With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning

Bolun Wang*, YuanshunYao, Bimal Viswanath§ Haitao Zheng, Ben Y. Zhao University of Chicago, * UC Santa Barbara, §Virginia Tech

USENIX Security 2018

Web Security & Privacy Lab

KAIST

What This Paper is About? – Summary

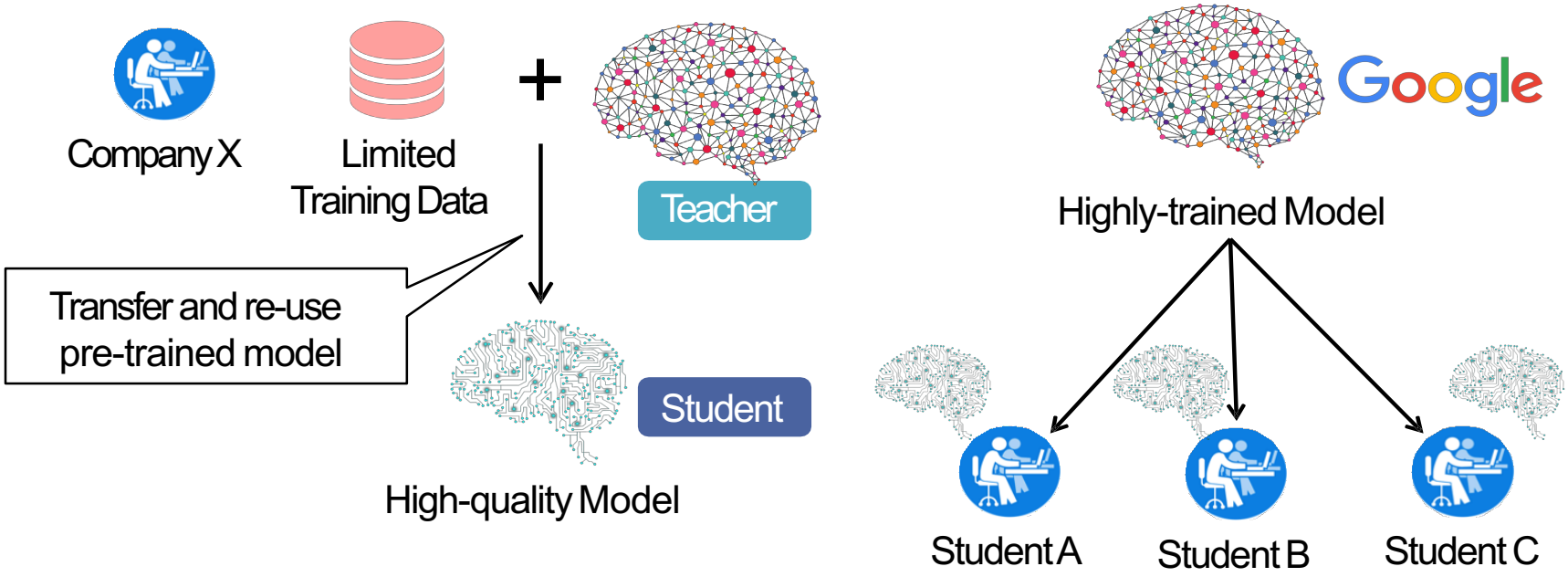
- Main question: “Is transfer learning safe?”
- Contribution
  - Adversarial attack in the context of Transfer Learning that shows 92% attack success rate
  - Fingerprinting method to identify the teacher model
  - Proposing solutions
    - Randomizing Input via Adding dropout layer
    - Injecting Neuron Distances

Background – Transfer Learning

- Motivation
  - High quality model need large labeled dataset
  - High quality model need lot of computational resources
    - Ex : ImageNet need 2 weeks to be train using 8 GPU
  - Small company may can't get a sufficient amount of data or resources.

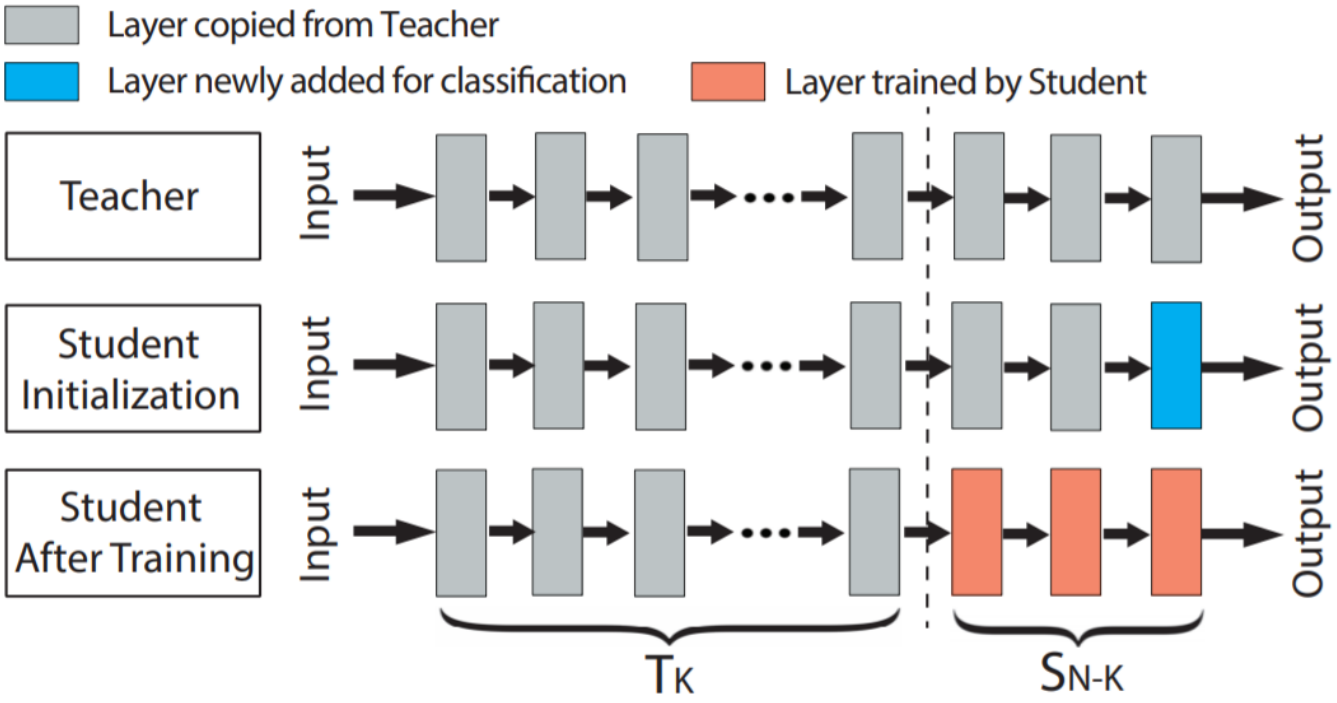
Background – Transfer Learning

- Key idea: "Reuse the pre-trained model!" + (fine-tuning on target dataset)



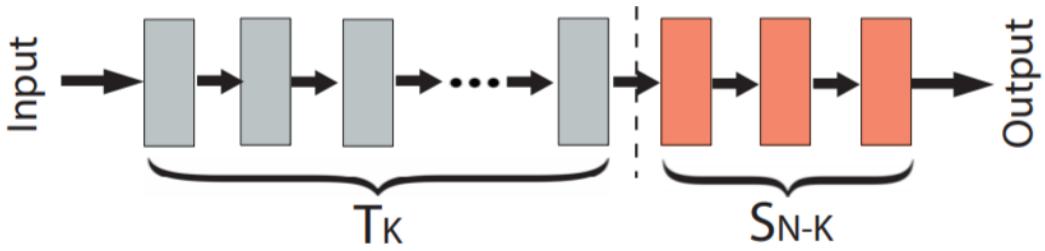
Background – Transfer Learning

- How does it work ?



Background – Transfer Learning

- Three types
  - Deep-layer Feature Extractor (K = N-1)
    - Student task is very similar to the teacher task
  - Mid-layer Feature Extractor (0 < K < N-1)
    - Student task is more dissimilar to the teacher task
    - More training data is available
  - Full-Model Fine-tuning (K = 0)
    - Student task differs significantly from the teacher task



Transfer Learning: Example

- Face recognition: recognize faces of 65 people

Company X

Student

10 images/person
65 people

Transfer 15 out of 16 layers

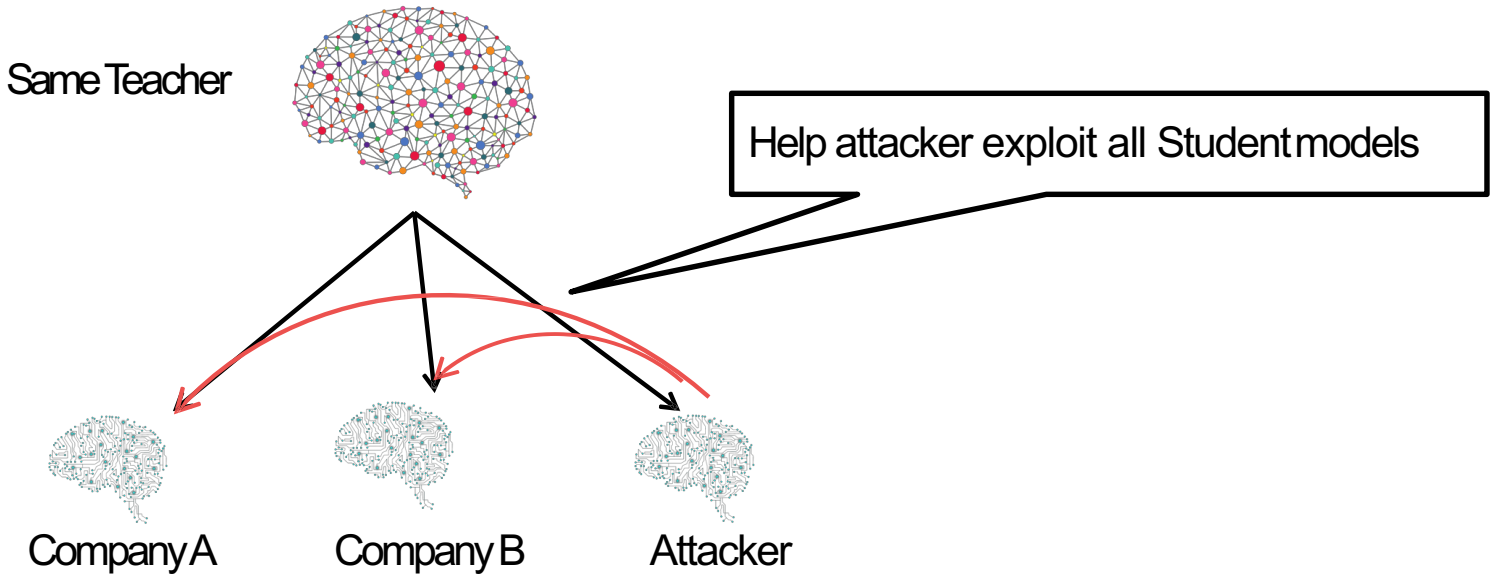
Teacher
(VGG-Face)

900 images/person
2,622 people

Classification Accuracy	
Without Transfer Learning	With Transfer Learning
1%	93.47%

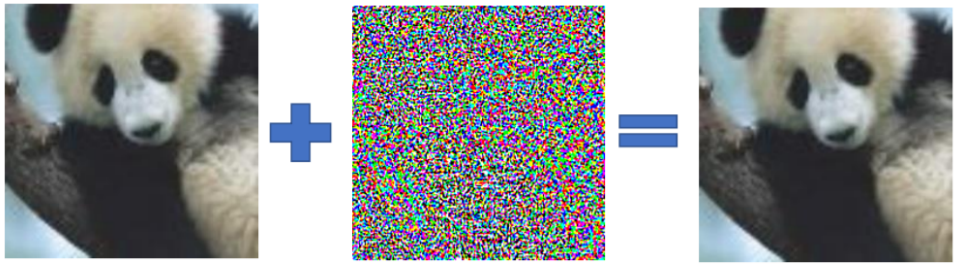
Is Transfer Learning Safe?

- Transfer Learning lacks diversity
  - Users have very limited choices of Teacher models
  - Teacher models are often published with their trained parameters, that mean that attackers have a whitebox access to the Teacher model.



Is Transfer Learning Safe?

- Adversarial attack
  - Misclassify inputs by adding carefully engineered perturbation



Classified as panda | Small adversarial noise | Classified as gibbon

Attack Model

Teacher

White-box

- Teachers are made public by popular DL services
- Model internals are known to the attacker

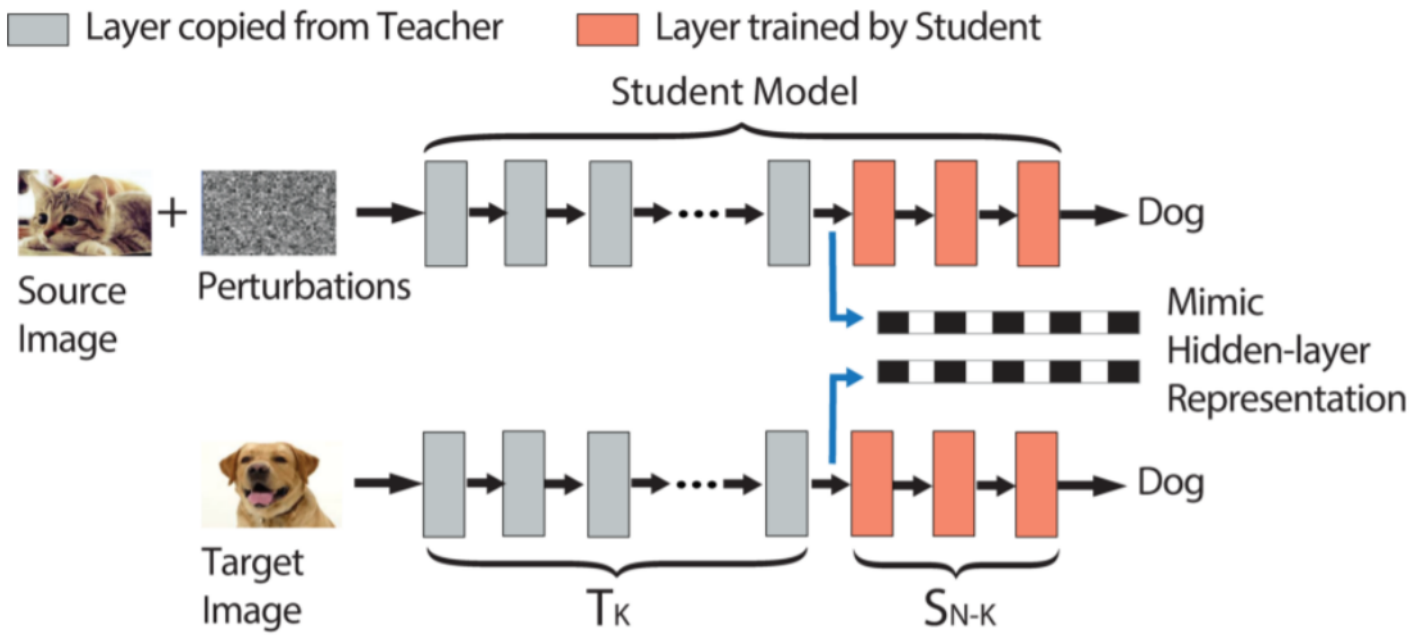
Student

Black-box

- Model internals are hidden and kept secure

Attack Methodology: Neuron Mimicry

- Attacker knows how much layers are frozen



Computing Perturbation

- Compute perturbation (Δ) by solving an optimization problem
  - Goal: mimic hidden-layer representation
  - Constraint: perturbation should be indistinguishable by humans

Targeted attack

$$\begin{aligned} \min \quad & D(T_K(x_s'), T_K(x_t)) \\ \text{s.t.} \quad & d(x_s', x_s) < P \end{aligned}$$

$x_s : source\ image$

$x_s' = x_s + \Delta : modified\ image$

$x_t : target\ image$

T_K : internal representation at layer K

D : distance between two internal representations. (L2)
d : is a distance function measuring the amount of perturbation added to x_s
P : constant to limit perturbations.

Computing Perturbation

Untargeted attack :
Use several target image and choose the one that have the minimum dissimilarity

$$\begin{aligned} \min \quad & min_{i \in I}\{D(T_K(x'_s), T_K(x_{ti}))\} \\ \text{s.t.} \quad & d(x'_s, x_s) < P \end{aligned}$$

$x_s : source\ image$

$x'_s = x_s + \Delta : modified\ image$

$x_{ti} : i^{th}\ target\ image$

T_K : internal representation at layer K

D : distance between two internal representations. (L2)
d : is a distance function measuring the amount of perturbation added to x_s
P : constant to limit perturbations.

Computing Perturbation

- Measuring the amount of perturbation
  - Lp fail to capture what humans perceive as image distortion.
  - Use DDSIM, which is an objective image quality assessment metric that closely matches with the perceived quality of an image
- Humans are sensitive to structural changes in an image, which strongly correlates with their subjective evaluation of image quality.
- DDSIM captures :
  - patterns in pixel intensities, especially among neighboring pixels
  - luminance
  - Contrast
- DDSIM values fall in the range [0,1]

- Optimization Function :

$$\min \quad D(T_K(x'_s), T_K(x_t)) + \lambda \cdot (\max(d(x'_s, x_s) - P, 0))^2$$

Computing Perturbation

- Measuring the amount of perturbation
  - Lp fail to capture what humans perceive as image distortion.
  - Use DDSIM, which is an objective image quality assessment metric that closely matches with the perceived quality of an image
- Humans are sensitive to structural changes in an image, which strongly correlates with their subjective evaluation of image quality.
- DDSIM captures :
  - patterns in pixel intensities, especially among neighboring pixels
  - luminance
  - Contrast
- DDSIM values fall in the range [0,1]

What do you think about it ?

- Optimization Function :

$$\min \quad D(T_K(x_s'), T_K(x_t)) + \lambda \cdot (\max(d(x_s', x_s) - P, 0))^2$$

Perturbation Budget

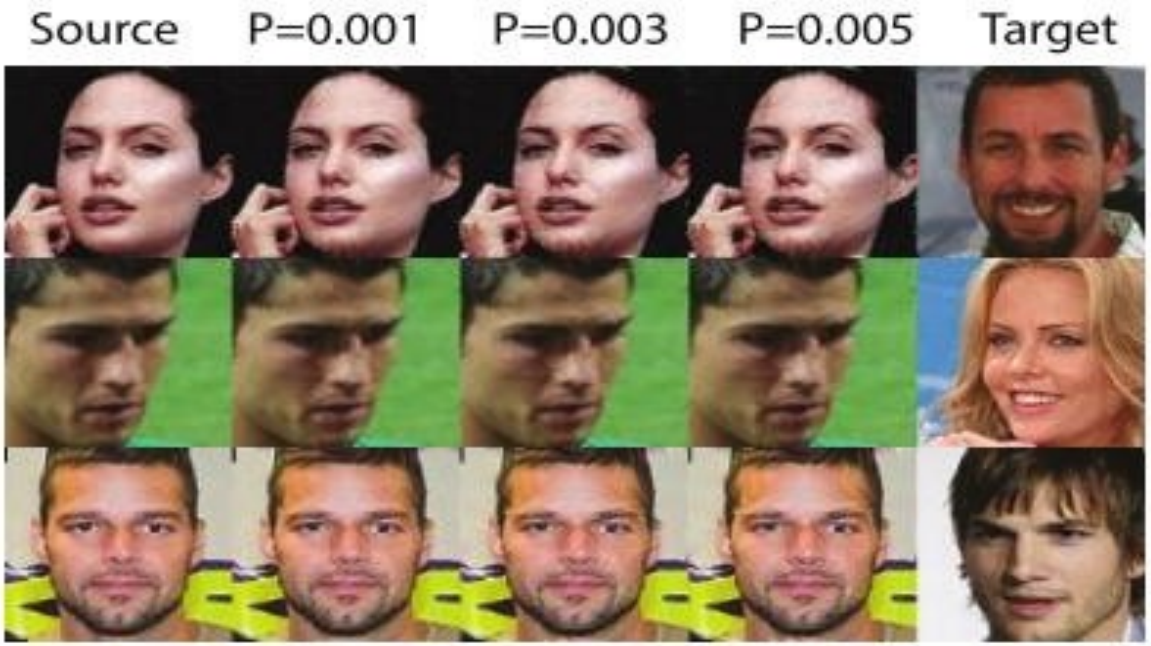
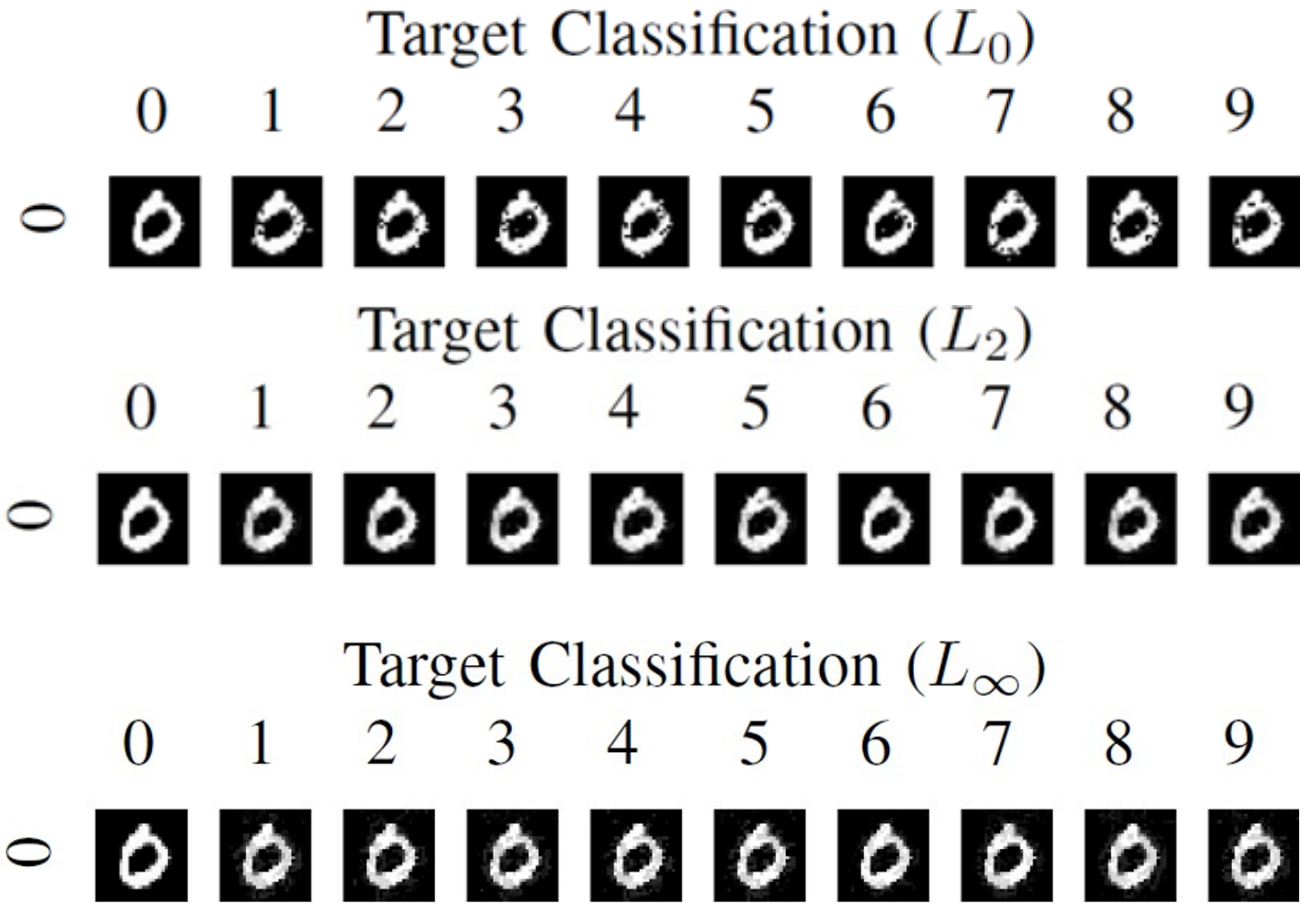


Figure 10: Adversarial examples generated from the same source image with different perturbation budgets (using *DSSIM*). Lower budget produces less noticeable perturbations.

Computing Perturbation



Dataset Review

	Teacher	Student
Face	16 layer VGG-Face model Trained on a dataset of 2.6M images (2,622 faces)	Trained on PubFig 90 face images belonging to each of the 65 people
Iris	16 layer VGG-Face model Trained on the ImageNet dataset (1.3M images)	Trained on the CASIA IRIS dataset 16,000 iris images associated with 1,000 individuals
Traffic sign	16 layer VGG-Face model Trained on the ImageNet dataset	Trained using the GTSRB dataset 39,209 images of 43 different traffic signs
Flower	ResNet50 model Trained on the ImageNet dataset	Trained on the VGG Flowers dataset 6,149 images from 102 classes

Attack Effectiveness

- Assumption
  - Attacker knows how many layers are frozen.
- Deep-layer Feature Extractor

Face recognition
92.6% attack success rate (P=0.003)

Iris recognition
95.9% attack success rate (P=0.005)

- Mid / Full -layer Feature Extractor

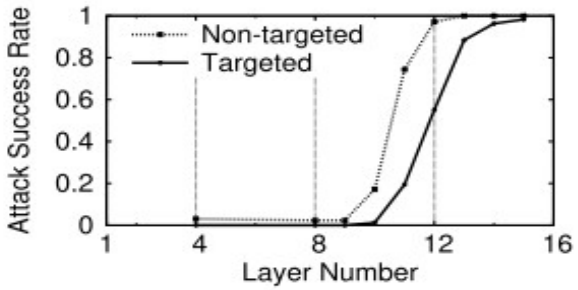
Sign recognition (mid-layer)
43,7% attack success rate (P=0.01)

flower recognition (full-layer)
1,1% attack success rate (P=0.01)

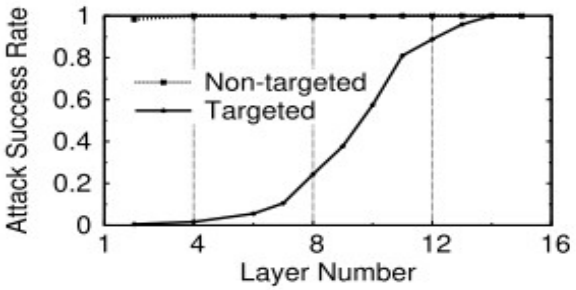
Targeted attack: Randomly select 1,000 source, target image pairs
Success rate: Percentage of images successfully misclassified into the target

Impact of The Attack Layer

- Deep-layer feature extraction

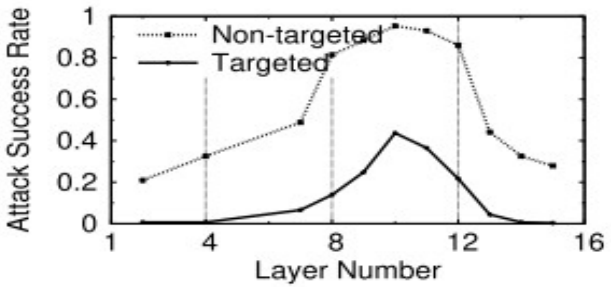


(a) Face

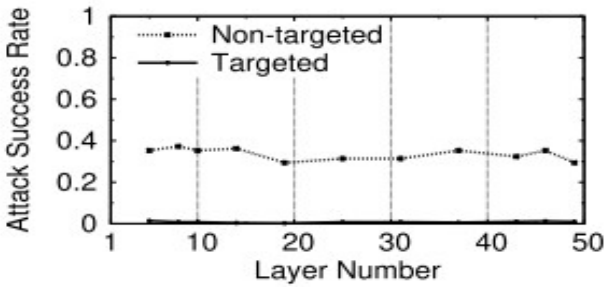


(b) Iris

- Mid / Full -layer Feature Extraction



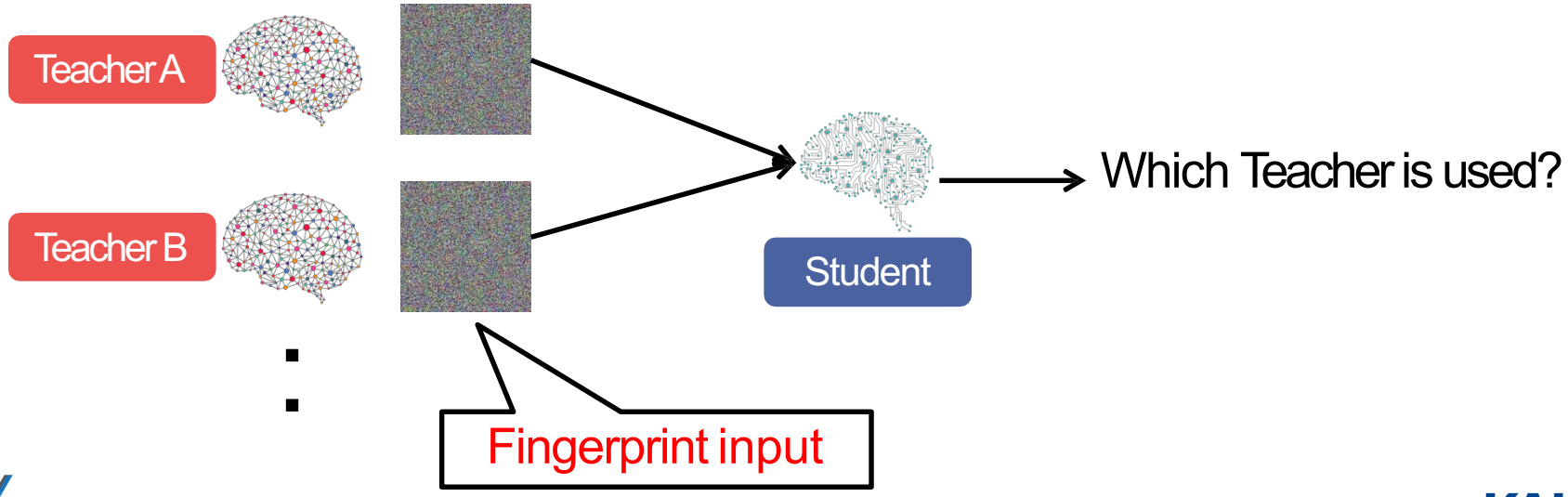
(c) Traffic Sign



(d) Flower

Challenges of Attack in the Wild

- Given a Student, can we identify the Teacher ?
- **Method**
  1. Craft "fingerprint" input for each Teacher candidate
  2. Query Student
  3. Identify Teacher among candidates base on the prediction results



Challenges of Attack in the Wild

- How to craft fingerprints
- Input (x) that nullify the teacher model to produce an all-zero vector in T_{N-1}

$$S(x) = W_N \times T_{N-1}(x) + B_N$$

Fingerprint input makes zero

W_N: the weight matrix of the dense layer
T_{N-1}: the function transforming the input x to neurons at layer N−1
B_N: Bias vector

Fingerprinting Method

- Key hypothesis: B_N shows lower dispersion compared to normal S(x) values

x_p is fingerprint value of teacher T

x_p is fingerprint value of teacher T

$$Dispersion(S_t(x_p)) \quad v.s. \quad Dispersion(S_{nt}(x_p))$$

$$Dispersion(B_N) \quad > \quad Dispersion(W_N \times T_{N-1}(x_p) + B_N)$$

The only difference is here: the internal representation to mimic is a zero-vector

Validation of Fingerprinting Method

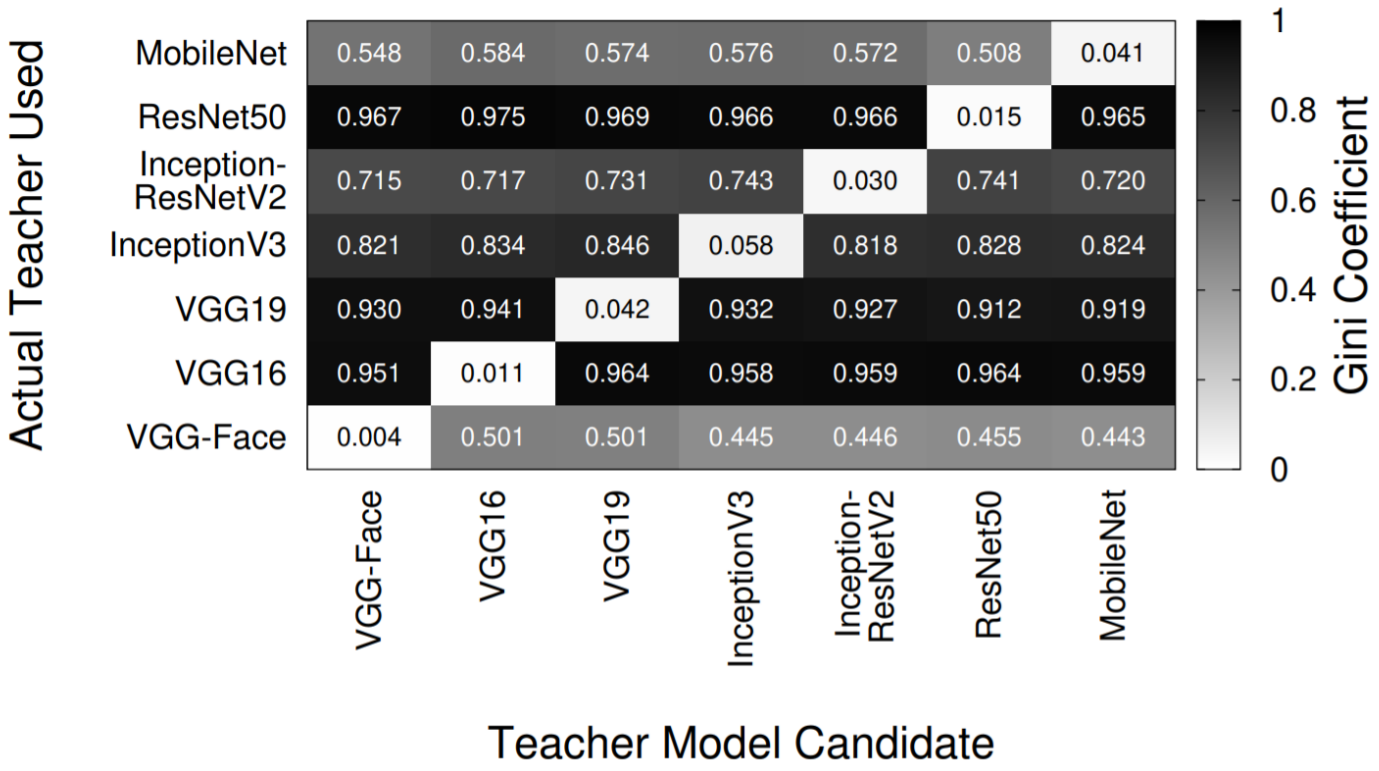
- 7 Student models using multiple popular public Teacher.
- Evaluate the dispersion of S(x), (measure B_N's dispersion)
  - Experimental method :
    - Set the output of the N-1th layer as a zero vector using a specific input. So only B_N is fed into the final prediction. Then feed students with random output.
    - Measure the Gini coefficient (0: complete equality, 1: complete inequality).
  - Result:
    - Fingerprinting input: < 0.011, Random input: 0.648~0.999

Validation of Fingerprinting Method

- Evaluate the effectiveness of the method
  - Experimental method :
    - Calculate 10 fingerprints for each Teacher
    - Feed these fingerprint to students and compute the average coefficient of S(x)

Validation of Fingerprinting Method

- Evaluate the effectiveness of the method
  - Result:



Challenges of Attack in the Wild

- Would this attack work on Students trained by real DL services?
  - Follow tutorials to build Student using following services
  - Attack achieves >88.0% success rate for all three services

Teacher	Google Cloud ML	Microsoft CNTK	PyTorch
Accuracy	89.3%	82.25%	???
Attack success rate	96.5% (P=0.001)	99.4% (P=0.003)	88.0% (P=0.001) (87.4% with Full-model fine tuning)
Recommended Model	Deep-layer Feature Extractor	Full Model Fine-tuning	Deep-layer Feature Extractor / Full Model Fine-tuning

Challenges of Attack in the Wild

- Q2: would attack work on Students trained by real DL services?
  - Follow tutorials to build Student using following services
  - Attack achieves >88.0% success rate for all three services

Q: Why the author redundantly run the attack evaluation on the real DL services?

What makes the student model from real DL service different from the previous model?

Defence

- Adding dropout layer
  - Intuition:
    - Effectiveness of attacks is heavily dependent on the level of perturbations
  - Method
    - Dropping a certain fraction of randomly selected input pixels
    - Introducing additional random perturbations to the image before classification

Defence

- Adding dropout layer - Result

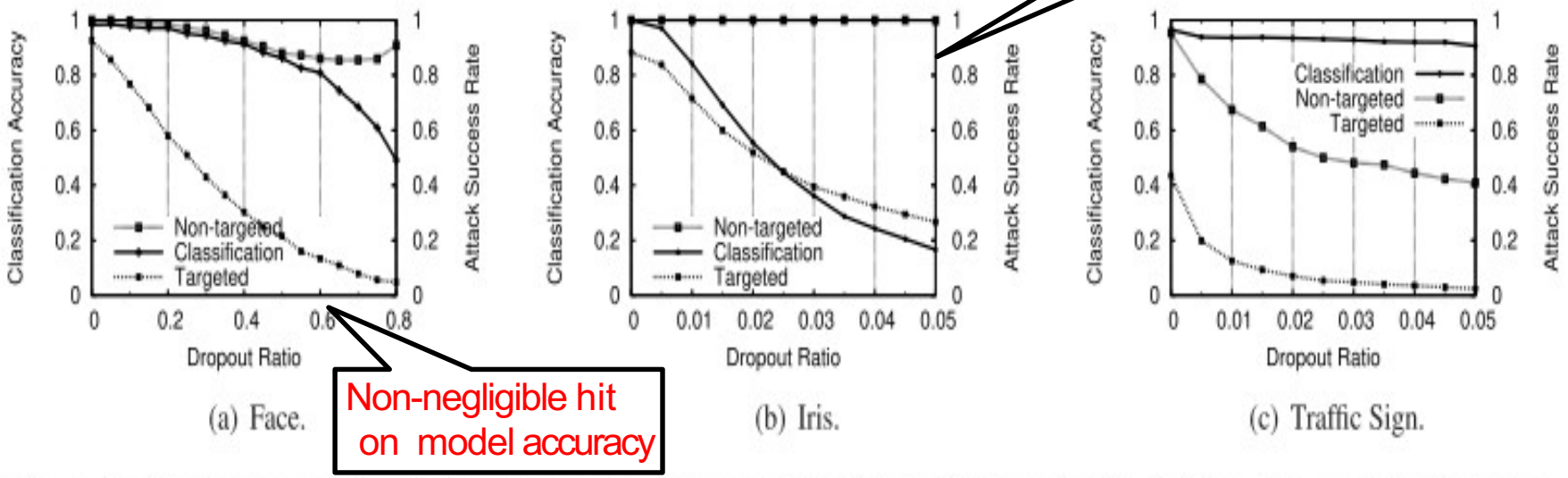
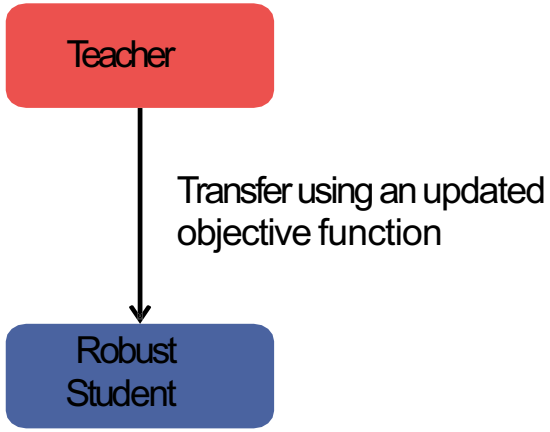


Figure 14: Performance of applying Dropout as defense with different Dropout ratio in Face, Iris, and Traffic Sign.

Defence

- Modify internal representation
  - Goal: Make Student Unpredictable
  - Modification should be unpredictable by the attacker without impacting classification accuracy



$$\begin{aligned} \min \quad & CrossEntropy(Y_{true}, Y_{pred}) \\ \text{s.t.} \quad & \sum_{x \in X_{train}} \| |W_s| \circ (T_K(x) - S_K(x)) \|_2 > D_{th} \end{aligned} \tag{4}$$

Effectiveness of Defense

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

Conclusion

- Transfer learning is effective, but is not safe.
  - Student models leverage knowledge of white-box Teacher models
  - Attacker can also leverage knowledge of white-box Teacher models
- Even if the information of teacher is hided from the attacker, he can identify the teacher models given a student model
- To defend the model against adversarial attack developer should make the student diverge from the Teacher.

Questions