Adversarial camera stickers: A physical camera-based attack on deep learning systems

Juncheng B. Li, Frank R. Schmidt, J. Zico Kolter
Bosch Center for Artificial Intelligence
Adversarial attacks: not just a digital problem

All existing physical attacks modify the *object*.

Sharif et al., 2016  
Etimov et al., 2017  
Athalye et al., 2017
QUESTION

All existing physical attacks modify the object, but is it possible instead to fool deep classifiers by modifying the camera?
This paper: A physical adversarial camera attack

- We show it is indeed possible to create visually inconspicuous modifications to a camera that fool deep classifiers

- Uses a small specially-crafted translucent sticker, placed upon camera lens

- The adversarial attack is *universal*, meaning that a *single* perturbation can fool the classifier for a given object class over multiple viewpoints and scales
The challenge

- (Inconspicuous) physical stickers are **extremely limited** in their resolution (can only create blurry dots over images)

- Need to both learn a model of allowable perturbations and create the adversarial image

Our solution

- A differentiable model of sticker perturbations, based upon alpha blending of blurred image overlays

- Use gradient descent to both fit the perturbation model to observed data, and construct an adversarial attack
Methodology

• Attack model consists of smoothed alpha blend between observed image and some fixed color (iterated to produce multiple dots)

\[
l_{x,y}^{\text{out}} = (1 - \alpha_{x,y}) \cdot l_{x,y}^{\text{in}} + \alpha_{x,y} \cdot c, \quad \alpha_{x,y} = \exp \left( -\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2} \right)
\]

• Parameters of attack include color \( c \), dot position \( (x_c, y_c) \) and bandwidth \( \sigma \)

• **Key idea:** use gradient descent over some parameters (e.g., color, bandwidth) to fit model to observed physical images, over other parameters (e.g. location) to maximize loss

\[
\min_{c,\sigma} \text{SSIM}(I^{\text{out}}, I^{\text{real}}), \quad \max_{x_c, y_c} \sum_{i=1}^{m} \text{Loss}(I^{\text{out}}, y_{\text{true}}, y_{\text{target}})
\]
How does a dot look like through camera lense?

Clean Camera View  Red Dot  Resulting Blur  Simulated Blur
Table 1. Performance of our 6-dot attacks on ImageNet test set

<table>
<thead>
<tr>
<th>Class</th>
<th>Attack</th>
<th>Prediction</th>
<th>Correct</th>
<th>Target</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>No</td>
<td>85%</td>
<td>15%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mouse</td>
<td>Yes</td>
<td>48%</td>
<td>36%</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>Street sign</td>
<td>No</td>
<td>64%</td>
<td></td>
<td>36%</td>
<td></td>
</tr>
<tr>
<td>Guitar Pick</td>
<td>Yes</td>
<td>32%</td>
<td>34%</td>
<td>34%</td>
<td></td>
</tr>
<tr>
<td>Street sign</td>
<td>No</td>
<td>64%</td>
<td></td>
<td>36%</td>
<td></td>
</tr>
<tr>
<td>50 random classes</td>
<td>Yes</td>
<td>18%</td>
<td>33%</td>
<td>49%</td>
<td></td>
</tr>
<tr>
<td>50 random classes</td>
<td>No</td>
<td>74%</td>
<td></td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>50 random classes</td>
<td>Yes</td>
<td>42%</td>
<td>31%</td>
<td>27%</td>
<td></td>
</tr>
</tbody>
</table>
This is a ResNet-50 Model implemented with pyTorch deployed on a Logitech C920 Webcam with clear lense.
This is a ResNet-50 model implemented with PyTorch deployed on a Logitech C920 WebCam with clear lense. It can recognize street sign at different angles with only minor errors.
Now we cover the camera with our adversarial sticker made by our proposed method to achieve the targeted attack. This should make a “street sign” misclassified as a “guitar pick”.
The Sticker results in very inconspicuous blurs in the view. We can achieve targeted attack most of the time at different angles and with different distances.
### Results: Real World Evaluation

<table>
<thead>
<tr>
<th>Original Class</th>
<th>Target Class</th>
<th>Prediction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Correct</td>
<td>Target</td>
</tr>
<tr>
<td>Keyboard</td>
<td>Mouse</td>
<td>271</td>
<td>548</td>
</tr>
<tr>
<td></td>
<td>Space bar</td>
<td>320</td>
<td>522</td>
</tr>
<tr>
<td>Street sign</td>
<td>Guitar Pick</td>
<td>194</td>
<td>605</td>
</tr>
<tr>
<td></td>
<td>Envelope</td>
<td>222</td>
<td>525</td>
</tr>
<tr>
<td>Coffee mug</td>
<td>Candle</td>
<td>330</td>
<td>427</td>
</tr>
</tbody>
</table>

*Table 2. Fooling performance of our method on two 1000 frame videos of a computer keyboard and a stop sign, viewed through a camera with an adversarial sticker placed on it targeted for these attacks.*
Summary

• Adversarial attacks don’t need to modify every object in the world to fool a deployed deep classifier, they just need to modify the camera

• Implications in self-driving cars, security systems, many other domains

To find out more, come see our poster

at  Pacific Ballroom #65
on  Tuesday, Jun 11th 06:30-09:00