

# **Computer Vision Applications**

**COMP 388-002/488-002 Computer Science Topics**

**Daniel Moreira**  
**Fall 2022**



**LOYOLA**  
UNIVERSITY CHICAGO

**RECAP**

# Face Recognition

COMP 388-002/488-002 Computer Science Topics  
Computer Vision Applications

**Daniel Moreira**  
Fall 2022



**LOYOLA**  
UNIVERSITY CHICAGO

# Practical Activity 1

**RECAP**

## Work in Pairs

Use Google Colab at <https://bit.ly/3FZFk6S>

Observe ArcFace's<sup>1</sup> distance

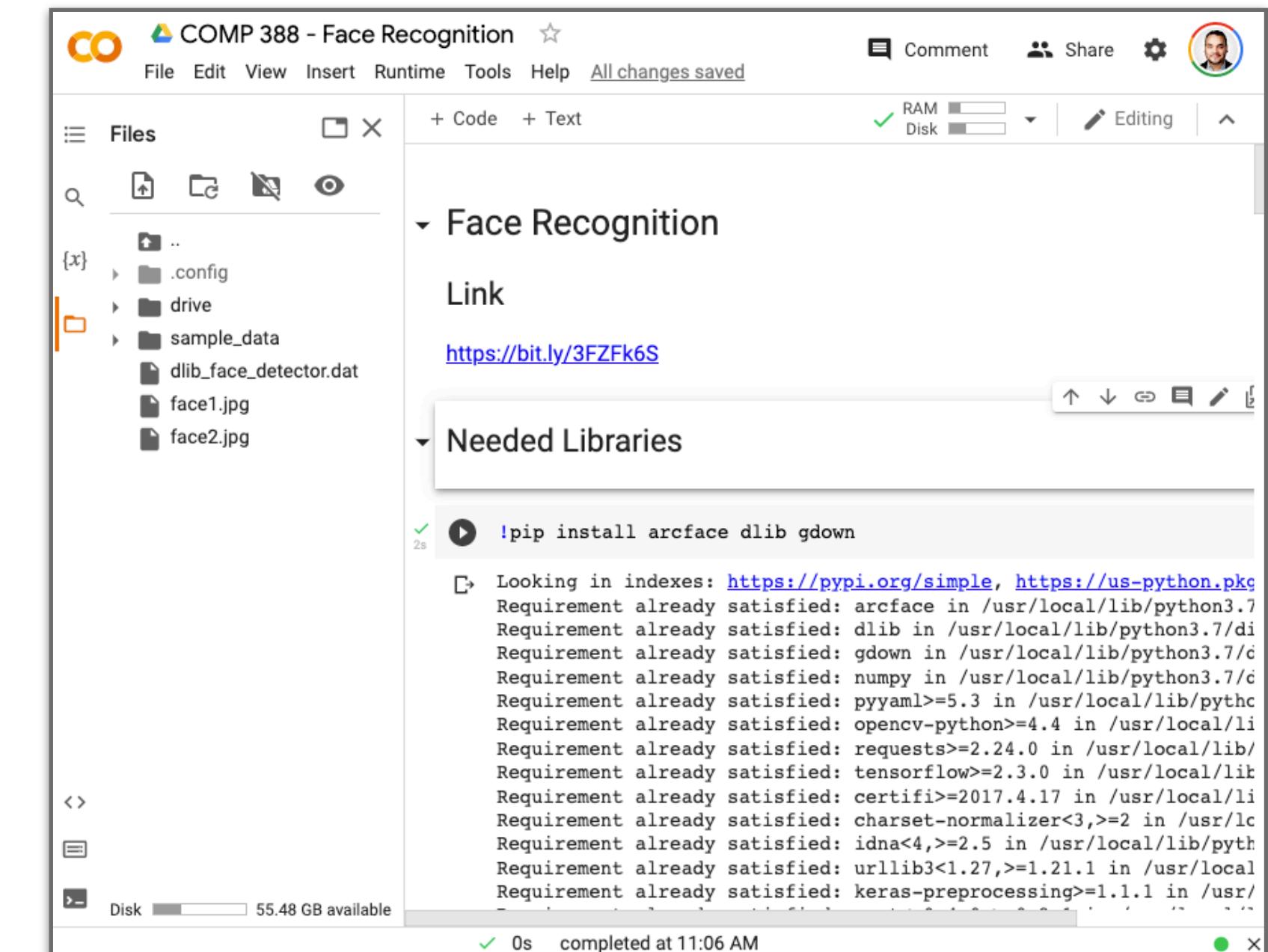
Different faces

Different poses

Different “accessories” (e.g., glasses, mask, etc.)

Contributions or Question?

1. <https://arxiv.org/abs/1801.07698>



The screenshot shows the Google Colab interface for a notebook titled "COMP 388 - Face Recognition". The left sidebar displays a file tree with a single folder named "Face Recognition" containing a file named "Link" which points to the URL <https://bit.ly/3FZFk6S>. The main workspace contains a code cell with the command `!pip install arcface dlib gdown`. Below the code cell, the output shows the results of the pip installation command, listing various dependencies such as arcface, dlib, gdown, numpy, pyyaml, opencv-python, requests, tensorflow, certifi, charset-normalizer, idna, urllib3, and keras-preprocessing.

```
!pip install arcface dlib gdown
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev
Requirement already satisfied: arcface in /usr/local/lib/python3.7
Requirement already satisfied: dlib in /usr/local/lib/python3.7/di
Requirement already satisfied: gdown in /usr/local/lib/python3.7/c
Requirement already satisfied: numpy in /usr/local/lib/python3.7/c
Requirement already satisfied: pyyaml>=5.3 in /usr/local/lib/pythc
Requirement already satisfied: opencv-python>=4.4 in /usr/local/li
Requirement already satisfied: requests>=2.24.0 in /usr/local/lib/
Requirement already satisfied: tensorflow>=2.3.0 in /usr/local/lit
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/li
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/lc
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/pyt
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/
```

# How about Attacks?

**RECAP**



The Guardian news article headline: "The fashion line designed to trick surveillance cameras". The article is from August 13, 2019. The URL is <https://www.theguardian.com/world/2019/aug/13/the-fashion-line-designed-to-trick-surveillance-cameras>.



<https://www.wired.com/story/10-year-old-face-id-unlocks-mothers-iphone-x/>

# Attacks

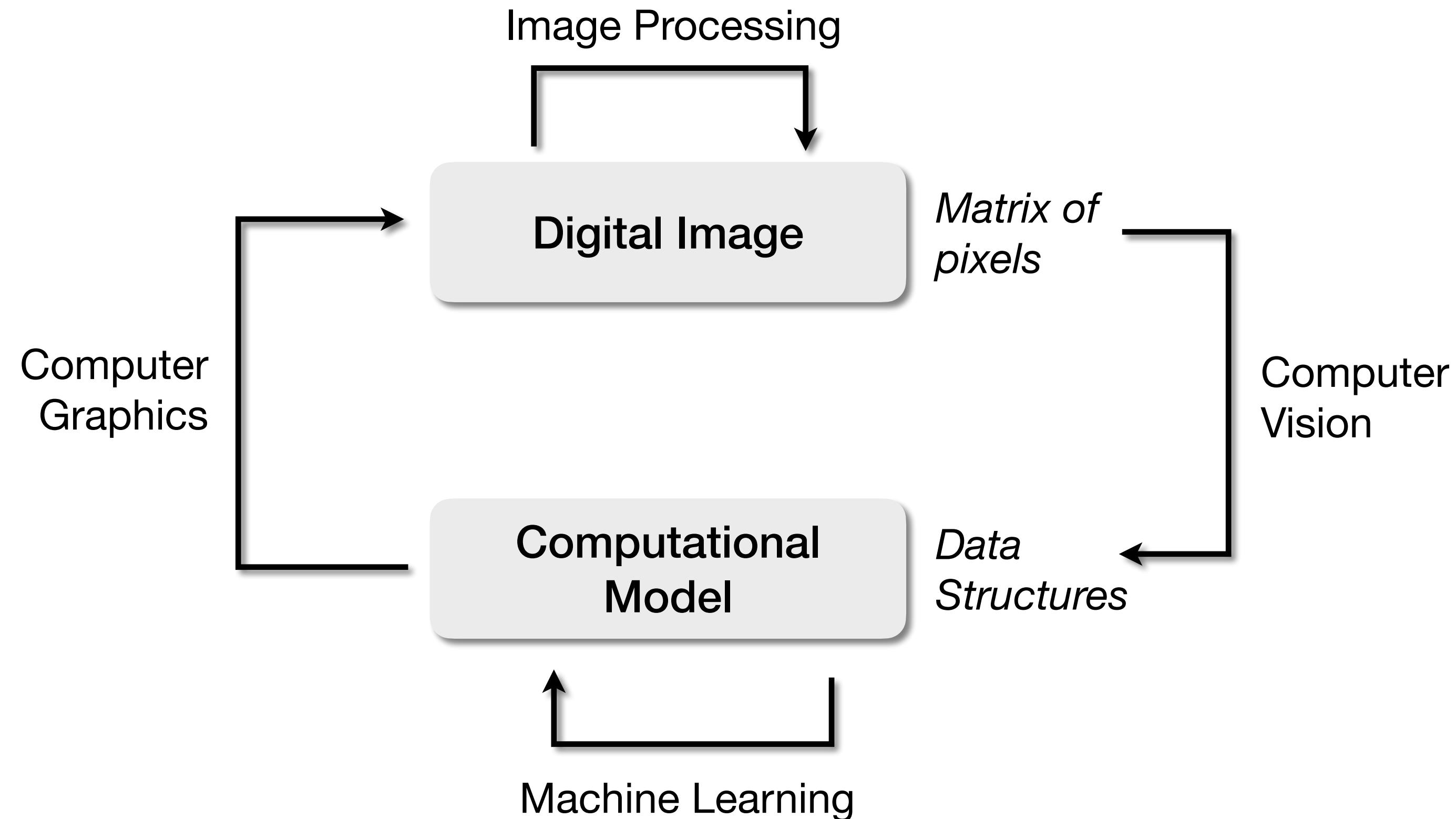
COMP 388-002/488-002 Computer Science Topics  
Computer Vision Applications

**Daniel Moreira**  
Fall 2022

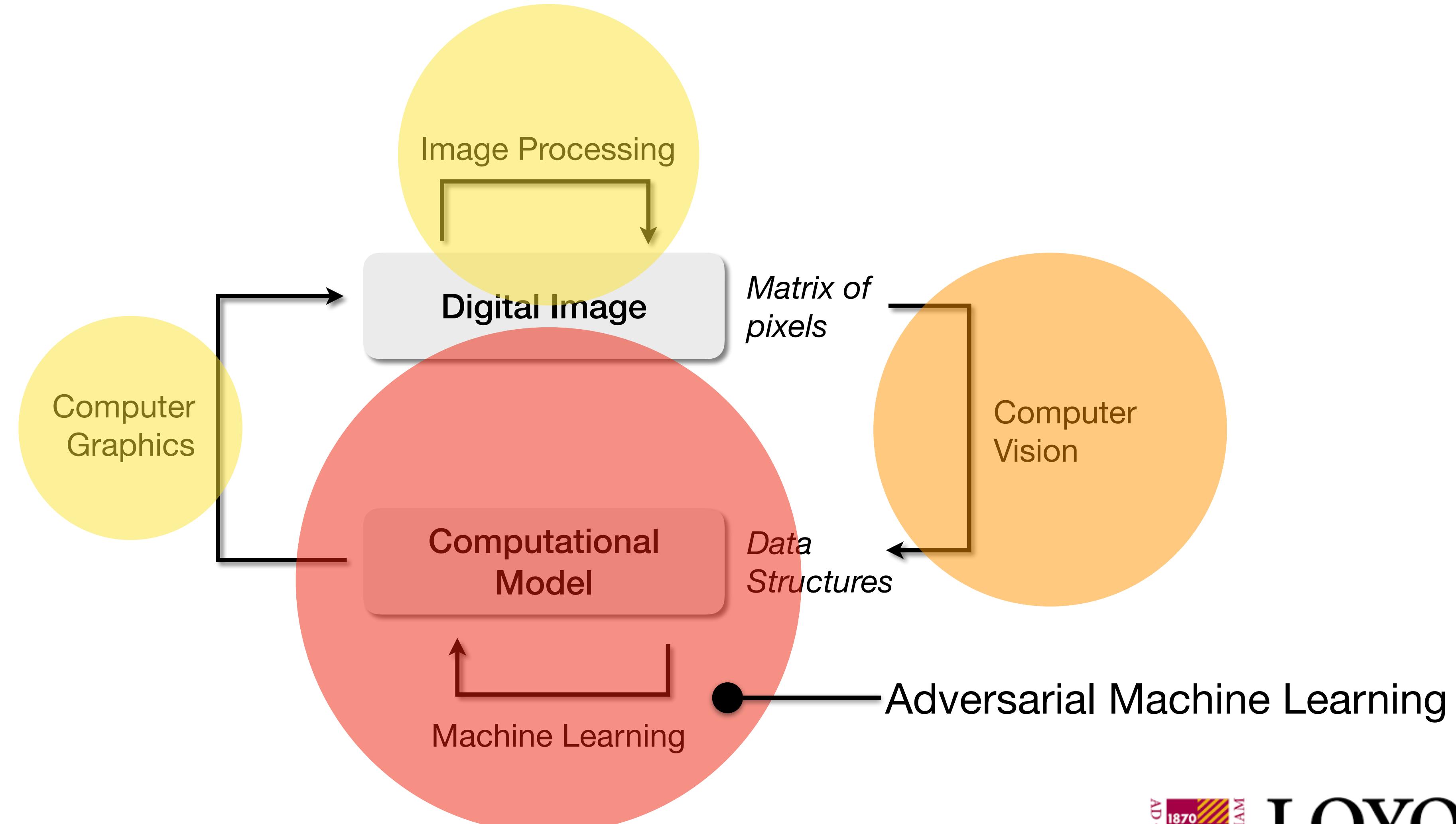


**LOYOLA**  
UNIVERSITY CHICAGO

# Computer Vision Attacks



# Computer Vision Attacks



# Adversarial Machine Learning

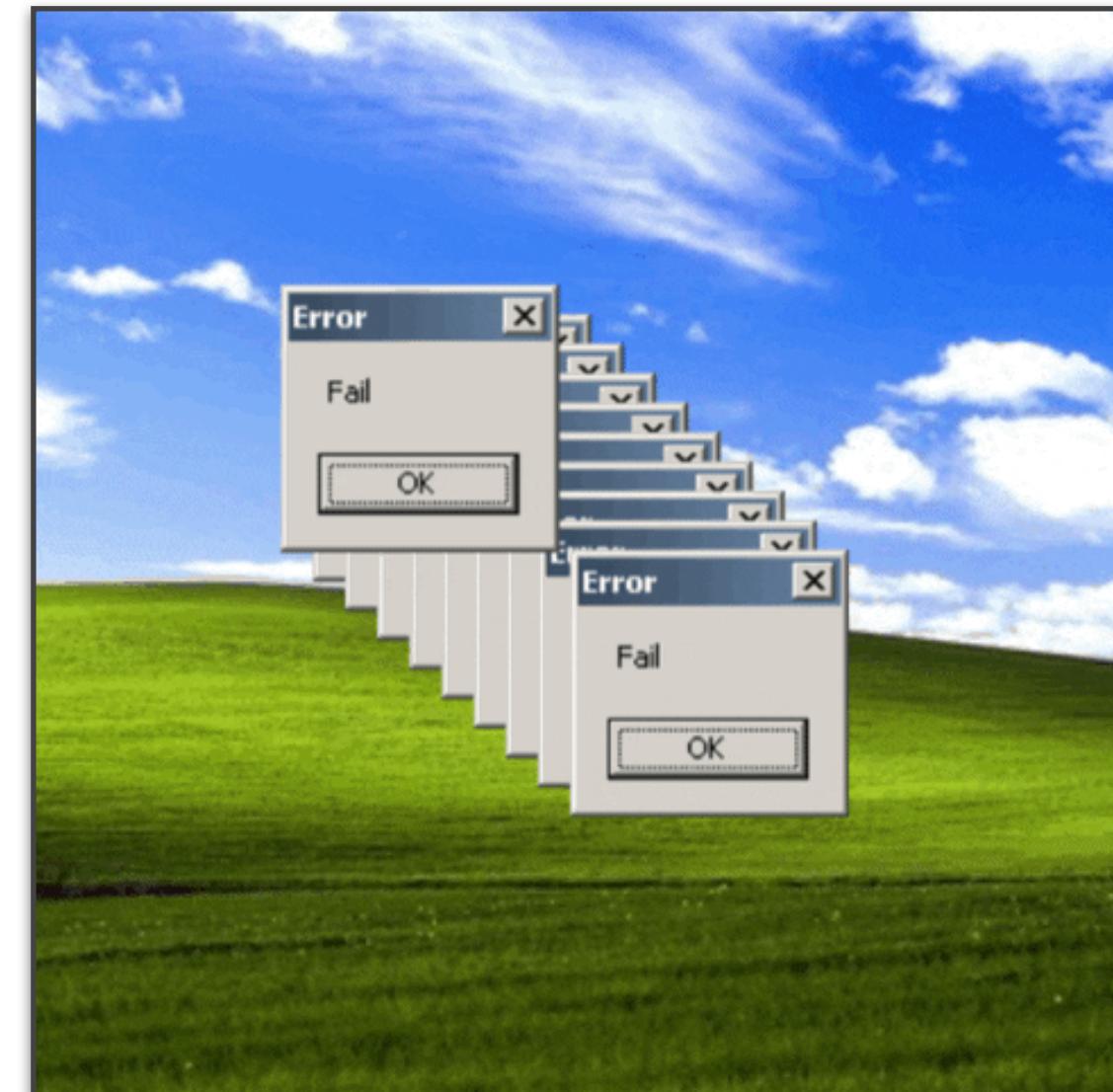
## Attacks and Defenses

### Attacks

Procedures performed by an **attacker** to explore the vulnerability of a computer system to make it behave out of specification.

### Defenses

Procedures to avoid, detect, or mitigate attacks.



# Adversarial Machine Learning

## Within Machine Learning

Data driven.

## Adversarial Data

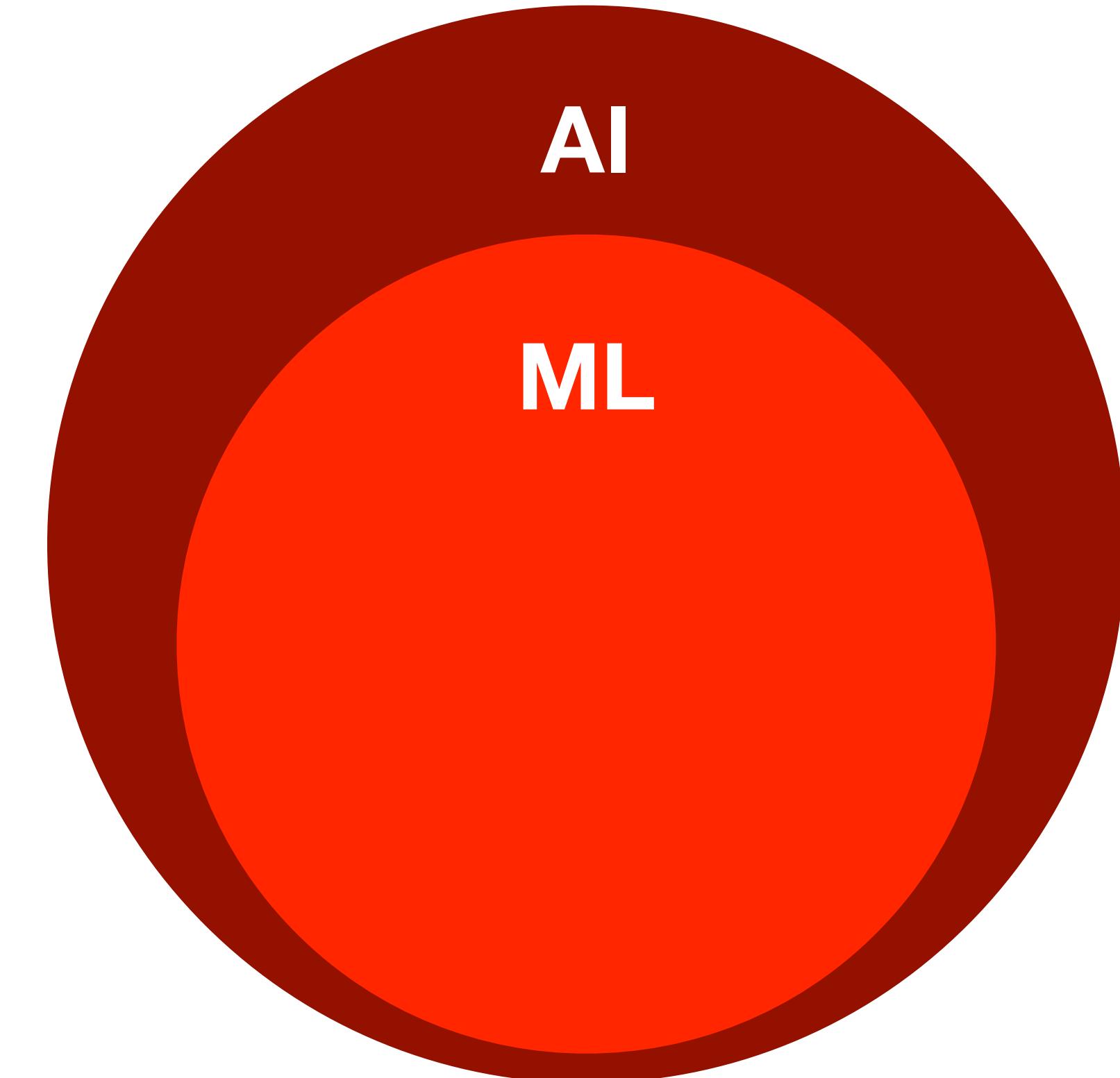
Synthetic samples pretending to be real.

Manipulated samples pretending to be pristine.

Data copies pretending to be original (spoofing).

## Two Attack Opportunities

At training time and at inference time.



# Adversarial Machine Learning

## Within Machine Learning

Data driven.

### Types of Attack

#### White Box

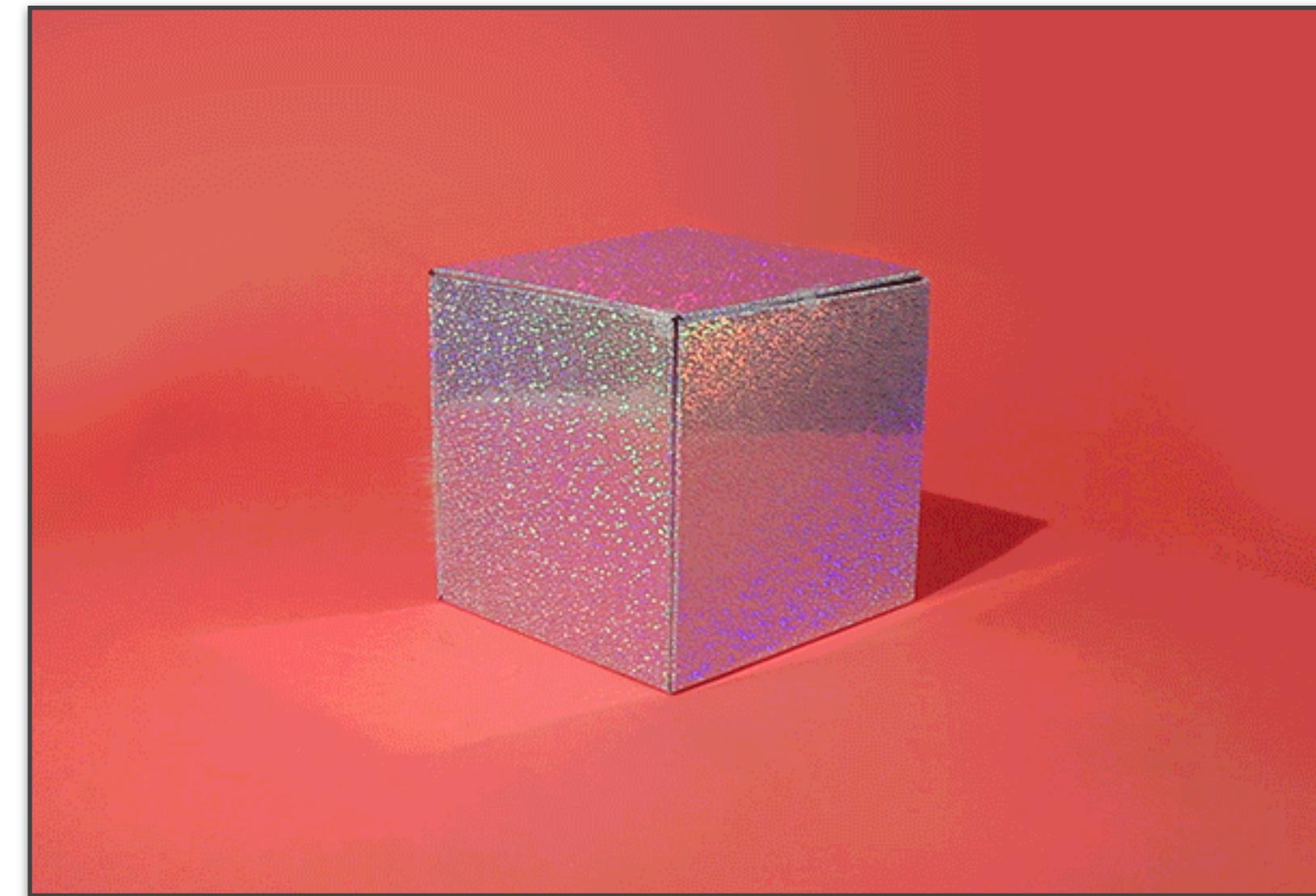
The attacker knows how the solution works and has access to its trained model.

#### Black Box

The attacker only sees the input and output of the solution; sometimes, even these are not clear.

#### Gray Box

The attacker has limited knowledge about the solution.



# Adversarial Machine Learning

## Within Machine Learning

Data driven.

## Attack Examples

### Evasion

The attacker avoids data detection  
(or causes misclassification) by the system,  
usually by presenting adversarial data  
(at inference time).



Example: evasion of text-based  
spam detection with images.

# Adversarial Machine Learning

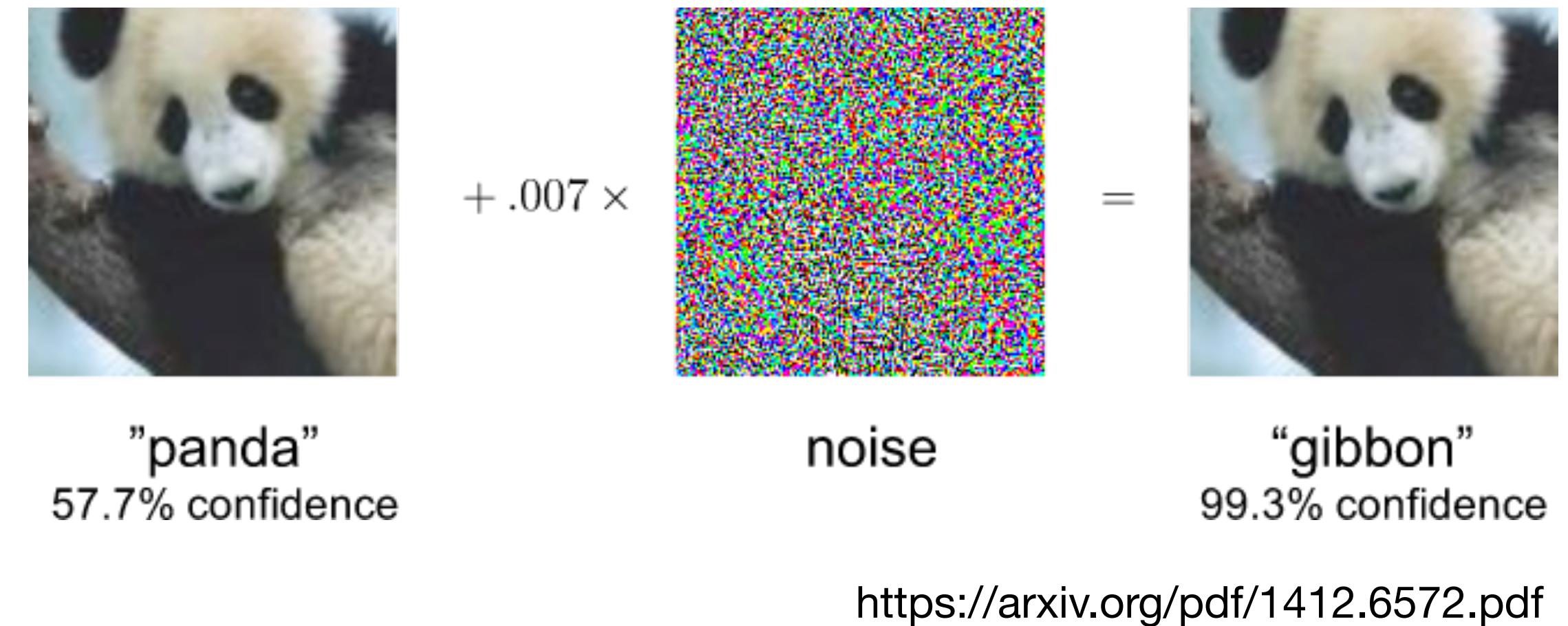
## Within Machine Learning

Data driven.

### Attack Examples

#### Evasion

The attacker avoids data detection (or causes misclassification) by the system, usually by presenting adversarial data (at inference time).



# Adversarial Machine Learning

## Within Machine Learning

Data driven.

### Attack Examples

#### Evasion

The attacker avoids data detection (or causes misclassification) by the system, usually by presenting adversarial data (at inference time).

(a) Image



(b) Prediction



(c) Adversarial Example



(d) Prediction



[https://openaccess.thecvf.com/content\\_ICCV\\_2017/papers/Metzen\\_Universal\\_Adversarial\\_Perturbations\\_ICCV\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_ICCV_2017/papers/Metzen_Universal_Adversarial_Perturbations_ICCV_2017_paper.pdf)

# Adversarial Machine Learning

## Within Machine Learning

Data driven.

## Attack Examples

### Evasion

The attacker avoids data detection (or causes misclassification) by the system, usually by presenting adversarial data (at inference time).

Daily **Mail**

Tesla cars tricked into autonomously accelerating up to 85 MPH in a 35 MPH zone while in cruise control using just a two-inch strip of electrical tape



[https://www.youtube.com/watch?v=4uGV\\_fRj0UA](https://www.youtube.com/watch?v=4uGV_fRj0UA)



**LOYOLA**  
UNIVERSITY CHICAGO

# Adversarial Machine Learning

## Within Machine Learning

Data driven.

## Attack Examples

### Evasion

The attacker avoids data detection  
(or causes misclassification) by the system,  
usually by presenting adversarial data  
(at inference time).

Repudiation



[https://www.youtube.com/watch?v=\\_PoudPCevN0](https://www.youtube.com/watch?v=_PoudPCevN0)

# Adversarial Machine Learning

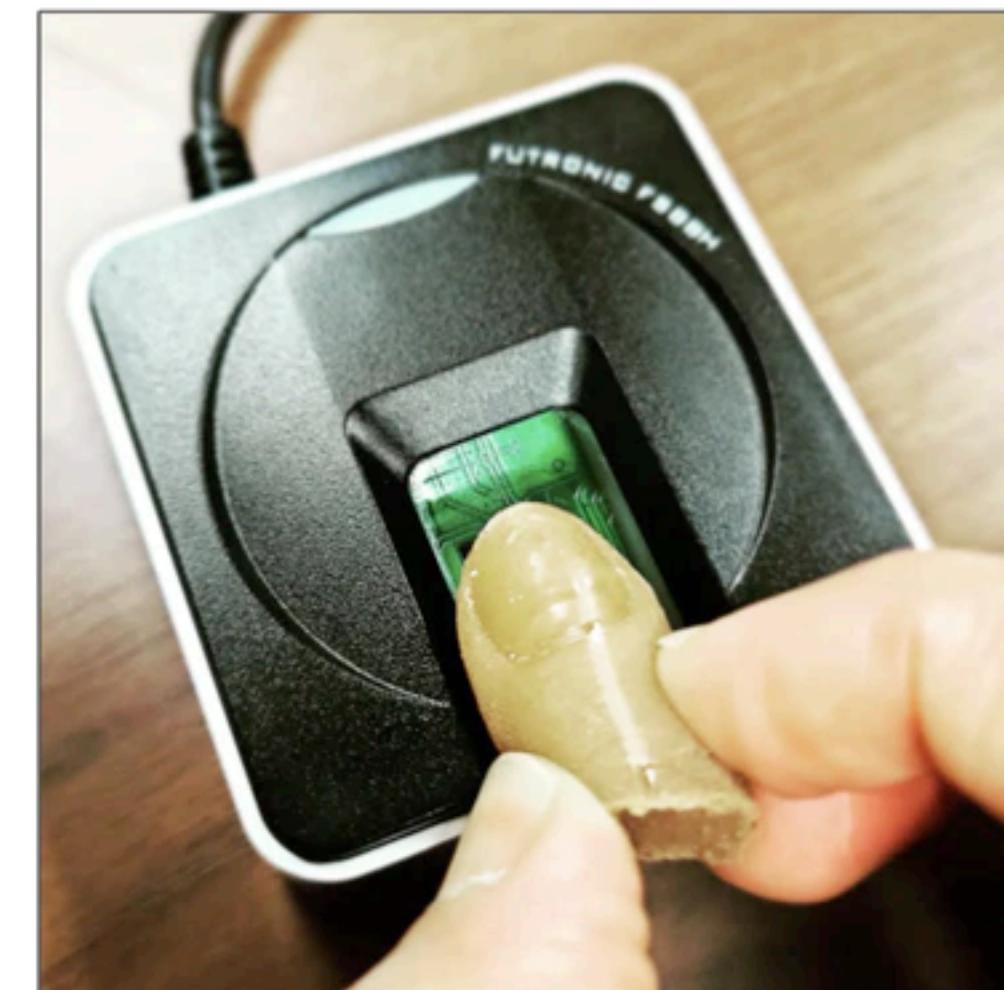
## Within Machine Learning

Data driven.

## Attack Examples

### Evasion

The attacker avoids data detection (or causes misclassification) by the system, usually by presenting adversarial data (at inference time).



Spoofing



# Adversarial Machine Learning

## Within Machine Learning

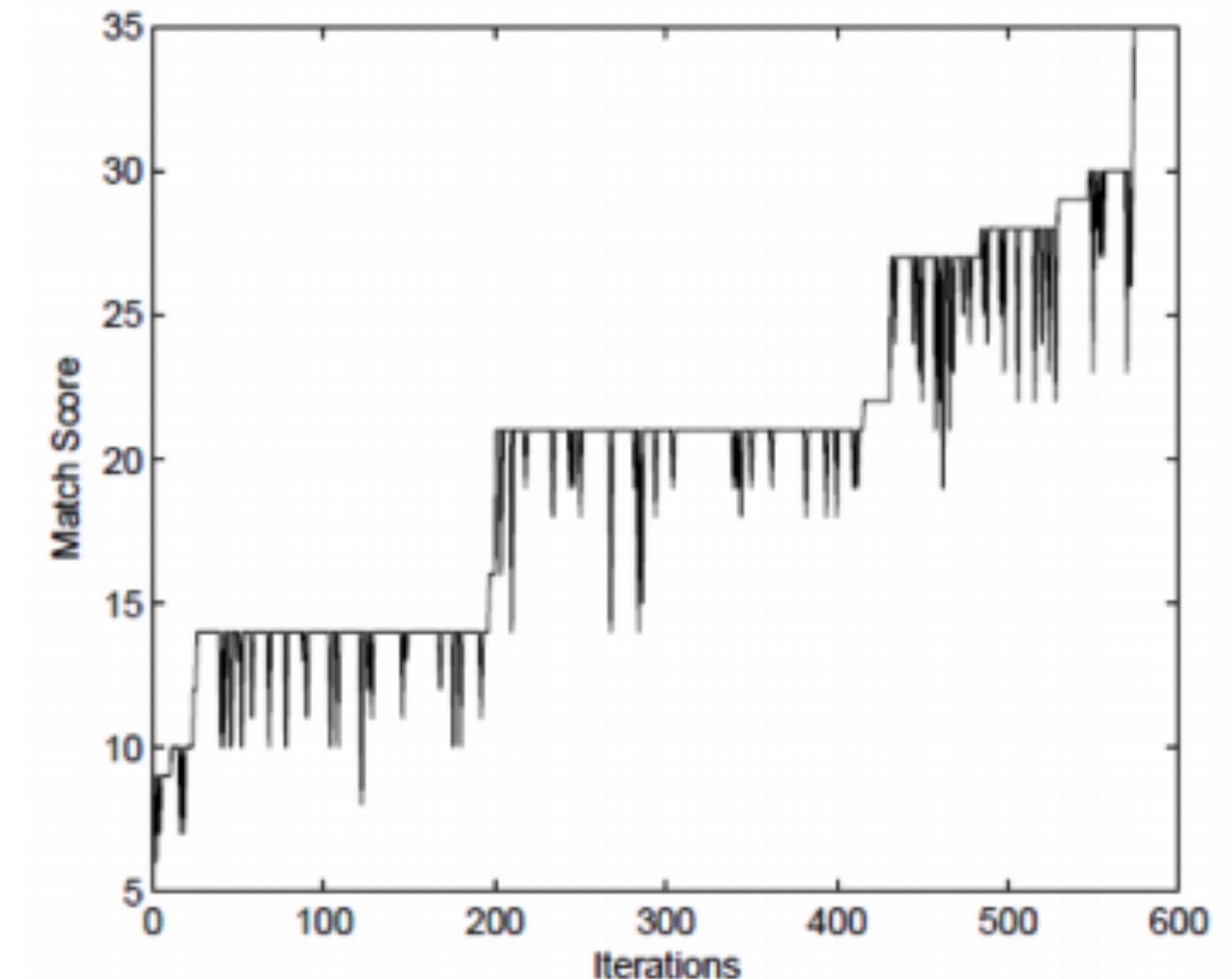
Data driven.

## Attack Examples

### Hill Climbing

The attacker iteratively provides synthetic adversarial samples to the system (at inference time).

At each iteration, the attacker observes how the output scores are progressing (gray-box attack).



Martinez-Diaz et al.  
*Hill-Climbing and Brute-Force Attacks on Biometric Systems: A Case Study in Match-on-Card Fingerprint Verification*  
IEEE ICCST, 2006



**LOYOLA**  
UNIVERSITY CHICAGO

# Adversarial Machine Learning

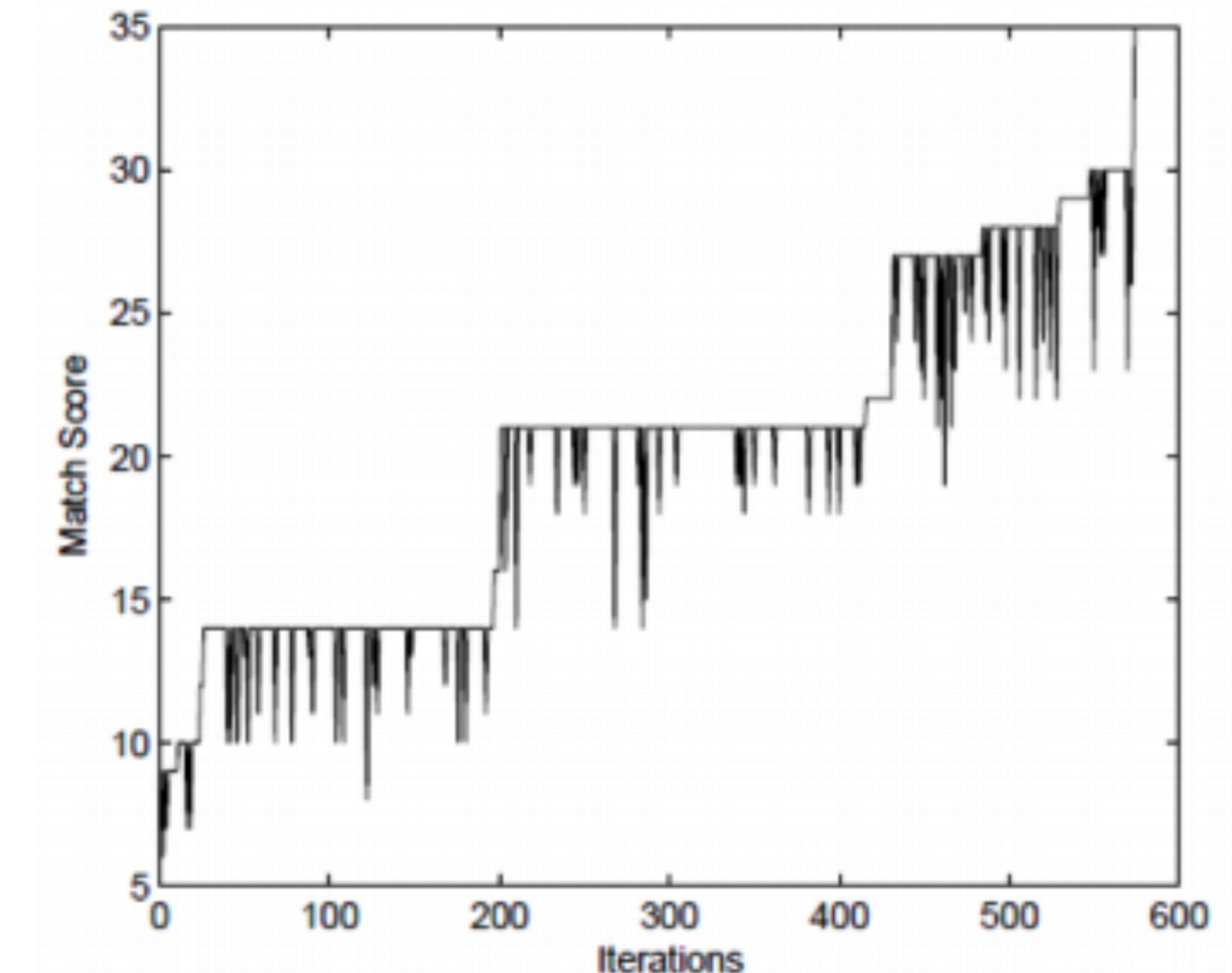
## Within Machine Learning

Data driven.

## Attack Examples

### Hill Climbing

With such progress feedback, the attacker can guide the generation of better and better synthetic data samples, up the point of trespassing the decision threshold.



Martinez-Diaz et al.  
*Hill-Climbing and Brute-Force Attacks on Biometric Systems: A Case Study in Match-on-Card Fingerprint Verification*  
IEEE ICCST, 2006



**LOYOLA**  
UNIVERSITY CHICAGO

# Adversarial Machine Learning

## Within Machine Learning

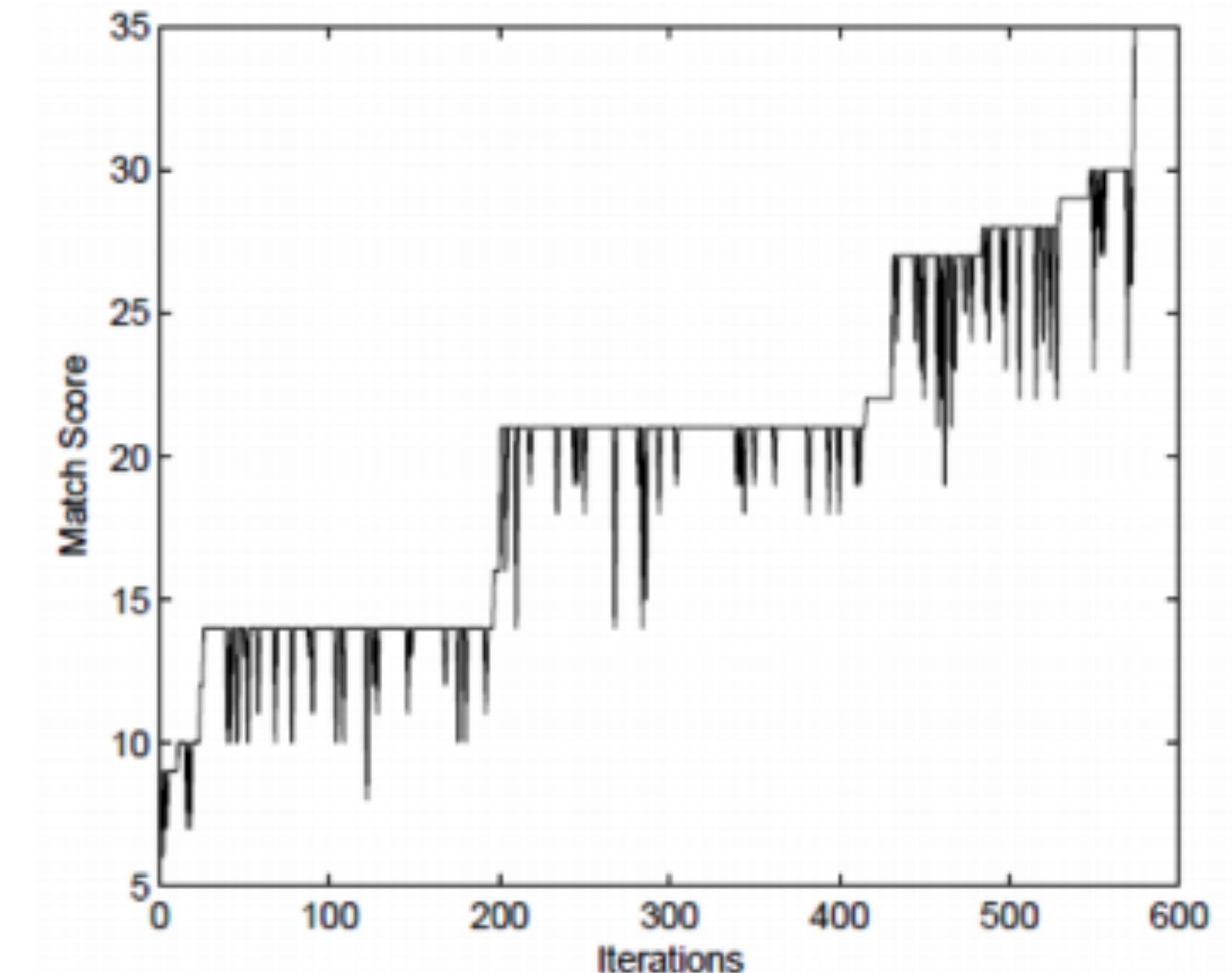
Data driven.

## Attack Examples

### Hill Climbing

With such progress feedback, the attacker can guide the generation of better and better synthetic data samples, up the point of trespassing the decision threshold.

One can make it fully data-driven with GANs.



Martinez-Diaz et al.  
*Hill-Climbing and Brute-Force Attacks on Biometric Systems: A Case Study in Match-on-Card Fingerprint Verification*  
IEEE ICCST, 2006



**LOYOLA**  
UNIVERSITY CHICAGO

# Adversarial Machine Learning

## Within Machine Learning

Data driven.

### Attack Examples

#### Master Key

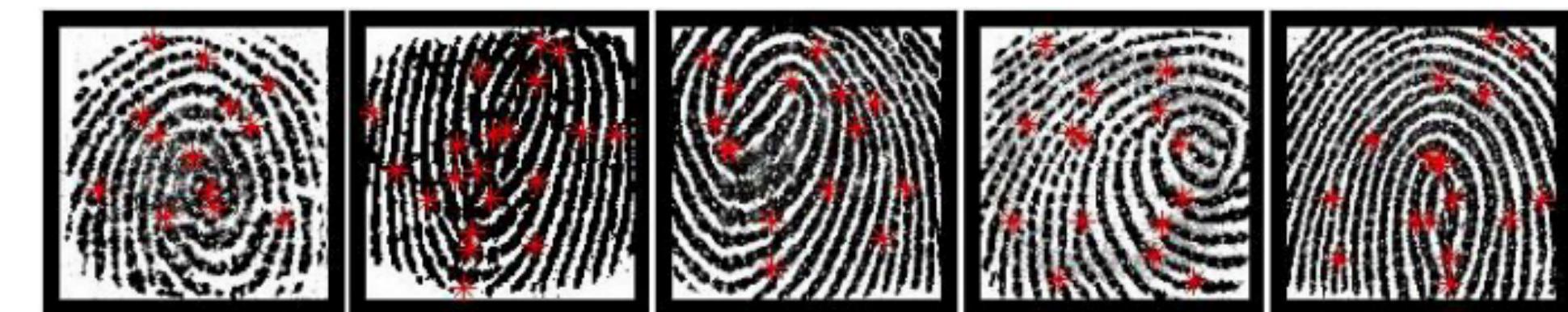
Similar to hill climbing, but the target is to generate an adversarial sample that matches multiple classes.

IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 12, NO. 9, SEPTEMBER 2017

2013

#### MasterPrint: Exploring the Vulnerability of Partial Fingerprint-Based Authentication Systems

Aditi Roy, *Student Member, IEEE*, Nasir Memon, *Fellow, IEEE*, and Arun Ross, *Senior Member, IEEE*



[https://www.cse.msu.edu/~rossarun/  
pubsRoyMemonRossMasterPrint\\_TIFS2017.pdf](https://www.cse.msu.edu/~rossarun/pubsRoyMemonRossMasterPrint_TIFS2017.pdf)

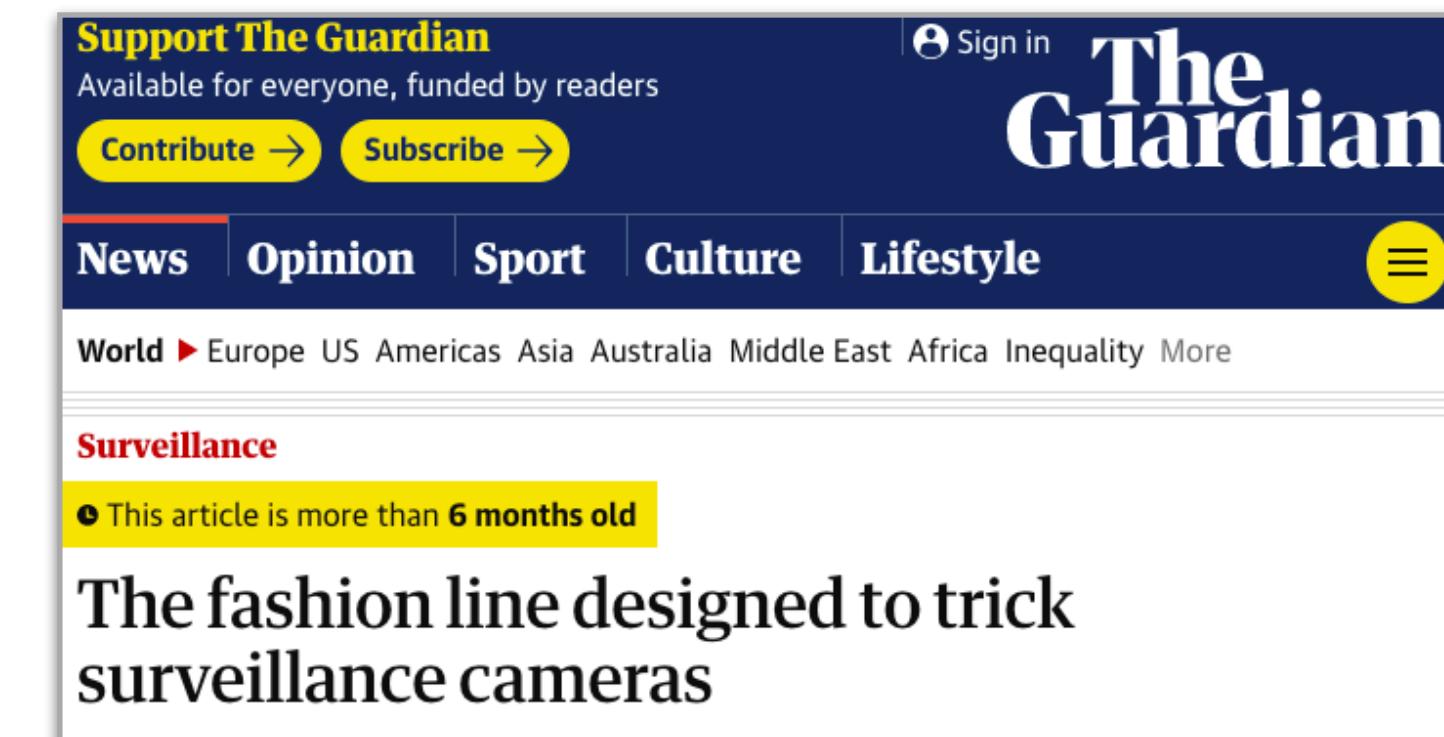
# Adversarial Machine Learning

## Within Machine Learning

Data driven.

### Attack Examples Denial of Service

The system is either fooled to mostly take its more intense processing path, or flooded with multiple input data to process.



The screenshot shows a news article from The Guardian. At the top, there are buttons for 'Support The Guardian' (Contribute and Subscribe), a sign-in link, and the 'The Guardian' logo. Below the header is a navigation bar with links for News, Opinion, Sport, Culture, and Lifestyle. A yellow banner indicates the article is 'more than 6 months old'. The main headline reads 'The fashion line designed to trick surveillance cameras'. The URL of the article is provided at the bottom: <https://www.theguardian.com/world/2019/aug/13/the-fashion-line-designed-to-trick-surveillance-cameras>.



# Adversarial Machine Learning

## Within Machine Learning

Data driven.

### Attack Examples

**Data Poisoning** (or Backdoor, or Trojan)

Mislabeled adversarial data are covertly included among the training samples.



Gu et al.

*BadNets: Evaluating Backdooring Attacks on Deep Neural Networks*

IEEE Access, 2019

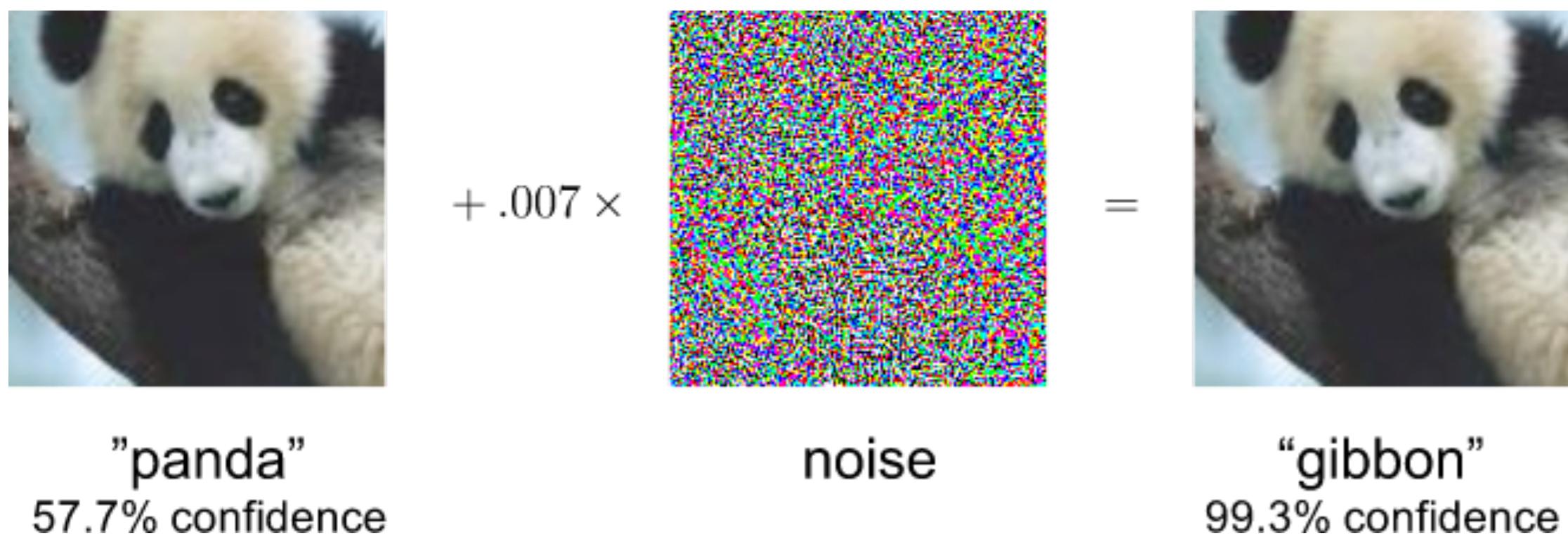
<https://ieeexplore.ieee.org/document/8685687>

# Adversarial Attacks on DL

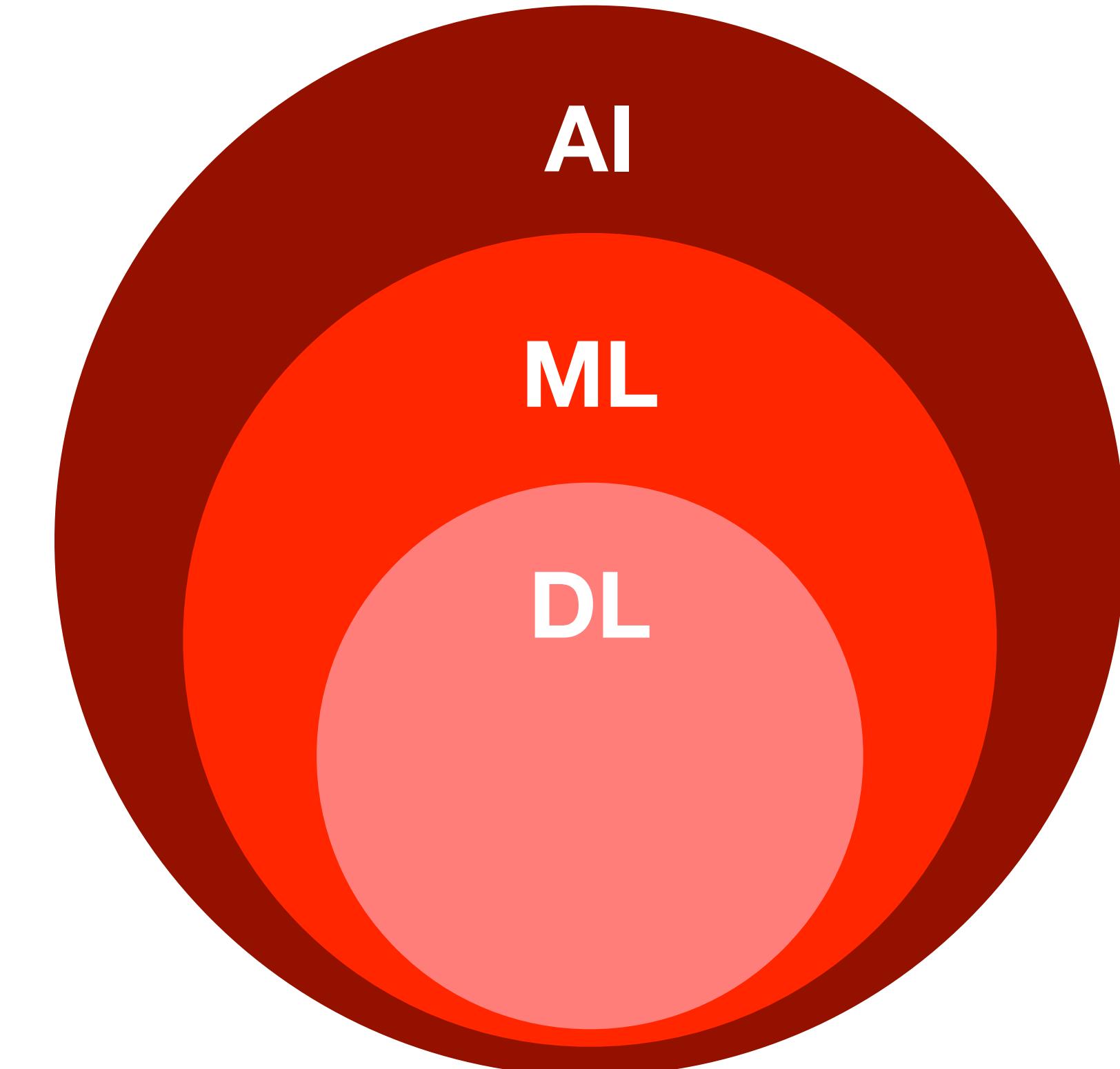
## Harnessing Adversarial Examples

### Objective

Learn how to add imperceptible noise to input data in a way that changes the model's class prediction, causing misclassification.



<https://arxiv.org/pdf/1412.6572.pdf>



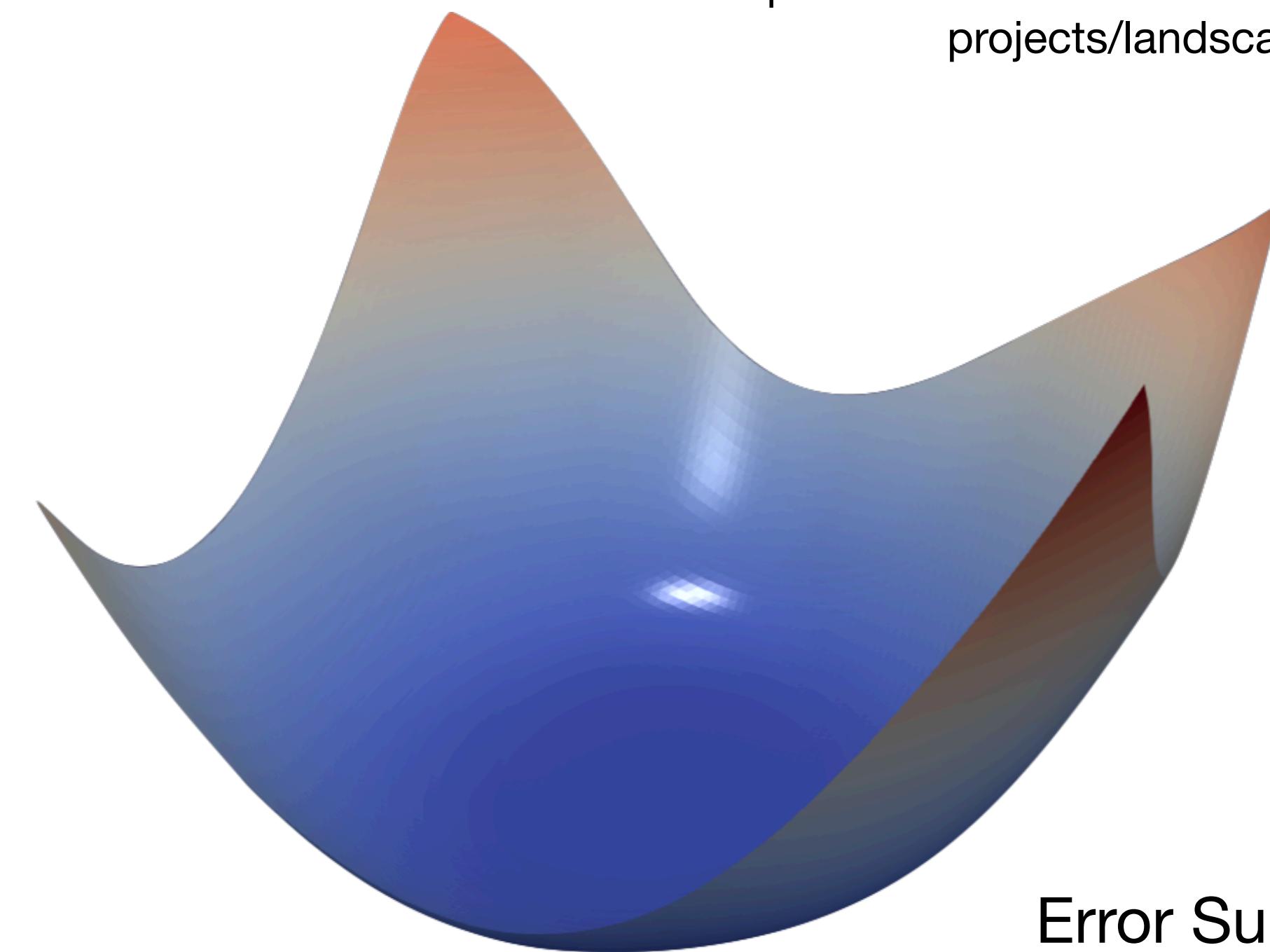
# Adversarial Attacks on DL

**Harnessing Adversarial Examples**

[https://www.cs.umd.edu/~tomg/  
projects/landscapes/](https://www.cs.umd.edu/~tomg/projects/landscapes/)

**Possible Solution**

***Fast Gradient Sign Method (FGSM)***



Error Surface

# Adversarial Attacks on DL

## Harnessing Adversarial Examples

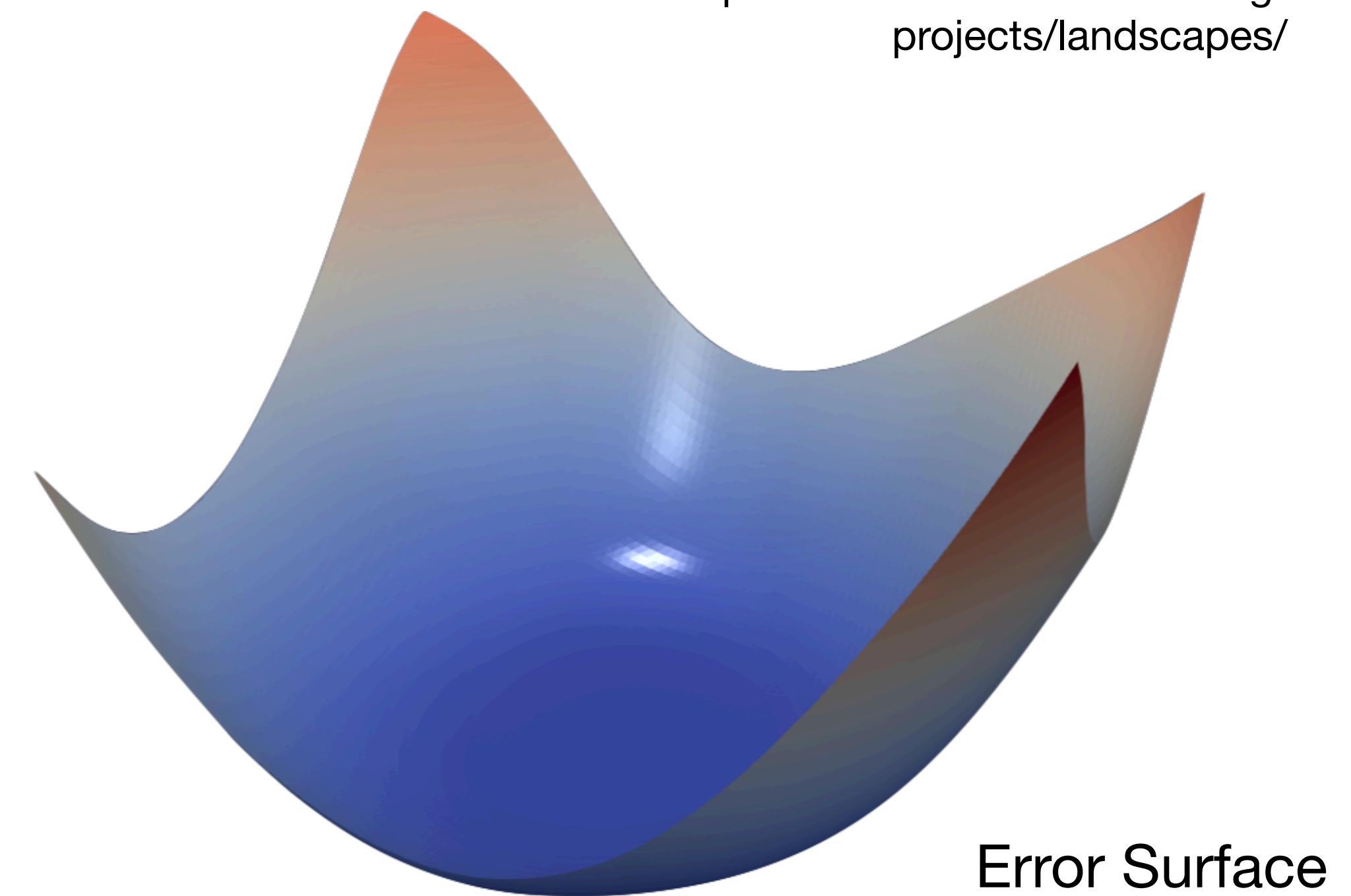
[https://www.cs.umd.edu/~tomg/  
projects/landscapes/](https://www.cs.umd.edu/~tomg/projects/landscapes/)

## Possible Solution

### *Fast Gradient Sign Method (FGSM)*

## Typical Training Solution

Minimize the error (actual label, predicted label) according to the training data batch by “walking” on the error surface to the opposite direction of the error gradient.



Error Surface

# Adversarial Attacks on DL

## Harnessing Adversarial Examples

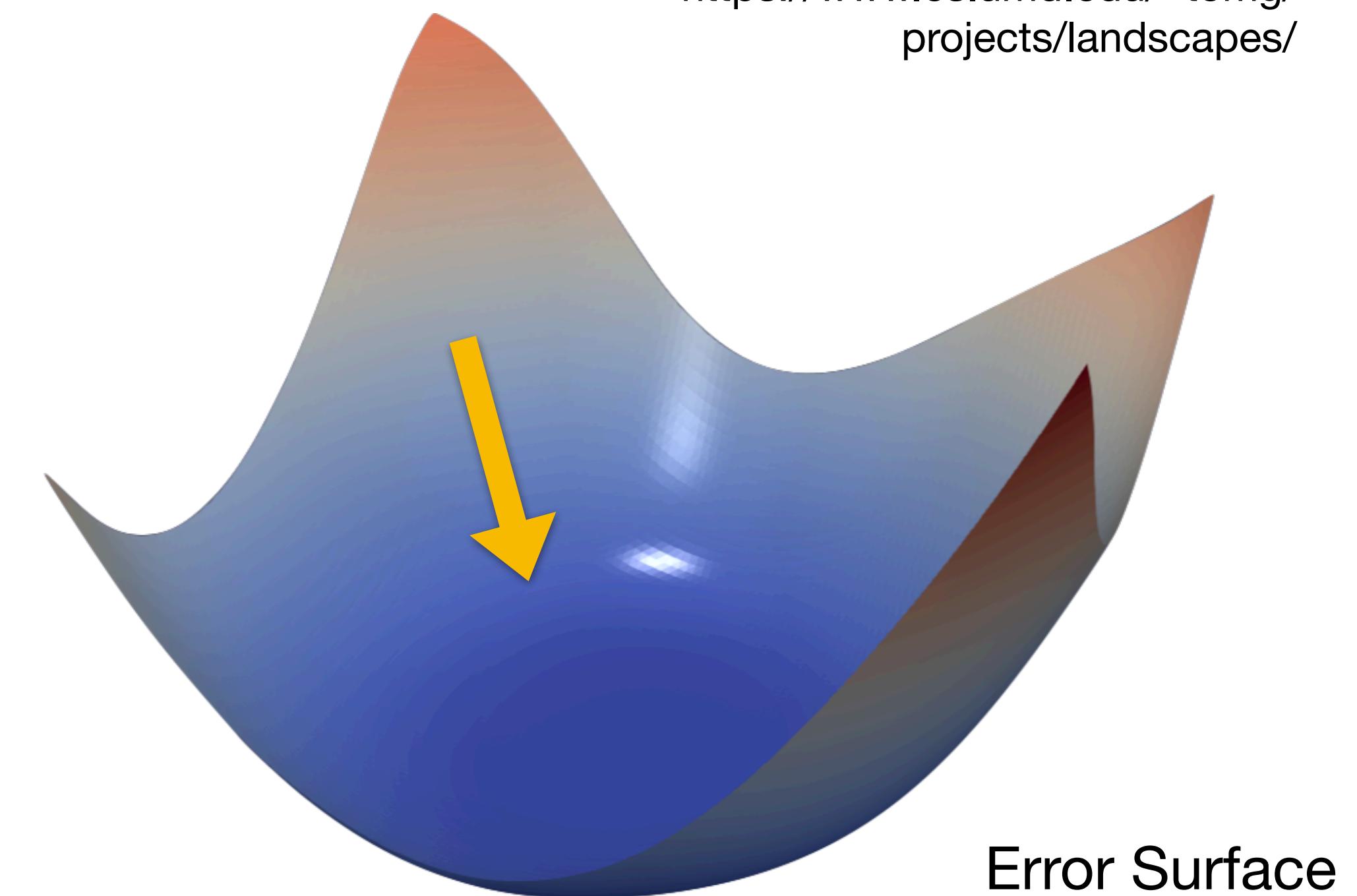
[https://www.cs.umd.edu/~tomg/  
projects/landscapes/](https://www.cs.umd.edu/~tomg/projects/landscapes/)

### Possible Solution

#### *Fast Gradient Sign Method (FGSM)*

### Typical Training Solution

Minimize the error (actual label, predicted label) according to the training data batch by “walking” on the error surface to the opposite direction of the error gradient.



$$\text{new\_weight} = \text{current\_weight} - \text{learning\_rate} \times \text{sign}(\nabla_{\text{batch\_error}})$$

# Adversarial Attacks on DL

## Harnessing Adversarial Examples

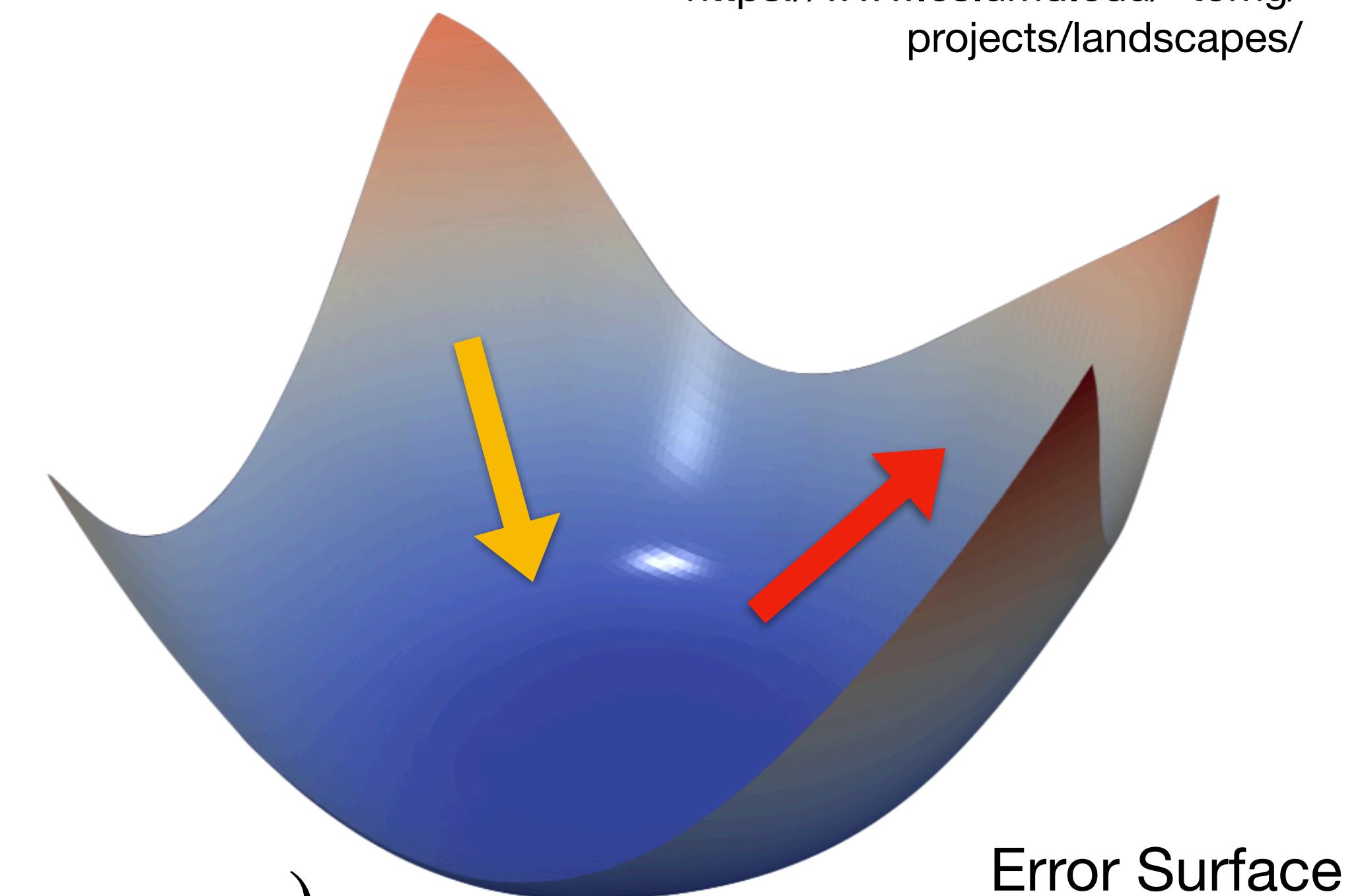
[https://www.cs.umd.edu/~tomg/  
projects/landscapes/](https://www.cs.umd.edu/~tomg/projects/landscapes/)

### Possible Solution

#### *Fast Gradient Sign Method (FGSM)*

##### **FGSM**

Given one sample (that will be the adversarial data), maximize the error by “walking” on the error surface to the direction of the error gradient.



$$\text{new\_sample} = \text{current\_sample} + \epsilon \times \text{sign}(\nabla_{\text{new\_sample}})$$

# Adversarial Attacks on DL

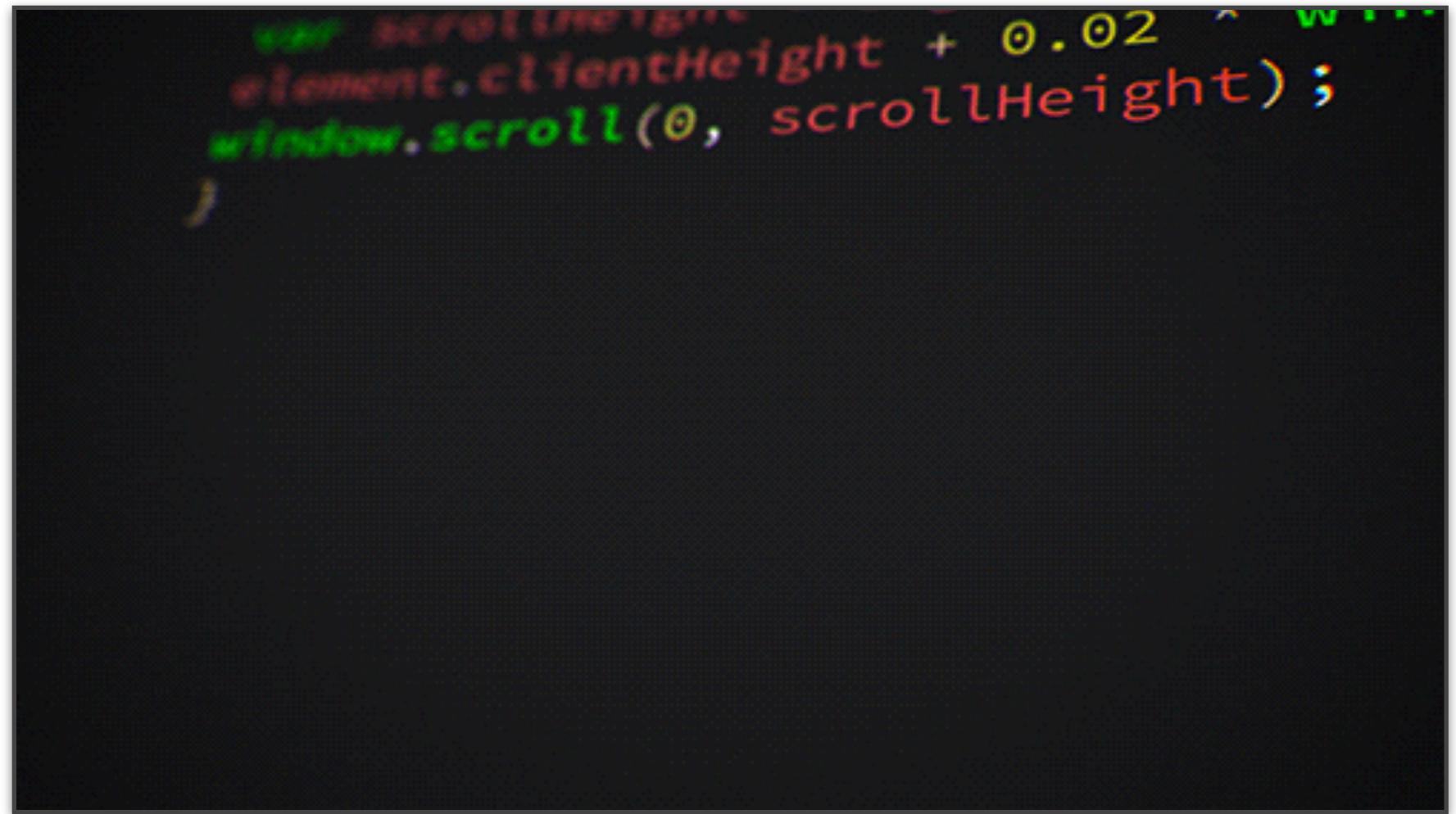
## Harnessing Adversarial Examples

### Possible Solution

#### *Fast Gradient Sign Method (FGSM)*

### Algorithm

1. Forward-propagate sample through the network.
2. Compute the (actual label, predicted label) error.
3. Back-propagate the error gradient to the sample.
4. Tweak the sample features in the direction that maximizes the error by adding “noise”.



# Adversarial Attacks on DL

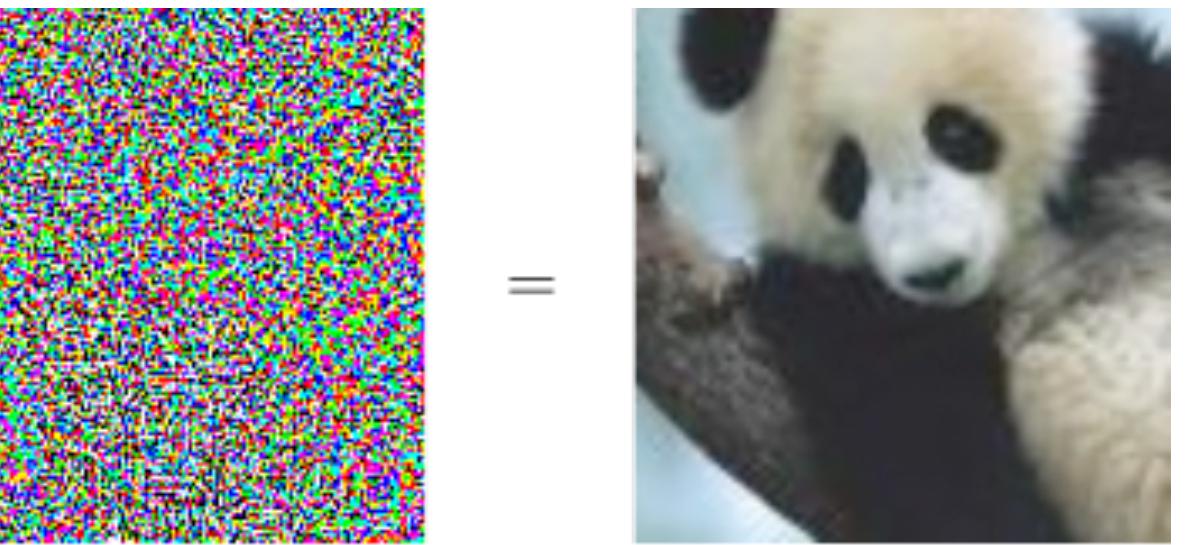
## Harnessing Adversarial Examples

### Possible Solution

#### *Fast Gradient Sign Method (FGSM)*

### Algorithm

1. Forward-propagate sample through the network.
2. Compute the (actual label, predicted label) error.
3. Back-propagate the error gradient to the sample.
4. Tweak the sample features in the direction that maximizes the error by adding “noise”.



"panda"  
57.7% confidence

noise

"gibbon"  
99.3% confidence

<https://arxiv.org/pdf/1412.6572.pdf>

$\epsilon \times \text{sign}(\nabla_{\text{new\_sample}})$ : noise.

$\text{sign}(\nabla_{\text{new\_sample}})$ : direction that maximizes the error.

$\epsilon$ : control the perceptive-mislabel trade-off.

# Practical Activity 2

## Work in Pairs

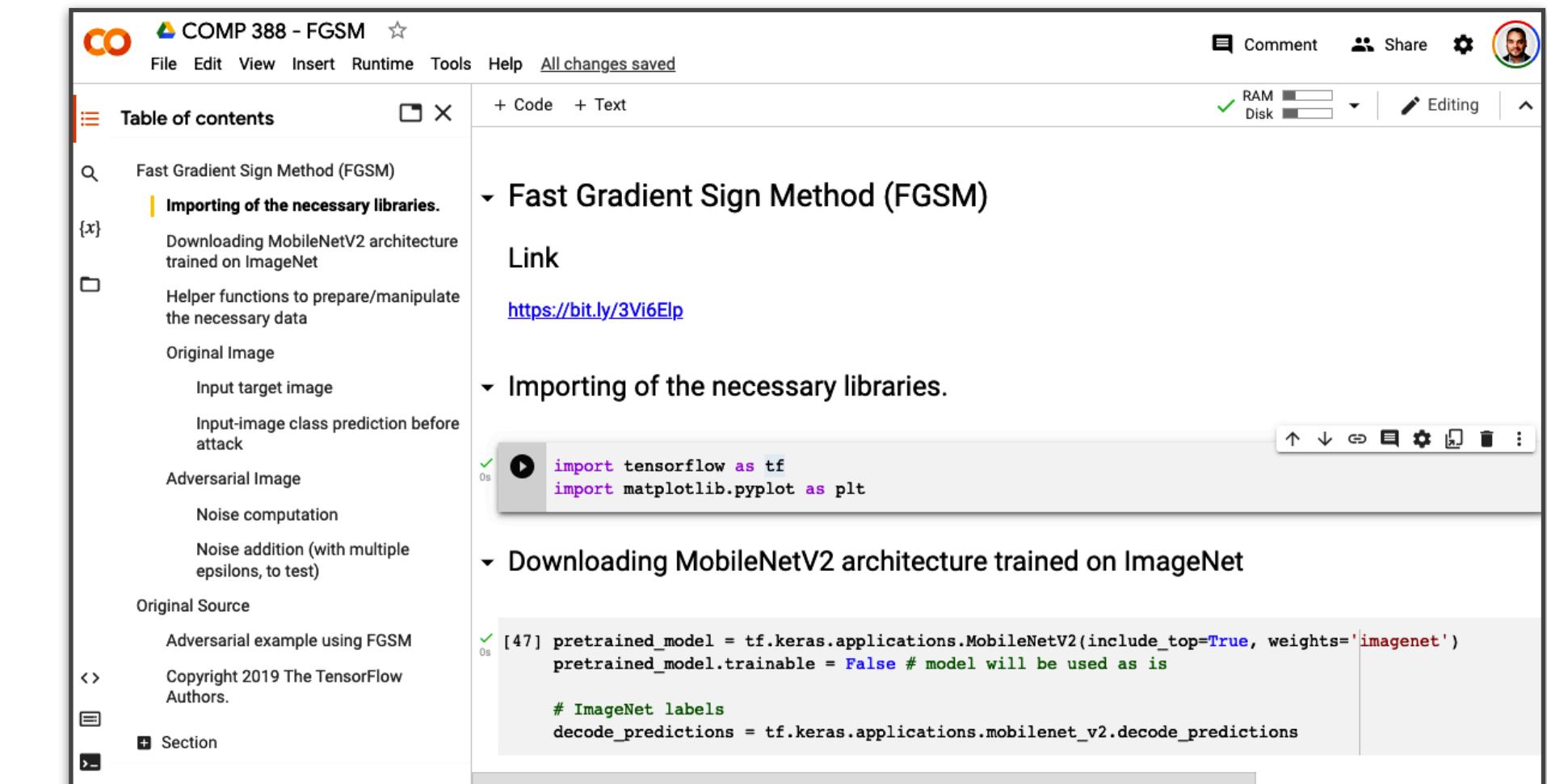
Use Google Colab at <https://bit.ly/3Vi6Elp>

Observe the epsilon's perceptive-mislabel trade-off.

Try with different images  
(source: <https://bit.ly/3OqEMsP>).

Contributions or Question?

1. <https://arxiv.org/abs/1801.07698>



The screenshot shows a Google Colab notebook titled "COMP 388 - FGSM". The left sidebar contains a "Table of contents" with sections like "Fast Gradient Sign Method (FGSM)", "Importing of the necessary libraries.", "Downloading MobileNetV2 architecture trained on ImageNet", "Helper functions to prepare/manipulate the necessary data", "Original Image", "Input target image", "Input-image class prediction before attack", "Adversarial Image", "Noise computation", "Noise addition (with multiple epsilons, to test)", "Original Source", "Adversarial example using FGSM", and "Copyright 2019 The TensorFlow Authors.". The main area shows code cells. The first cell, which is currently executing, contains:`import tensorflow as tf
import matplotlib.pyplot as plt`

```
The second cell, labeled [47], contains:
```

`pretrained_model = tf.keras.applications.MobileNetV2(include_top=True, weights='imagenet')
pretrained_model.trainable = False # model will be used as is

# ImageNet labels
decode_predictions = tf.keras.applications.mobilenet_v2.decode_predictions`

# Adversarial Machine Learning

## Discussion Time

### FGSM

What type of attack is FGSM  
(e.g., evasion, white box)?

It worked on MobileNet;  
would it work on other architectures?

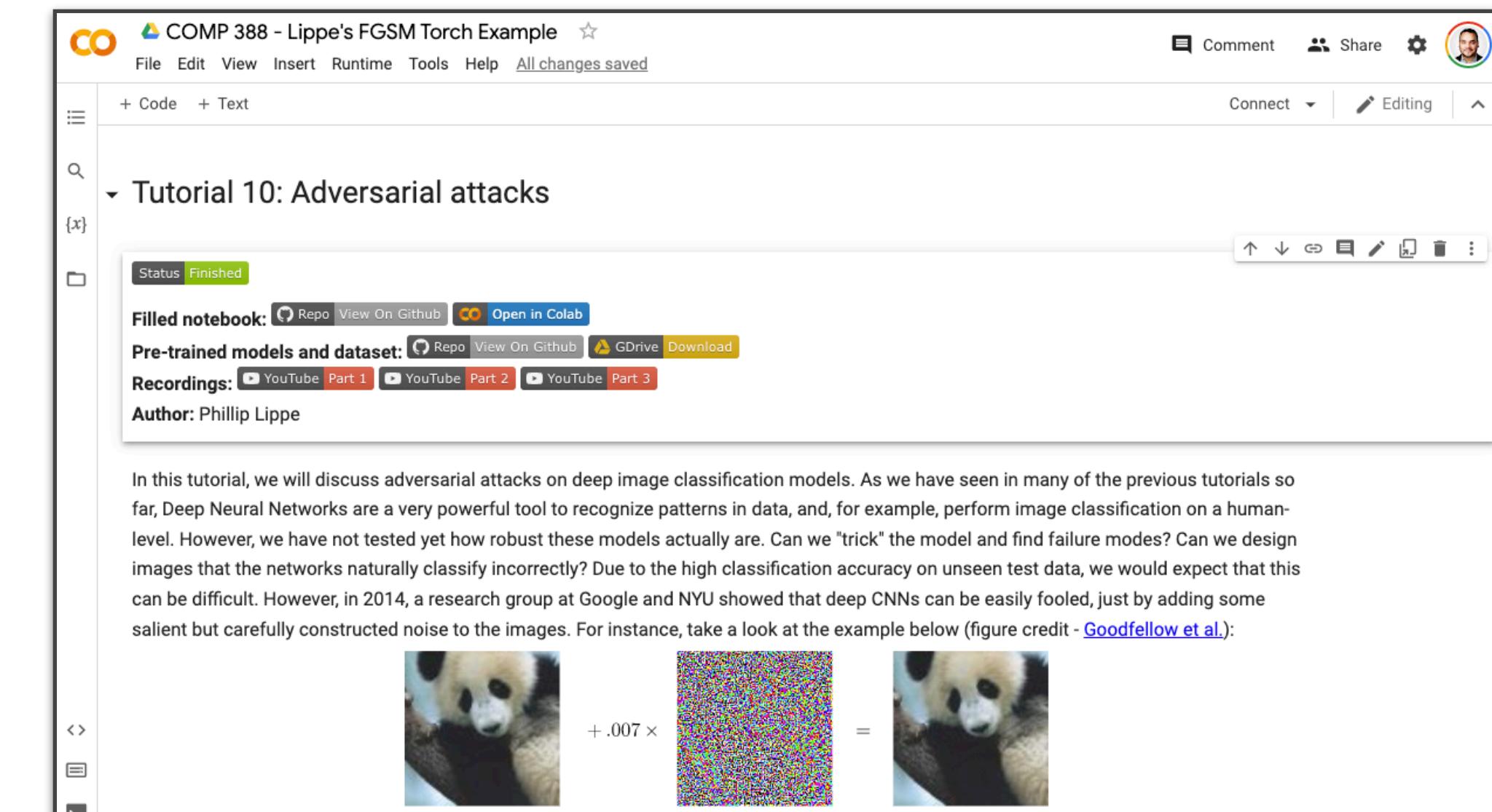


# PyTorch Example

## Watch it Later

Use Google Colab at  
<https://bit.ly/3V5VW1c>

Phillip Lippe's PyTorch example  
FGSM attack on ResNet-34  
Other attack methods are available



1. <https://arxiv.org/abs/1801.07698>

# Adversarial Machine Learning

## Discussion Time

### FGSM

What type of attack is FGSM  
(e.g., evasion, white box)?

It worked on MobileNet;  
would it work on other architectures?

### Defenses

Procedures to avoid, detect,  
or mitigate these attacks?



# Deep Fake Detection

## Discussion Time

**Can a GAN-generated image be detected?**

Yes, for now...

The discriminator always wins (the generator).

## Open Problems

Cheap fake detection.

GAN architecture attribution.

GAN in the wild.

