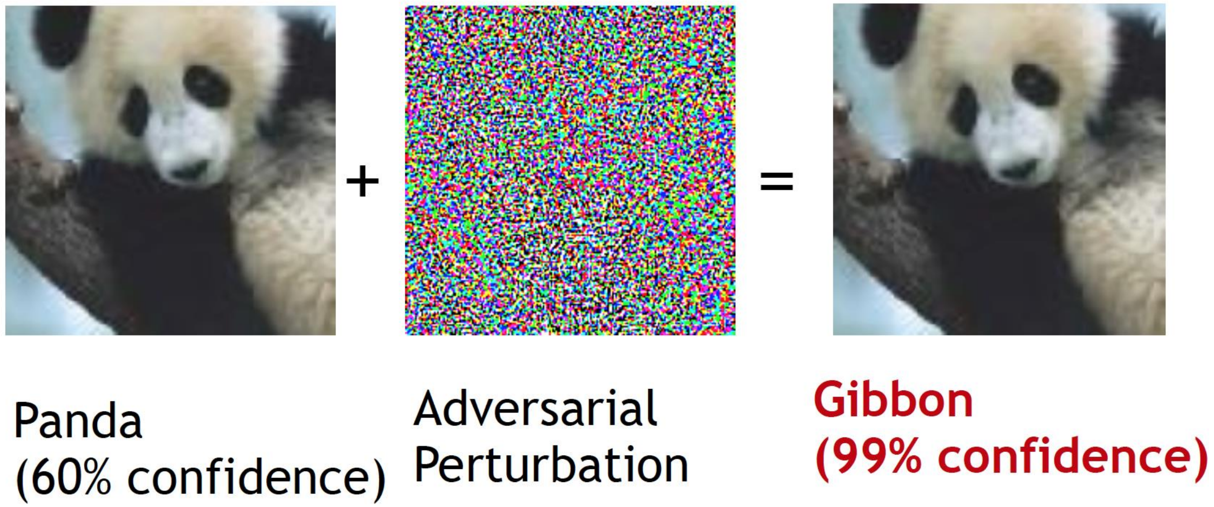
Adversarial Attacks and Defenses

Explainable and Trustworthy AI

Gabriele Ciravegna

Adversarial attacks – in brief

- They are malicously crafted example making the network provide a wrong classification
  - They can be created for a specific network or for many networks
  - The perturbation may be valid for many samples
  - We still don’t have know on how to completely defend from them



Summary

1. Why are adversarial attacks important?

2. Adversarial Attacks

3. Adversarial Defenses

Why are adverarial attacks important?

Is the threat real? Road Sign Attack

Adversarial sign

Normal sign

https://www.youtube.com/watch?v=xwKpX-5Q98o

Is the threat real? Adversarial 3-D Object

https://www.youtube.com/watch?v=piYnd_wYlT8

…But also attacks on:

CELL-PHONE CAMERA

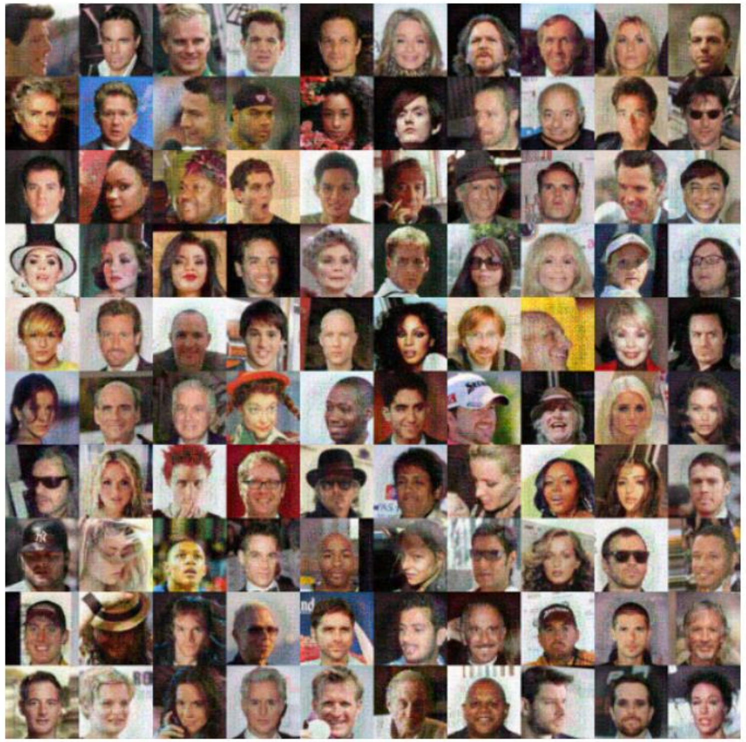
CYBERSPACE ATTACK

ROBOTIC VISION

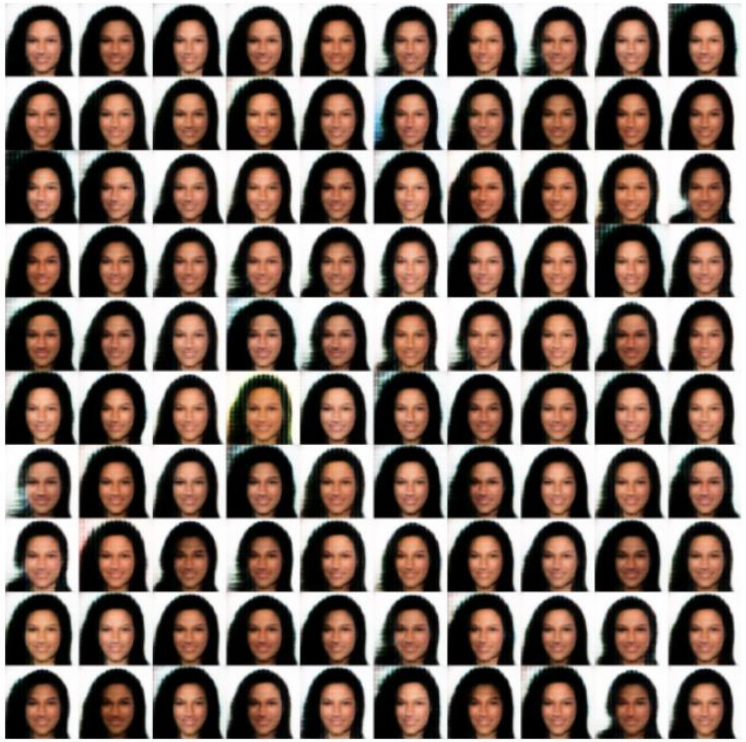
VISUAL Q&A

Only Concerns Object Recognition?

Attacks on Generative Models



AutoEncoder Input
(Adversarial)

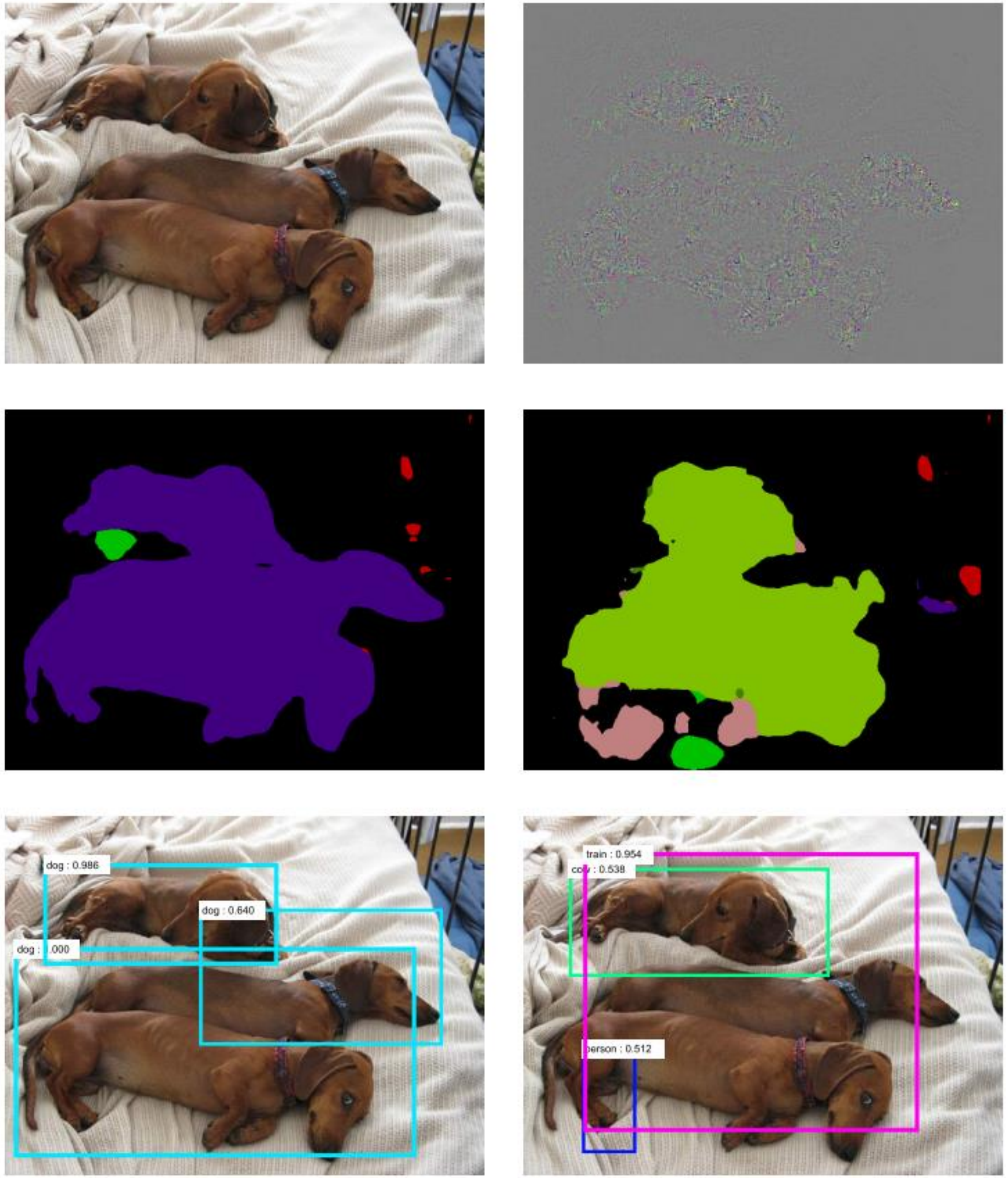


AutoEncoder Output

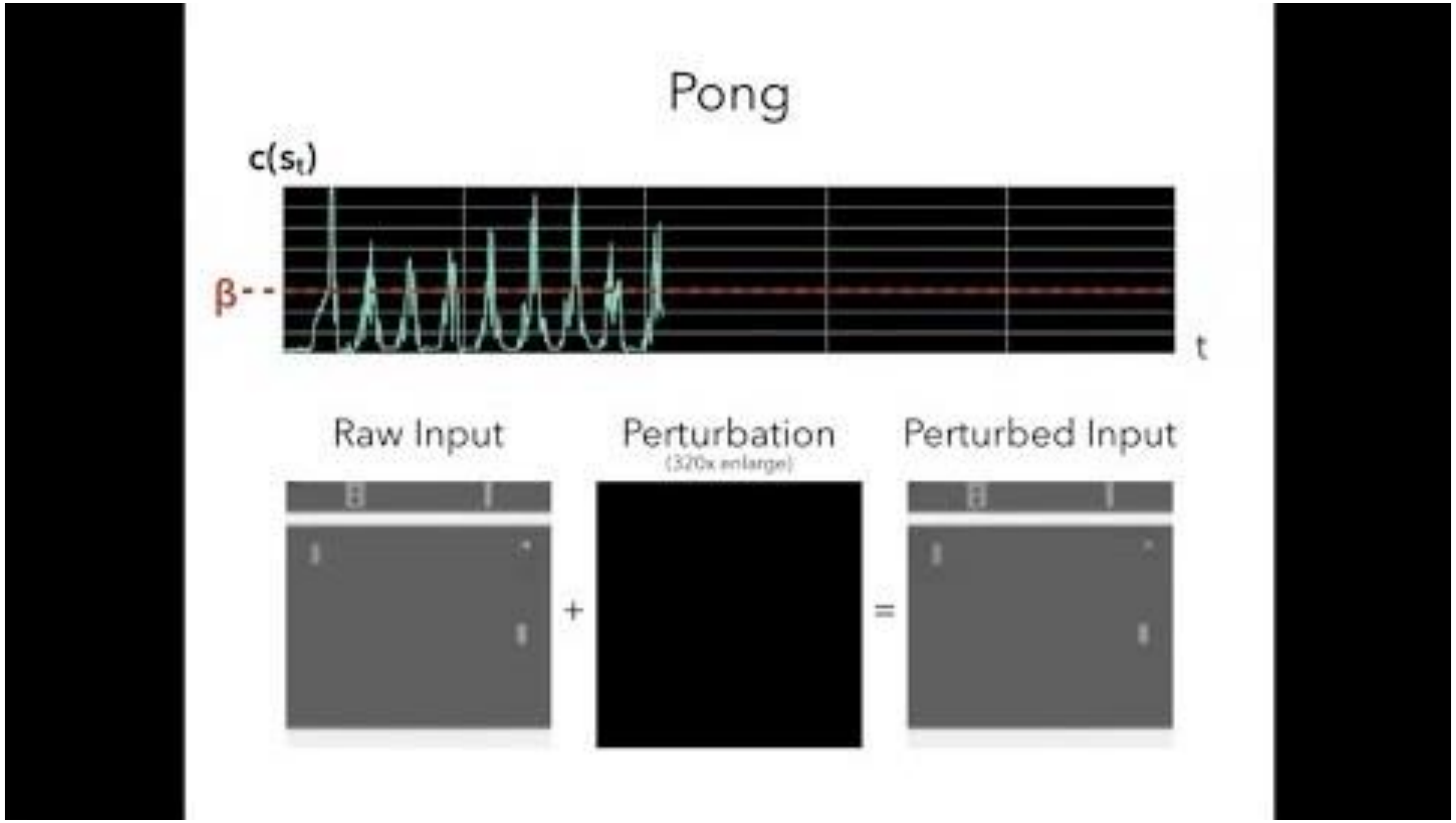
Attacks on Voice Recognition

Groundtruth	• **"The fact that a man can recite a poem does not show he remembers any previous occasion on which he has recited it or read it".**
G-Voice – original example:	• **"The fact that a man can decide a poem does not show he remembers any previous occasion on which he has work cited or read it."**
G-Voice – adversarial example:	• **"The fact that I can rest I'm just not sure that you heard there is any previous occasion I am at he has your side it or read it."**

Attacks on Semantic Segmentation



Attacks on Deep Reinforcement Learning



https://www.youtube.com/watch?v=gCMNRnWc-s0

Not Network Specific:
Good generalization capabilities

Adversarial examples often transfer well between different NNs

Allow many ‘Black Box’ attacks

(where we don’t have access to the real network but only to a recreated copy)

Why Adversarial Examples exist?

- «Supposed Reasons» (We don’t have an answer yet)

Structural reason:

‘Linearity Hypothesis’ (Goodfellow)

- Flatness of decision boundaries
- Low flexibility of the networks
- Many disagree: non-linearity of decision boundaries

Algorithmic reason:

‘Evolutionary Stalling’

- Positive samples stop contributing to the network update once correctly classified
- Weight regularization helps, but it is still present

There exists any effective defense?

- Yes, but existing defense methods have issues:

Most Defenses are attack-specific:

New attacks fool old defense methods

Counter-counter methods are possible: modifying an existing attack can make a defense vulnerable

Adverarial attacks

Types of attacks

Knowledge of the network

- **Black-box attack:**
  - The network is not analyzed inside, we check only its output
- **White-box attack:**
  - The activations and the gradient of the inner layers is used to maximize the attack

Specificity of the attack

- **Image specific:**
  - The single image get wrongly classified
- **Universal attack**:
  - The perturbation is valid for many different samples

Types of attacks

Iterations

- **Single-step attack:**
  - The attack requires a single forward of the network
- **Iterative attack:**
  - The attack is repeated many times until is successfull

Target

- **Untargeted attack**
  - The image get classified as a different class (no matter which)
- **Targeted-attack**:
  - The network classify the sample as belonging to a specific class

Attack Metrics

Fooling Rate: number of samples effectively fooled

Perturbation distance: distance of the adversarial sample from the original one

Time to attack: iterations/seconds required to complete the attack

Attack Evolution

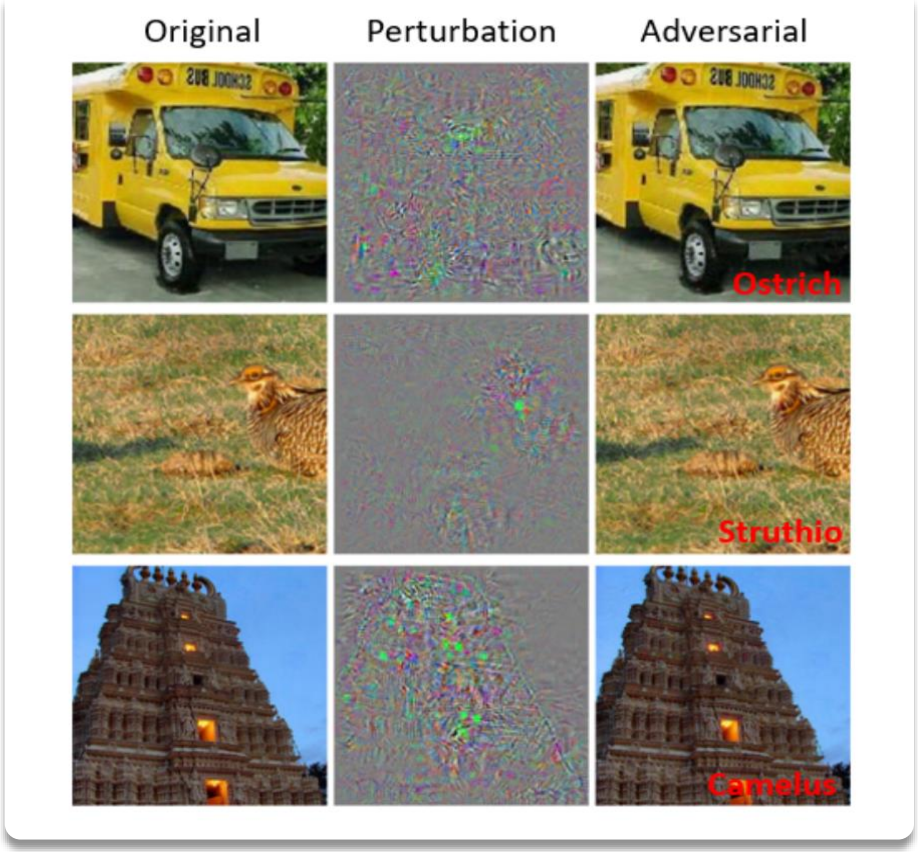
- White-Box Image Specific Single-Step Attacks
- White-Box Image Specific Iterative Attacks
- White-Box Universal Iterative Attacks
- Black-Box Attacks

Attack Evolution

White-Box Image Specific Single-Step Attacks

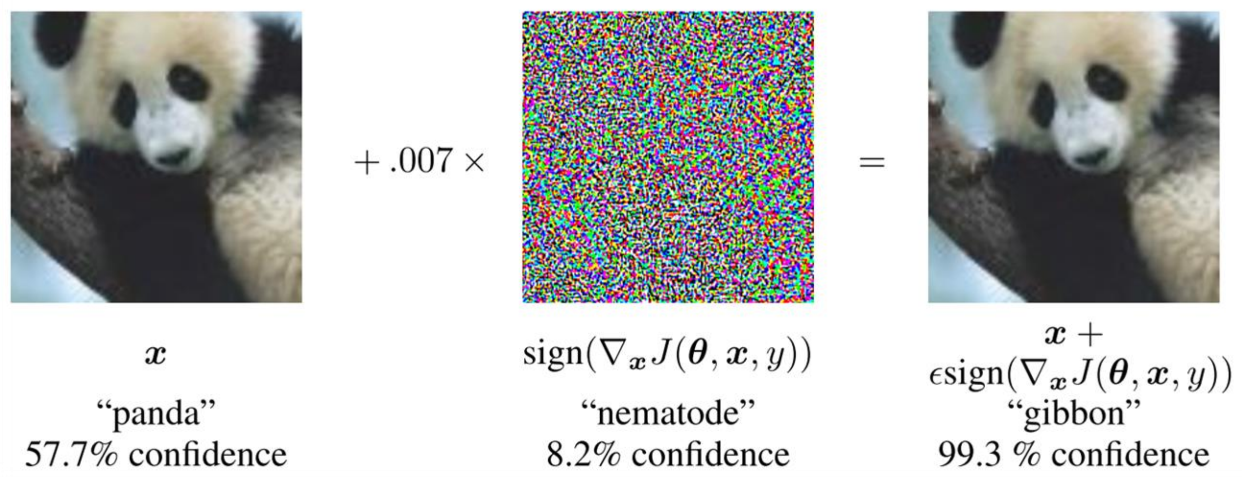
1) BOX-CONSTRAINED L-BFGS ATTACK

- First adversarial attack paper
  - “Intriguing properties of neural networks“ (Szegedy 2014)
- Optimization problem:
  - $\min||\rho||_2 : C(I_c + \rho) = l_{target}$
  - $\min\{ ||\rho||_2 + \mathcal{L}(I_c + \rho,\ l_{target})\}$
  - I_c: clean image, ρ: perturbation, l_{target} != $C(I_c)$ $C(\cdot)$: classification
  - L-BFGS optimization: approximate the inverse hessian matrix to guide the search in the variable space
  - --> not using the gradient through the network



2) Fast Gradient Sign Method (FGSM - Goodfellow)

- Optimization problem:
  - $\rho = \varepsilon * sign\,(\nabla J(\theta, I_C, l)$
  - l: current class
- It directly compute the gradient ∇J through the network weights θ
  - It allows fast computation
  - Exploits the linearity of the model
- Also, Introduced the adversarial training idea



Historical Evolution

White-Box Image Specific Single-Step Attacks

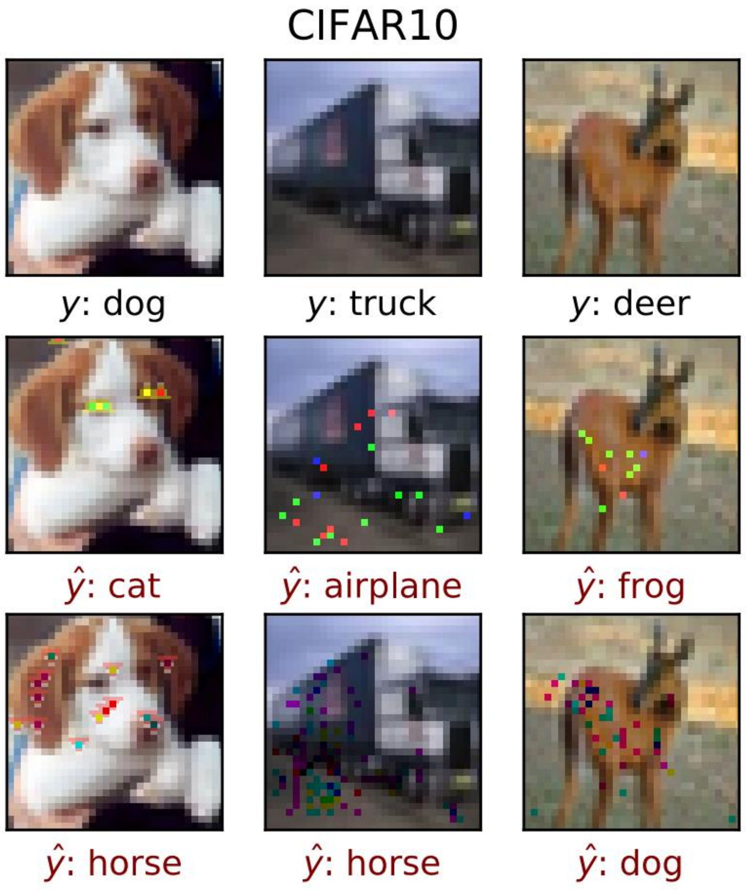
White-Box Image Specific **Iterative** Attacks

3) Basic Iterative Method (BIM) & Iterative Least-Likely Class Method (ILCM)

- Optimization problem:
  - $I_\rho^{i+1} = Clip_\varepsilon\{I_\rho^i + \alpha * sign\ (\nabla J(\theta, I_\rho^i, l)\}$
  - α: step size, $I_\rho^0 = I_c$, repeated n times
- BIM: l – untargeted attack
- More computationally expensive
- ILCM: l_{target} - targeted attack to predict even the least likely class

4) Jacobian-based Saliency Map Attack (JSMA)

- Algorithm based on the saliency map
  - Identify through a saliency map the most influential pixels for a class
  - Saturate the pixels with maximum and minimum values
  - Stop when the class has been predicted
- Objective $|\epsilon|_0$ rather than $|\epsilon|_\infty$: minimize the number of pixels modified
- Simple algorithm to determine strength of defense algorithms



Other Attacks

5) Deep Fool Attack

- Iteratively push an image to the current nearest decision boundary
  - Based on the gradient of the network
  - At each iteration assume the network is linear
  - The final perturbation corresponds to the sum of the perturbation

6) Carlini & Wagner Attacks (C&W)

- Again iterative method
- Minimize the difference between the loss of the original class and the loss of the second most likely class
- Fool several defense methods
  - Adversarial examples with minimal magnitude

7) Projected Gradient Descent (PGD)

- Extension of BIM
  - Iterative attack projecting the perturbation within a selected boundary
- Random initial perturbation
  - Several restart to get the best

Attack Evolution

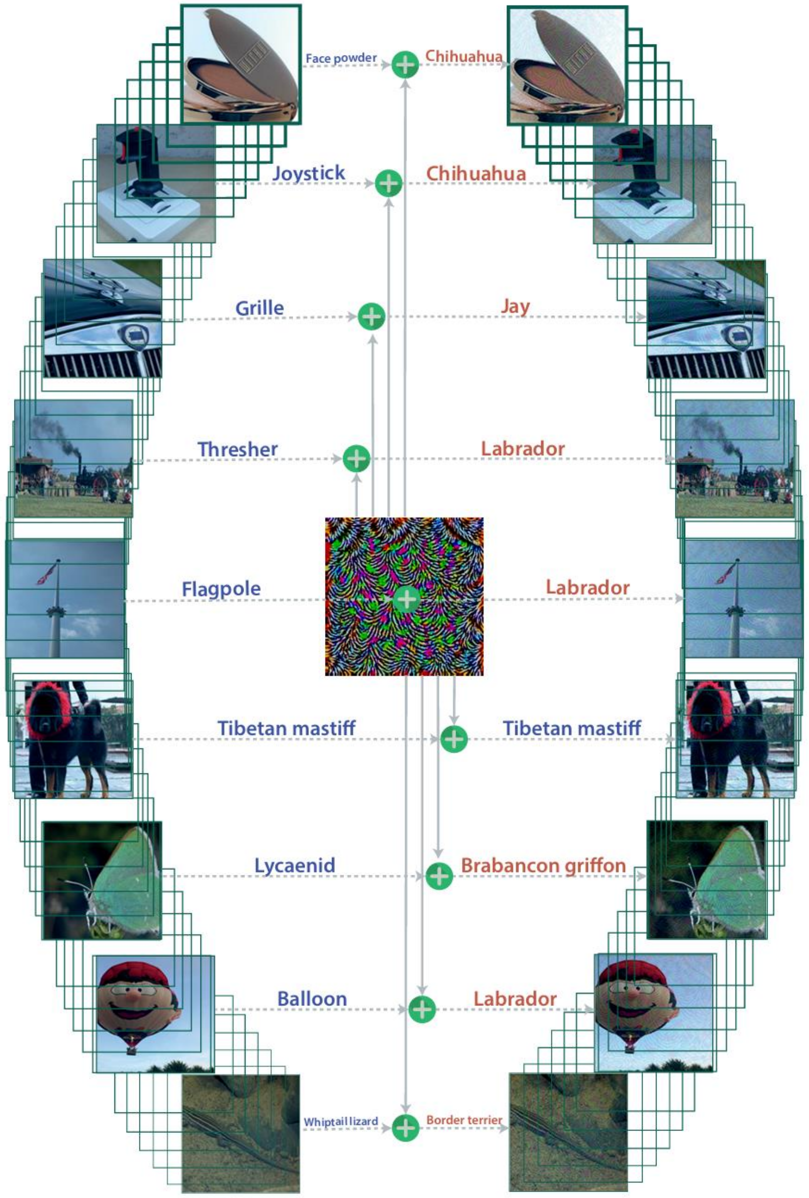
White-Box Image Specific Single-Step Attacks

White-Box Image Specific Iterative Attacks

White-Box **Universal** Iterative Attacks

7) Universal Adversarial Perturbation

- Fool a network on “any” image with the same perturbation
- $P\big(C(I_c) \neq C(I_c + \rho)\big) \geq \delta \ : \ ||\rho|| \leq \xi$
- They sum the perturbations needed on all images to create a single universal perturbation
- Untargeted attack: all images are wrongly classified for different classes

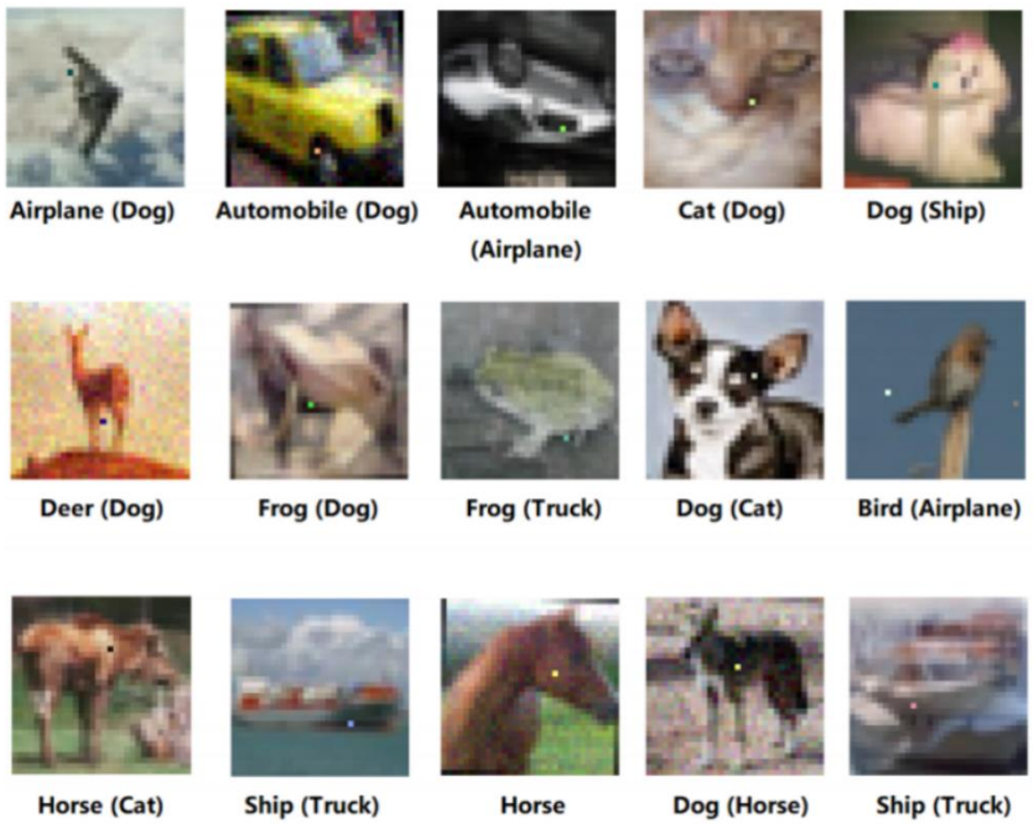


Attack Evolution

- White-Box Image Specific Single-Step Attacks
- White-Box Image Specific Iterative Attacks
- White-Box Universal Iterative Attacks
- **Black-Box** Attacks

8) One-Pixel Attack

- Only one pixel of the image is perturbed
- Evolutionary algorithm to select pixels and perturbations
  - based on reduction of the network classification precision
- Do not access internal parameters or loss of the net (BlackBox attack)
- Does not always work (low fooling rate)



9) UPSET, ANGRI

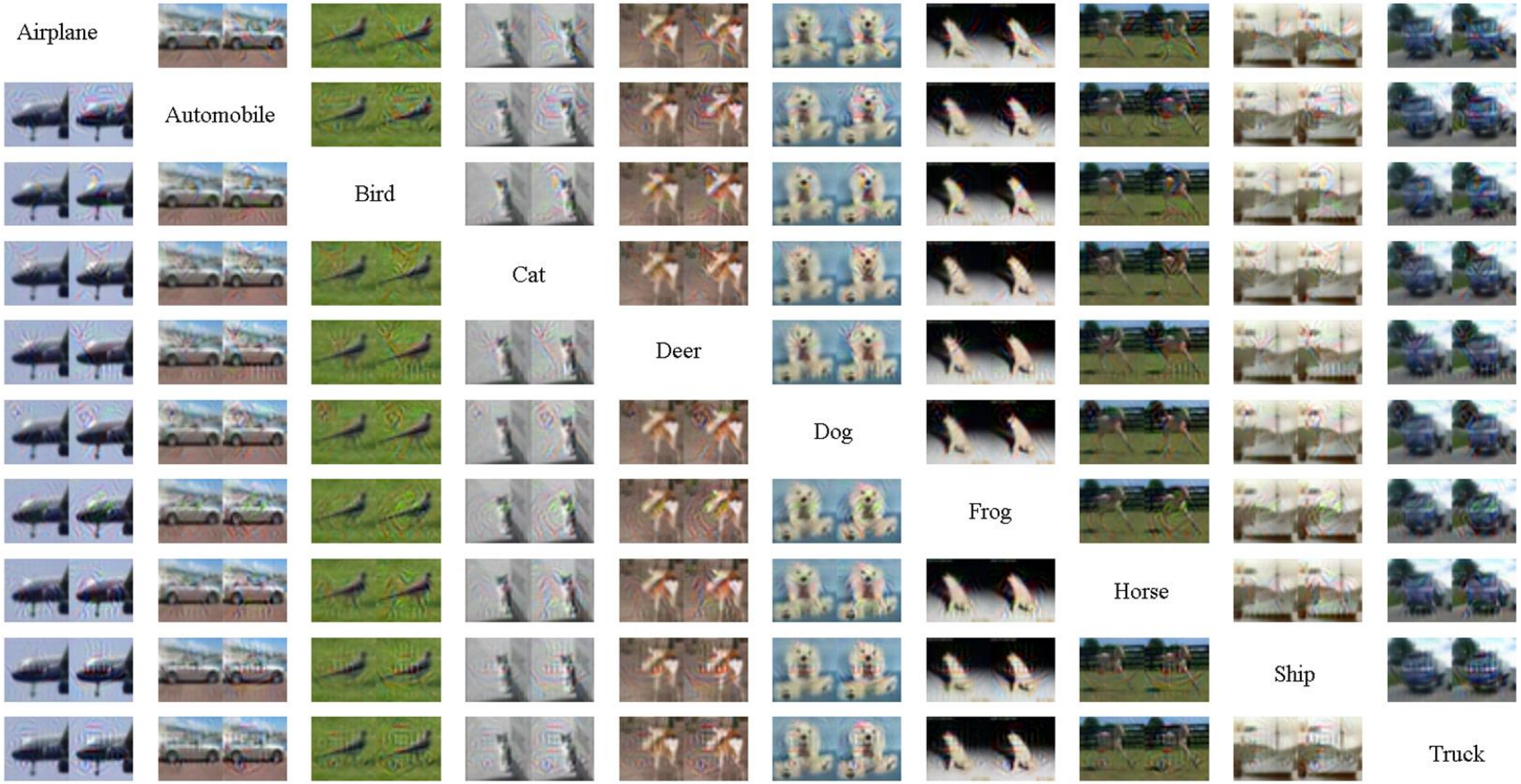
Residual Generating Network R(t):

- $I_p = \max(\min(sR(t) + I_c, 1), -1) : C(I_p) = l\_target$
- Generate n perturbation $I_{p,i}$ one for each class i
- They are added to the image and clipped between [-1, +1]

- ANGRI:
  - Antagonistic Network for Generating Rogue Images
  - Finds a targeted Image-Specific perturbation

- UPSET
  - Universal Perturbations for Steering to Exact Targets
  - i.e., a perturbation that when added to any image it makes it classify as belonging to a specific class

9) UPSET, ANGRI

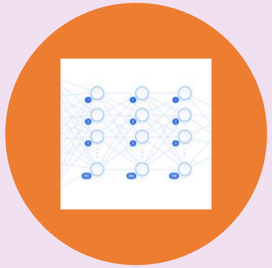


Adversarial Attack Framework?

- **Adversarial Robustness Toolbox (ART)**:
  1. **Features**:
     1. Supports popular machine learning frameworks (TensorFlow, Keras, PyTorch, MXNet, scikit-learn, etc.).
     2. Works with various data types (images, tables, audio, video, etc.).
     3. Covers a wide range of machine learning tasks (classification, object detection, speech recognition, generation, certification, etc.)
  2. **GitHub Repository**: You can find ART on GitHub.
- **Foolbox**:
  1. **Description**:
     1. Python library to run adversarial attacks against machine learning models, including deep neural networks.
     2. It works natively with models in PyTorch, TensorFlow, and JAX.
  2. **Website**: Explore Foolbox on the official website.

Defenses

Requirements



LOW IMPACT ON THE ARCHITECTURE

MAINTAIN SPEED OF THE NETWORK

MAINTAIN ACCURACY ON CLEAN DATA

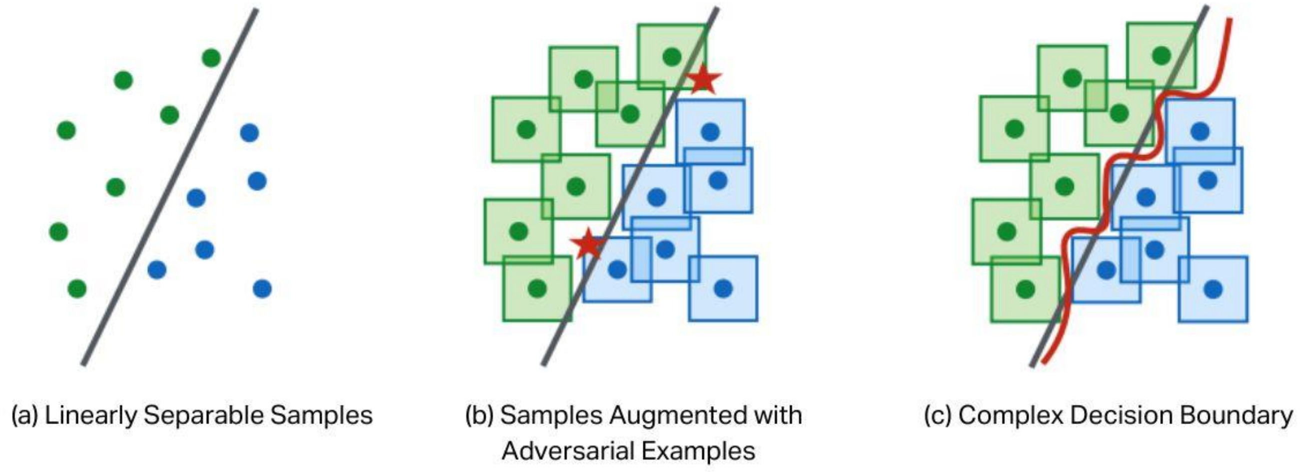
CORRECTLY CLASSIFY ONLY ADVERSARIAL EXAMPLES CLOSE TO THE REAL ONES

Types of Defense algorithms

Modified input data	• Defense, reduce the fooling rate
Modifying the network (layer or loss function)	• Defense, reduce the fooling rate • Detection, check for adv. input data
Network add-ons (external sub-module)	• Defense, reduce the fooling rate • Detection, check for adv. input data

Modified input data: Defense methods

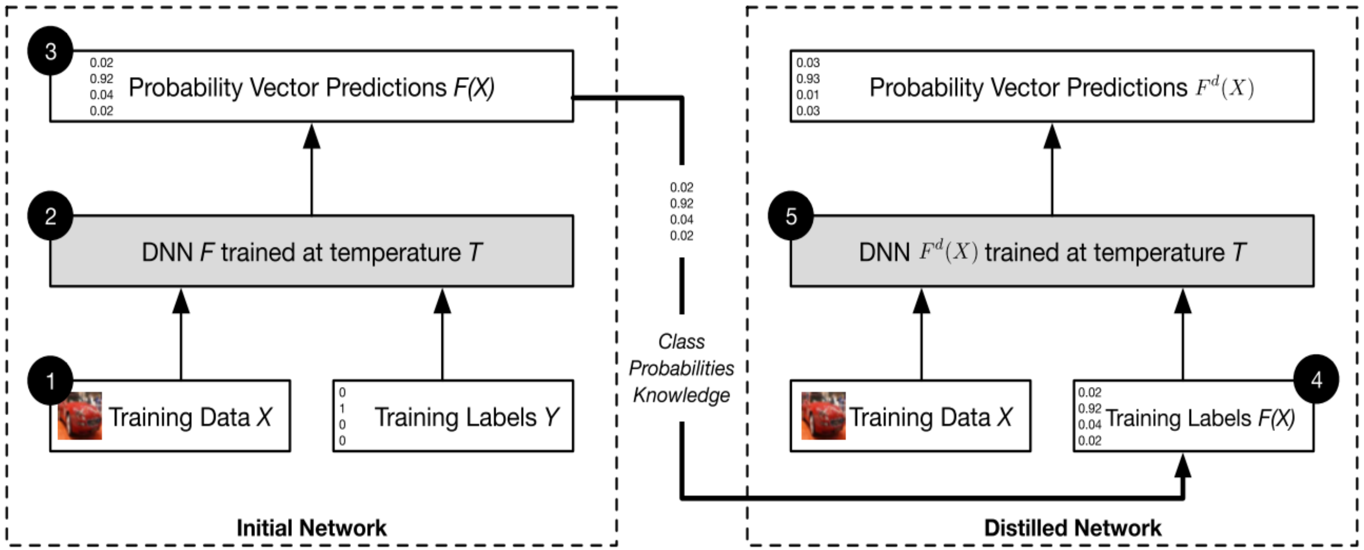
1. Adversarial Training
   1. One of the most effective techniques
   2. Regularize the network - Improves the robustness
2. Data Compression as a defense
   1. Both at training and test time
   2. JPEG compression & PCA/DCT
3. Foveation based defense
   1. Foveate several parts of the image at training time
   2. Focus only on important parts
4. Also data augmentation (less effective)



Augmenting Training Data with Adversarial Examples

Modifying the network: defense methods

1. Defense Distillation
   1. Based on Knowledge-distillation (teacher-student learning framework)
   2. Also very effective
2. Deep Contractive Network
   1. Regularization forcing the latent space to be insensitive to small input variations
3. Gradient Regularization
   1. Penalize output variations w.r.t. small input variations
   2. Parseval Networks: control lipschitz constant
4. Biologically Inspired Network
   1. Higly non-linear activations (like neuron dendrites) - > improve robustness



Defense Distillation

Modifying the network: Detection-Only Approaches

Classify adversarial examples

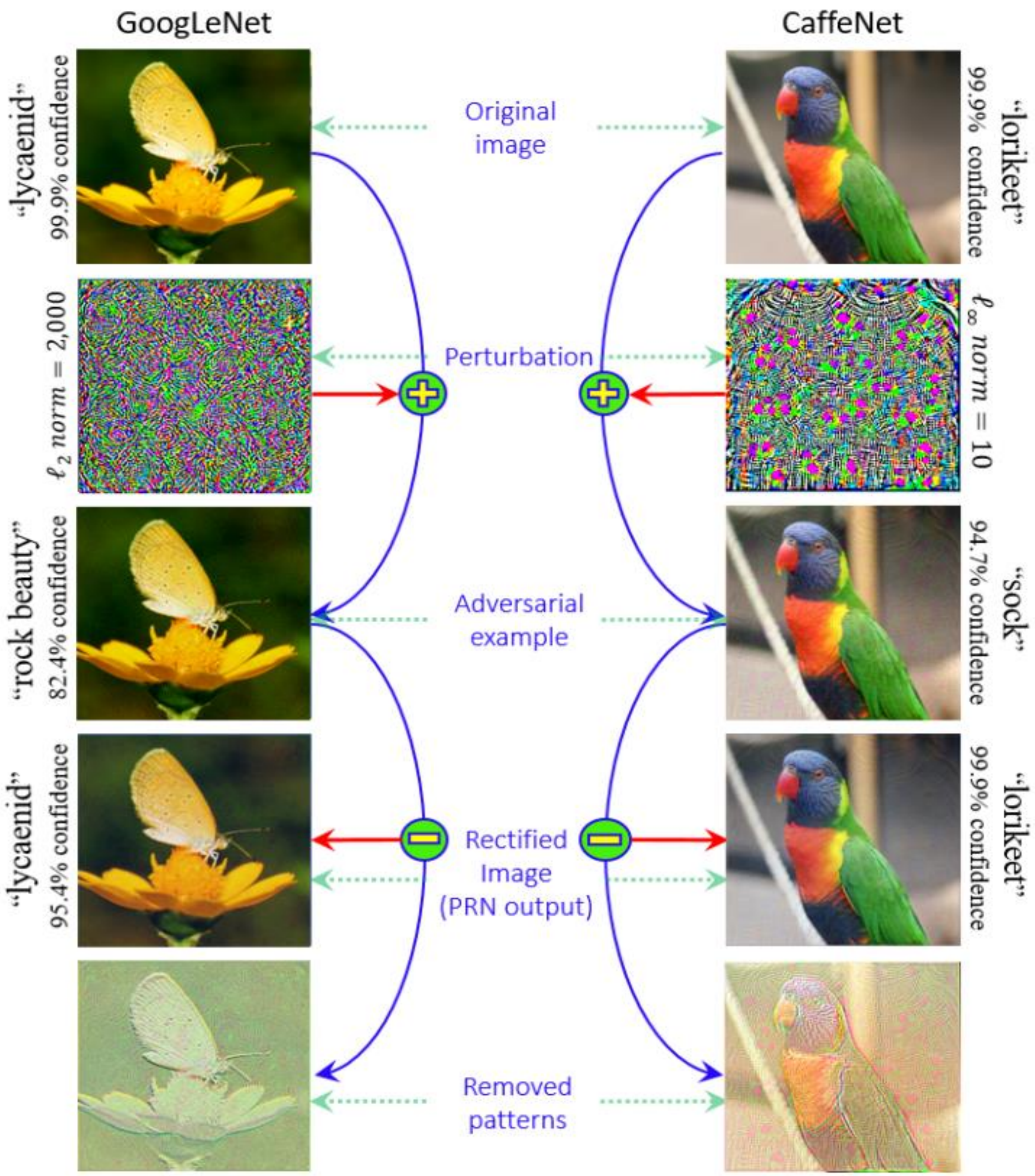
1. As an additional class
2. With a detector subnetwork pre-processing layer trained to discard adv. samples

Control activations statistics

1. RELU activations (Safety Net)
2. Convolutional filter activation

Network add-ons: Defense Algorithms

1. Defense Against Universal Perturbation
   - Detector + Pre-processing Rectifier Network
   - PRN: rectify inputs when and adv. sample is detected
2. GAN-based defense
   - Ad-hoc for the network to train
   - Need to correctly classify both adversarial and clean data - different from gan selector
   - Different from adv. training: the generator is trained to fool the network



Network add-ons: Detection-Only Approach

Feature squeezing

- Reduce pixel depth (# of colors that can be displayed)
- Perform spatial smoothing
- Classification Comparison of original and squeezed images

Magnet

- External model learn data manifold
- Reform near data and exclude far images

Conclusions

- Is The Threat Real? **YES**
- Does It Concern Only Computer Vision? **NO**
- Are Attacks Network Specific? **NO**
- Why Adversarial Examples Exists? **Unknown**
- There exists effective defense yet? **NO**

Thank you for your attention