Stealthy Adversarial Examples in the (Simulated) Physical World

Weilin Xu[1], Cory Corenelius[1], Luis Murillo Rojas[2], Marius Arvinte[1], Sebastian Szyller[1], Alvaro Velasquez[3], Jason Martin*[4], and Nageen Himayat[1]

[1]*Intel Labs*
[2]*Intel Corporation*
[3]*University of Colorado Boulder*
[4]*HiddenLayer*

Abstract

While the first adversarial example in the digital domain against deep learning-based computer vision models promises hardly imperceptible perturbations to human eyes, it has been difficult to produce similar adversarial examples in the physical world. Existing physically realizable adversarial examples typically contain obvious or even strange visual patterns, because the attack algorithms must over-constrain perturbations or over-approximate the non-differentiable distortion functions in visual sensing systems. In this work, we make adversarial examples effective and even stealthy in the (simulated) physical world. We introduce a novel approach to overcome the obstacle of non-differentiability by applying exact non-differentiable distortions in the forward pass of backpropagation and using differentiable rendering in the backward pass, which enables fast generation of ℓ_p bounded adversarial examples despite the non-differentiable distortions. The experiments in CARLA, a photo-realistic urban driving simulator using the non-differentiable renderer of Unreal Engine 4, show that our method not only produces effective adversarial examples but also stealthy ones. We open-source our attack implementation and urge the community to re-evaluate the threat of adversarial examples in the physical world.

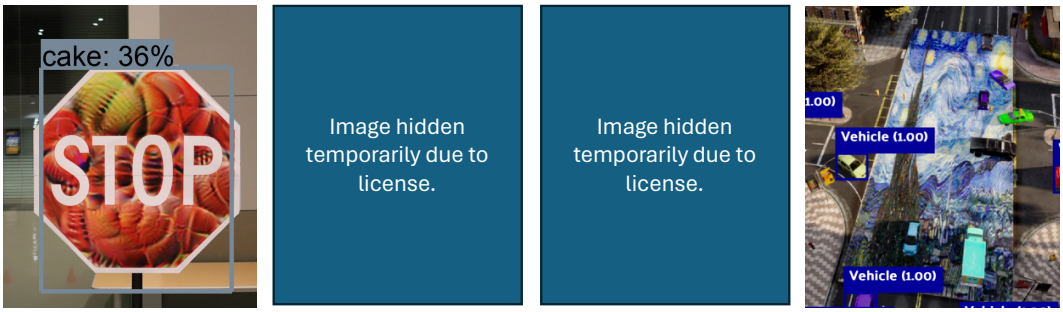
1 Introduction

Deep Neural Networks (DNNs) have been the de facto solutions to many computer vision tasks due to their exceptional accuracy compared with previous methods. However, DNNs are not perfect due to the weakness to adversarial examples. According to the original definition by Szegedy et al., "adversarial examples" are samples with *imperceptible* non-random perturbation which cause arbitrary changes to the network's prediction [20]. Though it is fairly easy to generate adversarial examples against any DNNs using gradient descent and back propagation— the same way as training DNNs, it has been rather difficult to launch such attacks against visual sensing systems in the physical world, because of the non-differentiable distortion functions in the imaging pipeline. This is why adversarial examples do not deter practitioners from deploying DNNs in security-sensitive scenarios, such as video surveillance systems and autonomous vehicles.

The goal of this work is to demonstrate that the threat of adversarial examples in the physical world is real. Users should re-evaluate the security implication to *Deep Neural Network* (DNN)-based visual sensing systems assuming that adversarial examples in the physical world are as effective as those in the digital domain.

*Jason Martin contributed to the work when he was at Intel Labs.

Preprint. Under review.



(a) ShapeShifter [7] (b) APRICOT [5] (c) Invisibility Cloak [21] (d) Our work

Figure 1: While most existing physically realizable adversarial examples exhibit strange visual patterns, our novel attack produces adversarial patches with *imperceptible* perturbations (See comparison with benign counterparts in Figure 2).

Most previous works on physically realizable adversarial examples loosen the definition of adversarial examples by allowing unbounded perturbations, so that perturbations have a better chance of surviving the non-differentiable distortions in the target system. Remedies like these bring about a side effect of obvious or even strange textures in the generated perturbations, making them easier to be spotted as anomalous as in Figure 1. In contrast, our work focuses on hardly *imperceptible* perturbations that are aligned with the original definition.

In this work, we develop a novel approach to generate adversarial examples against DNN-based visual sensing systems which distort images with non-differentiable functions, such as clipping of electrical signals caused by photodiodes or power amplifiers [6]. The key is to use non-differentiable components in the forward pass without any kind of approximation and use differentiable rendering in the backward pass to enable backpropagation.

Our experiments in the CARLA simulator show that the effectiveness of our method in the simulated physical world is comparable to that of classic attacks in the digital domain. Our method is so powerful that it even generates ℓ_p bounded adversarial examples with imperceptible perturbations that are effective in CARLA, as in Figure 2.

Contributions Our key contribution is proposing and evaluating a novel method to generate adversarial examples against DNN-based visual sensing systems that have non-differentiable distortion functions in the imaging pipeline. We instantiate the method in the CARLA urban driving simulator [9] and implement the attack algorithm in *Modular Adversarial Robustness Toolkit* (MART) [22]. Our experiments in CARLA show that the method produces adversarial examples that are effective in the simulated physical world.

2 Background

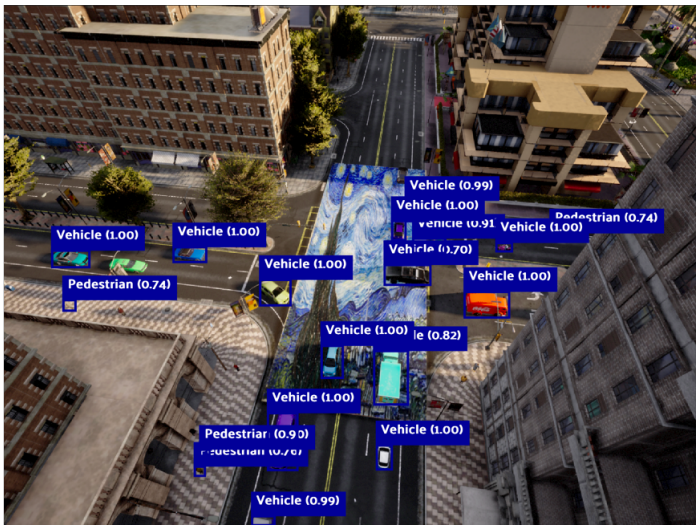
Adversarial examples represent a well-known weakness of neural network models that has not been fixed after many years of research [20]. By introducing unnoticeable perturbation to an image, an adversary can fool neural network models that are accurate on the benign (un-modified)3 input.

While it is trivial to generate ℓ_p bounded adversarial examples with imperceptible perturbation in the digital domain using gradient descent and backpropagation, such adversarial examples are not as effective in the physical space if we print them out, use a camera to capture them and feed the frames to the target model [13].

Adversarial examples in the digital domain do not transfer well to the physical world, because visual sensing systems distort images by passing through complex processing pipelines [6, 16] that many attack algorithms do not take into account. A straightforward solution is to model the distortion function of the printer (or screen) and the camera in the differentiable manner. But it requires non-trivial effort and it is usually deemed infeasible.

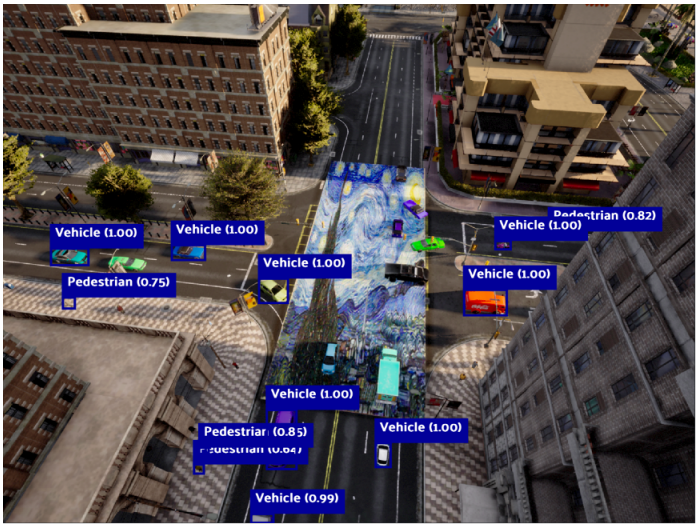
Instead, researchers have proposed various approaches in an attempt to overcome the hurdle of non-differentiability in generating adversarial examples in the physical world. We categorize such

(a) Benign patch



(b) Rendering benign patch in CARLA

(c) Adversarial patch $\ell_\infty = 10/255$



(d) Rendering adversarial patch in CARLA

Figure 2: Our method produces adversarial examples with barely imperceptible perturbations that survive non-differentiable distortions in CARLA. The adversarial patch in the background in Figure 2d makes all vehicles on top of it to be missed by the object detection model.

remedies into three groups: Perturbation Constraints, *Expectation over Transformation* (EoT) and Neural Rendering. Recent work on physically realizable adversarial examples often combines techniques from multiple categories [19, 23].

Perturbation Constraints. Printers or screens do not produce 100% accurate colors [3]. Even if they would do so, imaging systems (e.g., digital cameras) do not capture pixels perfectly. Recognizing the limitation of rendering and imaging pipelines, we may constrain adversarial examples in some way to lower the discrepancy between a digital image and the corresponding camera frame, so that the target model gets similar input as the adversary expects. In the very first work on physically realizable adversarial examples, Sharif et al. proposed several physical constraints on adversarial examples, including allowing only several printable colors, encouraging smooth textures with the total variance term, and using manual color management to calibrate the printing colors [17]. Such physical constraints enable adversarial eyeglasses frames that fool a face recognition system.

Expectation over Transformation (EoT). Although modeling the exact distortion functions in the non-differentiable components is non-trivial, we can make adversarial examples robust to distortions in the digital domain. As a result, the robustness to simulated distortions may generalize well to the actual distortions in the target visual sensing system. Athalye et al. proposed EoT to approximate unknown distortion functions with random image transformations, including rescaling, rotation, lightening, darkening, Gaussian noising, and translation etc [2]. Chen et al. later adapted EoT to attacking object detection models in the ShapeShifter work [7].

Neural Rendering. Since neural networks can approximate arbitrary functions, we may approximate the non-differentiable distortion functions using differentiable DNNs. Jan et al. proposed to train an image-to-image model that simulates the distortions of an imaging pipeline that consists of a printer and a camera [11]. The method can go a long way as the DNN community keeps improving the performance of image-to-image models.

However, as all methods add burden to the adversarial perturbation budget, it becomes difficult to generate ℓ_p bounded adversarial examples with imperceptible perturbations in the physical world. In fact, many attack algorithms loosen the conventional ℓ_p constraint on pixel values in favor of unbounded perturbations. As a result, existing adversarial examples that are effective in the physical world often contain obvious or even strange visual patterns as in Figure 1, which leads to many defense solutions dedicated to detecting such abnormal textures [14, 12].

3 Methodology

We introduce a novel attack method that overcomes non-differentiability in attacking DNN-based visual sensing systems and present an instance of such an attack in the CARLA environment.

3.1 Differentiable Approximation of Non-differentiable Sensing

We formalize a typical visual sensing system as follows: given a target model $f(\cdot)$ and a camera $g(\cdot)$, an adversary can arbitrarily perturb x, the digital texture of an object of interest, but it cannot change the surrounding environment e. The adversary uses a device $h(\cdot)$, either a printer or a screen, to realize the digital texture x. The whole visual sensing system can be expressed as

$$f(g(h(x), e)) \tag{1}$$

The adversary intends to influence the output of $f(\cdot)$ by changing x. If both $g(\cdot)$ and $h(\cdot)$ are continuous and differentiable, generating adversarial examples against the system should be as easy as an attack in the digital domain. However, the camera $g(\cdot)$ is a non-differentiable function that distorts captured frames due to the imperfect sensor and lens. The function of the display device $h(\cdot)$ is not differentiable either and it distorts x by presenting inaccurate pixel values. Thus the adversary can not backpropagate through $g(\cdot)$ and $h(\cdot)$ to get the gradient of x for generating adversarial examples.

A straightforward solution would be implementing $g(\cdot)$ and $h(\cdot)$ in a fully differentiable manner. But it is usually deemed too expensive or even infeasible.

Alternatively, we can replace the non-differentiable $g(h(x), e)$ with $g_d(x, e)$ such that $g_d(x, e)$ is differentiable with respect to x. We use a differentiable rendering engine $r(\cdot)$ and a stop-gradient

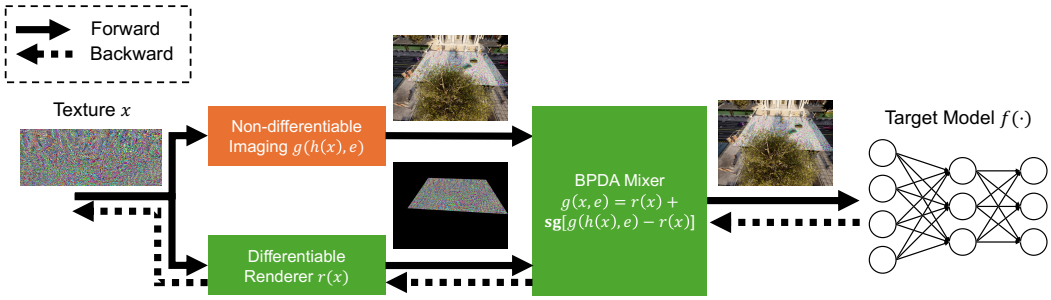


Figure 3: Our attack overcomes the non-differentiable obstacle by aligning the simplified differentiable rendered image with the more complex non-differentiable rendered image.

($\mathtt{sg}[\cdot]$) operation to construct a differentiable imaging pipeline as

$$g_d(x, e) = r(x) + \mathtt{sg}[g(h(x), e) - r(x)] \tag{2}$$

It is important to note that

1. $g_d(x, e)$ outputs the same values as $g(h(x), e)$ does in the forward pass, because $r(x)$ cancels out $-r(x)$ numerically.
2. $g_d(x, e)$ is differentiable with respect to x through differentiable rendering $r(x)$.
3. The differentiable renderer $r(x)$ ignores the surrounding environment e.
4. The difference between the non-differentiable $g(h(x), e)$ and differentiable $r(x)$ is detached as a constant for the autograd engine in the backward pass by the $\mathtt{sg}[\cdot]$ operation.

The effect of $g_d(x, e)$ is similar to the straight-through estimator [4] (a.k.a. *Backward Pass Differentiable Approximation* (BPDA) in the adversarial machine learning literature [1]), which uses the non-differentiable output in the forward pass, but an identity function in the backward pass to approximate the gradient. The difference is that in our backward pass, we need a more complex differentiable renderer $r(x)$ instead of an identity function. As long as $r(x)$ renders x at exactly the same location as in $g(h(h(x), e))$, we have the property $g(h(x), e) \approx r(x)$ for all pixels of x, thus $\nabla_x g(h(h(x), e)) \approx \nabla_x r(x)$. It does not matter to the adversary if $g_d(x, e)$ is differentiable with respect to e or not, as the adversary has no control over the surrounding environment. The gradient of e is not helpful to the adversary anyway.

3.2 Instance in CARLA

We depict one instance of Equation (2) in Figure 3, in which an adversary perturbs a rectangular patch to influence the model prediction in the CARLA simulator. CARLA uses the non-differentiable renderer of Unreal Engine 4 that supports complex lighting and material rendering. The differentiable renderer $r(x)$ in this case is implemented as a perspective transformation function that takes coordinates of the four vertices of the rectangle as parameters [2]. Even though the lights and shadows make some regions brighter or darker, the non-differentiable rendering of the texture is similar to that of differentiable rendering. Intuitively, if we make a pixel of x brighter, both render a brighter pixel, so the sign of the gradient never gets wrong. It would make a perfect gradient approximation if the adversary uses gradient sign instead of gradient to update perturbations. Even if the adversary has to approximate the magnitude of gradients with some error, the iterative updates often tolerate such errors in practice. We understand that some parts of the patch are blocked by other objects such as trees, vehicles or pedestrians in the scene, resulting in entirely different pixels in the two rendering modes. However, the gradient of those blocked pixels does not matter, since the adversary cannot influence the model prediction with those pixels whatsoever. We evaluate the attack in CARLA in Section 4.

[2] `https://pytorch.org/vision/main/generated/torchvision.transforms.functional.perspective.html`

4 Experiments

We use the CARLA driving simulator [9] to set up visual sensing systems for the evaluation. Even though CARLA is not a physical world environment, it is an advanced and photo-realistic simulation of the physical world. More importantly, CARLA shares one property with the physical world— the non-differentiable visual sensing system that is a big challenge to physically realizable adversarial examples. The simulation-based implementation also allows readers to fully reproduce our results in this paper.

4.1 Experimental Setup

Threat Model. We assume that an adversary has full knowledge of a visual sensing system, which comprises a camera and a DNN model. The adversary can arbitrarily change the texture of a green screen in the scene captured by the camera, but it cannot change other elements including the lighting condition and other objects. The objective of the adversary is to lower *mean Average Precision* (mAP) of the target object detection model.

Target Model. We use the baseline object detection model provided in ARMORY [18] as the target model. The Faster R-CNN model is trained on the MS-COCO dataset [15] and finetuned on the CARLA training dataset offered in ARMORY.

Dataset. We use a self-collected dataset that is similar to the DARPA GARD 7th evaluation dataset for object detection, which consists of 20 images with green screens that adversaries can perturb to influence the target model. Due to the lack of essential CARLA metadata in the released GARD dataset, we have to recreate the scenes of every image using the OSCAR Datagen Toolkit [8] so that we can render custom textures on the green screens in CARLA. We present all 20 CARLA images in Figure 4, which are similar to those in the official DARPA GARD 7th evaluation dataset.

Attacks. We implement the attacks in this paper using the MART [22] framework. Following the parameters of the example attack in ARMORY [3], we use the Adam optimizer to maximize the training loss for 500 iterations with the step size $12.75/255$. We compare two attacks in the experiment: one is the digital attack that does not take into account the non-differentiable rendering of CARLA; the other is our novel attack that uses Equation (2) to overcome the non-differentiable hurdle. We visualize both attacks in Figure 5.

4.2 Results

We present the results of two attacks in Table 1. Without adversarial examples, the target model achieves 67% mAP on both the benign frames and frames with green screens. The digital attack appears to be effective, as it significantly drops mAP of the target model to 8%. However, as we render the digital patches in CARLA to evaluate them in the simulated physical world, the mAP increases back to 59%. In contrast, the adversarial patches we generated using our novel attack are significantly more effective in lowering mAP when being evaluated in CARLA(16% compared to 59%). The small mAP gap, 16% over 8%, between our attack and the digital attack is not surprising, because our patch pixels have to obey the (simulated) optics laws in physics, leading to smaller perturbation budget to an adversary. For example, a realistic patch does not allow pure black pixels, while a digital patch could.

Table 1: Our novel attack produces effective adversarial patches in CARLA simulation. The results are measured on 20 self-collected test images.

Attack	**mAP**
Benign	0.67
Greenscreen	0.67
Digital Attack	0.08
Re-inserted to CARLA	0.59
Simulation-in-loop	0.16

4.3 Stealthy Adversarial Examples

Our attack is so powerful that it makes adversarial examples stealthy again in the (simulated) physical world. While the famous adversarial example of a panda is composed with invisible perturbations [10], most adversarial examples in the physical world contain obvious or even strange visual patterns, as a

[3] `https://github.com/twosixlabs/armory/blob/v0.19.2/scenario_configs/eval7/carla_overhead_object_detection/carla_obj_det_adversarialpatch_targeted_undefended.json`

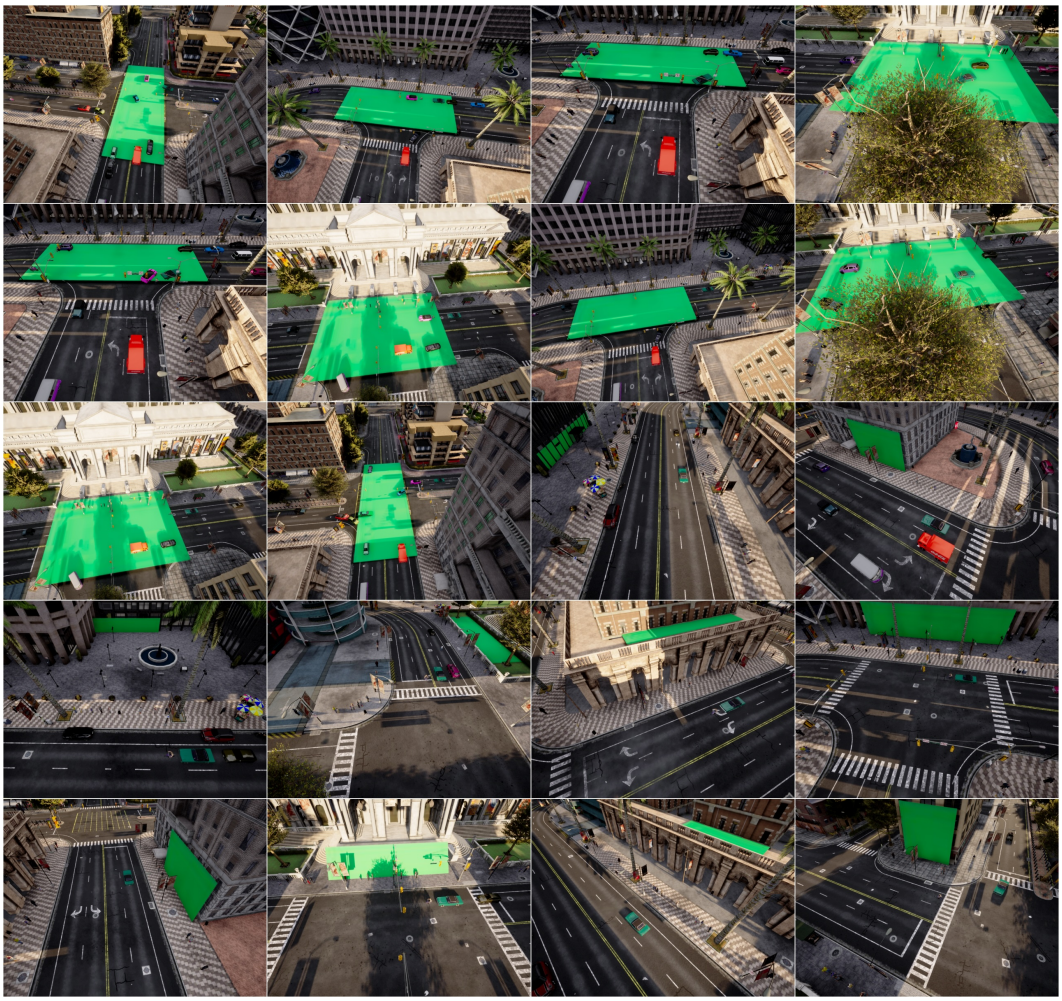


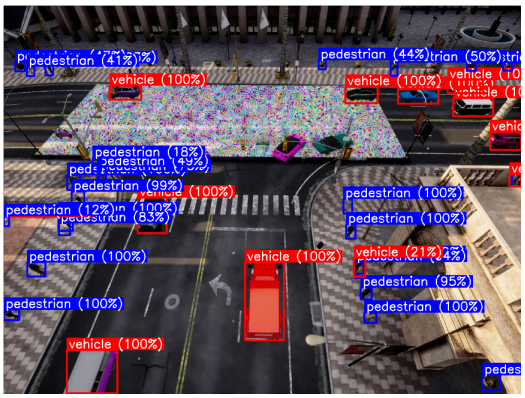
Figure 4: We recreated the scenes of all 20 images of the DARPA GARD 7th evaluation dataset for object detection, so that we can render adversarial patches on the green screens in CARLA.

result of over-approximating the image transformation in the non-differentiable imaging pipeline. Since our attack algorithm uses the exact transformations in non-differentiable rendering without any kind of approximation, the adversary can use the whole perturbation budget to fool the target model, yielding similar results in the realm of digital adversarial examples. In practice, we are able to produce physically realizable adversarial examples constrained with very small ℓ_p bounds, as shown in Figure 2c. We render both benign and adversarial patches in CARLA in Figure 2 for comparison. The adversarial patch in Figure 2d looks identical to the benign counterpart in Figure 2b, but it causes many hallucinations to the target object detection model.

5 Conclusion

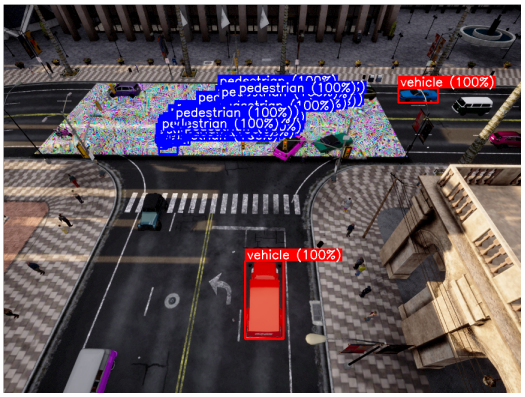
The effectiveness of our attack against DNN-based visual sensing systems with non-differentiable distortions raises an alarm that adversaries may be more capable than what was previously thought. We need to understand more about the technique, because adversaries may use it to attack visual sensing systems in the real world. In contrast to existing physically realizable adversarial examples, the adversary may be able to produce stealthy adversarial examples, making it harder to detect in the physical world. Even though we only demonstrate attacks with rectangular patches, it is mathematically and practically straightforward for adversaries to use the technique to generate adversarial camouflages, as long as they can align the texture with differentiable rendering during the

(a) A digital adversary generates adversarial examples while ignoring the CARLA distortions.



(b) CARLA rendering distorts the digital texture, destroying efficacy of the adversarial patch.

(c) Our attack generates adversarial patches with the non-differentiable distortions.



(d) CARLA rendering distorts the patch in the same way, but our adversarial patch remains effective.

Figure 5: While CARLA distorts a digitally composed adversarial patch, resulting in non-effective adversarial examples, our attack is aware of the CARLA distortions and remains effective if we use CARLA to render the adversarial patch.

generation process. It will be important to evaluate existing defense solutions against such attacks to better understand the landscape of the threat.

Acknowledgments and Disclosure of Funding

This work is partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001119S0026.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[3] Farhan A Baqai, J-H Lee, A Ufuk Agar, and Jan P Allebach. Digital color halftoning. *IEEE Signal Processing Magazine*, 22(1):87–96, 2005.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[5] Anneliese Braunegg, Amartya Chakraborty, Michael Krumdick, Nicole Lape, Sara Leary, Keith Manville, Elizabeth Merkhofer, Laura Strickhart, and Matthew Walmer. Apricot: A dataset of physical adversarial attacks on object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 35–50. Springer, 2020.

[6] Mark Buckler, Suren Jayasuriya, and Adrian Sampson. Reconfiguring the imaging pipeline for computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 975–984, 2017.

[7] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 52–68. Springer, 2019.

[8] Cory Cornelius, Luis Murillo, and Weilin Xu. Oscar datagen toolkit. `https://github.com/IntelLabs/OSCAR/tree/main/lib/oscar-datagen-toolkit`, 2023.

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[11] Steve TK Jan, Joseph Messou, Yen-Chen Lin, Jia-Bin Huang, and Gang Wang. Connecting the digital and physical world: Improving the robustness of adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 962–969, 2019.

[12] Melanie Jutras, Ethan Liang, Sara Leary, Chris Ward, and Keith Manville. Detecting physical adversarial patch attacks with object detectors. In *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7. IEEE, 2022.

[13] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[14] Bin Liang, Jiachun Li, and Jianjun Huang. We can always catch you: Detecting adversarial patched objects with or without signature. *arXiv preprint arXiv:2106.05261*, 2021.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[16] Ali Mosleh, Avinash Sharma, Emmanuel Onzon, Fahim Mannan, Nicolas Robidoux, and Felix Heide. Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7529–7538, 2020.

[17] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[18] David Slater and Lucas Cadalzo. armory, January 2023.

[19] Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. Dta: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15305–15314, 2022.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[21] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 1–17. Springer, 2020.

[22] Weilin Xu, Cory Cornelius, and Luis Murillo. Modular adversarial robustness toolkit. `https://github.com/IntelLabs/MART`, 2023.

[23] Jiawei Zhou, Linye Lyu, Daojing He, and Yu Li. Rauca: A novel physical adversarial attack on vehicle detectors via robust and accurate camouflage generation. *arXiv preprint arXiv:2402.15853*, 2024.