

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)

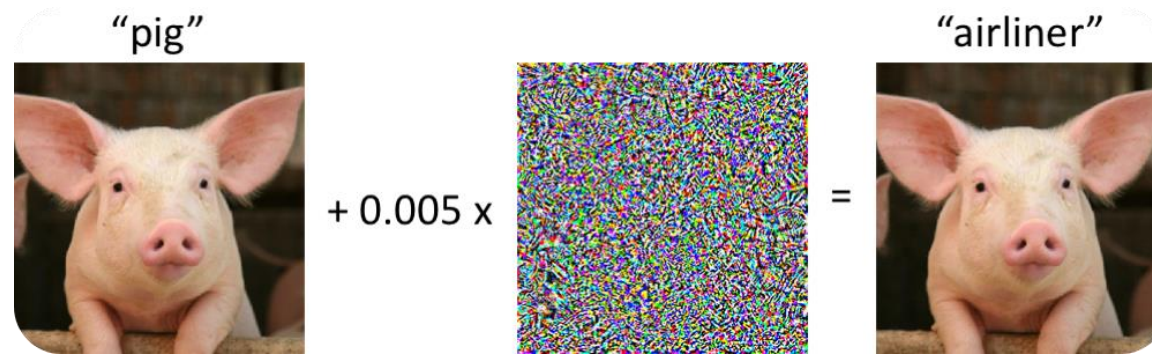


(b) Tesla Autopilot

Deep Learning for Autonomous Driving

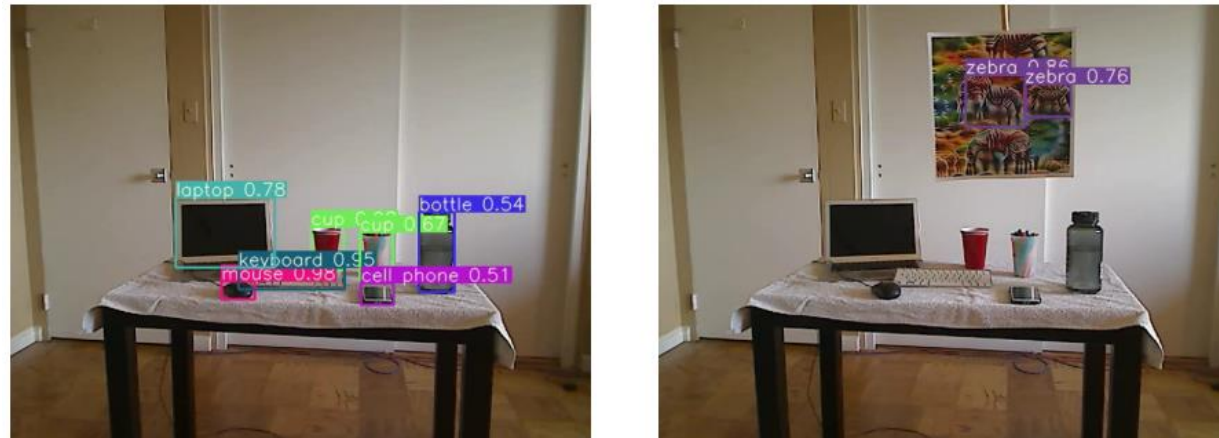
Background – Deep Learning

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



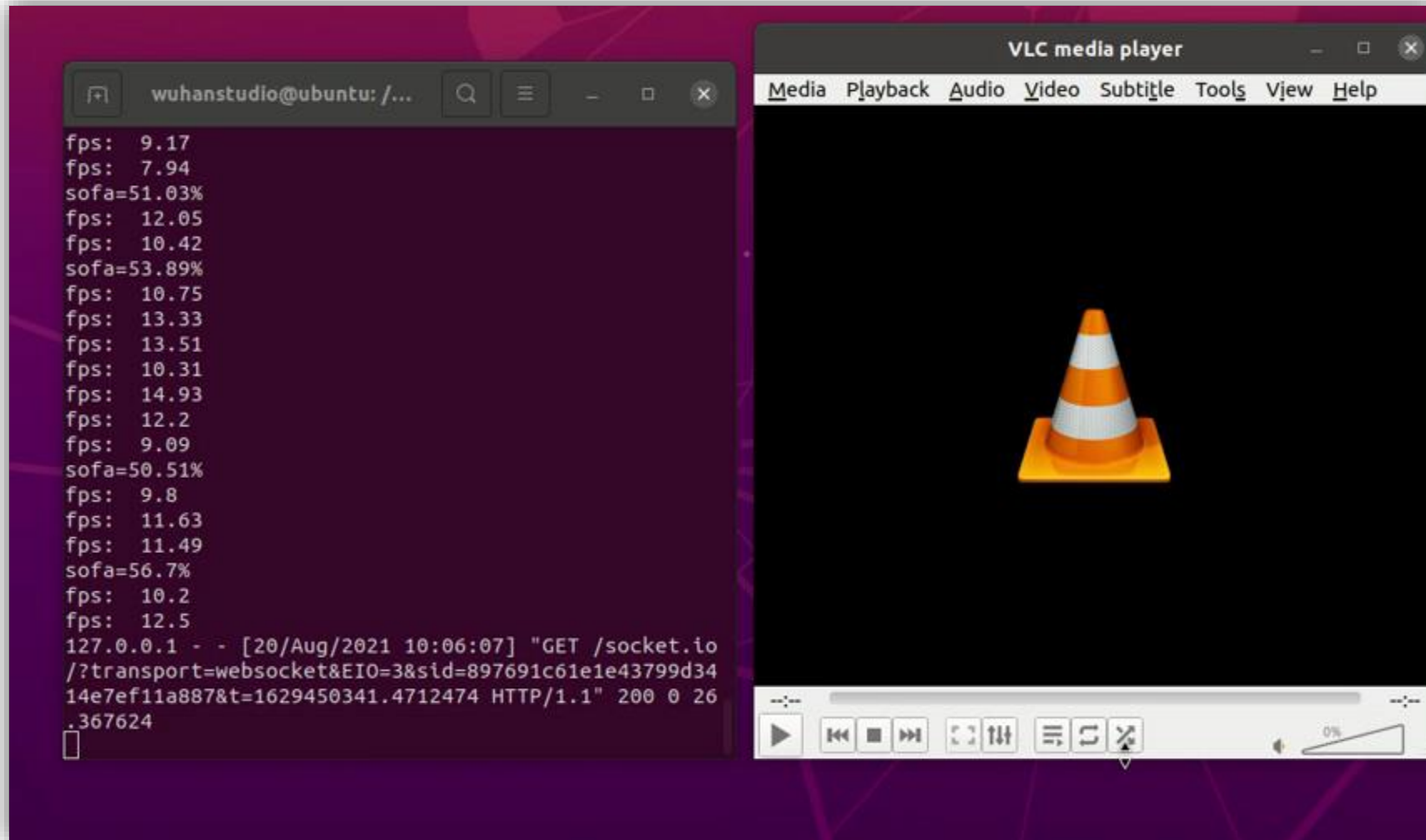
Adversarial attacks against image classification

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



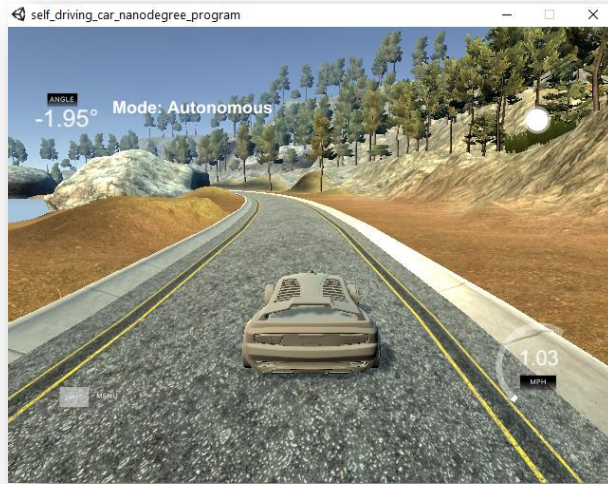
Adversarial attacks against object detection.

Demo – Adversarial Filter

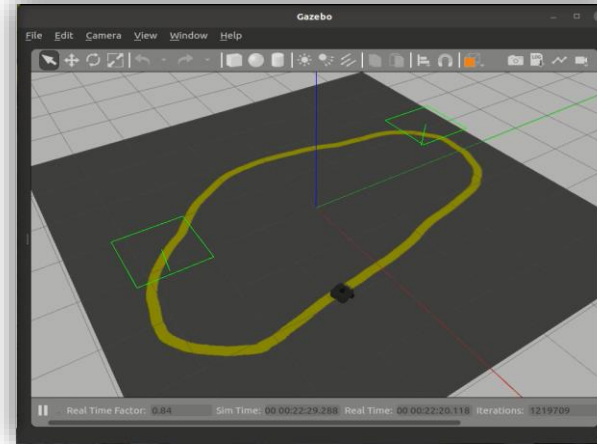


A fake camera that fools the object detection model.

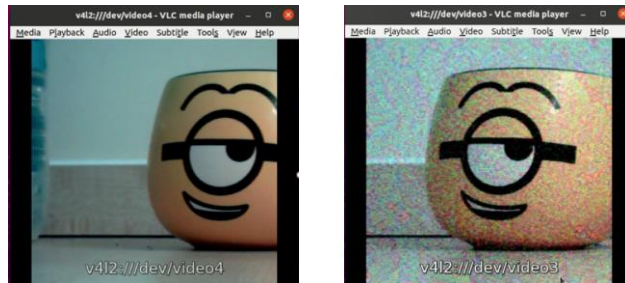
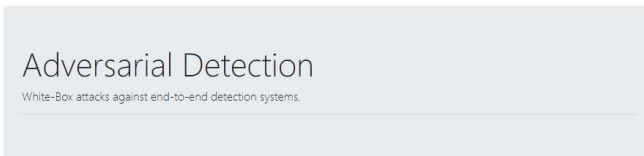
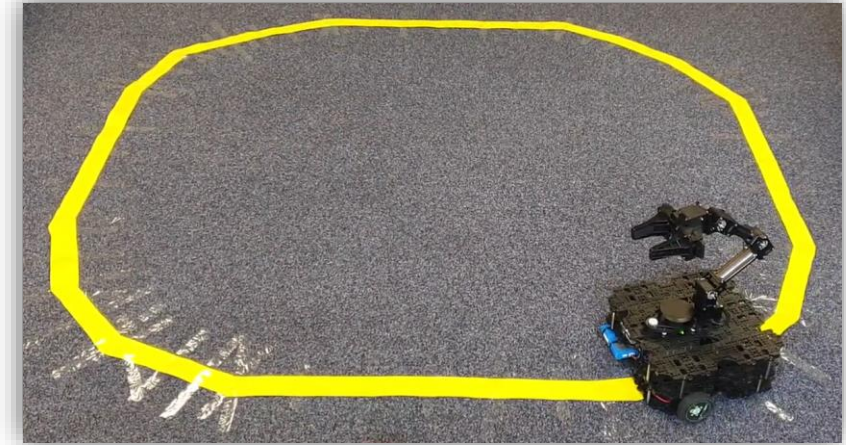
Overview



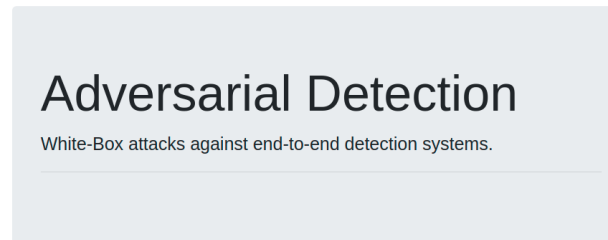
Adversarial Driving



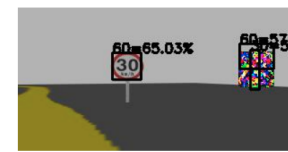
Adversarial ROS Driving



Adversarial Detection



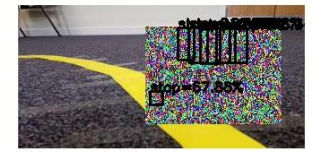
Input Image



Adversarial Image



Input Image



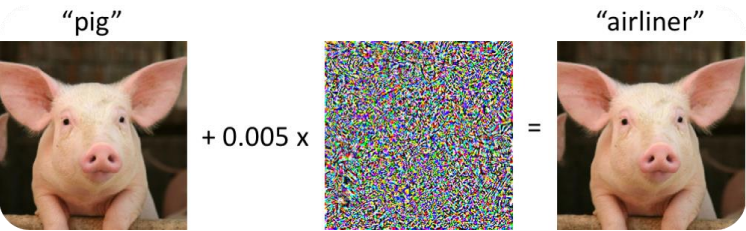
Adversarial Image

Adversarial ROS Detection

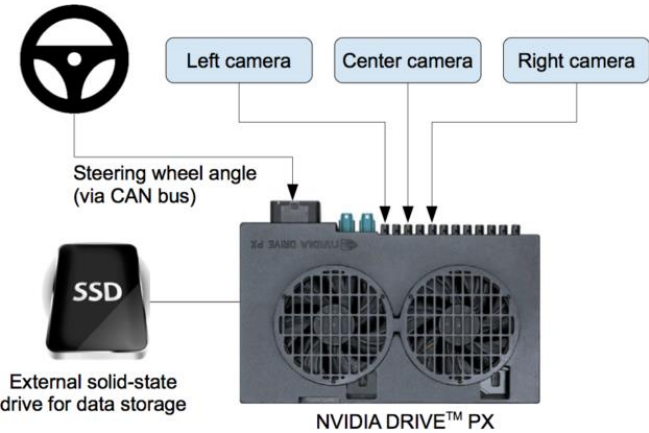
Project 1: Adversarial Driving

Project 1: Adversarial Driving^[4]

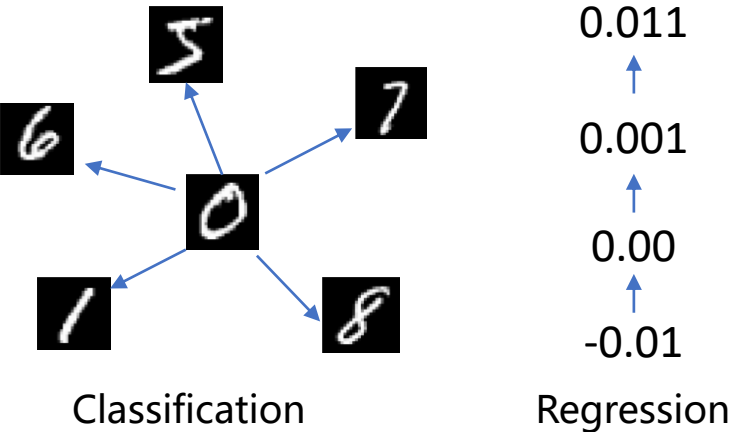
White-Box Adversarial Attack against Autonomous Driving



Fast Gradient Sign Method (FGSM)



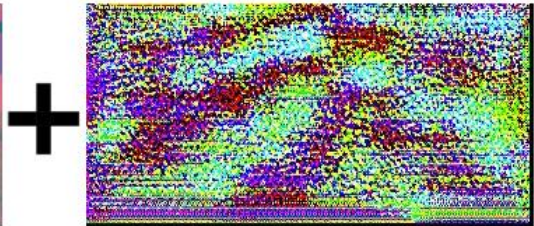
Nvidia end to end self driving



Camera Image



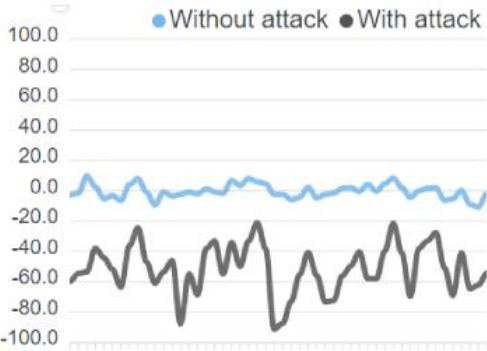
Input Image



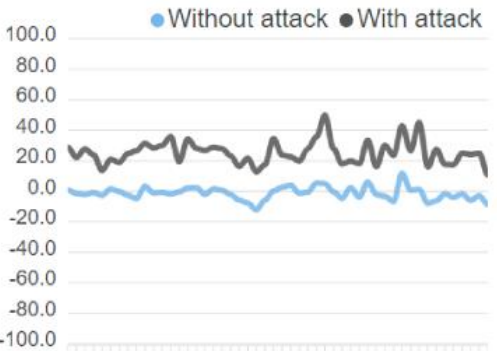
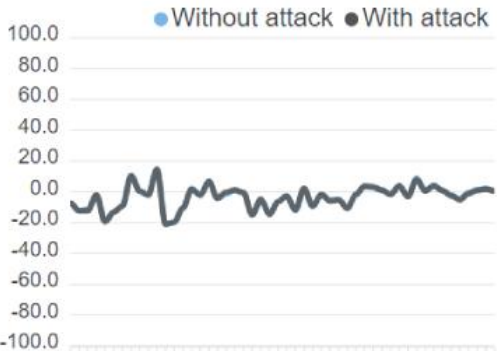
Perturbation



Adversarial Image



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



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

Project 1: Adversarial ROS Driving

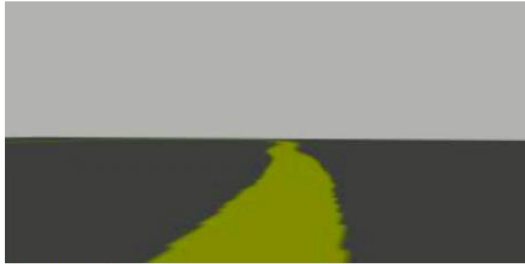
Adversarial Driving

White-Box attacks against end-to-end autonomous driving systems.


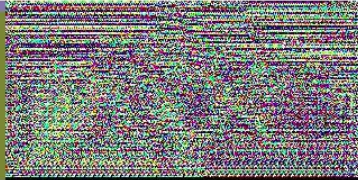

☐ Random Noise
☐ FGSMr: Left
☒ FGSMr: Right

☐ Attack

Left Right



Camera Image

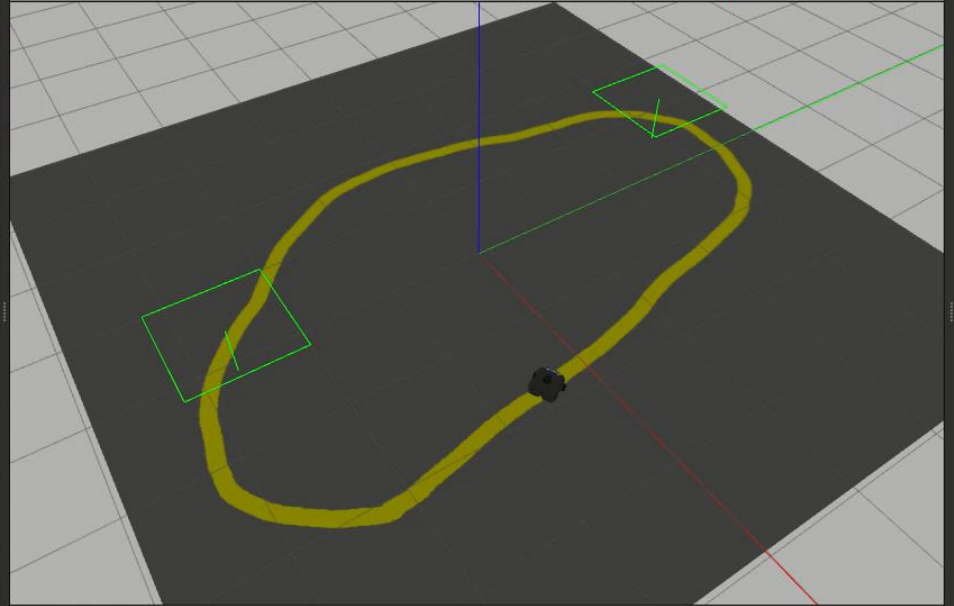


Input Image Perturbation Adversarial Image

```
wuhanstudio@ubuntu: ~  
wuhanstudio@ubuntu: ~ 95x16  
angle: 0.40356886  
angle: 0.28236178  
angle: 0.048892427  
angle: -0.22312714  
angle: -0.37631822  
angle: -0.40785757  
angle: -0.34947565  
angle: -0.021428447  
angle: 0.18881893  
angle: 0.38506278  
angle: 0.3791555  
angle: 0.3148877  
angle: 0.10411697  
angle: -0.02642906  
[0] 0:python3* "ubuntu" 15:52 29-Jun-23
```

Gazebo

File Edit Camera View Window Help



Real Time Factor: 0.88 Sim Time: 00 00:24:32.128 Real Time: 00 00:24:46.500 Iterations: 1342549

Project 1: Adversarial ROS Driving


Adversarial Driving

White-Box attacks against end-to-end autonomous driving systems.


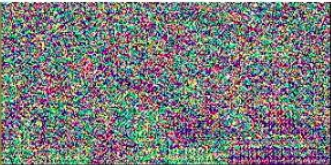
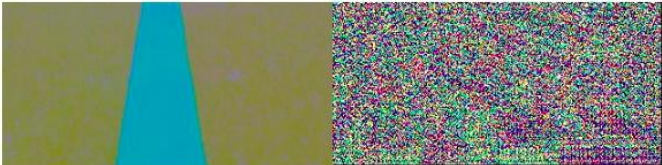
☐ Random Noise
☒ FGSMr: Left
☐ FGSMr: Right

☐ Attack

Left Right



Camera Image



Input Image Perturbation Adversarial Image

Steer

Steering Angle

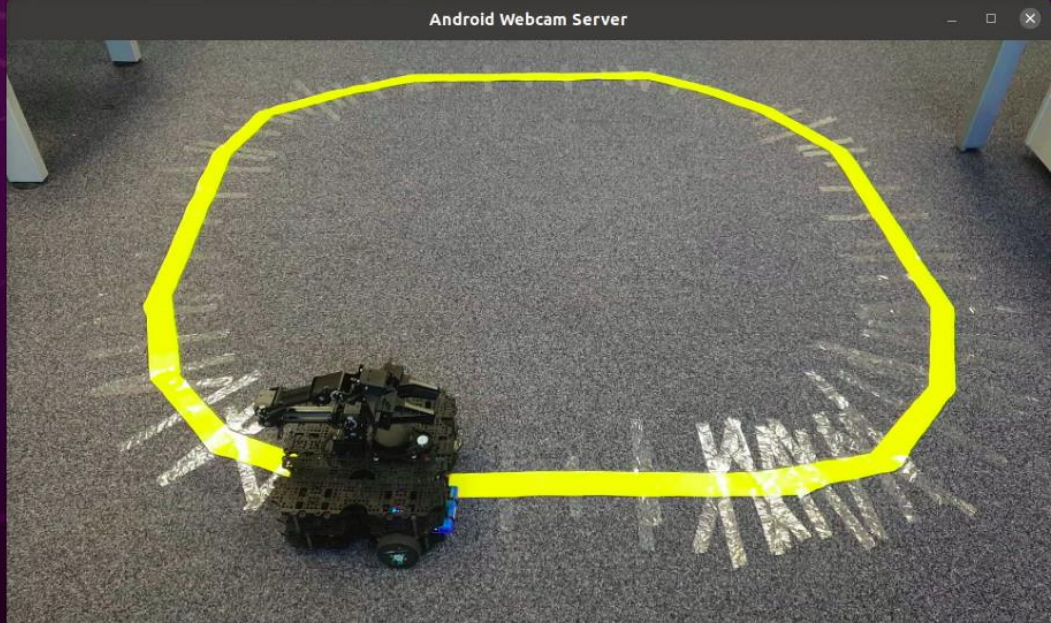
200.0

Without attack With attack

```
wuhanstudio@ubuntu: ~/adversarial-ros-driving
wuhanstudio@ubuntu:~/adversarial-ros-driving$ ls
client doc model README.md ros_ws
wuhanstudio@ubuntu:~/adversarial-ros-driving$ cd client/
wuhanstudio@ubuntu:~/adversarial-ros-driving/client$ ls
client go.mod go.sum main.go web
wuhanstudio@ubuntu:~/adversarial-ros-driving/client$ ./client
Listening on port 3333
```

| | |
|---------------------|--|
| angle: -0.09687273 | 2021-08-04 15:16:56+0100 [-] [INFO] [1628086 |
| angle: -0.06875558 | 616.475762]: [Client 2] Subscribed to /pertu |
| angle: -0.047367826 | rb_img |
| angle: 0.0028021778 | 2021-08-04 15:16:56+0100 [-] [INFO] [1628086 |
| angle: 0.025827363 | 616.535466]: [Client 2] Subscribed to /adv_i |
| angle: 0.03150396 | mg |
| angle: 0.034957677 | 2021-08-04 15:16:56+0100 [-] [INFO] [1628086 |
| angle: 0.0027756686 | 616.593207]: [Client 2] Subscribed to /cmd_v |
| angle: 0.005768552 | el_attack |
| | ^F |

[0] 0:python3* "ubuntu" 15:25 04-Aug-21




Android Webcam Server

Project 2: Adversarial Detection


Project 2: Adversarial ROS Detection

Adversarial Detection


White-Box attacks against end-to-end detection systems.




Input Image



Perturbation



Adversarial Image



Patch

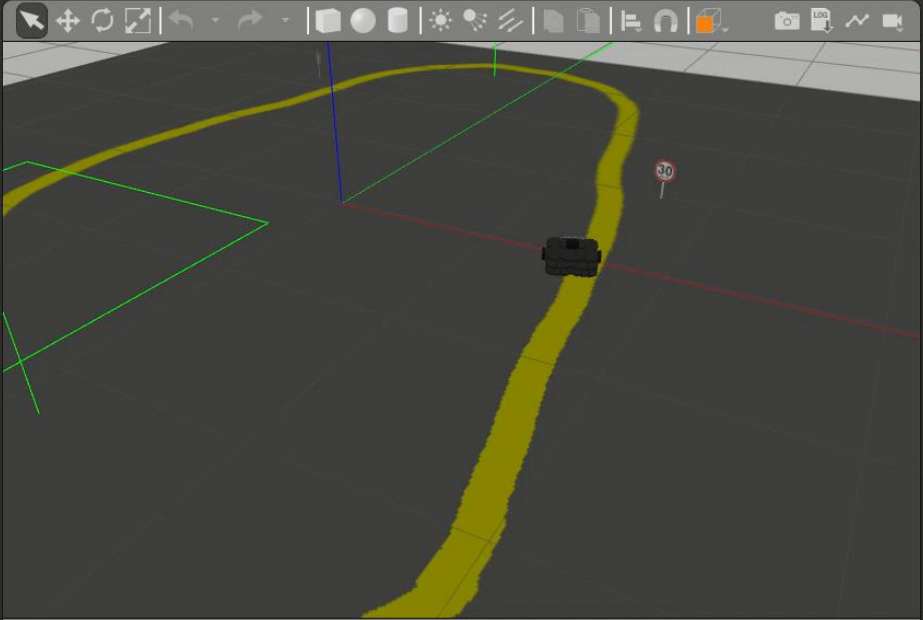
☐ Fix patch

Generating adversarial patch is as easy as drag and drop.

```
wuhamstudio@ubuntu: ~  
wuhamstudio@ubuntu: ~ 95x16  
fps: 13.51 [0] No object  
fps: 14.49 [0] No object  
fps: 13.33 [0] No object  
fps: 13.16 [0] No object  
fps: 14.29 [0] No object  
fps: 13.33 [0] No object  
30=55.8% [0] No object  
fps: 14.08 [0] No object  
30=54.26% [0] No object  
fps: 13.89 [0] No object  
30=54.77% [0] No object  
fps: 13.33 [2] Deaccelerate  
30=55.45% [2] Deaccelerate  
fps: 12.99 [2] Deaccelerate  
[0] 0:python3* "ubuntu" 16:06 29-Jun-21
```

Gazebo

File Edit Camera View Window Help




Real Time Factor: 0.92 Sim Time: 00 00:17:27.047 Real Time: 00 00:11:20.886 Iterations: 636744


Project 2: Adversarial ROS Detection


Adversarial Detection

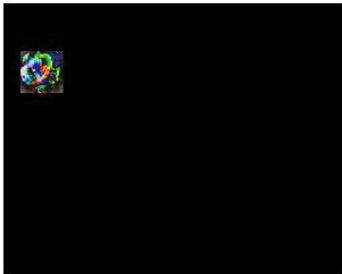
Adversarial Detection

White-Box attacks against end-to-end detection systems.









Clear Patch

☐ Fix patch

Generating adversarial patch is as easy as drag and drop.

wuhanstudio@ubuntu: ~

2021-08-09 17:07:44+0100 [-] [INFO] [1628525264.876195]: [Client 3] Subscribed to /adv_img


Listening on port 3333

2 -0.022919140225179123
182.67101147028154 22.6710114702815
4 -0.02267101147028154
181.87785448751993 21.8778544875199
26 -0.02187785448751993
181.75066737853712 21.7506673785371
16 -0.02175066737853712

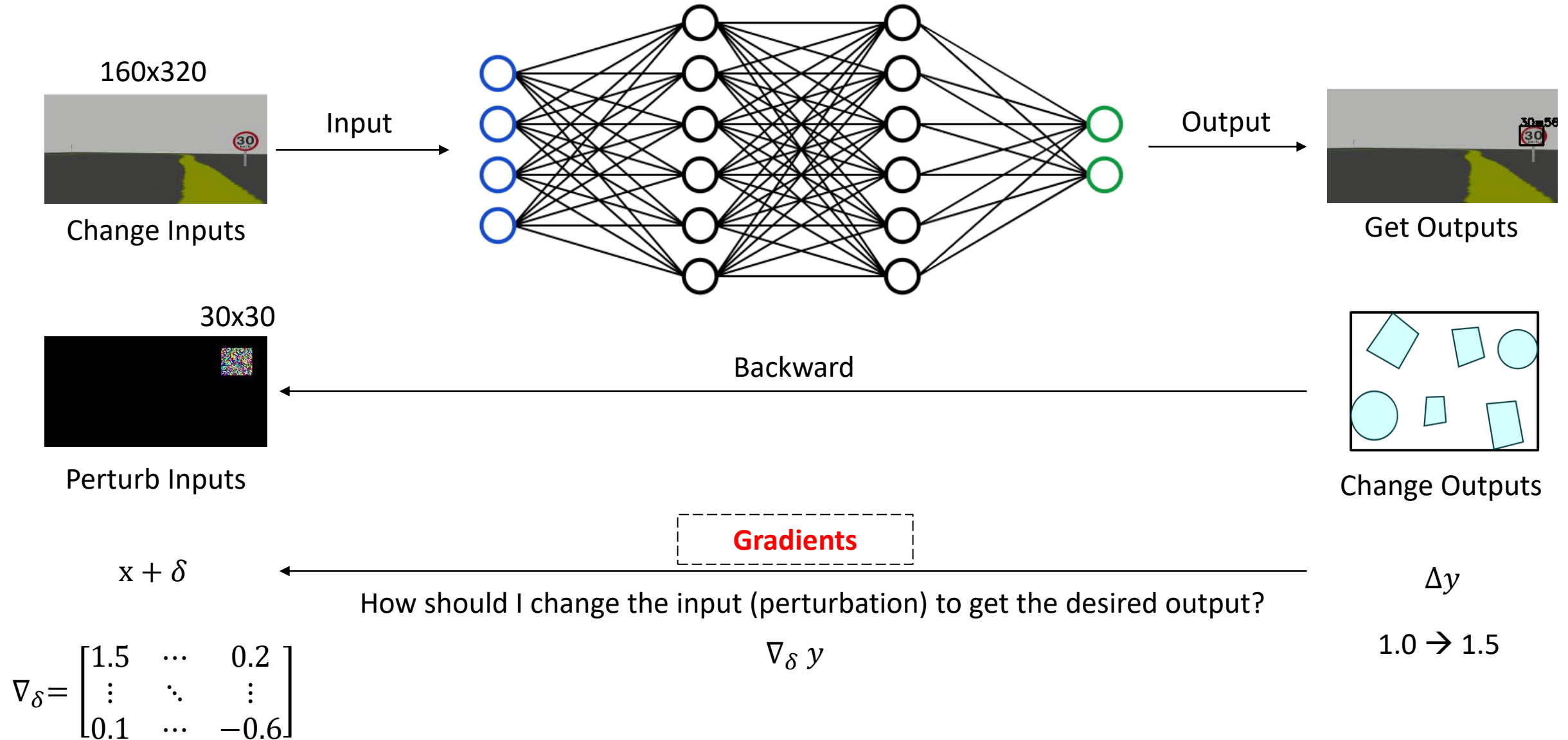
False
[0] No object
False
[0] No object

[0] 0:python3* "ubuntu" 17:49 09-Aug-21

Android Webcam Server



Project 2: Adversarial Detection



Take the 30x30 part from the 160 x 320 gradient

Project 2: Adversarial Detection

Gradients

$x + \delta$

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

Δy

$1.0 \rightarrow 1.5$

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

$\nabla_{\delta} y$

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

$1.0 \rightarrow 1.5$

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

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

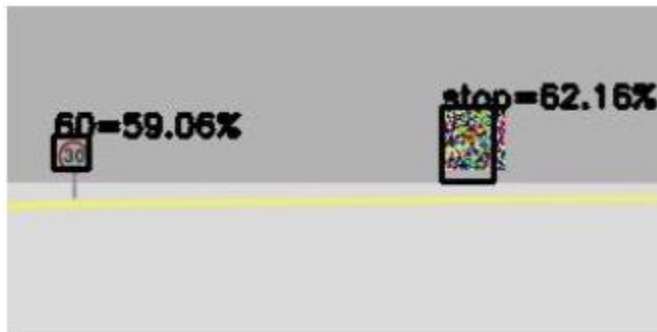
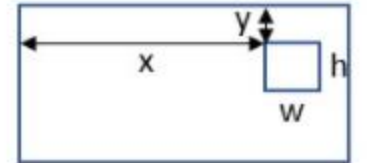
Now it's a stop sign



Δy

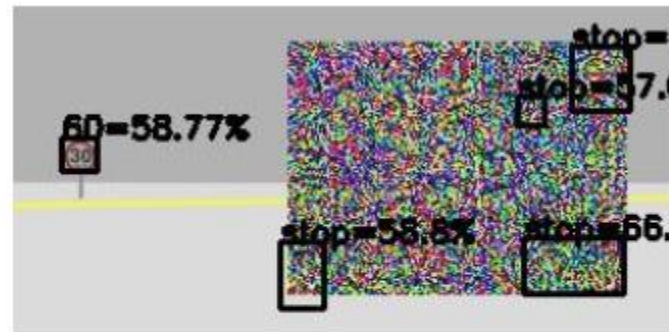
1.0 \rightarrow 1.5

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



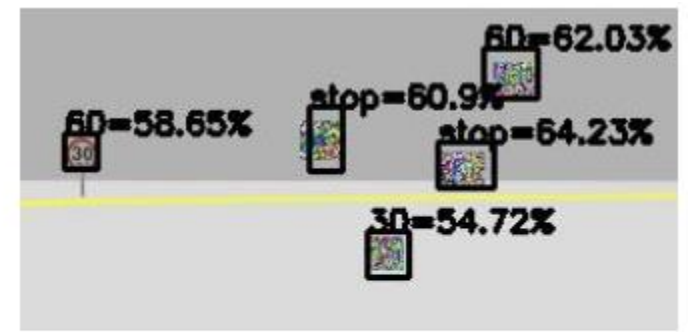
$$J_1(x, \delta, y_h) = \max(\sigma(c) * \sigma(p_0))$$

One Targeted Attack



$$J_2(x, \delta, y_h) = \sigma(c) * \sigma(p_0)$$

Multi Targeted Attack



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

Multi Untargeted Attack

Thank you!

Q&A