Perception Deception: Physical Adversarial Attack
Challenges and Tactics for DNN-based Object Detection

Zhenyu (Edward) Zhong, Yunhan Jia, Weilin Xu, Tao Wei
Our Team X-Lab

Chief Security Scientist
Dr. Tao Wei

Dr. Yunhan Jia

Dr. Zhenyu (Edward) Zhong
edwardzhong [at] baidu DOT com

Weilin Xu

AI Security Research

System Security Research

MesaTEE
https://www.mesatee.org

RUST · SGX
https://github.com/baidu/rust-sgx-sdk

MesaLock
https://github.com/mesalock-linux

MesaLink
https://github.com/mesalock-linux/mesalink

MesaPy
https://github.com/mesalock-linux/mesapy

Scan Me
• This talk doesn’t target any commercial autonomous driving systems.

• We don’t provide any comments to the vulnerabilities of the perceptions of existing autonomous driving systems.

• We focus on state-of-the-art object detection methods, all the results/techniques are proof-of-concept.
Car Safety – Unintended Acceleration


Toyota "Unintended Acceleration" Has Killed 89

Unintended acceleration in Toyota vehicles may have been involved in the deaths of 89 people over to the massive...

The New York Times

Toyota Will Pay $1.6 Billion Over Faulty Accelerator Suit

By Jaclyn Trop
July 19, 2013

...the defects we found were linked to unintended Acceleration through vehicle testing...

"unprotected critical variables."...


...the large throttle ope submitted VOQs could... does not mean it could not occur...
Uber Self-Driving Car That Struck, Killed Pedestrian Wasn’t Set to Stop in an Emergency

Pedestrian tested positive for methamphetamine and marijuana
SALT LAKE CITY -- The driver of a Tesla electric car had the vehicle's semi-autonomous Autopilot mode engaged when she slammed into the back of a Utah fire truck over the weekend, in the latest crash involving a car with self-driving features. The 28-year-old driver of the car told police in suburban Salt Lake City that the system was switched on and that she had been looking at her phone before the Friday evening crash.

Tesla's Autopilot system uses radar, cameras with 360-degree visibility and sensors to detect nearby cars and objects. It's built so cars can automatically change lanes, steer, park and brake to help avoid collisions.

The auto company markets the system as the "future of driving" but warns drivers to remain alert while using Autopilot and not to rely on it to entirely avoid accidents. Police reiterated that warning Monday.
Object Detection:
a technology related to computer vision and image processing that deals with instances of semantic objects of certain class in digital images and videos.

State-of-the-Art Vision-based Object Detection

YOLO (You Only Look Once)

numAnchors x (5 + numClasses)
### Accuracy on MS COCO

<table>
<thead>
<tr>
<th>backbone</th>
<th>AP</th>
<th>AP(_{50})</th>
<th>AP(_{75})</th>
<th>AP(_{S})</th>
<th>AP(_{M})</th>
<th>AP(_{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Two-stage methods</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faster R-CNN+++ [3]</td>
<td>34.9</td>
<td>55.7</td>
<td>37.4</td>
<td>15.6</td>
<td>38.7</td>
<td>50.9</td>
</tr>
<tr>
<td>Faster R-CNN w FPN [6]</td>
<td>36.2</td>
<td>59.1</td>
<td>39.0</td>
<td>18.2</td>
<td>39.0</td>
<td>48.2</td>
</tr>
<tr>
<td>Faster R-CNN by G-RMI [4]</td>
<td>34.7</td>
<td>55.5</td>
<td>36.7</td>
<td>13.5</td>
<td>38.1</td>
<td>52.0</td>
</tr>
<tr>
<td>Faster R-CNN w TDM [18]</td>
<td>36.8</td>
<td>57.7</td>
<td>39.2</td>
<td>16.2</td>
<td>39.8</td>
<td>52.1</td>
</tr>
<tr>
<td><strong>One-stage methods</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YOLOv2 [13]</td>
<td>21.6</td>
<td>44.0</td>
<td>19.2</td>
<td>5.0</td>
<td>22.4</td>
<td>35.5</td>
</tr>
<tr>
<td>SSD513 [9, 2]</td>
<td>31.2</td>
<td>50.4</td>
<td>33.3</td>
<td>10.2</td>
<td>34.5</td>
<td>49.8</td>
</tr>
<tr>
<td>DSSD513 [2]</td>
<td>33.2</td>
<td>53.3</td>
<td>35.2</td>
<td>13.0</td>
<td>35.4</td>
<td>51.1</td>
</tr>
<tr>
<td>RetinaNet [7]</td>
<td>39.1</td>
<td>59.1</td>
<td>42.3</td>
<td>21.8</td>
<td>42.7</td>
<td>50.2</td>
</tr>
<tr>
<td>RetinaNet [7]</td>
<td>40.8</td>
<td>61.1</td>
<td>44.1</td>
<td>24.1</td>
<td>44.2</td>
<td>51.2</td>
</tr>
<tr>
<td>YOLOv3 608 × 608</td>
<td><strong>33.0</strong></td>
<td><strong>57.9</strong></td>
<td><strong>34.4</strong></td>
<td><strong>18.3</strong></td>
<td><strong>35.4</strong></td>
<td><strong>41.9</strong></td>
</tr>
</tbody>
</table>

---

# Pick Our Target – YOLOv3

## Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD321</td>
<td>28.0</td>
<td>61</td>
</tr>
<tr>
<td>DSSD321</td>
<td>28.0</td>
<td>85</td>
</tr>
<tr>
<td>R-FCN</td>
<td>29.9</td>
<td>85</td>
</tr>
<tr>
<td>SSD513</td>
<td>31.2</td>
<td>125</td>
</tr>
<tr>
<td>DSSD513</td>
<td>33.2</td>
<td>156</td>
</tr>
<tr>
<td>FPN FRCN</td>
<td>36.2</td>
<td>172</td>
</tr>
<tr>
<td>RetinaNet-50-500</td>
<td>32.5</td>
<td>73</td>
</tr>
<tr>
<td>RetinaNet-101-500</td>
<td>34.4</td>
<td>90</td>
</tr>
<tr>
<td>RetinaNet-101-800</td>
<td>37.8</td>
<td>198</td>
</tr>
<tr>
<td>YOLOv3-320</td>
<td>28.2</td>
<td>22</td>
</tr>
<tr>
<td>YOLOv3-416</td>
<td>31.0</td>
<td>29</td>
</tr>
<tr>
<td>YOLOv3-608</td>
<td>33.0</td>
<td>51</td>
</tr>
</tbody>
</table>


Definition:
For an input image $x$,

$$\text{minimize } D(x, x + \delta), \ s.t. \ C(x + \delta) = t, x + \delta \in [0,1]^n$$

The most well-studied distance metric: $L_p$ Norm Perturbations

- $L_\infty$ -- each pixel is allowed to be changed by up to a limit
- $L_0$ -- number of pixels altered that matter most
- $L_2$ -- many small changes to many pixels
Adversarial Examples & $L_\infty$ Norm Perturbations Impact to DNN

**Intuition:** each pixel is allowed to change by up to a limit

$\text{FGSM } L_\infty$ based Perturbation Method

$x' = x - \epsilon \cdot \text{sign}(\nabla \text{Loss}_{F,\ell}(x))$

Still in Digital Context
Adversarial Examples & $L_\infty$ Norm Perturbations Impact to DNN

**Source Image** + **Perturbations** → **YOLOv3 Detection**

**FGSM** $L_\infty$ based Perturbation Method

Intuition: each pixel is allowed to change by up to a limit

$x' = x - \epsilon \cdot \text{sign}(\nabla \text{Loss}_{F,t}(x))$

Still in Digital Context
Adversarial Examples & $L_0$ Norm Perturbations Impact to DNN

Source Image + Perturbations → Perturbed Image

*JSMA $L_0$ based Perturbation Method*  
*Intuition: # of pixels altered that matter the most*

Still in Digital Context
Adversarial Examples & $L_0$ Norm Perturbations Impact to DNN

Source Image + Perturbations → YOLOv3 Detection

$JSMA$ $L_0$ based Perturbation Method

Intuition: # of pixels altered that matter the most

Still in Digital Context
Adversarial Examples & $L_2$ Norm Perturbations Impact to DNN

Source Image + Perturbations → Perturbed Image

$CW2$  $L_2$ based Perturbation Method  Intuition: many small changes to many pixels  

$\text{minimize } \| x - x' \|^2_2 + c \cdot f(x')$

Still in Digital Context
Adversarial Examples & $L_2$ Norm Perturbations Impact to DNN

Source Image + Perturbations → Perturbed Image

$CW2$ $L_2$ based Perturbation Method

Intuition: many small changes to many pixels

$\text{minimize} \; \|x - x'\|_2^2 + c \cdot f(x')$

Still in Digital Context
Digital Perturbations Realistic Enough?

Alter any pixel value in the image?

FGSM

Alter pixel value of the tree?

JSMA

Alter pixel value of the sky?

CW2

Feasible
Identify Opportunities by Completely Understanding YOLOv3 Inference Mechanism
Deep Dive into YOLOv3

Input
[416x416x3]

YOLO v3
Object Detection Model
[147 Layers, 62M Parameters]

Output
[10,647 Bounding Boxes]

Image: http://media.nj.com/traffic_impact/photo/all-way-stop-sign-that-flashes-in-montclairjpg-30576ab330660eff.jpg
Common Objects in Context

80 Classes: person, [car, truck, bus], [bicycle, motorcycle], [stop sign, traffic light], etc.
YOLOv3 Prediction

Anchor Boxes

Prediction Vector

Bounding Box Objectness 80 Class Confidence

\[
(t_x, t_y, t_w, t_h, p_{obj}, c_1, c_2, ..., c_{79}, c_{80})
\]

Center Point

Object Size

\[
\begin{align*}
(c_x, c_y) &= (11, 2) \\
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h}
\end{align*}
\]

13 x 13 Grid

116x 90

156x 198

373x326

stop sign 99%
car 0.01%
Threat Model: Physical Image Patch Attack

Image Patches
• Input Patch Construction
  • Differentiable to craft adversarial examples

• Attack Objectives
  • Make YOLOv3 detect fake object
  • Make object disappear in front of YOLOv3
Differentiable Input Patch Construction

Resize → Perspective Transformation

Company Logo
Our Physical Attack Approach & Objectives

• Input Patch Construction
  • Differentiable to craft adversarial examples

• Attack Objectives
  • Object Fabrication: make YOLOv3 detect fake object
  • Object Vanishing: make object disappear in front of YOLOv3
A. Naive Fabrication
   • Push more detections towards a certain object

```python
1  tgt_cls_id = self.model.class_names.index("car")
2  loss_box_class_conf = -tf.reduce_mean(y_box_class_probs[:, tgt_cls_id])
3  loss_box_conf = -tf.reduce_mean(y_box_confidence)
4  loss_final = loss_box_class_conf + loss_box_conf
```

B. Precise Fabrication
   • Produce fake object at specific location

```python
1  loss_boxes = 0
2  idx_pred_dict = self.yolo3_calc.calculate_box_preds(x1_y1_x2_y2)
3  for idx, pred in idx_pred_dict.items():
4      loss_boxes += tf.losses.mean_squared_error(pred, y_box_preds[idx])
```
Make a certain object class disappear in the whole image.

```
1  tgt_cls_id = self.model.class_names.index("car")
2  loss_box_class_conf = tf.reduce_mean(y_box_class_probs[:, tgt_cls_id])
3  loss_box_conf = tf.reduce_mean(y_box_confidence)
4  loss_final = loss_box_class_conf + loss_box_conf
```
Challenges to the Success of Physical Attack

1. Controlled Perturbation Area

2. Object appearance changes at various distances, angles

3. Various Light conditions: e.g. glaring, dimming

4. Color Distortion on various devices
   - Digital color palette 32 x 21
   - Kyocera Taskalfa 3551 ci

5. Inaccurate Patch Location

Captured by iPhoneX from a distance
Tactics to the Challenges:

- **[Controlled Perturbation Area]** Image-patch based Attack
- **[Color Distortion]** Color Management with the **Non-Printability Loss (NPS)**
- **[Inaccurate Patch]** Random **Transformation (RT)** during optimization iterations
- **[Various Distances & Angles]** **RT + Total Variation** regularization instead of Expectation-Over-Transformation
- **[Various Light Condition]** Get a stable environment
- More ...
Given $P \subset [0,1]^3$, a set of printable RGB triplets. $NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|$

For the perturbated $\delta$, $NPS(\delta) = \sum_{\hat{p} \in \delta} NPS(\hat{p})$. $NPS(\delta) \downarrow$, color reproducibility $\uparrow$

---

Accessorize to a Crime: Real and Stealthy Attacks on State-Of-The-Art Face Recognition.
Random Transformation During Optimization Iterations

- Introduce Random Perspective Transformation
- Random Transformation During Optimization Iterations
- Perfectly positioned?

Generated Perturbation Patch

- $\text{Iteration}_i$
- $\text{Iteration}_j$
- $\text{Iteration}_k$
• **Random Transformation + Total Variance Regulation**: a different approach from EOT

Simulate the transformations using RT + TV for various distances & angles instead of drawing from a distribution
Put Everything Together: An Iterative Optimization

Iteration$_i$  Iteration$_j$  Iteration$_k$
• With careful setup, physical attacks are achievable against DNN-based object detection methods in a white box setting

• Defense is hard, a good safety and security metric has to be explored

• We call out efforts for a robust, adversarial example resistant model that is required in safety critical system like autonomous driving system