ORCA Project

Towards the Accountable Learning-enabled Autonomous Systems.

Is deep learning secure for robots?

Presenter: Han Wu, Dr Wenjie Ruan

Han Wu

Ph.D. Student
University of Exeter

Syed Yunas

Postdoc
University of Exeter

Dr. Wenjie Ruan

Senior Lecturer
University of Exeter

Exeter Trustworthy AI Lab
https://trustai.uk/

Is deep learning secure for robots?

- Background
- Project 1: Adversarial Driving
- Project 2: Adversarial Detection

Background – Robotics

Advances in deep neural networks have opened a new era of robotics, **intelligent robots**.

(a) Amazon Kiva Robot

(b) Alibaba Quicktron Robot

Intelligent robots possess a more comprehensive **perception** of environments.

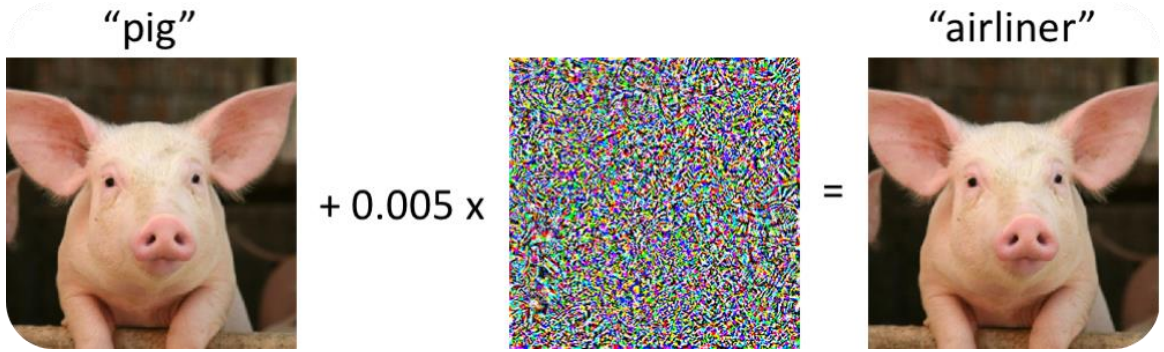
(a) Waymo (formerly Google self-driving project)

(b) Tesla Autopilot

Deep Learning for Autonomous Driving

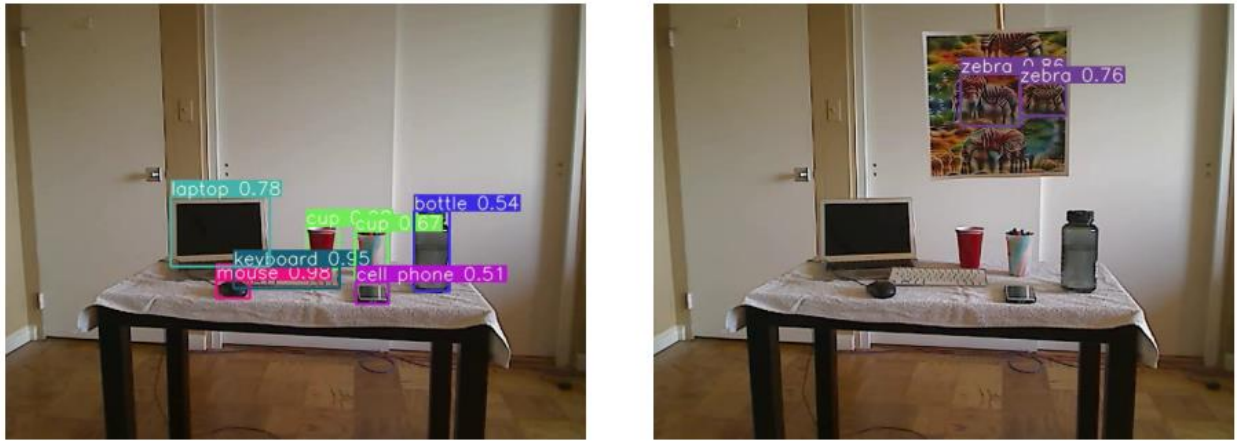
Background – Deep Learning

Deep neural networks are vulnerable to **adversarial attacks** in various tasks.



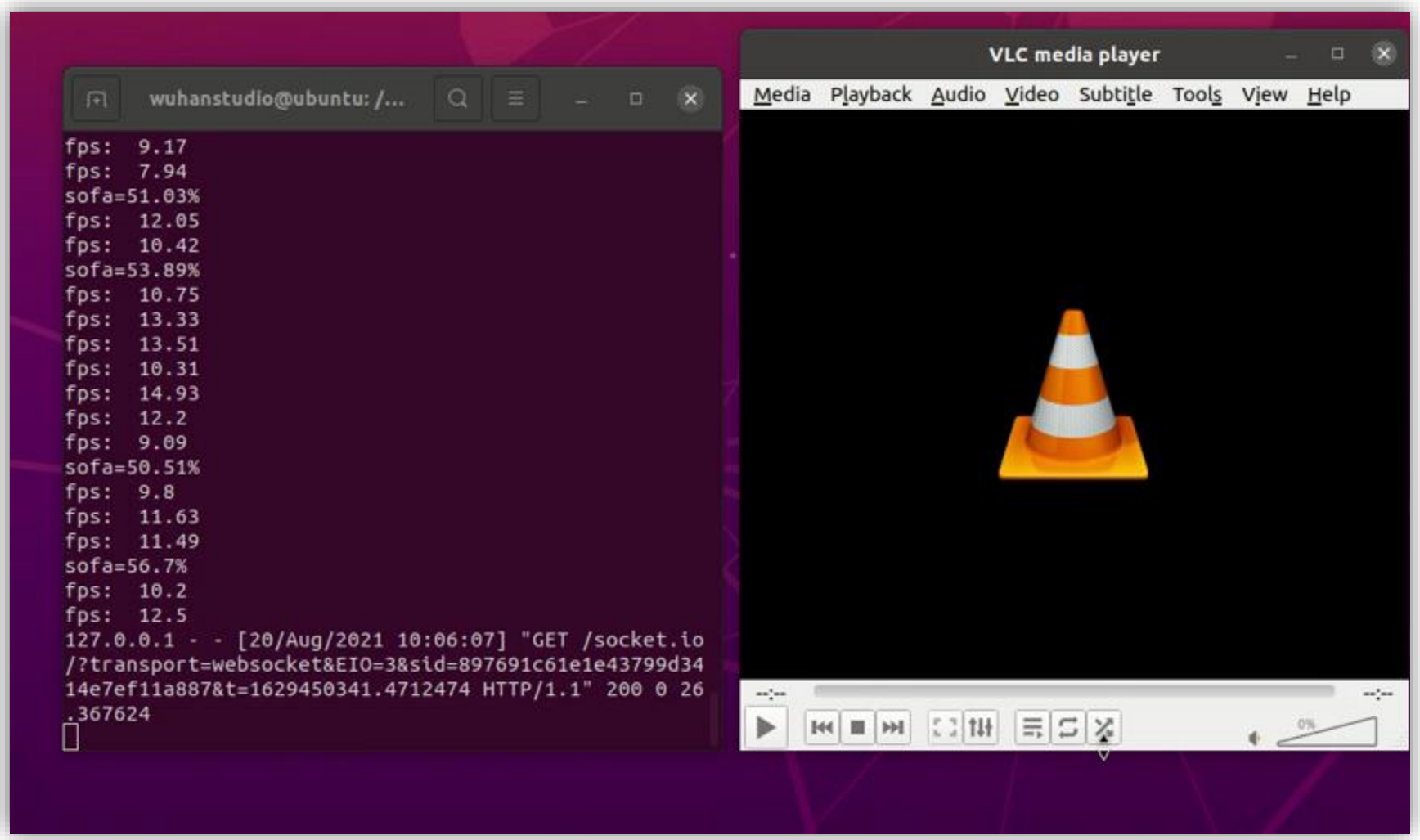
Adversarial attacks against image classification

Instead of **minimizing** the loss function, the adversarial attack **maximizes** it.



Adversarial attacks against object detection.

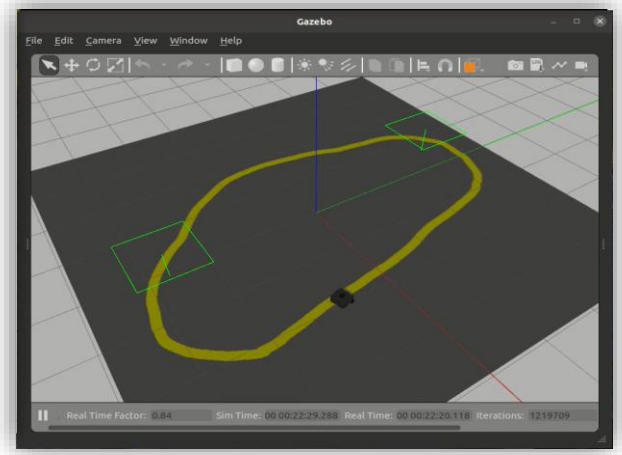
Demo – Adversarial Filter

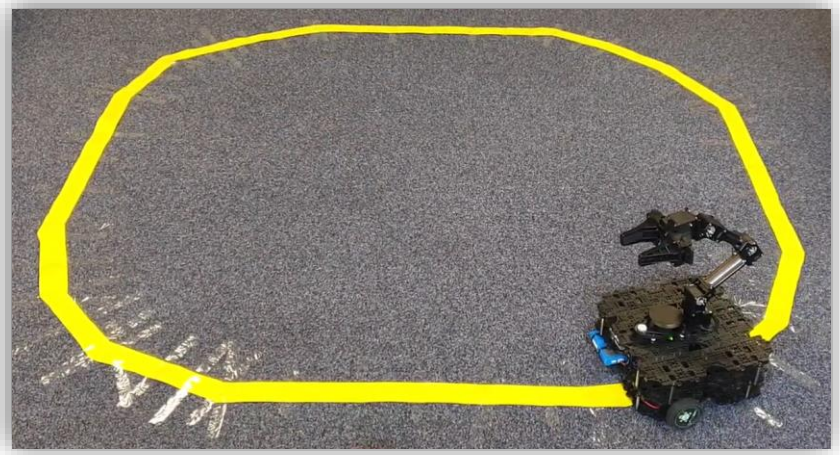


A fake camera that fools the object detection model.

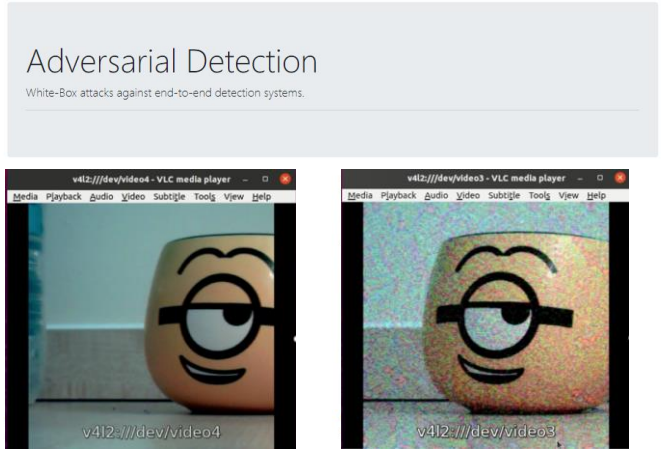
Overview

Adversarial Driving

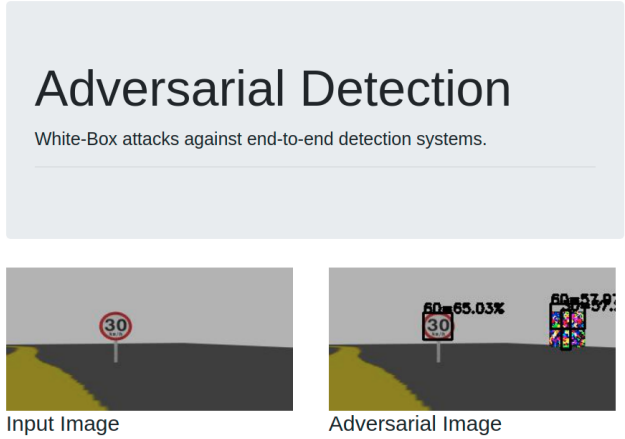




Adversarial ROS Driving

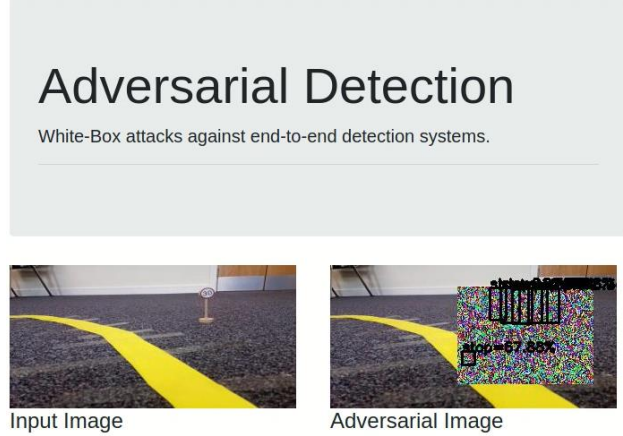


Adversarial Detection



Input Image

Adversarial Image



Input Image

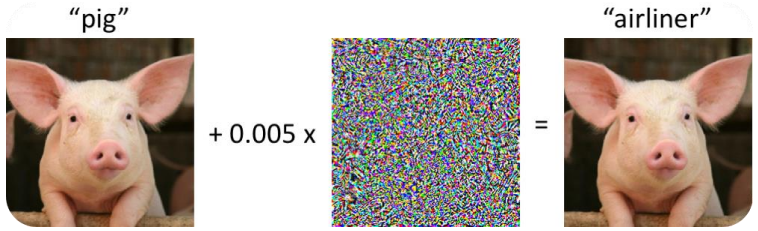
Adversarial Image

Adversarial ROS Detection

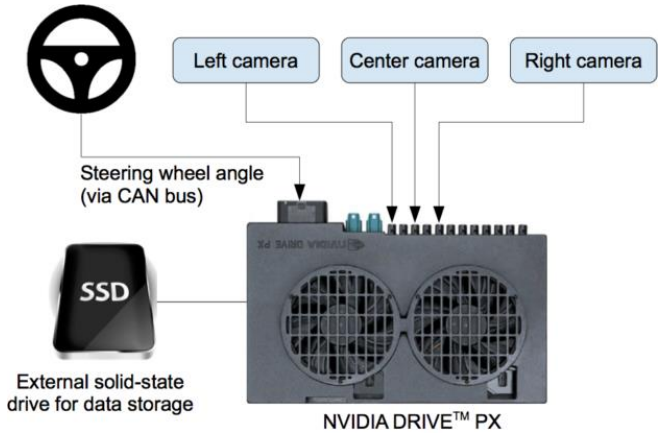
Project 1: Adversarial Driving

Project 1: Adversarial Driving[4]

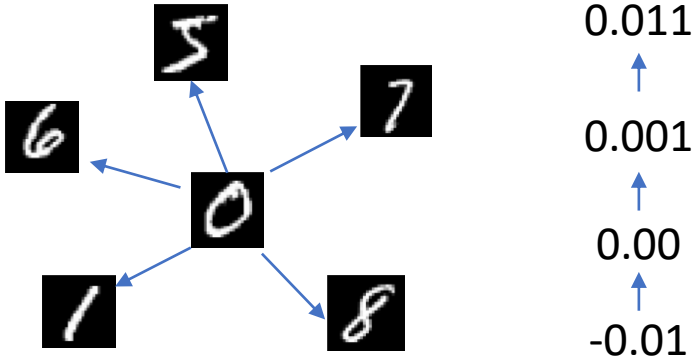
White-Box Adversarial Attack against Autonomous **Driving**



Fast Gradient Sign Method (FGSM)



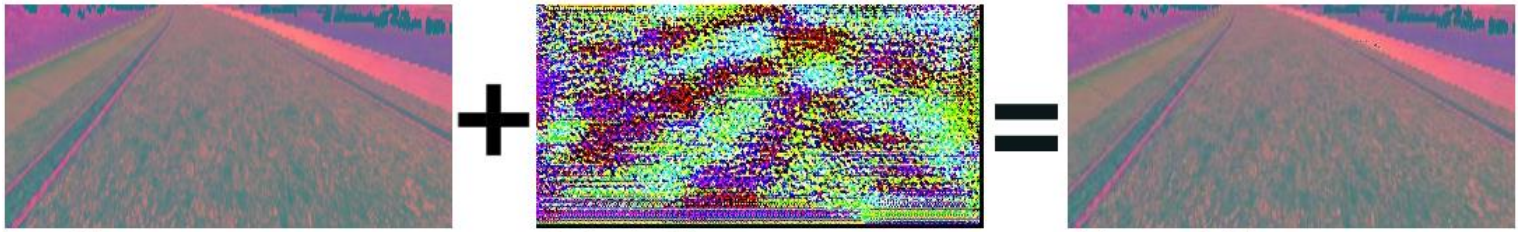
Nvidia end to end self driving



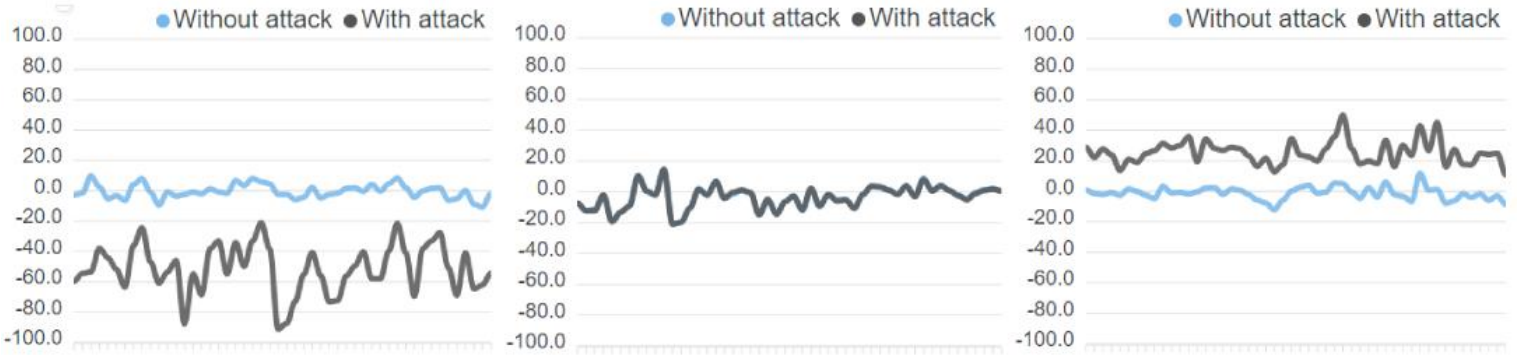
Classification

Regression

Camera Image



Input Image | Perturbation | Adversarial Image

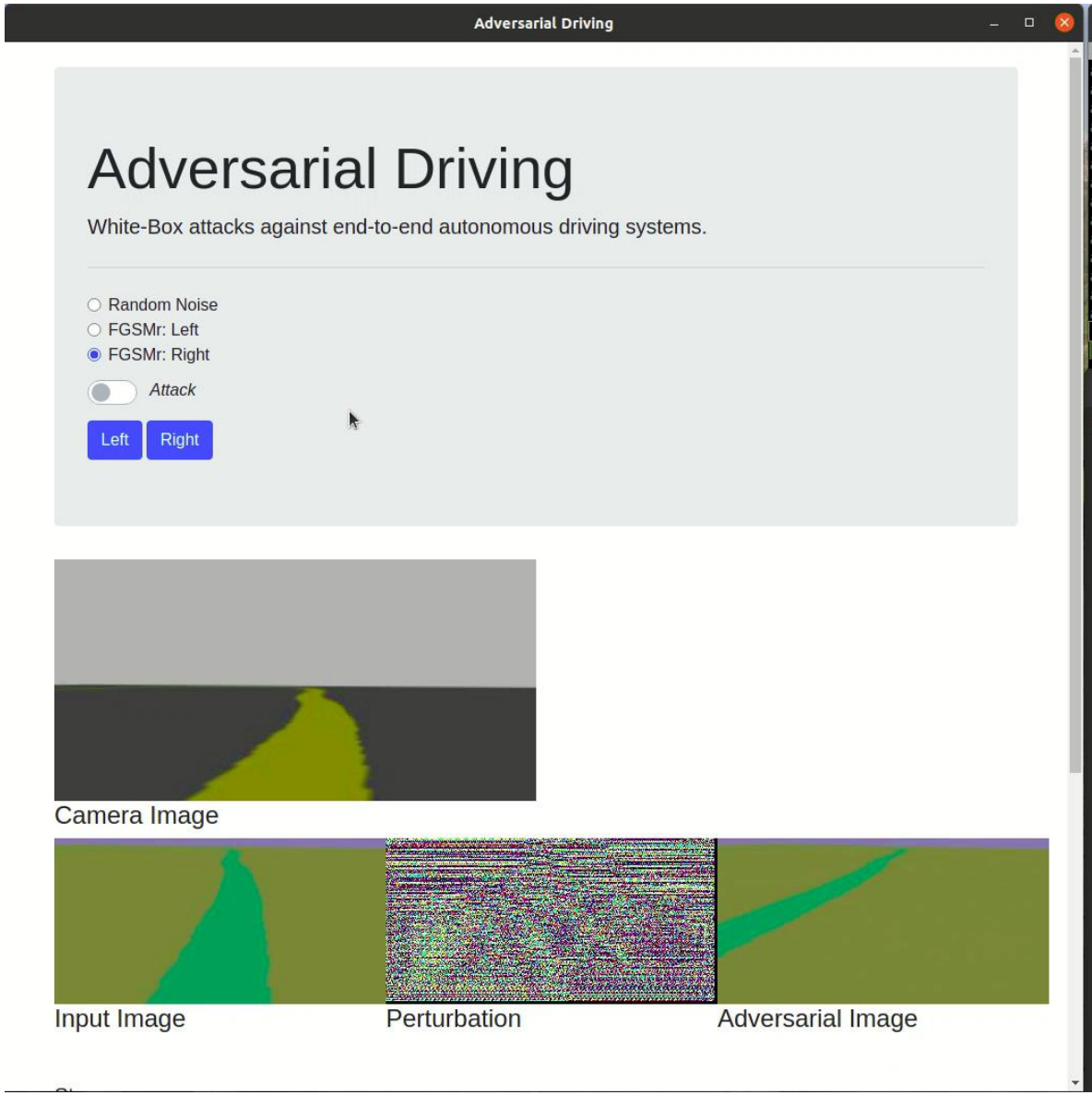


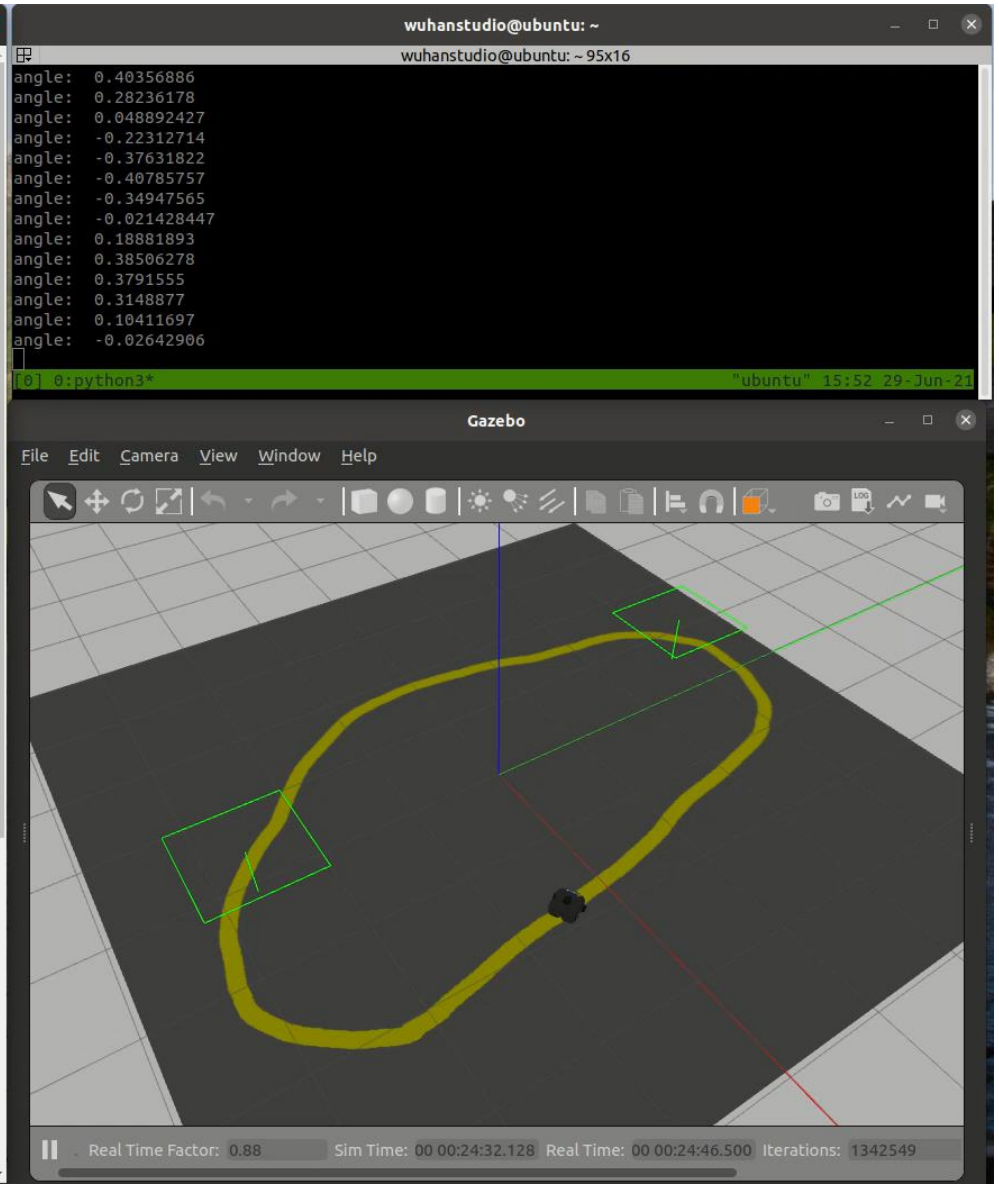
Attack to the Left (Decreasing) | Random Noise | Attack to the right (Increasing)

FGSM: $\eta = \epsilon\ sign(\nabla_x \boldsymbol{J}(\theta, x, \boldsymbol{y}))$

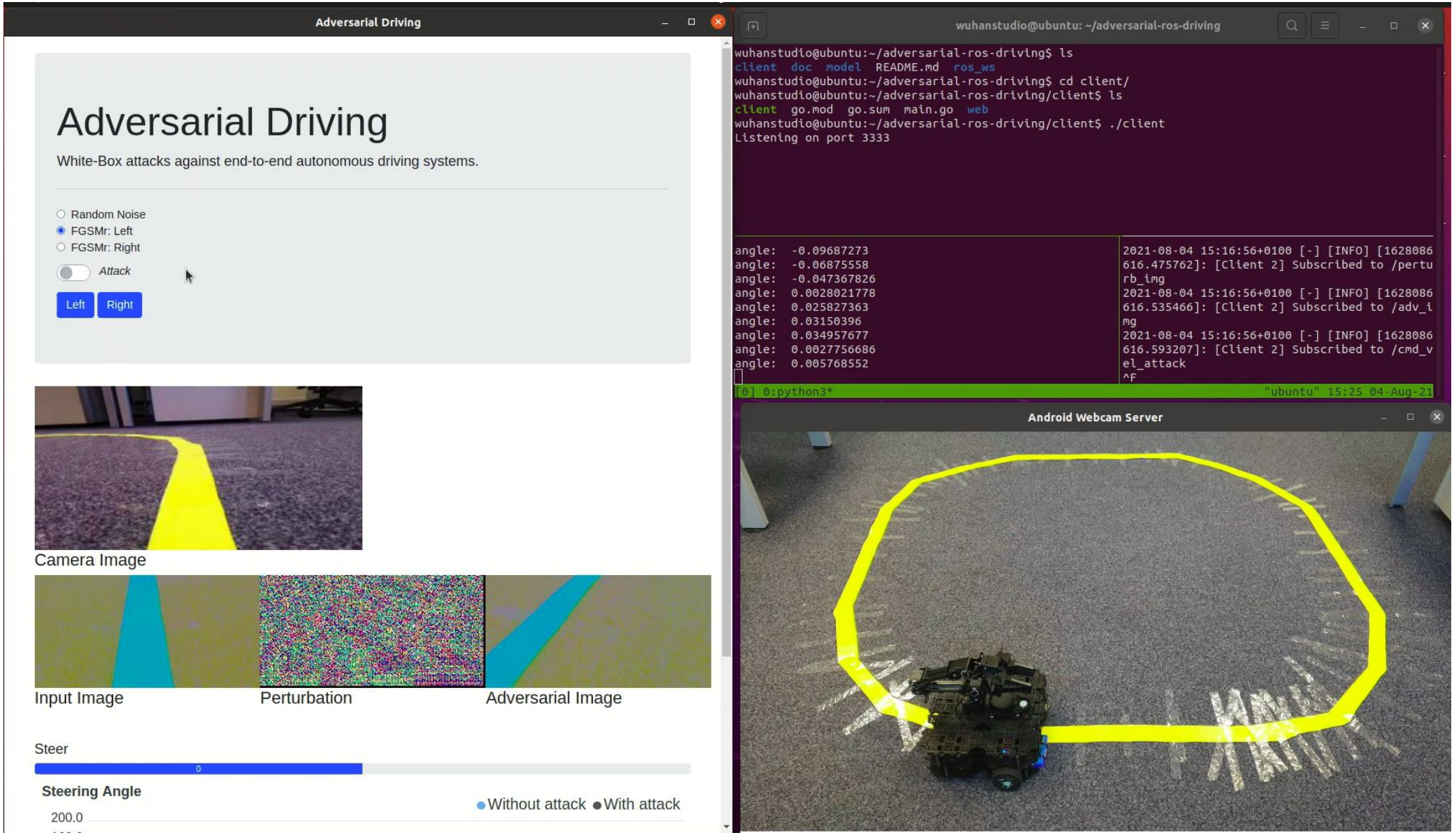
FGSMr: $\eta = \epsilon\ sign(\nabla_x \pm \boldsymbol{f}(\theta, x))$

Project 1: Adversarial ROS Driving



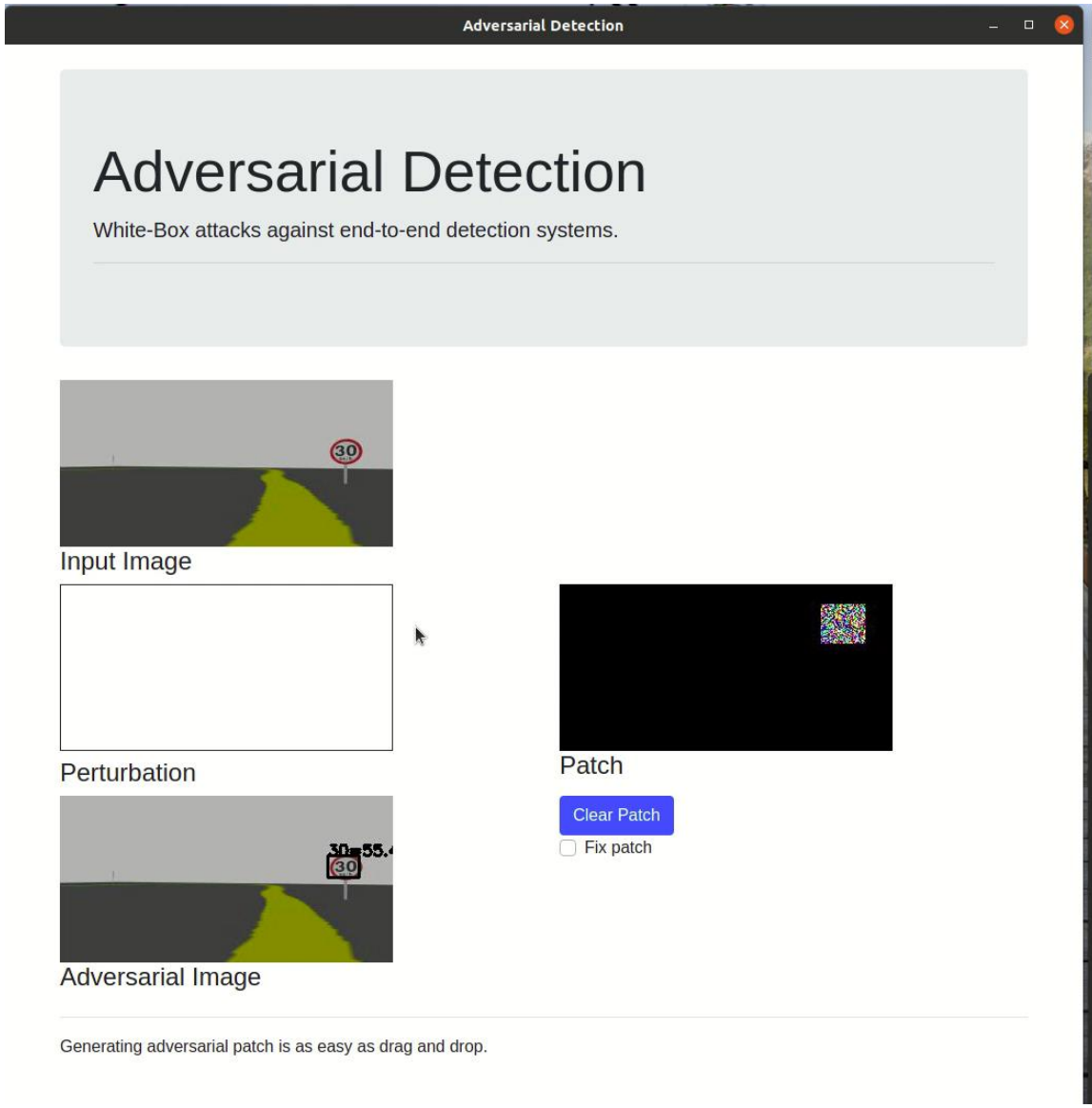


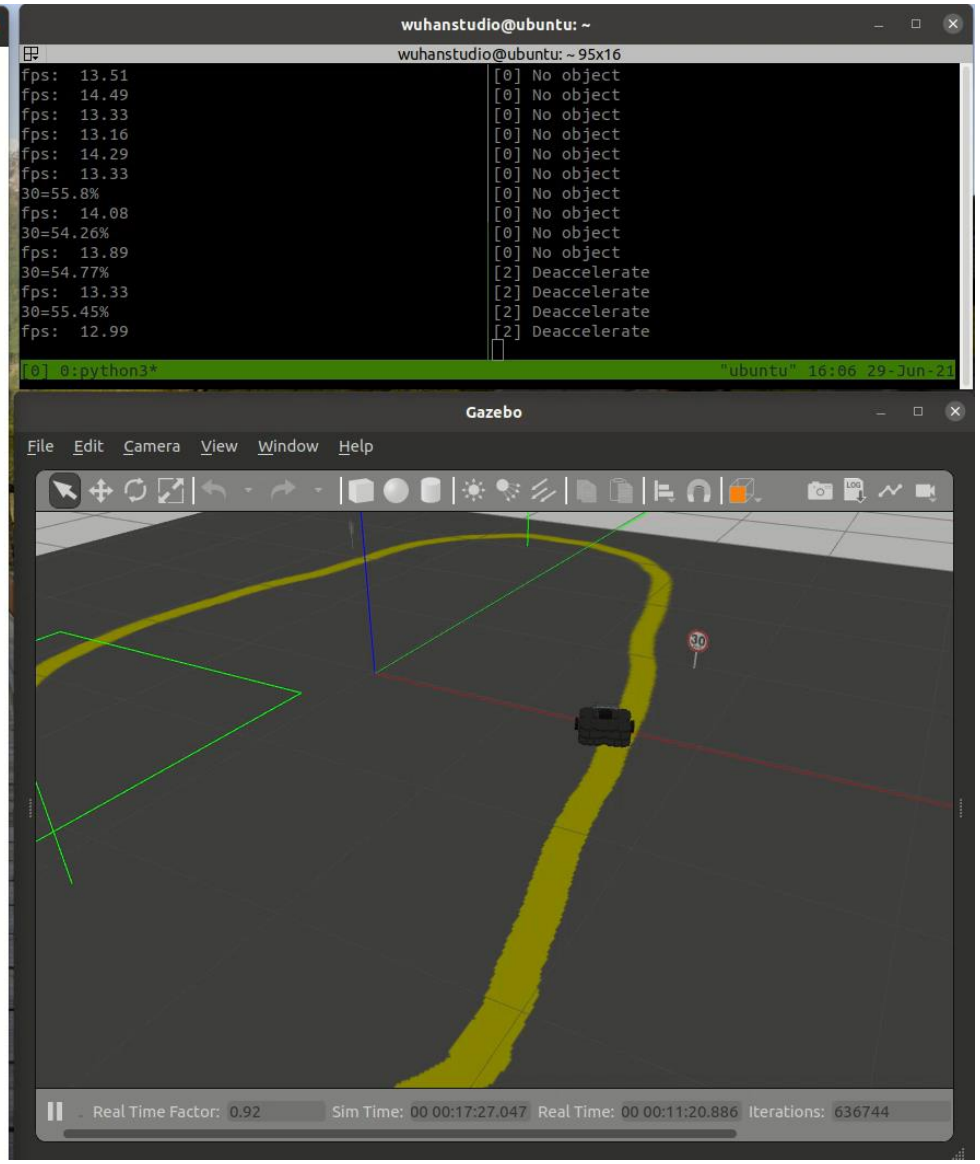
Project 1: Adversarial ROS Driving



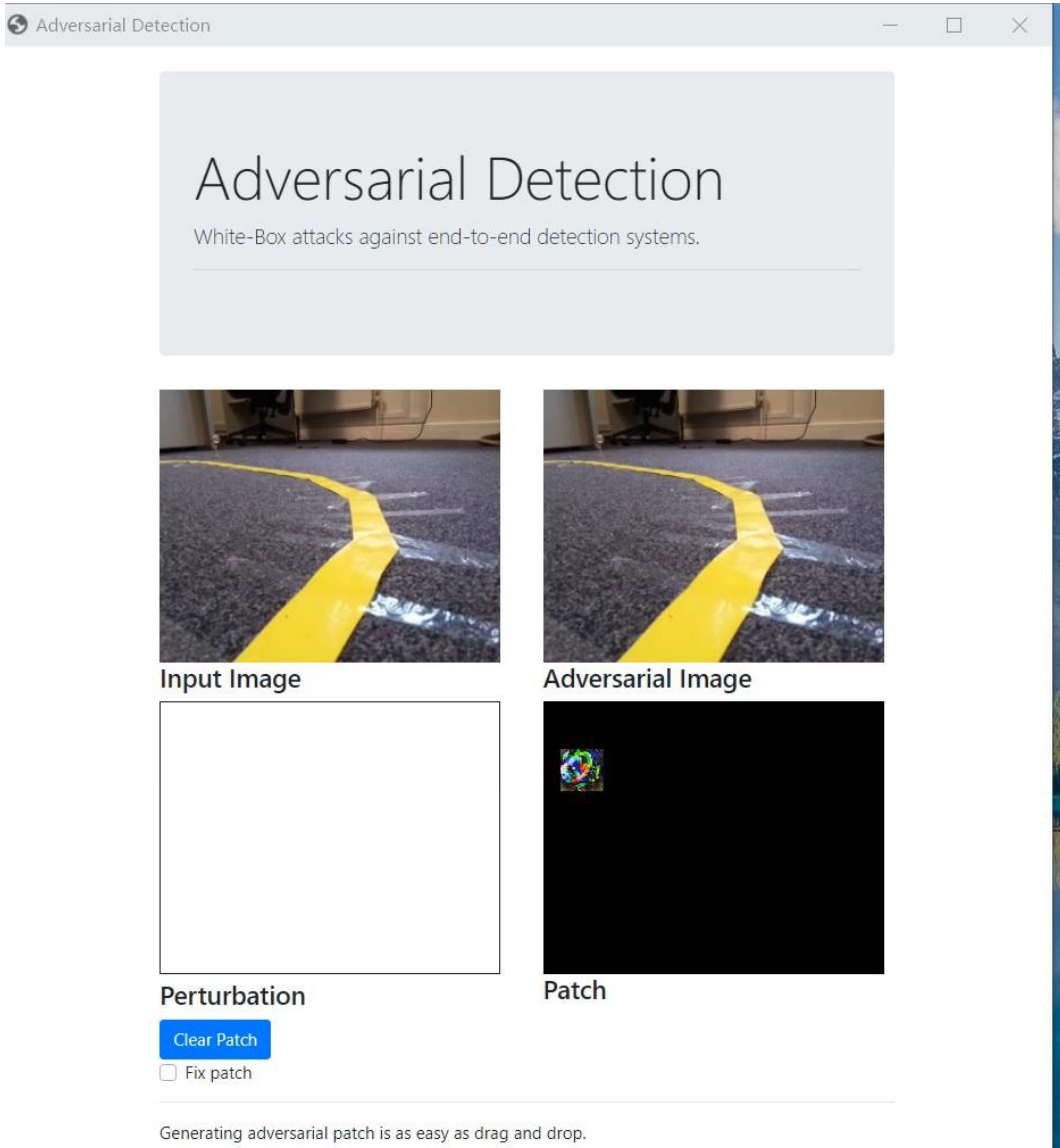
Project 2: Adversarial Detection

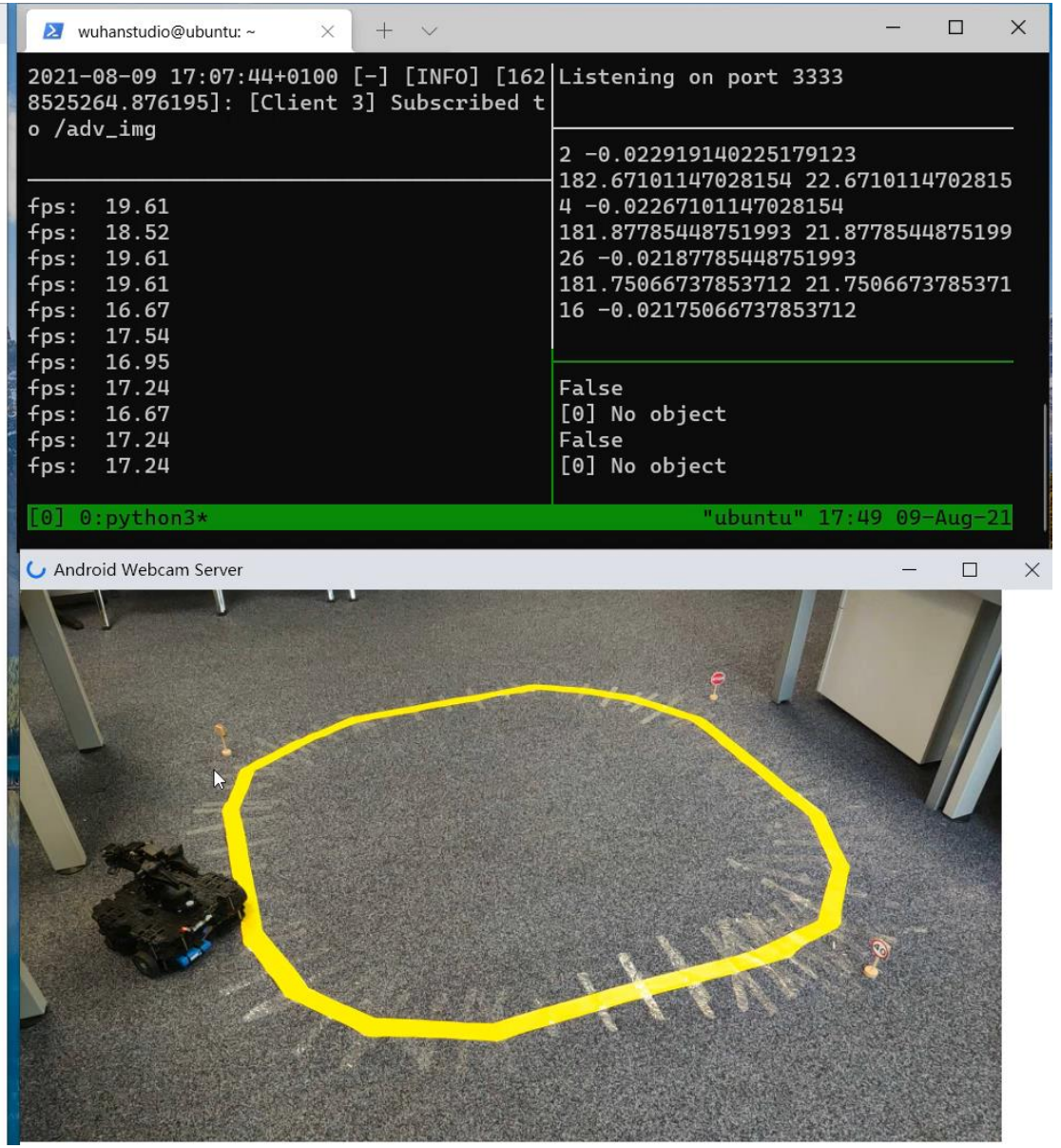
Project 2: Adversarial ROS Detection





Project 2: Adversarial ROS Detection



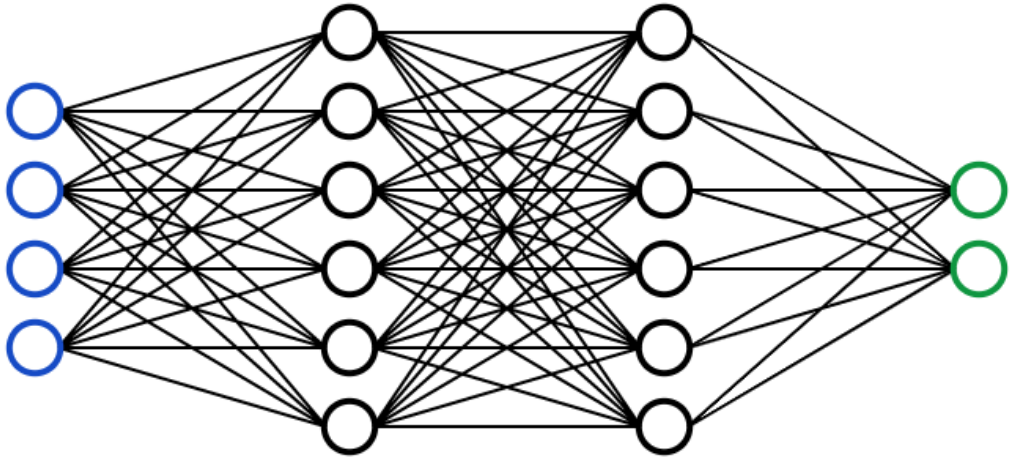


Project 2: Adversarial Detection

160x320

Change Inputs

Input



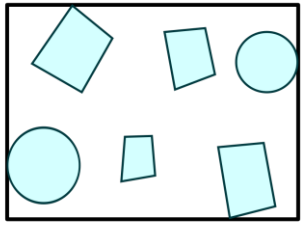
Output

Get Outputs

30x30

Perturb Inputs

Backward



Change Outputs

Gradients

$x + \delta$

Δy

How should I change the input (perturbation) to get the desired output?

$\nabla_\delta y$

1.0 → 1.5

$$\nabla_\delta = \begin{bmatrix} 1.5 & \cdots & 0.2 \\ \vdots & \ddots & \vdots \\ 0.1 & \cdots & -0.6 \end{bmatrix}$$

Take the 30x30 part from the 160 x 320 gradient

Project 2: Adversarial Detection

$x + \delta$ ← **Gradients** ← Δy

How should I change the input (perturbation) to get the desired output?

$\nabla_\delta y$

1.0 → 1.5

$$\nabla_\delta = \begin{bmatrix} 1.5 & \cdots & 0.2 \\ \vdots & \ddots & \vdots \\ 0.1 & \cdots & -0.6 \end{bmatrix}$$

Take the 30x30 part from the 160 x 320 gradient

Forward →

1. Generate a zero-initialized patch, feed forward

It's not a stop sign

Gradients ←

2. Retrieve the backward gradient

It will be recognized as one if

Δy

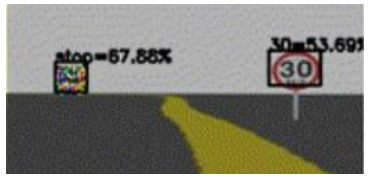
1.0 → 1.5

Forward →

3. Update the patch using the gradient

$\nabla_\delta y$

Now it's a stop sign



Project 2: Adversarial Detection

Forward

3. Update the patch using the gradient

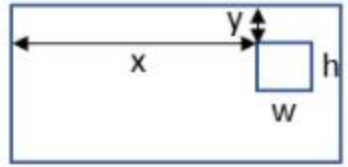
$\nabla_\delta y \quad \rightarrow \quad \nabla_\delta J(h_\theta(x,\delta),y)$

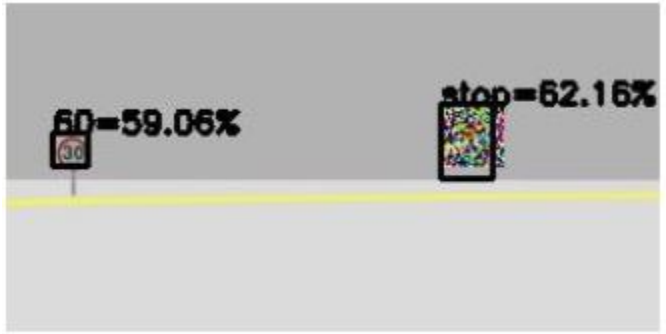
Now it's a stop sign

Δy

1.0 → 1.5

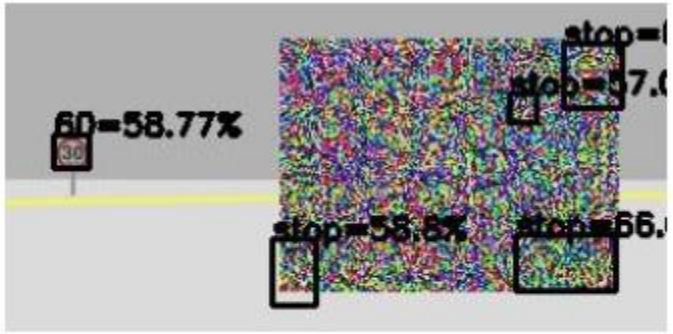
[None, x, y, w, h, c, p0, p1, p2]





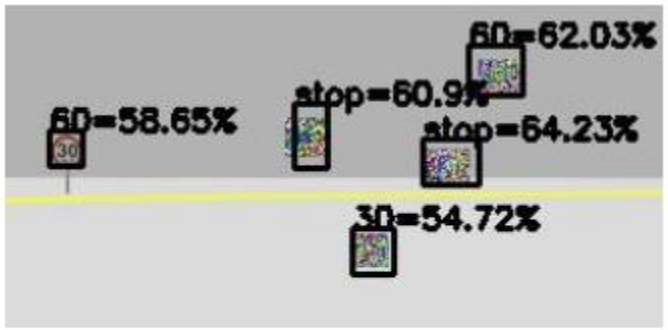
$J_1(x,\delta,y_h) = \max(\,\sigma(c) * \sigma(p0)\,)$

One Targeted Attack



$J_2(x,\delta,y_h) = \sigma(c) * \sigma(p0)$

Multi Targeted Attack



$J_3(x,\delta,y_h) = \sigma(c) * \sum\sigma(pi)$

Multi Untargeted Attack

Thank you!

Q&A