GRAPHITE: Generating Automatic Physical Examples for Machine-Learning Attacks on Computer Vision Systems

Ryan Feng\textsuperscript{1}, Neal Mangaokar\textsuperscript{1}, Jiefeng Chen\textsuperscript{2}, Earlence Fernandes\textsuperscript{2}, Somesh Jha\textsuperscript{2}, Atul Prakash\textsuperscript{1}

\textsuperscript{1}University of Michigan, \textsuperscript{2}University of Wisconsin

Euro S&P 2022
Robust Physical Perturbation Attacks

- Physical attacks such as RP₂ [1] enable sticker attacks on physical objects

  - **Key idea**: physical attacks are more *practical*
    - Easier to attack a real system, harder to defend

  - **Limitations**: current methods still require
    - Manual mask experimentation
    - White-box access to model weights / architecture

Motivation: A Framework for Practical Attacks

• **Goal: Generate Practical Attacks**
  • *Automatically* generate masks
  • Apply attacks as *physical* stickers
  • Can work with just *hard-label* access

• Automatic attack generation tools can assist with adversarial testing and defense design
Graphite Framework

Key idea: jointly optimize mask size and transform-robustness

\[
\arg\min_{\delta, M} \lambda \cdot ||M||_0 - \mathbb{E}_{t \sim T} \left[ F(t(x + M \cdot \delta)) = y_{\text{tar}} \right]
\]

Small mask size \hspace{1cm} High transform-robustness

\(x\): Input image
\(y_{\text{tar}}\): Target label
\(\delta\): Perturbation
\(F\): Model
\(M\): Mask (Patch Area)
\(T\): Transformation Distribution
\(\lambda\): Weight parameter

Algorithm 1 General Graphite Framework

Input: Victim Image \(x\), Target Image \(x_{\text{tar}}\), Initial Mask \(M_{\text{init}}\), Model \(F\), Target Label \(y_{\text{tar}}\)
Output: Attacked Image \(A\), Mask \(M\), Perturbation \(\delta\)

1: \(M \leftarrow M_{\text{init}}\)
2: \(\delta, g \leftarrow \text{INIT\_PERT\_+_GRAD}(x, x_{\text{tar}}, M, F, y_{\text{tar}})\)
3: while not done do
4: \(S \leftarrow \text{SELECT\_PIXELS}(x, x_{\text{tar}}, M, \delta, y_{\text{tar}}, g)\)
5: \(M \leftarrow \text{REMOVE\_PIXELS}(M, S)\)
6: \(A, \delta, g \leftarrow \text{ATTACK}(x, x_{\text{tar}}, M, \delta_{\text{init}}, F, y_{\text{tar}})\)
7: \(A, \delta \leftarrow \text{Last Successful Attack}\)
White-box Version of GRAPHITE

• Start with C&W $\ell_0$ attack [1]
  • Alternates between C&W $\ell_2$ attack [1] and removing the pixel with least impact

• Replace the C&W $\ell_2$ attack with an EoT PGD attack [2, 3]

• Avg. 78% transform-robustness, 9% mask size

White-box GRAPHITE attacks can be generated.

What about black-box (hard-label) GRAPHITE attacks, where only the top-1 prediction label is available (no gradients, no probabilities)?
Hard-label Baselines

• Simple combinations of C&W $\ell_0$ [2], EoT [3], and OPT Attack [4] poor
  • Issues included: Poor transform-robustness, large masks, query inefficiency

<table>
<thead>
<tr>
<th>$\ell_0$ and OPT</th>
<th>$\ell_0$ and OPT w/ EoT</th>
<th>$\ell_0$ and Boosting</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

• Pixel ordering by impact as in C&W $\ell_0$ [2] breaks down without gradients
• Distance minimizing hard-label attacks query-inefficient with EoT

Hard-label Version of GRAPHITE

- Simplify to a two-step optimization – Mask Generation and Boosting

\[
\begin{align*}
\argmin_M & \quad \lambda \cdot ||M||_0 - \mathbb{E}_{t \sim T} \left[ F\left( t(x + M \cdot \delta_{\text{tar}}) \right) = y_{\text{tar}} \right] \\
\text{s.t.} & \quad \mathbb{E}_{t \sim T} \left[ F\left( t(x + M \cdot \delta_{\text{tar}}) \right) = y_{\text{tar}} \right] \geq tr_{\text{lo}} \\
\argmax_\delta & \quad \mathbb{E}_{t \sim T} \left[ F\left( t(x + M \cdot \delta) \right) = y_{\text{tar}} \right]
\end{align*}
\]

Victim Image (x) -> Heatmap Estimation -> Coarse-grained Reduction -> Fine-grained Reduction -> Boosting -> Perturbation Generation -> Output Image (x + M • δ)

Ryan Feng
rtfeng@umich.edu
**Physical World Results**

**TABLE 8. GTSRB field test results. Physical robustness results are calculated over 5 pictures each at the following spots: 5 ft × {0°, 15°, 30°, 45°}, 10 ft × {0°, 15°, 30°}, 15 ft × {0°, 15°}, 20 ft × {0°, 15°}, 25 ft, 30 ft, 40 ft. Each example was tested 3 times: outdoors, indoors with indoor lights turned off, and indoors with indoor lights turned on.**

<table>
<thead>
<tr>
<th>Victim</th>
<th>Target</th>
<th>Digital GRAPHITE attack</th>
<th>Physical GRAPHITE attack (outdoors)</th>
<th>Dig. TR (100 xforms)</th>
<th>Phys. TR (Indoors, lights off)</th>
<th>Phys. TR (Indoors, lights on)</th>
<th>Phys. TR (Outdoors)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Stop Sign" /></td>
<td><img src="image2.png" alt="30 MPH Sign" /></td>
<td><img src="image3.png" alt="Stop Sign" /></td>
<td><img src="image4.png" alt="Stop Sign" /></td>
<td>86%</td>
<td>92.9%</td>
<td>94.3%</td>
<td>100%</td>
</tr>
<tr>
<td><img src="image5.png" alt="Stop Sign" /></td>
<td><img src="image6.png" alt="Pedestrian Sign" /></td>
<td><img src="image7.png" alt="Stop Sign" /></td>
<td><img src="image8.png" alt="Stop Sign" /></td>
<td>79%</td>
<td>97.1%</td>
<td>85.7%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Tuning GRAPHITE

- 3 parameters to trade off: query count, transform-robustness, and mask size
- In the extreme, we can find attacks with as few as 500 queries with lower transform-robustness
Attacking PatchGuard

• GRAPHITE can defeat PatchGuard [1]
  • Tested on 100 CIFAR-10 examples
  • Avg. Transform-robustness: 68%
  • Avg. Query Count: 155.8k
  • Avg. Mask size: 193.81 pixels

• Example on right: 10 pixel attack to misclassify a dog as a cat

Conclusion

• GRAPHITE: first automatic physical hard-label attack

• We hope GRAPHITE guides future defense work against practical attacks

• Code available to try it out:
  https://github.com/ryan-feng/GRAPHITE
Thank you!

• Contact Information
  • Ryan Feng
  • rtfeng@umich.edu
  • http://www-personal.umich.edu/~rtfeng/
  • https://twitter.com/ryantfeng

• Paper Links
  • https://github.com/ryan-feng/GRAPHITE