Poster: Perceived Adversarial Examples

Yanmao Man, Ming Li
Dept. of ECE, University of Arizona
{yman, lim}@email.arizona.edu

Ryan Gerdes
Dept. of ECE, Virgina Tech.
rgerdes@vt.edu

***Abstract*—Adversarial examples in deep learning were first discovered for the digital domain and later effected in the physical domain. In this work, we demonstrate adversarial examples in the perception domain; i.e., adversarial examples that are introduced by compromising the sensing mechanism of an image sensor. Our proposed attack relies on the injection of electromagnetic interference in a remote manner, and does not require physical modifications to the object, which makes the attack easier to launch and harder to detect.**

***Index Terms*—Deep Learning, Adversarial Examples, Sensing Mechanism**

I. Introduction

Deep neural networks (DNN) [1] have been successfully applied to various tasks such as image recognition, natural language processing, etc. Recently, multiple researchers from ML/security communities have introduced the concept of adversarial examples (e.g., [2], [3]) whereby images are perturbed in slight ways that cause DNN-based object classifiers to misclassify the contents of the images. This can lead to severe consequences when DNNs are deployed for safety-critical autonomous platforms [4].

Existing attacks mainly target either *digital domain* or *physical domain*. See Figure 1 for their differences. For digital domain attacks, the adversarial perturbation is directly applied to digital images [3]. In physical domain attacks, objects of interest are modified to cause misclassification, e.g., putting stickers on a stop sign [4].

The drawbacks of physical modifications are that they may be noticeable to human observers and they may cause damage to the object permanently. Also, it may not be easy to gain physical access to the object in many cases. To address these issues, we propose a new class of attacks that target the *perception domain* in which the adversary remotely affects the sensing mechanisms of the sensor that is used to acquire the image (e.g., a camera in this work, or signal transceivers, LiDAR sensors, etc.). The adversary is equipped with an electromagnetic signal generator, such as a light source, that emits EM signals (light) which will interfere with the targeted sensor so that the captured signals (images) will be altered and then misclassified.

The main challenge of realizing perception domain attacks is that, unlike two other types of attacks, the injected interference signals are often difficult to control as they exhibit randomness due to variability of the signal sources and channel effects, thus the adversary is not able to manipulate each pixel deterministically or precisely.

This work is supported in part by NSF grants CNS-1801402, CNS-1410000, and CNS-1801611.

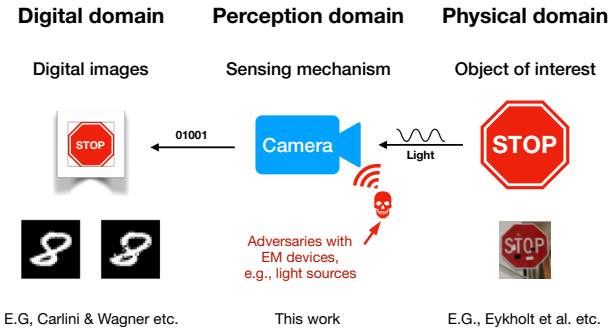

Figure 1: Three categories of attacks.

To address this challenge, we first find the optimal random noise for three RGB channels, and secondly we realize it in the perception domain. In order to find the optimal random noise, we develop a framework that determines the optimal parameters of the noise distributions from which the noise will be drawn. For the second step, the challenge is how to generate the actual noise physically such that the end-to-end noise captured by the imager will affect the captured image as we expect. We conduct a preliminary real-world experiment to demonstrate that it is in fact feasible to realize our attacks in the perception domain.

II. System and Threat Model

We denote a neural network as $y = f_\theta(x)$, where $x \in \mathbb{R}^{w\times h\times 3}$ (width, height and RGB three channels) is the input image, $y \in \mathbb{R}^m$ is the output vector, and θ is the parameters of the network (which is fixed thus we omit it for convenience). The classification result is $C(x) = \arg\max_i y_i$. Also, the inputs to the softmax layer are called *logits* and denoted as $Z(x)$. An adversarial example is denoted as x', where $x' = x + \delta$. Here, δ is slight, additive noise that has the same dimensionality with x. Given a benign image x and a target label t, an adversary wants to find a small δ such that $t = C(x + \delta)$, i.e., *targeted attacks*.

We assume that the adversary has access to the target neural network f, including the architecture (hyperparameters) and the well-trained, fixed parameters θ, i.e., we assume a *white-box* threat model. The adversary has access to the testing dataset only. Moreover, the adversary is unable to manipulate each pixel individually or deterministically; instead, the adversary wants to affect each channel independently with random noise drawn from the optimal adversarial distributions, which are calculated based on the testing dataset, rather than the images

of objects that are later fed to the network because the adversary does not have access to those images.

III. Optimal Adversarial Distributions

Consider the noise δ captured by the imager to be a random vector, whose elements are drawn from some independent distributions. We assign an independent Gaussian distribution to each RGB channel, i.e., $\delta = [\delta_R, \delta_G, \delta_B]$, where $\delta_{\{R,G,B\}} \in \mathbb{N}^{w \times h}$, and $\delta_R \sim \mathcal{N}(\mu_R, \sigma_R^2)$, $\delta_G \sim \mathcal{N}(\mu_G, \sigma_G^2)$, and $\delta_B \sim \mathcal{N}(\mu_B, \sigma_B^2)$. Here, $\mu_{\{R,G,B\}}$ and $\sigma_{\{R,G,B\}}$ are the means and the standard deviations of the three Gaussian distributions, respectively.

We measure the magnitude of δ using the expectation of its L_p norm, i.e., $\mathbb{E}(\|\delta\|_p) = \sqrt[p]{n}(|\mu_R| + |\mu_G| + |\mu_B|)$. Following the formulation in [3], we have our optimization problem

$$\underset{\mu_{\{R,G,B\}}, \sigma_{\{R,G,B\}}}{\text{minimize}} \quad \mathbb{E}(\|\delta\|_p) + \sigma + c \cdot \mathbb{E}(g(x')), \tag{1}$$

where $x' = [\tanh([x_R + \delta_R, x_G + \delta_G, x_B + \delta_B] + 1)]/2$ being a random vector, $\mathbb{E}(g(x')) = \max_{i \neq t}\{\mathbb{E}(Z_i(x'))\} - \mathbb{E}(Z_t(x'))$ measuring the likelihood of succeeding the attack, and c balances which objective is more important, making the noise small on average (by μ) and at peak (by σ), or making the attack succeed. Unfortunately, due to the complexity of neural networks computation, $\mathbb{E}(g(x'))$ is very hard to express analytically; we therefore use Monte Carlo methods to approximate it: $\hat{\mathbb{E}}(g(x')) = \sum_{j=1}^{r} \left[\max_{i \neq t} \left\{Z_i(x'_j)\right\} - Z_t(x'_j)\right]$, where r is the number of trials, and x'_j is the perturbed image at j-th trial.

We use the Adam Optimizer [5] to solve our problem. For every iteration the algorithm tunes μ and σ so that the noise drawn from $\mathcal{N}(\mu, \sigma^2)$ is more likely to fool the neural network to classify x' as the target class t.

Because each channel has noise added to it of a different mean, the resulting adversarial example x' looks like as though a color filter has been applied (see Sec. IV).

IV. Attack Implementation and Evaluation

The architecture of the targeted neural network was the same as the one in [3]: four convolutional layers followed by two fully-connected layers. We omit the details of the network due to the page limit. We implemented the network using TensorFlow and trained it with the distillation defense [6] using the CIFAR-10 dataset, and it achieved 80% accuracy.

To evaluate our attacks in the digital domain, for each pair of a benign image and a target class, we solved Eq. (1) for the noise distributions. We set $r = 100$ and $c = 100$. Resulting adversarial examples are shown in Fig. 2, where few of them were best-effort results (unsuccessful examples). Overall, the attack success rate was 83% among $10 \times 9 \times 10 = 900$ instances (there are ten classes and for each class, ten benign images were randomly chosen). This means optimal adversarial distributions are actually able to be used to generate successful adversarial examples.

For perception domain, we implemented our attacks in the real-world using a projector as the noise source. See Fig. 3b for the setup from the camera's point of view. We took Fig. 3a as a benign image and "frog" as the target label into Eq. (1)

Figure 2: Targeted adversarial examples. The image at i-th row and j-th column is an example that was perturbed from class i towards class j. When $i = j$, it is a benign image.

(a) A "ship" (b) Flare and "blooming" effects (c) A "frog"

Figure 3: (a) was displayed on a laptop shown in (b). (c) was cropped from (b).

and then we calculated the optimal random noise distribution of each RGB channel. We generated the noise (looked mostly green), using the projector to shine the noise directly to the camera. Due to the flare effect and the "blooming" effect (see Figure 3b), the benign image was overlapped with the green noise. As a result, Fig. 3a was misclassified as a "frog". We also did an experiment on traffic signs. Results (shown in the poster) showed that we were able to perturb a Pedestrian Crossing sign into a Signal Ahead sign.

V. Conclusion and Future Work

We proposed a class of adversarial examples that target the perception domain, where the adversary attacks the camera remotely with random noise so that the captured image is misclassified. Our remote attacks can be conducted with portable devices and they are not easy to be detected. In the future, we will systematically evaluate our physical implementation of the noise generator and also make the attack less conspicuous.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, 2015.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*. IEEE, 2017, pp. 39–57.

[4] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *CVPR*, 2018, pp. 1625–1634.

[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[6] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *S&P*. IEEE, 2016, pp. 582–597.