black hat ASIA 2020

OCTOBER 1-2, 2020
BRIEFINGS

Attacking and Defending Machine Learning Applications of Public Cloud

Hao Xin, Senior research fellow of Baidu Security

Our Team X-Lab

AI Security Research

百度安全
有 AI 更 安 全

Baidu Security
Advanced AI , Stronger Security

Open Source Projects:

AdvBox

MesaLock MesaLink MesaPy

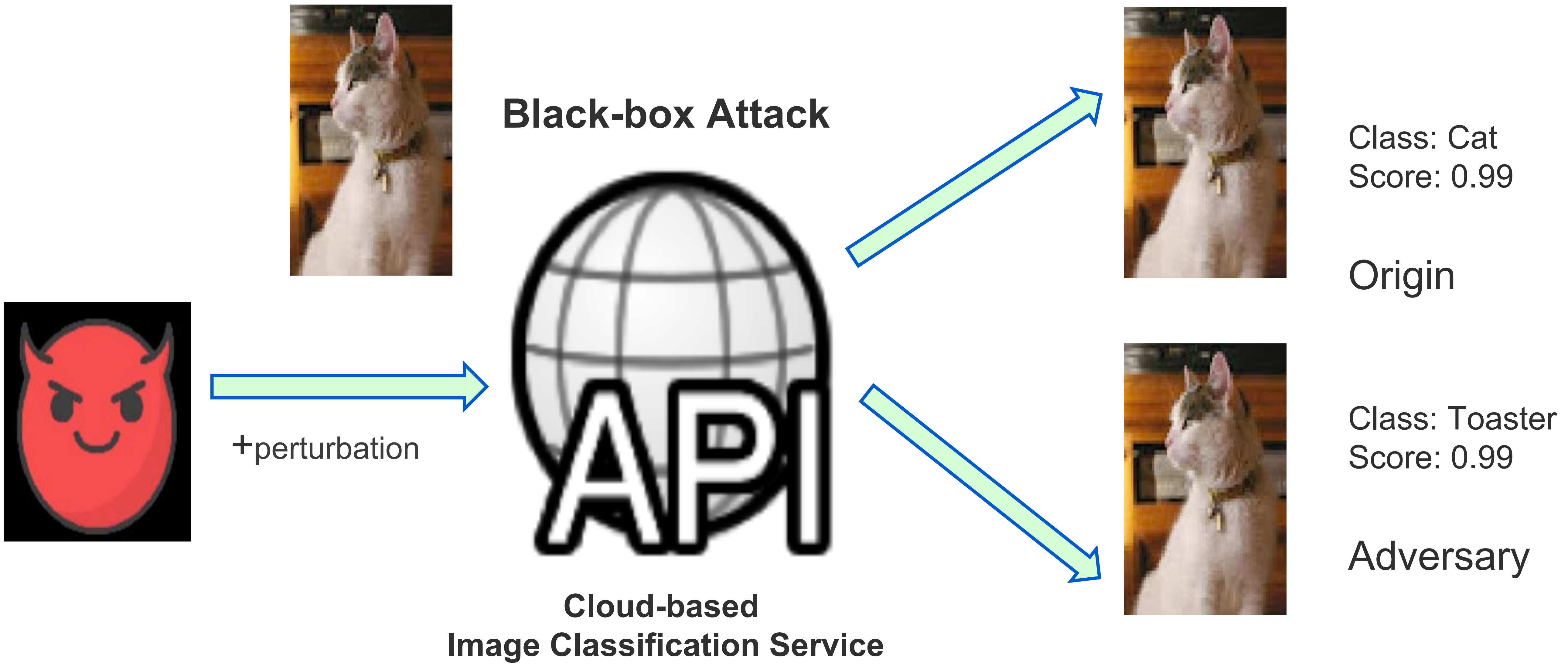
OpenRASP

https://github.com/baidu/AdvBox

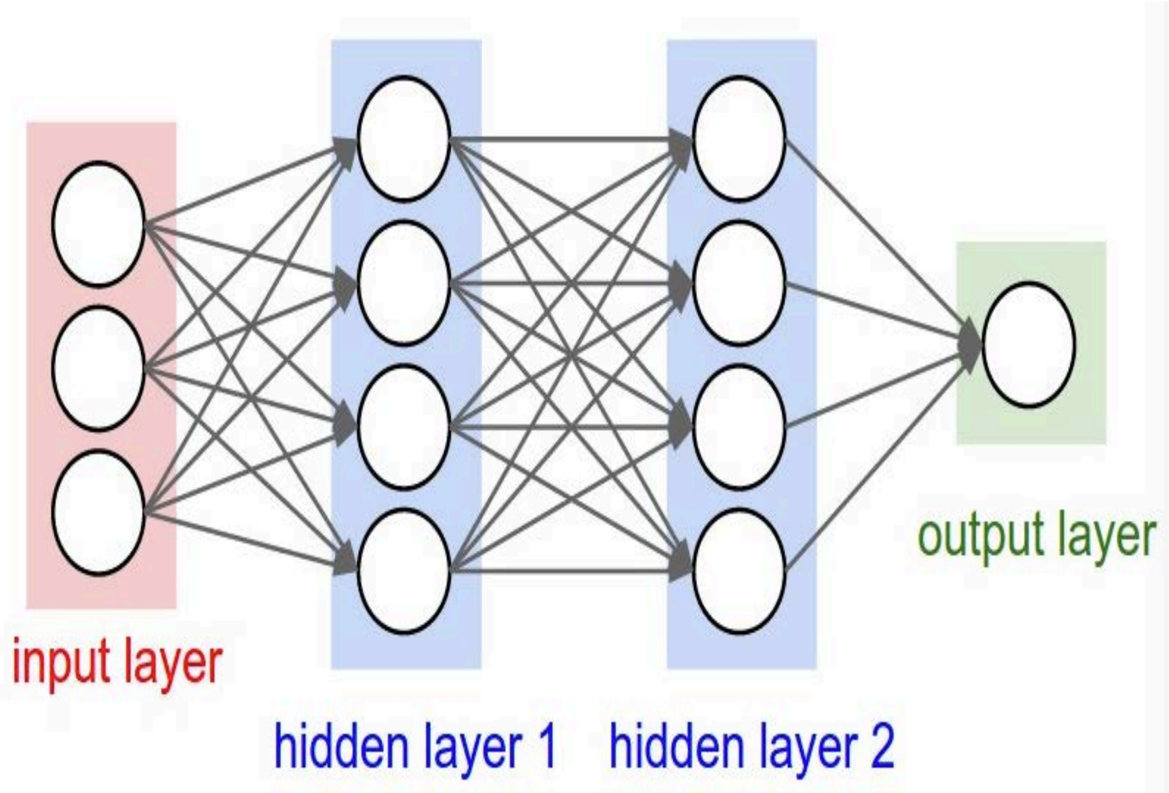
https://github.com/baidu/openrasp

Transfer attack against AI cloud service

Today's Topics



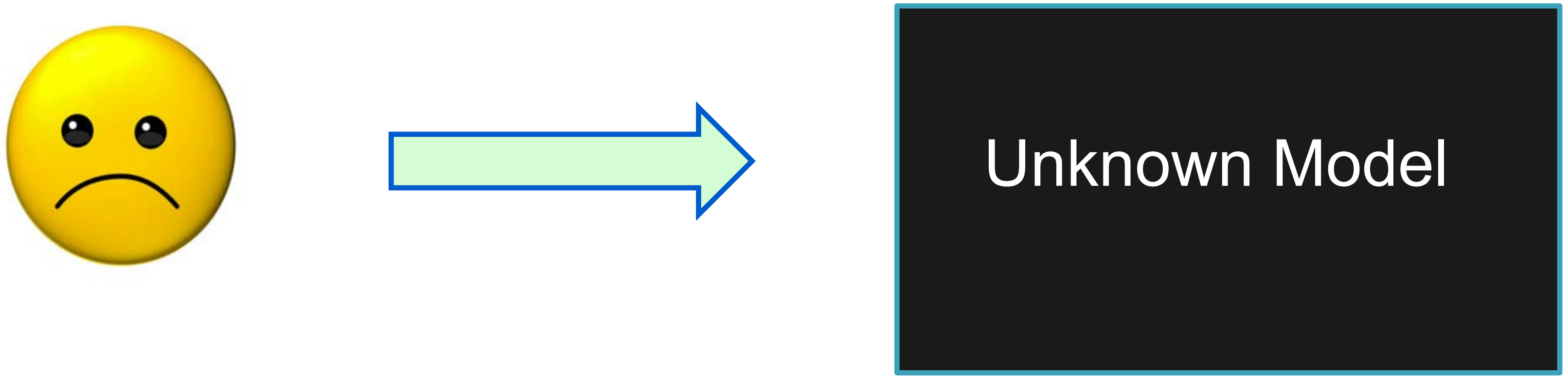
White-box Attack is Easy



The attacker knows the network structure and parameters, and has unlimited access to model input

White-box Attack is Easy

Unknown parameters
Unknown network structure



The attacker can unlimited access to model input

Attack Cloud-based Image Classifier Service is More Difficult

resize,crop,blur,…

Unknown parameters
Unknown network structure

Unknown Preprocessing

Unknown Model

The attacker can only access to model input with
unknown preprocessing and limited queries

Method 1:Query-based Attacks

- The thousands of queries are required for low-resolution images.
- For high-resolution images, it still takes tens of thousands of times.
- For example, they achieves a 95.5% success rate with a mean of 104342 queries to the black-box classifier.

slow

costly

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Query-efficient black-box adversarial examples (superceded). arXiv preprint arXiv:1712.07113, 2017.

Method 2:Transfer Learning Attacks

- Adversarial samples have transferability in DNN with similar structure
- White Box Attacks on Open Source Models with Same Function

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	22.83	0%	13%	18%	19%	11%
ResNet-101	23.81	19%	0%	21%	21%	12%
ResNet-50	22.86	23%	20%	0%	21%	18%
VGG-16	22.51	22%	17%	17%	0%	5%
GoogLeNet	22.58	39%	38%	34%	19%	0%

Panel A: Optimization-based approach

The cell (i, j) indicates the accuracy of the adversarial images generated for model i (row) evaluated over model j (column).
Yanpei Liu, Xinyun Chen, Liu Chang, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. 2016.

The challenge is to find open source models with the same functionality

Keeping model in cloud provides a FALSE sense of security

Demo: Fool Google Image Search

Demo: Fool Google Image Search

Attacks Overview

- We propose Fast Feature map Loss PGD(FFL-PGD) untargeted attack based on Substitution model with AutoDL, which achieve a high evasion rate with a very limited number of queries.

- Instead of millions of queries in previous studies, our method find the adversarial examples using average only one or two of queries.

- No need to find open source models with the same functionality.

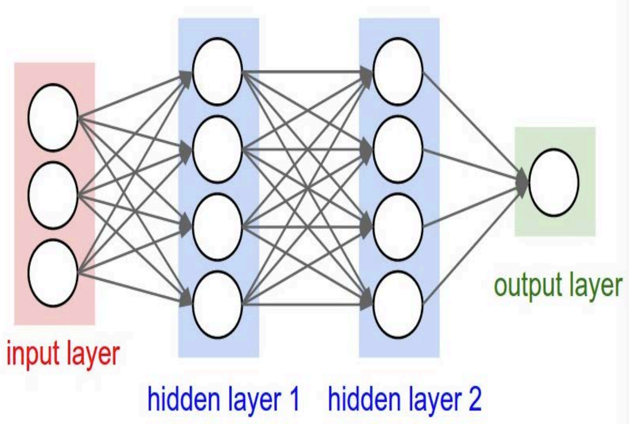
Steps of Our Attack

Step1:Substitute Model Training

Step2:Adversarial Sample Crafting

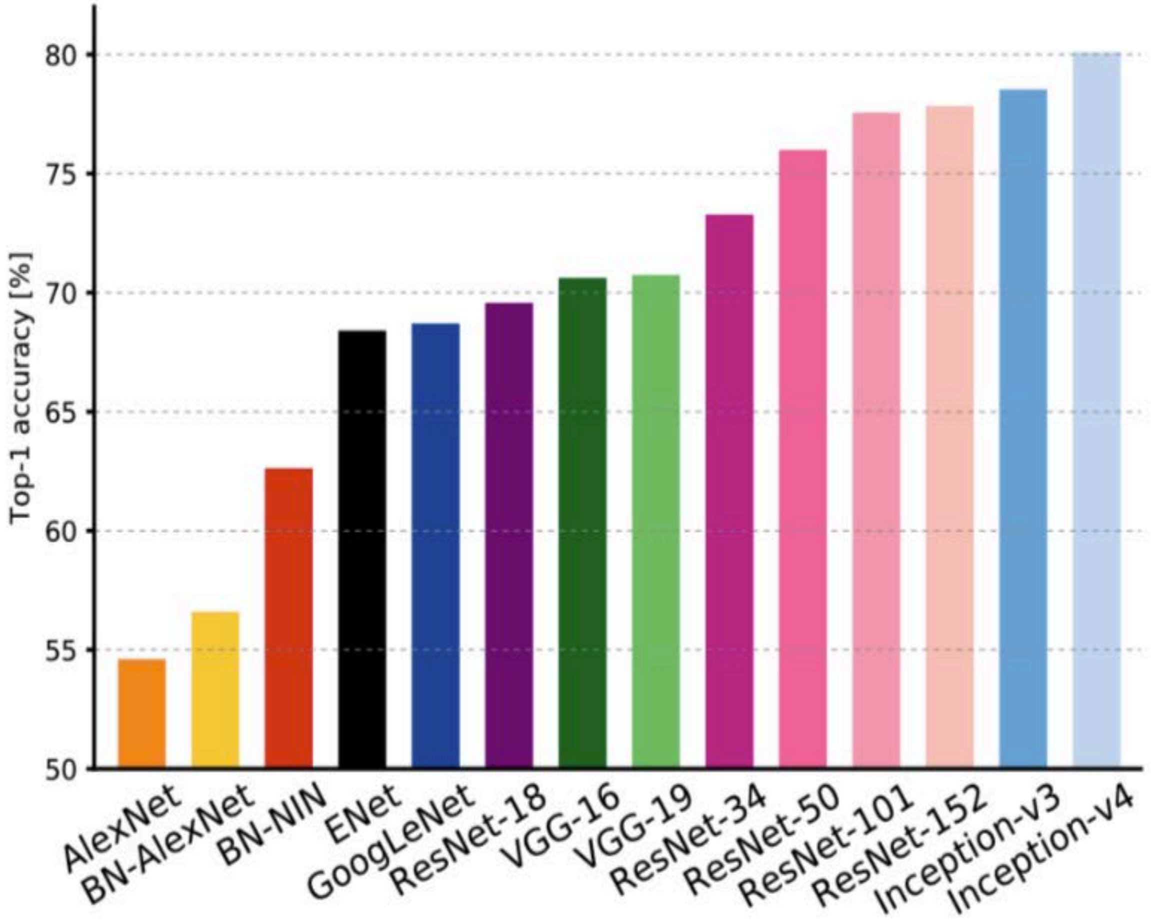
Step 2

Step 1



Substitution Model Training

- We select DNNs which pretrained on ImageNet as our substitute model .
- Better top-1 accuracy means stronger feature extraction capability.



Top1 vs. network. Top-1 validation accuracies for top scoring single-model architectures (Img from https://arxiv.org/abs/1605.07678v1)

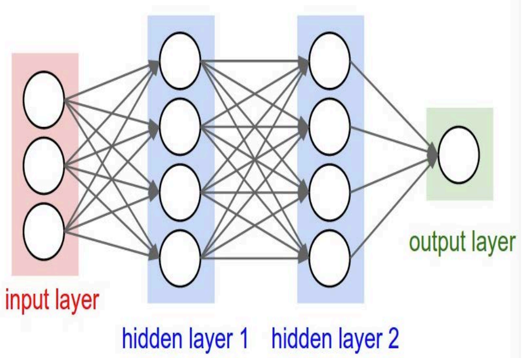
Substitution Model Training

- We simplify untargeted attack into binary classification problem :Cat or not?
- We fix the parameters of the feature layer and train only the full connection layer of the last layer.

fixed → feature layer → trainable → The last FC layer → Cat:0.99 Other:0.01

Substitution Model Training

Key point:
we use images which we will attack as our training set.



Train DNNs on S'

Initial training set S

Label the substitute training set

Labeled training set S'

Substitution Model Training

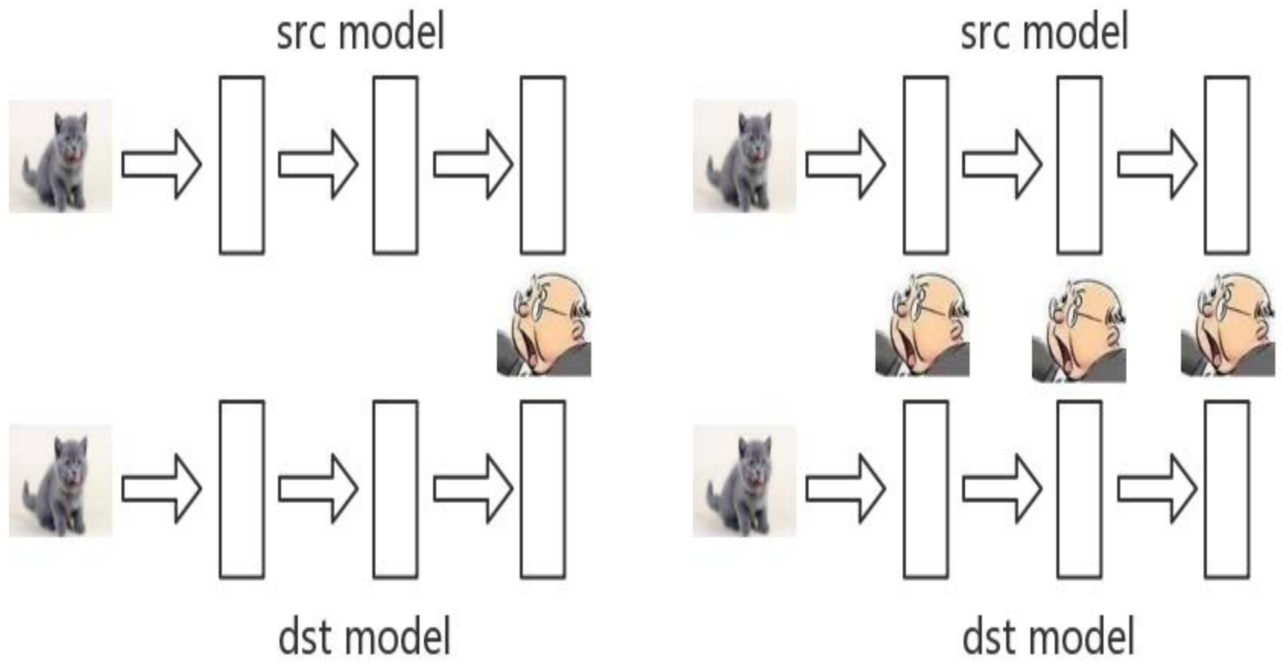
Smaller and fewer parameters mean more robust and better generalization ability

$$\min_{w} \sum_{i=1}^{n} L(z(x_i, \omega), y_i) + \lambda \cdot \Omega(\omega)$$

ω be the d-dimensional parameter vector containing all parameters of the target model. The optimization object with regularization.

Substitution Model Training

AutoDL Transfer regularize the behavior of the networks and considers the distance between the outer layer outputs of the two networks



you can experience it on the easy AutoDL of the Baidu AI website.

Adversarial Sample Crafting

We propose Fast Feature maps Loss PGD attack which has a loss function to improve the success rate of transfer attack.

The loss function L is defined as:

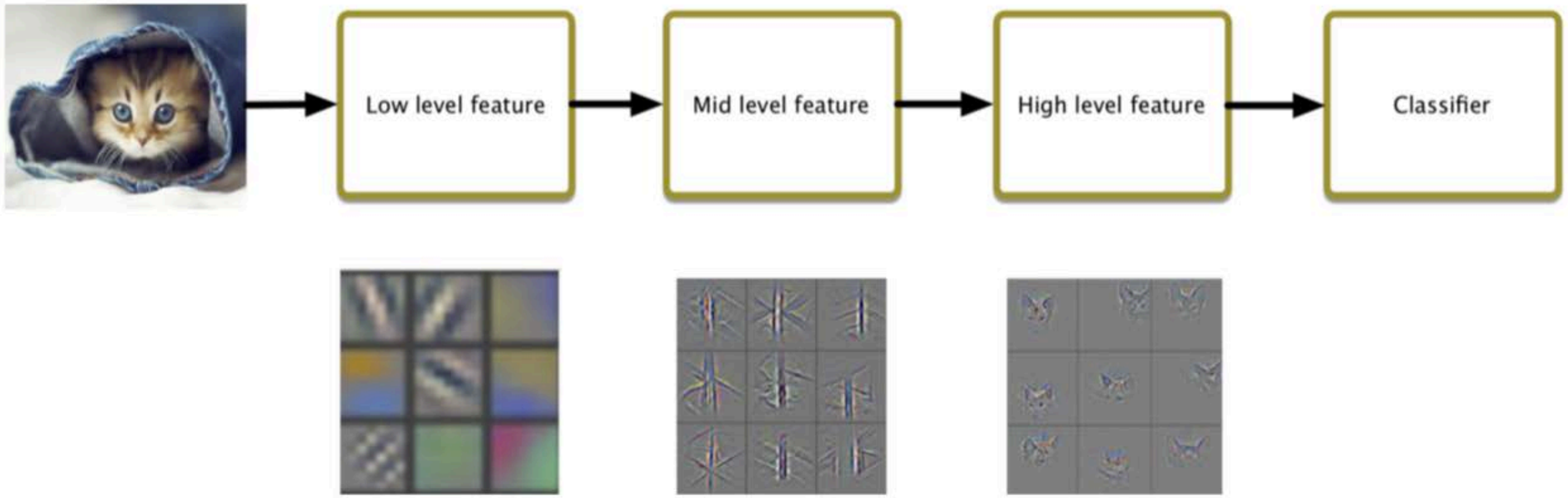
$$L = class\_loss + \beta * FeatureMaps\_loss$$

Adversarial Sample Crafting

- Class Loss makes the result of classification wrong
- FeatureMap Loss which is the output of the convolution layer of the substitute model, represents the high level of semantic features of the convolution layer and improves transferability of adversarial sample
- Different layers have different attention weights

Adversarial Sample Crafting

Illustration of cat recognition, the first convolution layer mainly recognizes low level features such as edges and lines. In the last convolution layer, it recognizes high level features such as eyes and nose.



Adversarial Sample Crafting

We assume the original input is O, the adversarial example is ADV , and the feature map loss can be simplified as:

$$FeatureMap\_loss(ADV, O) = \|L_n(ADV) - L_n(O)\|_2$$

Different layers have different attention weights

Datasets and Preprocessing

- 100 cat images and 100 other animal images are selected from the ImageNet val set.
- Images are clipped to the size of 224×224×3
- Image format is RGB

Datasets and Preprocessing

- We use these 100 images of cats as original images to generate adversarial examples and make a black-box untargeted attack against real-world cloud-based image classification services.

- We count the number of top-1 misclassification to calculate the escape rate.

Attack Evaluation

- We choose ResNet-152 as our substitute model
- We launche PGD and FFL-PGD attacks against our substitute model to generate adversarial examples.
- We compare FFL-PGD with PGD and ensemble-model attack , which are considered to have good transferability .

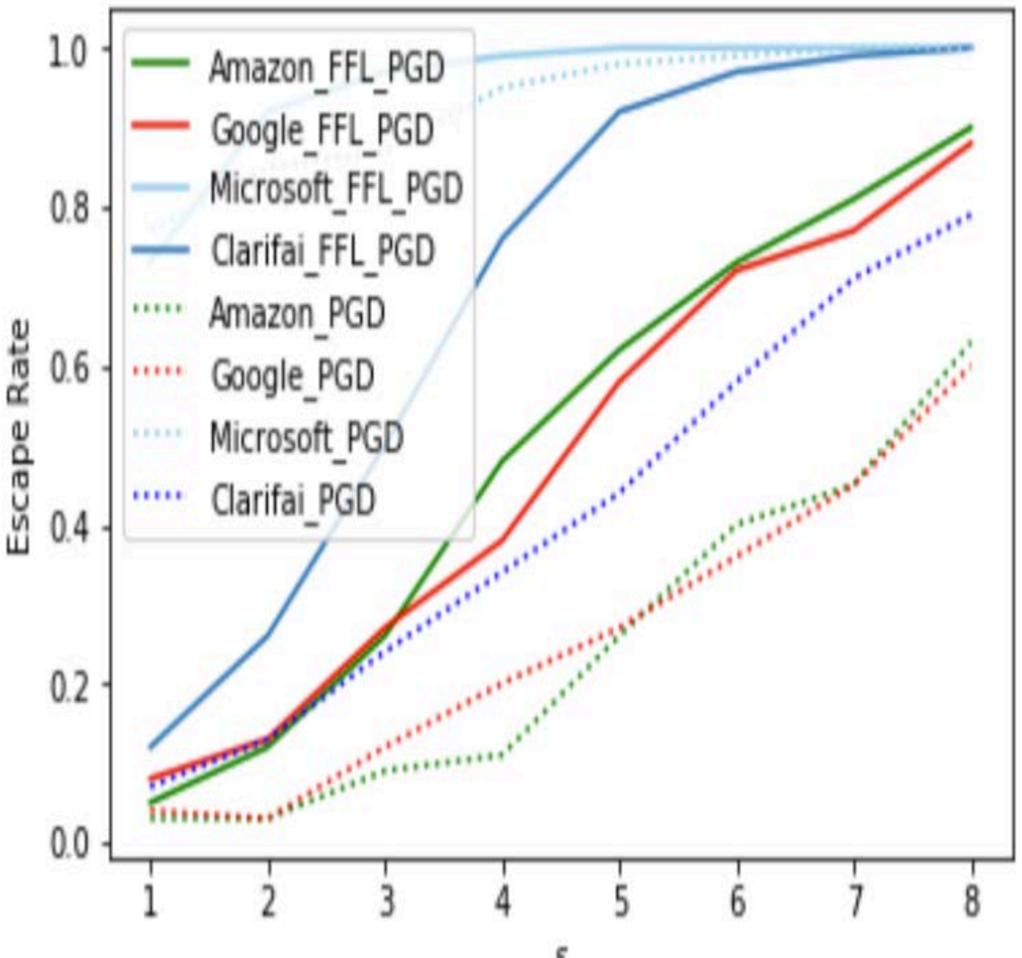
Attack Evaluation

We assume the original input is O, the adversarial example is ADV
We use Peak Signal to Noise Ratio (PSNR) to measure the quality of images.

$$PSNR = 10log_{10}(MAX^2/MSE)$$

We use structural similarity (SSIM) index to measure image similarity.

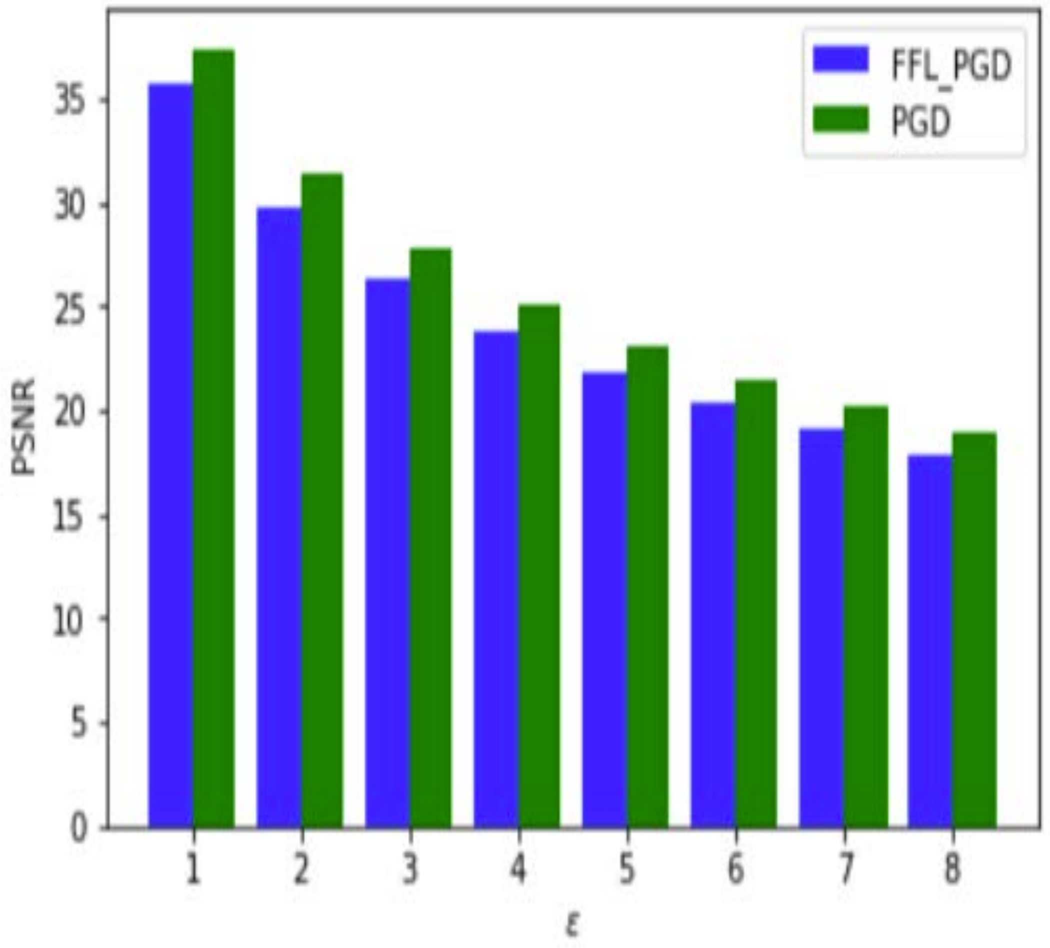
Attack Evaluation: Escape Rates



We increase step size ε from 1 to 8, the figure records the escape rates of PGD and FFL-PGD attacks

- FFL-PGD attack has a success rate over 90% among different cloud-based image classification services.
- Our FFL-PGD has a better transferability than PGD

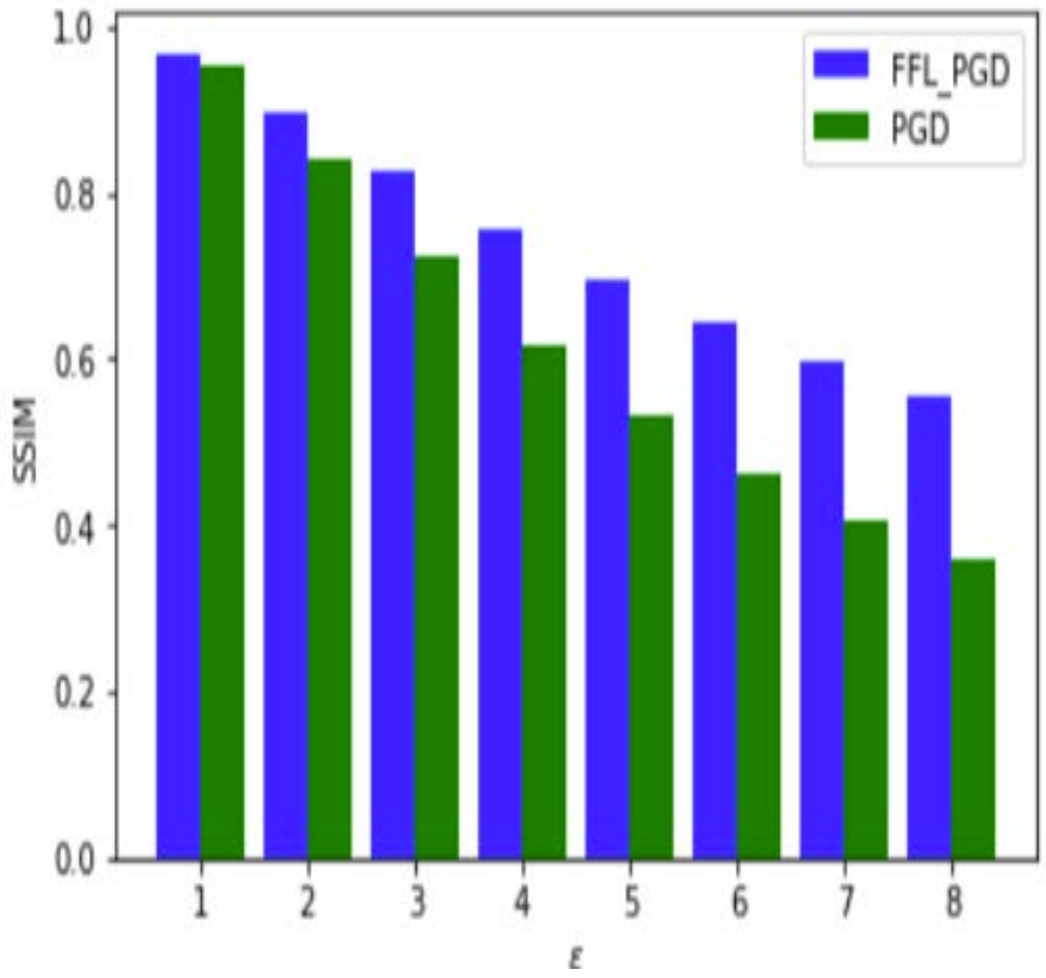
Attack Evaluation: PSNR



The figure records the PSNR of PGD and FFL-PGD attacks

- PGD has a higher PSNR ,which is considered as better image quality .But both of them higher than 20dB when ε from 1 to 8, which means both of them are considered acceptable for image quality.

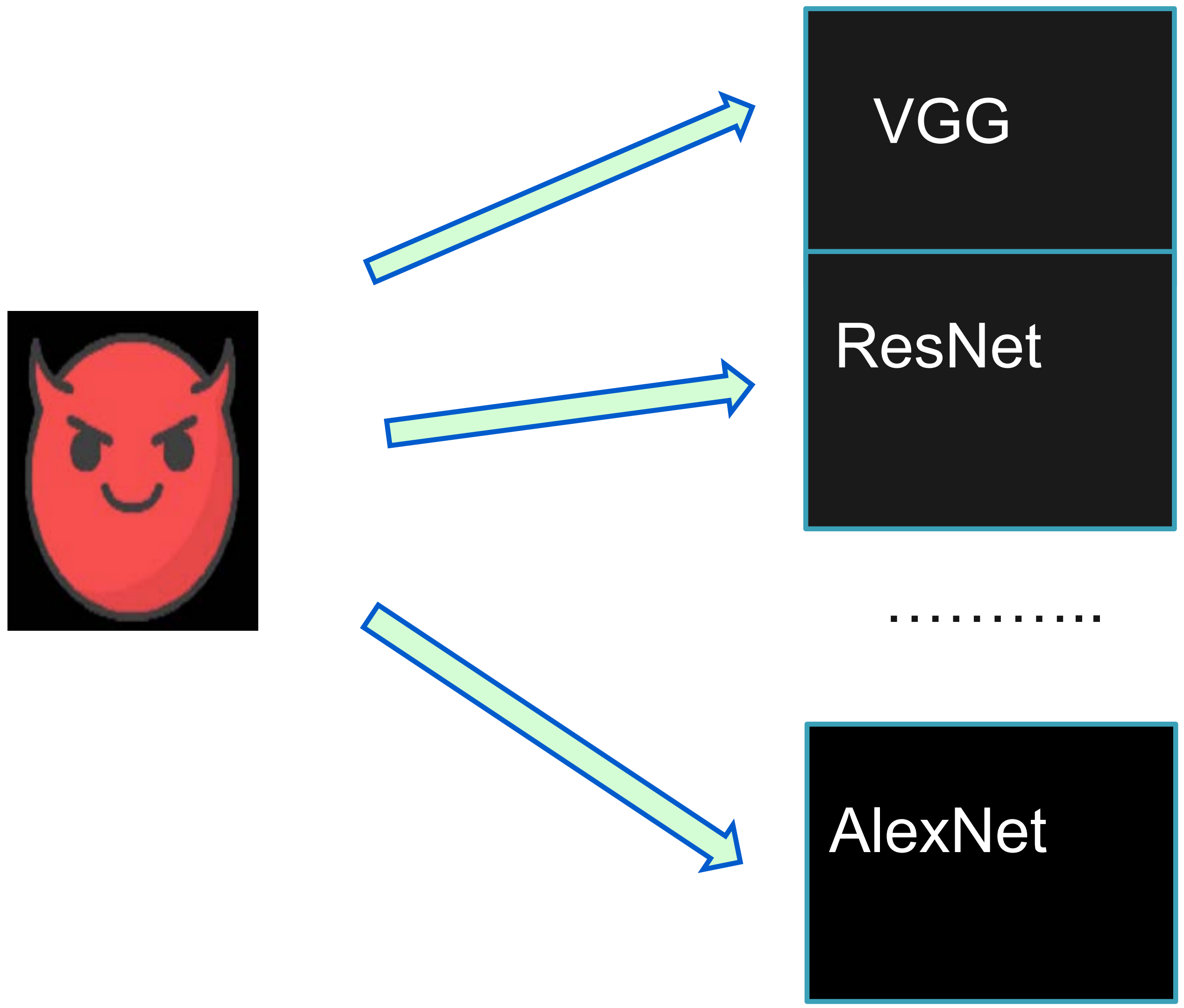
Attack Evaluation: SSIM



The figure records the SSIM of PGD and FFL-PGD attacks

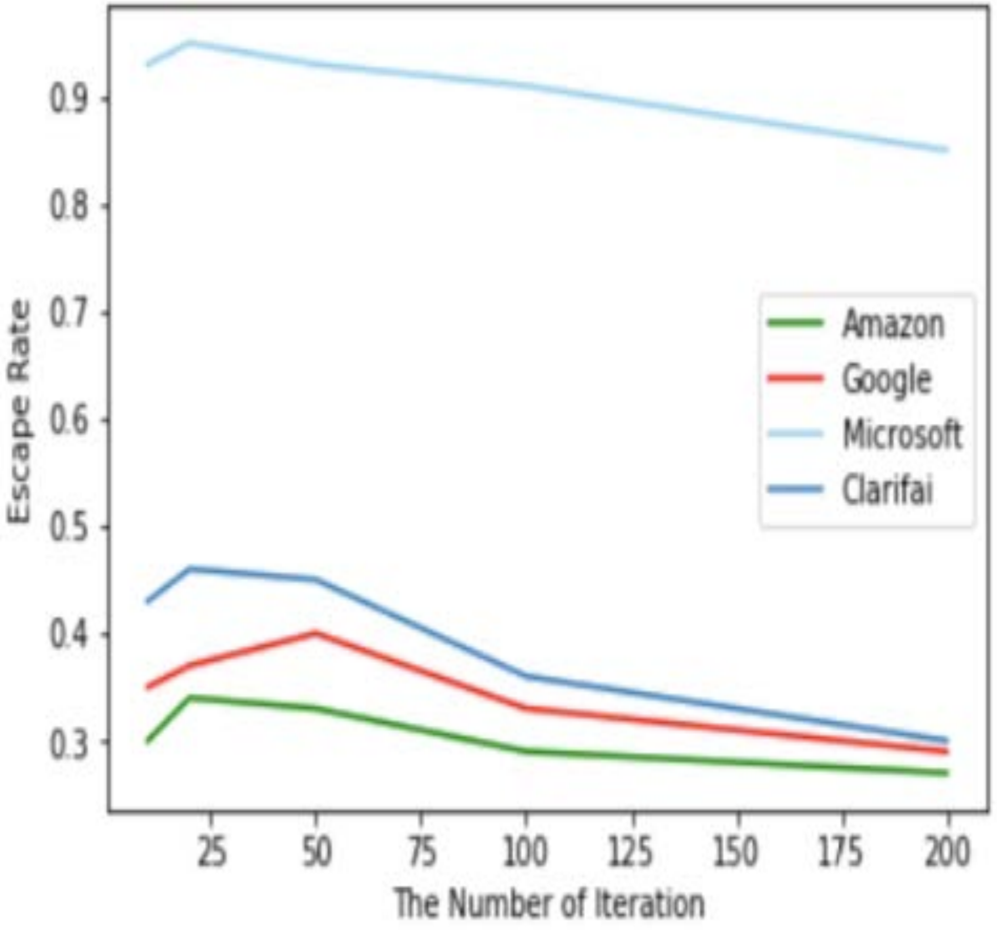
- FFL-PGD has a higher SSIM ,which is considered as better image similarity

Attack Evaluation: Ensemble-model Attack



- Ensemble-model attack a lot of DNNs to generate adversarial examples which can fool them at once

Attack Evaluation: Ensemble-model Attack

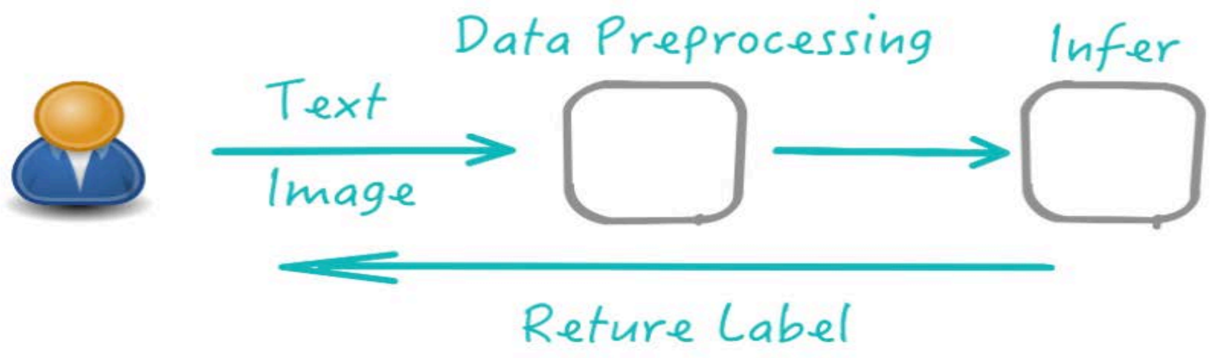


model attack

- The escape rates of Amazon, Google and Clarifai are below 50%
- The transferability decreases in the face of the pre-processing of cloud services, such as resizing,cropping

Adversarial attack mitigation

Defense strategies



- Reactive: detect adversarial examples after deep neural networks are built, e.g., Adversarial Detecting, Input Reconstruction, and Network Verification.
- Proactive: make deep neural networks more robust before adversaries generate adversarial examples, e.g., Network Distillation, Adversarial training, and Classifier Robustness.

Defense methods

Stages of ML-as-a-service	Mitigation Methods
Input Preprocessing	Feature Squeezing & Spatial Smoothing(Xu et al., 2017) Randomization(Xie et al., 2017a) Blurring(Hosseini et al., 2017a)
Prediction	PGD Adversarial Training(Madry et al., 2017a) Gaussian Augmentation(Zantedeschi et al., 2017) Ensembling Adversarial Training(Tramèr et al., 2017) Adversarial Logit Pairing(Kannan et al., 2018) Regularizing Input Gradients(Ross & Doshi-Velez, 2017) Randomized Adversarial Training(Araujo et al., 2019) Feature Denoising(Xie et al., 2018) Attention and Adversarial Logit Pairing(Goodman et al., 2019b)

Methods and parameters of Adversarial training

Stage	Method	Parameters
Training	Random Rotation(degree range)	(0,360)
	Random Grayscale(probability)	0.5
	Random Horizontal Flip(probability)	0.5
	Random Resize and Crop(image size)	224
	Gauss Filter(ksize)	29
	Median Filter(ksize)	11
Image preprocessing	Median Filter(ksize)	11
	Grayscale	N/A

Defense results

Attack	w/o Defense	w/ Defense
Gaussian Noise	0.60	**0.80**
Rotation	0.70	**0.80**
Salt-and-Pepper Noise	0.50	**0.95**
Monochromatization	0.4	**0.80**

Our defense technology can effectively resist known Spatial Attack, such as Gaussian Noise, Salt-and-Pepper Noise, Rotation, and Monochromatization.

Security testing for model robustness

Robustness evaluation

(a) *noise* (b) *brightness* (c) *contrast* (d) *blur*

(e) *rotation* (f) *raining* (g) *snowing*

Network	original	gaussian_noise	brightness	contrast	gaussian_blur	rotation	raining	snowing
InceptionV3	77%	52%	60%	55%	20%	30%	51%	40%

- **security-related**

Using the model gradient to stack perturbation to attack.

FGSM, PGD, C/W, etc

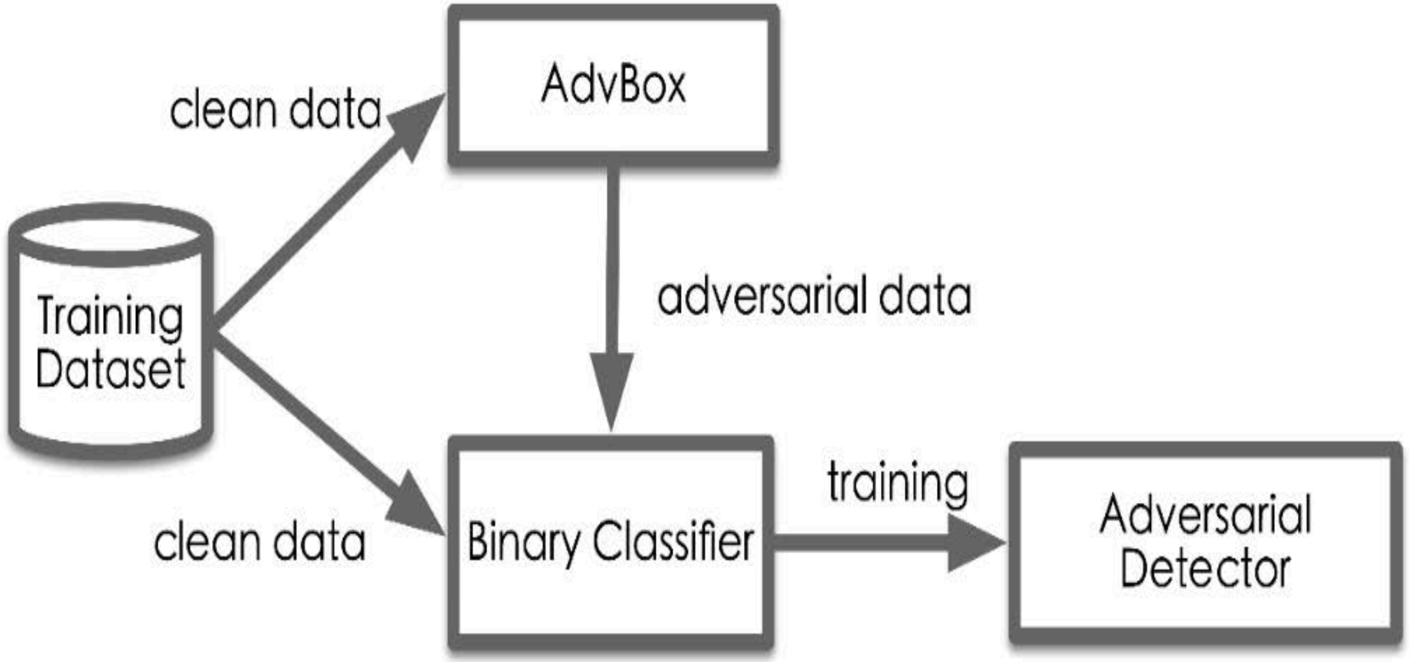
- **safety-related**

Using adversarial examples formed by spatial transformation or image corruption.

scaling, light transformation, weather, blur, shake, etc

Adversarial attack detection

Attack Detection: training binary classifier



- training deep neural network-based binary classifiers as detectors to classify the input data as a legitimate (clean) input or an adversarial example

Conclusion

Conclusion

- Cloud services may still be subject to adversarial attacks.

- Combined with the characteristics of adversarial attacks and cloud services, the security development cycle for machine learning applications is introduced, including adversarial detection, model defense, and robustness evaluation.

- The application of these methods in the development will greatly improve the security of the model and help developers build more secure software.

About Communication

OPEN SOURCE

- AdvBox

Advbox is a toolbox to generate adversarial examples that fool neural networks in PaddlePaddle、PyTorch、Caffe2、MxNet、Keras、TensorFlow and Advbox can benchmark the robustness of machine learning models.

https://github.com/advboxes/AdvBox
https://github.com/advboxes/perceptron-benchmark

CONTACTS

- Wang Yang

Leader of AI Model Security Team
Research Area: AI Security; Web Security

- Hao Xin

Senior researcher of AI Model Security Team
Research Area: AI Security; Web Security

black hat ASIA 2020

Thank you