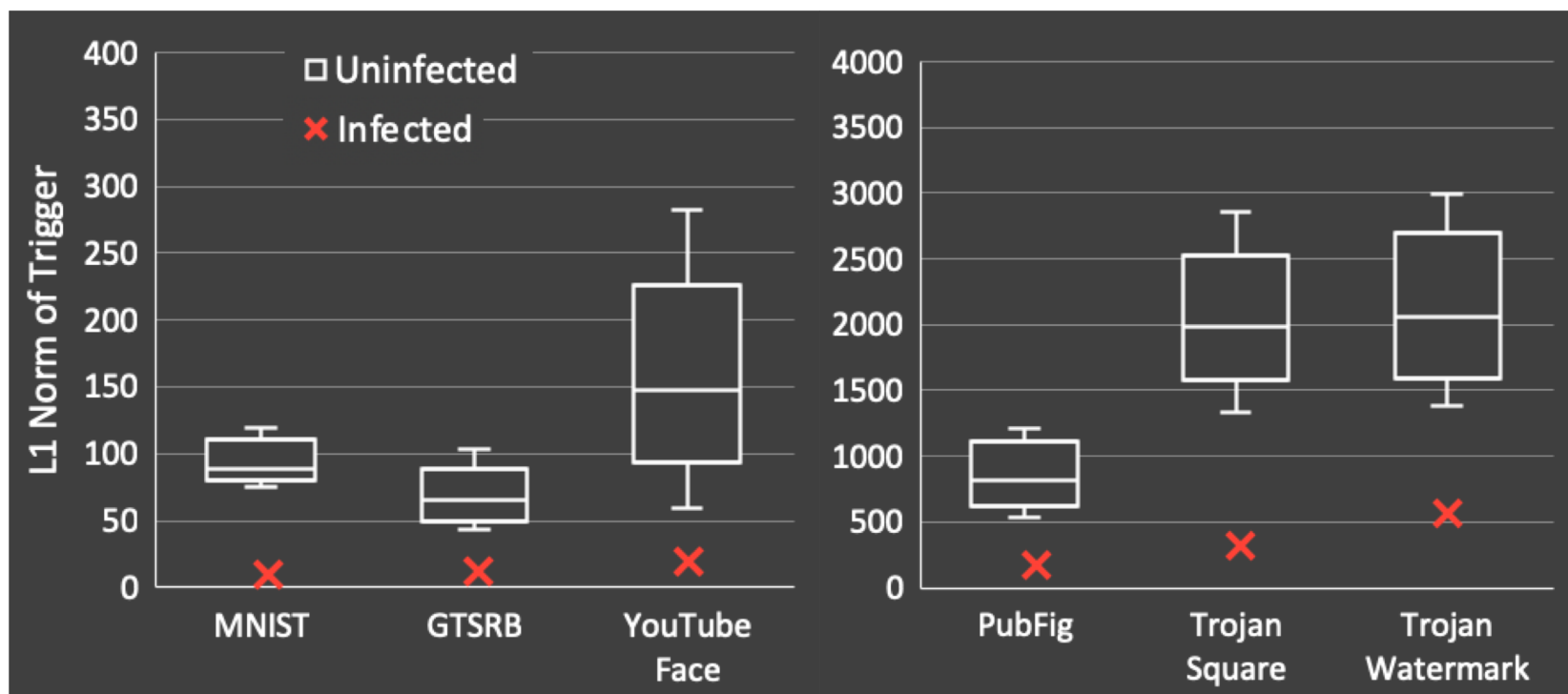
Backdoor Detection (1/3)

- Q1: Is the DNN infected?



Backdoor Detection (2/3)

- Q2: Which label is the target label?



Backdoor Detection (3/3)

- Q3: What is the trigger used by the backdoor?

