NaturalADV: An Exploratory Framework to Balance Adversarial Strength and Stealth in Autonomous Driving Environments

Meriel von Stein, LESS Lab @ University of Virginia

Motivation

Deep Neural Networks (DNNs) have become integral to various real-world autonomous mobile systems, from self-driving cars to food delivery robots. These systems rely on learned components, which are often susceptible to unexpected inputs from the set of realistic deployment conditions [1], resulting in system misbehaviors.

Figure 1. A DeepBillboard [2] in-situ patch attack.

However, current adversarial attack techniques often focus on maximizing the attack strength at the cost of naturalness, leading to examples that are easily detected by humans or deviate significantly from the expected input distribution. This trade-off between adversarial effectiveness and natural appearance presents a critical challenge in ensuring the robustness and reliability of DNNs in practical settings. This poster presents the Natural Adversarial DNN Validation (NaturalADV) framework for balancing the trade-off between adversarial strength and naturalness of a perturbation.

Perturbation Strength-Naturalness Tradeoff

Adversarial perturbations in deep neural networks (DNNs) often involve a fundamental tradeoff between strength and naturalness. The strength of an adversarial attack refers to how effectively the perturbation induces misclassification or otherwise disrupts the model's behavior. Naturalness [1] refers to how well the adversarial example aligns with the expected distribution of inputs – whether it "looks real" to humans or conforms to what the model would typically encounter under realistic deployment circumstances. Naturalness is often used interchangeably with stealthiness, which refers to the perceptual imperceptibility of the perturbation; a stealthy attack introduces minimal visual artifacts, making it hard for a human observer to detect any manipulation. Striking a balance between these two properties of strength and naturalness is critical, especially for real-world adversarial scenarios where overly artificial perturbations may be unrealistic, implausible, or easily detectable.

Perturbation Generation Loop

Figure 2 depicts a high-level overview of the generation loop for the NaturalADV framework. It takes in two images of the patch region of the deployment environment, one with the original adversarial perturbation known to have high adversarial strength under a given deployment [3] and a natural target patch, an image set `imgs` taken from an ADS navigating a driving environment without an adversarial patch, a navigation `DNN`, iterations of gradient ascent `iters`, a differentiable image similarity metric `similarity`, and `weights w1` for the image similarity loss and `w2` for the perturbed prediction loss between the original perturbation and the target patch.

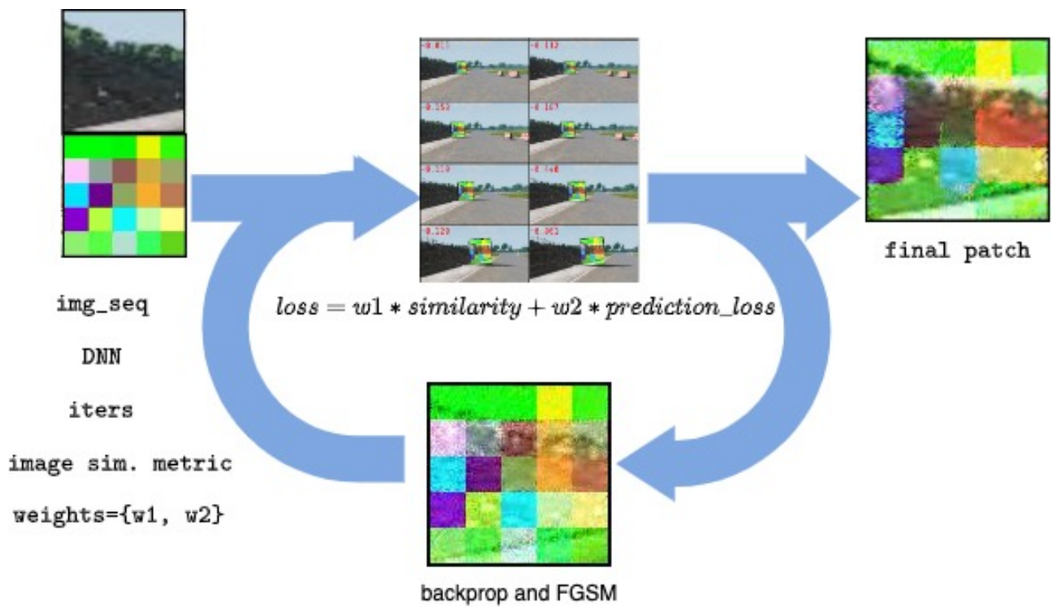

Figure 2. NaturalADV perturbation generation loop.

The framework alternates between calculating the loss function (see Equation 1) and backpropagation combined with Fast Gradient Sign Method (FGSM) to adjust the similarity of the patch to match the target image, while still retaining the adversarial strength of the original high-strength patch. Similarity and strength are prioritized according to parameterized `weights`. After `iters` loops, the generation loop exits and returns the final patch for injection into a driving environment.

NaturalADV Loss Function

The perturbation loop relies on a loss function for which preserving image similarity versus perturbation strength has been parameterized:

$$\begin{aligned} loss = w1 \times similarity(img_{natural}, img_{originial\_patch}) \\ +w2 \times L1(DNN(imgs + img_{originial\_patch}), DNN(imgs + img_{natural\_patch})) \end{aligned} \quad (1)$$

where $similarity$ is any differentiable image similarity metric (e.g. SSIM, German-McClure, Welsch, etc.), $img_{natural}$ is the natural patch we want the adversarial patch to resemble, $img_{originial\_peturbation}$ is the high-strength but unnatural-looking (or un-stealthy) original adversarial patch, $DNN(imgs + img_{originial\_peturbation}$ is the DNN prediction output for the original high strength perturbation, $DNN(imgs + img_{natural\_peturbation}$ is the DNN prediction output for the current version of the natural perturbation, and w_x are weights to prioritize image similarity or DNN prediction loss.

Experiments

Perturbation Strength

We explore a range of weights for similarity and prediction (see Equation 1) and report several performance metrics. Column 2 shows the resulting structural similarity index measure (SSIM) score when comparing the benign target patch and the generated adversarial patch. Column 3 shows the crash rate under simulated deployment like in Figure 1. Column 4 shows the average deviation in the vehicle's trajectory from the road center.

Weights (sim, pred)	SSIM Score	Crash Rate	Avg. traj. deviation
0.00, 1.00	0.24	53%	2.75m
0.10, 0.90	0.21	22%	2.45m
0.25, 0.75	0.33	18%	2.37m
0.50, 0.50	0.46	2%	2.31m
0.75, 0.25	0.51	0%	2.21m
0.90, 0.10	0.70	0%	2.24m
1.00, 0.00	1.00	0%	2.23m

Table 1. Performance metrics for a range of weights using NaturalADV.

Table 1 shows perturbation strength diminishes inversely to SSIM score, where the generated patch resembles more and more closely the benign target patch. However, the generated patch still retains the ability to crash the vehicle in deployment when image similarity loss and prediction loss are equally weighted.

Perturbation Naturalness

((a)) 0.00, 1.00 ((b)) 0.10, 0.90 ((c)) 0.25, 0.75 ((d)) 0.50, 0.50 ((e)) 0.90, 0.10 ((f)) 1.00, 0.00

Figure 3. Patch appearances from original perturbation to benign target patch

Naturalness is dictated by the choice of metric, in this study SSIM. SSIM is designed to compare two images in terms of luminance, contrast, and structure, or edge detection, and thus preserves colors of the original perturbation well into the $(0.90, 0.10)$ weighting.

NaturalADV Framework Contributions

NaturalADV can incorporate a number of differentiable naturalness metrics, works with various gradient traversal algorithms, and scales to attacks represented in multiple sensor readings. Our contributions are:

- a framework to balance the trade-off between adversarial strength and naturalness for in-situ adversarial patch attacks;
- a proof of concept study showing the naturalness-strength tradeoff for the motivating example; and
- an open-source repository with tool and data for reproducibility available at `https://github.com/MissMeriel/NaturalADV`.

References

[1] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021.

[2] H. Zhou, W. Li, Z. Kong, J. Guo, Y. Zhang, B. Yu, L. Zhang, and C. Liu, "Deepbillboard: Systematic physical-world testing of autonomous driving systems," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 347–358, 2020.

[3] M. von Stein, D. Shriver, and S. Elbaum, "Deepmaneuver: Adversarial test generation for trajectory manipulation of autonomous vehicles," *IEEE Transactions on Software Engineering*, 2023.