





How Many Cameras Do You Need? Adversarial Attacks and Countermeasures for Robust Perception in Autonomous Vehicles

Tu Anh Ngo^(✉) , Reuben Jon Chia, Jonathan Chan,
Nandish Chattopadhyay, and Anupam Chattopadhyay 

Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
{anhtu.ngo, anupam}@ntu.edu.sg,
{rchia013, jona0028, nandish001}@e.ntu.edu.sg

Abstract. Deep neural networks have been established by researchers to perform significantly better than prior algorithms in multiple domains, notably in computer vision. Naturally, this resulted in its deployment as a perception module in modern Autonomous Vehicle (AV) and in general for Advanced Driver Assistance Systems (ADAS). ADAS relies heavily on perception module, which harnesses various sensors such as camera, LiDAR, radar, ultrasonic sensor to make navigational decisions. By drawing from the adversarial attacks, which undermine a lot of machine learning applications, recent research shows that the AV perception modules are also vulnerable to adversarial attacks. Suggested countermeasures for these attacks include increasing the number of sensors, which incurs cost overhead and does not present any formal guarantee of protection. Hence, in this paper, we study the robustness and practicality of such a countermeasure. We demonstrate that it is still possible to spoof multiple cameras through adversarial object though, the attack success considerably reduces. Furthermore, the possibility of alternative countermeasures like dimensionality reduction and feature squeezing are investigated. Our study shows that these techniques, when applied together, significantly enhances the robustness of the AV perception system.

Keywords: ADAS · AV · Neural network · Adversarial attack · Adversarial defense

1 Introduction

Recent decades have witnessed a booming in the automotive industry, especially with major technological breakthroughs in autonomous driving. The level of automation in a vehicle has improved significantly, from manual operation to high level of automation. This is achieved mainly with the help of machine learning, which contributes to almost every modules of AV such as perception, localization, planning, prediction, etc. Perception is a fundamental element of

AVs, involving in most decisions made by other modules. In an AV’s perception, sensors like cameras and LiDARs gather information about the surrounding environment such as obstacles, pedestrians and traffic signs. One wrong information from the perception module can lead to consequentially wrong decisions from other modules, which can result in fatal outcomes. Thus, a considerable amount of research on state-of-the-art deep neural networks (DNNs) have been carried out since the introduction of AlexNet [17], winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012.

However, being equipped with state-of-the-art neural networks does not ensure a perception system that is resilient against adversaries. Extensive research is being done to identify various attack vectors in AV’s neural networks [6, 14, 25]. Many such attacks, however, target single perception source like single camera or single LiDAR. On the contrary, many commercial AVs provide multitude of sensors all working in conjunction [6]. With such a multi-sensor setup and a realistic assumption that not all the perception modules are attacked simultaneously, it is concluded in recent studies that multiple sensors present a robust defense against a determined attacker. Cao et al. [6] explored a very interesting way of attacking into both LiDAR and camera, using a 3D printable adversarial object. The authors also believe that using more cameras or LiDARs could improve the robustness of the perception model against this attack.

The growth in usage of multiple sensors can be accredited to the improved availability of public datasets published by major companies, such as [11], nuScenes [5], Argoverse [28], etc. The new public datasets provide a full 360° view of the surroundings, creating many overlapping field-of-views (FoV). With various viewing angles on a single object, it could increase the chance that an object can be detected by the model, like the side of a vehicle as compared to the front. An example of a production-grade AV being used on the road would be the Electric Car company, Tesla. Tesla utilizes a series of modern cameras in the Electric Vehicles for their Autonomous Driving (AD) capabilities [15].

In this paper, we investigate whether increasing the number of cameras helps AV against adversarial object. Furthermore, we look into a few simple countermeasures involving image feature manipulation such as dimensionality reduction and color depth reduction. The rest of the paper is organized as follows: Sect. 3 and 4 details the attack methodology and proposed countermeasures, respectively. Section 5 describes the experiments conducted. In Sect. 6, some limitations of the presented study are discussed, and conclusions are drawn in Sect. 7.

2 Background

2.1 Adversarial Attacks on Image Recognition

Traditional attacks on image recognition systems used strong extra sources of light to physically blind a camera [19, 21]. Recently, as deep learning models are becoming more powerful, research trends shifted to attacks on the DNNs of perception system. The pioneering works from Szegedy et al. [27] discovered

that state-of-the-art DNNs are susceptible to adversarial attacks. Since then, more researchers investigated adversarial attacks in computer vision domain. In 2017, researchers from Google used adversarial stickers called “Adversarial Patch” [4] with particular properties that can fool machine learning models. These “patches” can be attached to any objects on the street, e.g. road signs, to cause camera perception system to make wrong decisions. In that same year, Eykholt et al. [10] were able to generate robust adversarial perturbations in the forms of only black and white stickers attached on stop signs. This attack achieved high efficacy in both image and video sign classification tasks.

The higher level of automation in self-driving car leads to the use of multiple kinds of sensor. Many AV makers nowadays use both cameras and LiDARs for perception systems, adding more robustness to the object detection performance. Many researchers have studied the vulnerability of LiDAR-based object detectors to 3D adversarial objects. However, there were not a lot of such studies done on the effect of 3D adversarial objects to camera-based object detectors until 2021. Abdelfattah et al. [1] proposed a kind of attack that when they place an adversarial object on top of a car, that car evades being detected by both LiDAR-based detector and camera-based detector. Another work from Cao et al. [6] involves generating a 3D printable adversarial object that can deceive LiDAR-based and camera-based perception models, causing vehicle crashing into it. In most of these prior works, a common countermeasure suggestion is to increase the number of cameras for detection. However, the question remains is whether that suffices as a countermeasure and if yes, how many cameras do we need?

2.2 Motivation

The idea of fooling LiDAR-camera perception model with adversarial 3D object [6] is recent and is a very active area of research. We try to find out whether such kind of adversarial object is still effectively hidden from vehicle's perception system if we use more sources of sensing and manipulate input's features. In our study, we make use of multi-camera setup with overlapping FoVs. One reason to use multiple cameras is that cameras are much more budget-friendly than LiDARs. Furthermore, when an object appears in different camera views, there are distortions in the textures such as color and lighting, which might affect the attack efficacy. Using camera images also allows alternative countermeasures such as feature squeezing and dimension reduction, which we also study in this work.

2.3 Contributions

In this paper, we study the robustness of AV's camera perception model in the event of adversarial attacks. Then, we propose some countermeasures in order to prevent AV's camera perception model from being deceived by 3D adversarial objects. In summary, this work makes the following contributions:

- Studying the vulnerability of multi-camera system to 3D adversarial objects.
- Applying dimensionality reduction [7] and Feature Squeezing [29] to camera images, as potential countermeasures.
- Fusing the above techniques into one unified pipeline for robust countermeasure.

3 Spoofing Multiple Cameras with Overlapping FOV

We use the original attack idea from Cao et al. [6] and extend it to check if it is possible to spoof the perception module from various angles. The corresponding object generation procedure is an optimization process, which is briefly explained in the following subsections for completeness. Interested readers can refer to the detailed methodology in [6]. The goal of this attack is to create an object that is invisible to perception model, which is visualized in Fig. 1.



Fig. 1. Attack goal is to create an adversarial object that is invisible to camera model

3.1 Object Detection Output

Popular deep learning-based 2D object detectors can be classified into two categories: two-stage and one-stage detectors. For two-stage detectors, eminent networks are region-based detectors such as RCNN [13] and its more efficient variants, Fast RCNN [12] and Faster RCNN [23]. The two stages of these algorithms can be divided into region proposal and object detection with bounding-box regression. Two-stage detectors have good localization and object recognition performance. However, regarding inference speed, one-stage detectors clearly

outperform the two-stage counterparts. Some of the most prominent one-stage detectors include YOLO and SSD. One-stage detectors jointly detect and localize using one unified neural network, without the region proposal stage.

Due to their simplicity, one-stage detectors are suitable to be used in real-time applications. Over the past years, recent improvements have enhanced one-stage detectors’ performance, which makes them superior to two-stage ones in terms of speed while preserving respectable accuracy. Some popular open-source autonomous driving platforms employ one-stage detectors in their perception modules, for example, Autoware [16] use YOLOv3 for their camera perception, and Baidu Apollo [3] also utilizes the 3D version of YOLO for the same purpose.

As this attack targets YOLOv3 for camera models, we review a bit on its output here. Given an image, YOLOv3 runs a single CNN to detect objects at three different scale of the original image, aiming to handle small, medium and big objects. At each scale, image is divided into $S \times S$ grid cells. And each cell makes prediction for B different anchor boxes, whereas every box’s prediction has $5 + C$ elements, representing:

- 4 values for box center offsets and width/height scales (x, y, w, h) .
- 1 value for box confidence/objectness score P_0 .
- C values for class scores P_1, P_2, \dots, P_C .

Therefore, at every scale, the prediction’s output has the shape $(S, S, B \times (5 + C))$. For YOLOv3, $B = 3$ because it uses 3 anchor boxes per scale. The attack in [6] adds perturbation to the object’s shape so as to minimize the box confidence score P_0 in accordance with it, hence the object’s disappearance from the camera object detector.

3.2 Formulation of Attack Objective

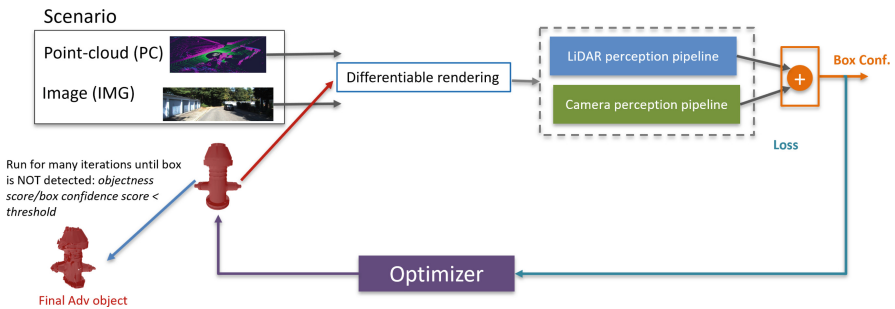


Fig. 2. Attack overview

Figure 2 visualizes the fundamental attack flow. An object is represented by its face-vertex meshes $(v - f)$. Let S denote the benign object and S^a the generated adversarial object. Therefore, the objective of the optimization process is to

change the position of the object’s vertices to minimize the box confidence to less than a threshold for it to be detected. The objective function is:

$$J = \mathcal{L}_a(S^a, \mathcal{R}^l, \mathcal{R}^c, \mathcal{P}, \mathcal{M}) + \lambda \cdot \mathcal{L}_r(S^a, S)$$

Hence, the optimization problem is:

$$\min_{S^a} J \tag{1}$$

subject to:

$$\Delta(S^a, S) \leq \epsilon \tag{2}$$

in which \mathcal{R}^l , \mathcal{R}^c are differentiable rendering functions for LiDAR and camera respectively, \mathcal{P} is the pre-processing approximation function and \mathcal{M} is the Multi-Sensor Fusion algorithm. The total loss J is the weighted sum of the two losses: adversarial loss \mathcal{L}_a for achieving attack goal, or to minimize the bounding box’s confidence value mentioned in Sect. 3.1, and realizability loss \mathcal{L}_r for improving surface smoothness, which is useful for 3D-printing.

Equation 1 is a constrained optimization problem, to solve it, Cao et al. [6] uses Projected Gradient Descent (PGD). The optimal value for this problem is achieved by optimizing the shape of the adversarial object S^a , more specifically by changing its vertices’ position. The constraint Eq. 2 is to ensure that S^a still has a recognizable shape to human’s eye and does not deviate too much from the original object S .

3.3 Robust Adversarial Object Generation

To improve robustness for this attack, it is necessary that the model can be fooled from various angles and distances. Cao et al. [6] apply Expectation over Transformation [2]. Equation 1 becomes

$$\min_{S^a} \mathbb{E}_{t \sim T} J$$

in which T is a set of random 3D transformation to S^a , including rotation and position shifting.

In [6], the authors slightly shift the object’s yaw angles to 5° , 10° , 15° . However, we could not find the EoT implementation from their public source code. Hence, we implement the EoT concept from scratch. First, we render the benign object in front of one front-center camera image. Let (x, y, ψ) be a set containing the distance between the object and the vehicle, the object’s horizontal distance and the angle of the object’s yaw rotation, respectively. In every iteration, we generate five random sets of changes $\{(\Delta x_i, \Delta y_i, \Delta \psi_i) | i \in \mathbb{N}, i \in [1, 5]\}$ that are applied to the object’s original position, resulting in five positions $\{(x + \Delta x_i, y + \Delta y_i, \psi + \Delta \psi_i) | i \in \mathbb{N}, i \in [1, 5]\}$. We select a wider range for yaw rotation changes since we want to produce a robust attack against multiple cameras, specifically $-40^\circ \leq \Delta \psi_i \leq 40^\circ$.

3.4 Spoofing Multiple Cameras

It is quite challenging for an adversary to fool the camera model from various viewing angles. In [20], the author demonstrated that a stop sign cannot consistently fool the camera model if it is viewed from various angles. However, with the use of EoT, the attack robustness is improved significantly. To check the attack efficacy, we randomly select 100 frames from Argoverse dataset. In each frame, we place the object at 3 m / 4 m / 5 m / 6 m in front of the front center camera and 0 m / 1 m to the right, hence a total of 800 ($100 \times 4 \times 2$) scenarios. We calculate the attack success rate (ASR) over all scenarios. We also experiment with the benign case in which we render the benign fire hydrant at the same positions as in the adversarial case. Then, we evaluate the benign detection rate (BDR) for the fire hydrant. In the following evaluations, a good result is the one with high benign detection rate and low attack success rate.

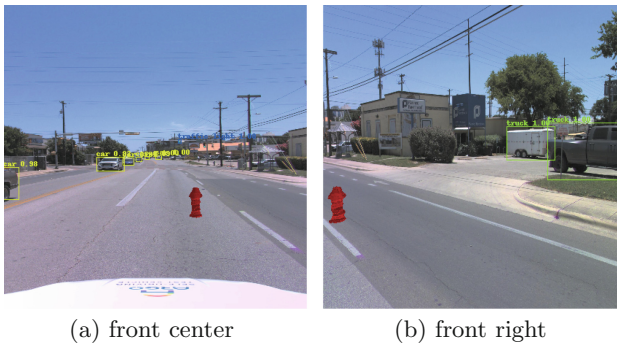


Fig. 3. Multi-cam setup is more robust but not sufficient: in some scenarios adversarial object can fool both cameras

Table 1. Attack evaluation on multi-cam setup

Cam setup	Benign det. rate (%)	Attack success rate (%)
Front center	75.75	78
Front center, Front right	98.38	43.75

Table 1 shows that using multiple cameras with overlapping FoV is more robust than just relying on one camera. We think it is still not enough to guard the camera model from being fooled. Figure 3 shows a scenario when both front center and front right camera cannot detect the fire hydrant. Therefore, we explore a few additional countermeasures in the next section.

4 Additional Countermeasures

In above section we demonstrate that fusing multiple cameras with overlapping FoVs improve system’s robustness, however, there are rooms for improvement. In this section, we discuss a couple of orthogonal countermeasures namely, dimensionality reduction and feature squeezing. We focus on manipulating image feature such as reducing image’s dimension or color, since these solutions were shown effective against adversarial examples in the literature. Furthermore, these countermeasures modify input features, which directly improves data bandwidth/storage and more straightforward than modifying the neural network architecture.

4.1 Dimensionality Reduction

This defense is inspired by the effect of the *curse of dimensionality*, which is one of the key causes facilitating the creation of adversarial examples. In [7], dimension reduction is demonstrated effective against adversarial objects, especially in classification problem. This has not been tested in object detection task, specifically when adversarial examples are to affect the bounding box confidence score. Since this can be a potential countermeasure boosting the perception module robustness, we applied the dimensionality reduction flow to camera images and studied its efficacy.

4.2 Feature Squeezing: Color Depth Reduction

There is little research on the effects of color to deep learning models. In [9], color quantization, which reduces color depth, is shown to affect the performance of convolutional neural networks. One hypothesis is color distortions affect the way neural networks perceive the input, due to the shift in image distribution. Indeed, according to [29], a neural network perceives the input space as continuous due to its differentiable manner. However, computers only support discrete representation of data. A digital image is represented by a pixel array, where each pixel is represented by numbers as a color code. Color bit depth is a feature in image representation that might affect the performance of a neural network. Therefore, we consider of color depth reduction as a feasible countermeasure mitigating the effect of adversarial examples. In general, color depth reduction is bracketed within a family of countermeasures termed as feature squeezing [29].

5 Experiments

5.1 Dataset

Due to the lack of real-hardware setup, we make use of readily available datasets. We choose Argo AI’s Argoverse 2 dataset [28], which is both open-source and provided by reliable institutions for our experiment. We use the *Sensor Dataset*

from Argoverse 2, which consists of 1,000 scenarios from 7 ring cameras, 2 stereo cameras and 2 LiDARs. One notable feature from Argoverse dataset is that each camera has overlapping FoV with its nearby camera. The overlapping areas and the position shift between two neighbor cameras are big enough to make two images disparate, which facilitates object detection from multiple viewing angles. As visualized in Fig. 4, the ring front left camera and the ring front right camera have significant overlapping FoVs with the ring front center camera. We also considered other well-known datasets such as KITTI [11], Waymo Open Dataset [26] and nuScenes [5]. However, there are some disadvantages of camera features in these datasets that do not suit our approach. For example, KITTI provides camera images with very limited position shifts, Waymo Open Dataset and nuScenes do not really provide camera images with overlapping FoV.

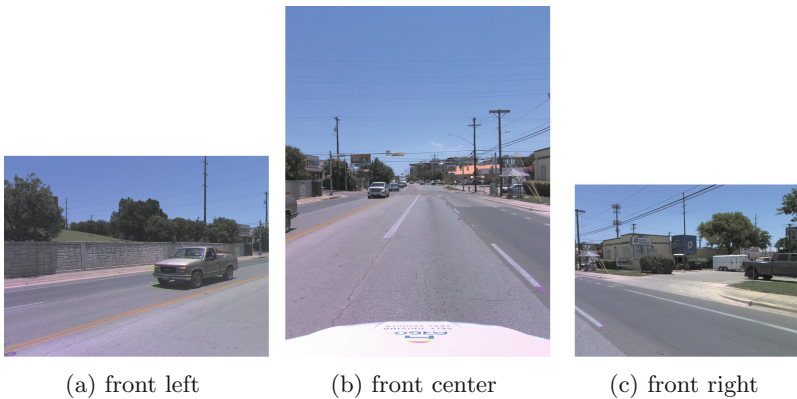


Fig. 4. Views from Argoverse ring front cameras

5.2 Choice of Objects

The first step is to pick a 3D benign object that can be fed into our optimization pipeline. Since we want to evaluate the object detection performance of one particular model, we have to use the objects that appear in the training set on which the model is trained. Here we evaluate our attack on YOLOv3 [22], which is pre-trained on COCO dataset [18]. We prefer to choose objects with not too complex texture and pretty symmetrical shape. There are quite a lot of websites that provide 3D object models, such as <https://free3d.com>, which has both free and paid 3D objects.

After obtaining a 3D object, we slightly process it using Blender [8], an open-source 3D graphics software.

5.3 Experimental Setup

Perception Models. This paper is inspired by the work from [6], which focuses on white-box attack. The targeted object detection models we choose are Baidu Apollo [3] for LiDAR and YOLOv3 [22] for cameras, the same as in original work [6]. Baidu Apollo is one of the most prominent open-source AV platforms and YOLOv3 is a popular real-time 2D object detector, which is still included in open-source AV platforms such as Autoware.AI [16] and Baidu Apollo [3]. In this study, our focus is on the vulnerability of multi-camera system, hence we use Baidu Apollo v2.5 instead of more recent versions for the sake of better memory usage. This is because the images and 3D point clouds in Argoverse 2 dataset are much more detailed than those in KITTI, therefore, we need to utilize our limited resources better.

Object Rendering and Placement. We experiment with attacking into the ring front center and ring front right cameras using the Argoverse 2 Sensor Dataset, as object can solely be visible to two cameras with overlapping FoVs at a time. We do not make use of scenes from the two stereo cameras, as there is no significant distinction between them. We render the object so that it appears in front of the ring front center and ring front right camera. As the color of an object also affects the detection performance, we mimic the typical color of real fire hydrants, which is mostly red.

5.4 Evaluation

As mentioned in Sect. 3, we selected 100 frames from the Argoverse 2 Sensor Dataset in which there are no objects with the same type as the injected object and rendered it to the aforementioned positions. There are a total of 800 scenarios.

Dimensionality Reduction: One popular method to reduce dimension is Singular Value Decomposition (SVD). From Chart 5, it can be observed that dimensionality reduction does not help much in guarding the model against adversarial attack. With less singular values, the model fails to recognize not only adversarial object, but also the benign one. Keeping just a small number of singular values drastically lowers the detection performance on both adversarial and benign objects. Retaining more singular values is safer for detection performance, however, it is still not useful against adversarial objects.

Color Depth Reduction: We use color quantization technique to reduce a 24-bit image to 8-bit image. The results are consistent for the reduction of various number of colors. From Chart 6, it is obvious that color quantization does indeed resist against adversarial objects, to some extent. Note that if the number of

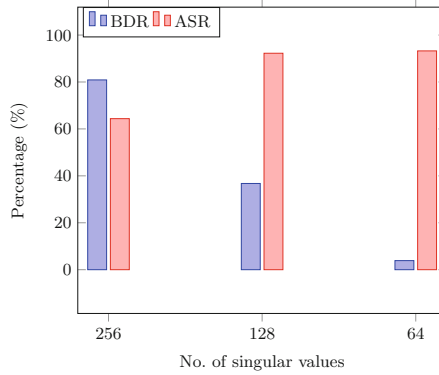


Fig. 5. Dimensionality reduction using SVD

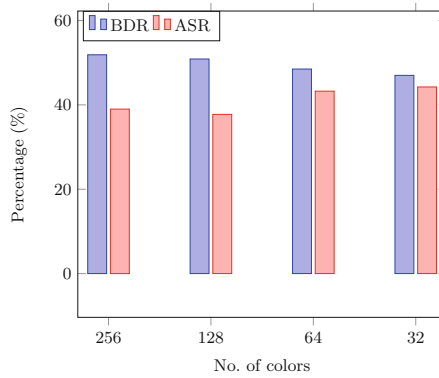


Fig. 6. Color depth reduction

colors is drastically reduced, then the object detection performance also drops accordingly. Hence, a hybrid approach of combining both 24-bit and 8-bit image is adopted (Table 2).

Table 2. Using both color depth reduction and multiple-camera system

Color depth	Benign det. rate (%)		Attack success rate (%)	
	Single-cam	Multi-cam	Single-cam	Multi-cam
24-bit (orig.)	75.75	98.38	78	43.75
8-bit	51.88	84.38	39	17.6
24-bit, 8-bit	78.75	99	37.25	13

A Unified Countermeasure Pipeline: Due to the low effectiveness of dimension reduction, we only combine color depth reduction and multi-camera setup

as one unified countermeasure. One downside of YOLOv3 is that it does not perform well on small objects. In our study, the farther the object’s distance to the vehicle is, the higher chance it is not detected by YOLOv3. This is the reason why in the benign case, there are scenarios where the model misses the fire hydrant, which results in a detection rate of merely 75.75%. Regarding color depth reduction, our study shows that it can mitigate the ASR in adversarial case. However, for benign fire hydrant, the detection performance drops dramatically to 51.88% if we only use the 8-bit images. We decide to fuse the original image (24-bit) and the 8-bit one together: whenever there is a detection happens in either image - it is considered a true detection. Regarding the multiple-camera setup, we find it more robust to guard the camera model than the single camera setup. Our results show that combining multiple-camera setup and color depth reduction technique together leads to a much more robust camera perception system, results in 99% benign detection rate and just 13% attack success rate in the adversarial case.

6 Limitations

Physical-World and Simulated Experiment. In this work, we extend the original work [6] and use multi-camera perception system as an attack vector as well as a feasible defense. One major drawback of our study is that we did not try out our concept on a real AV in the physical world due to cost concerns. Furthermore, we did not have the chance to experiment with AV simulators such as LGSVL [24] due to limited time, and due to LG’s announcement that they will suspend active development of SVL Simulator from 2022.

Multi-camera Object Projection. In Argoverse 2 Sensor Dataset, like other public datasets, all the calibration parameters and matrices are provided along the data itself. When we render the object with 3D information into 2D images from the dataset, we have to make use of the calibration matrices. We observed that when projecting the object onto side cameras, the final image might not completely reflect the true position of the object. In our belief, it is likely because there are some auxiliary parameters that we did not take into account or there are some misalignment in the cross-camera projection. This flaw does not affect the experiment, at large; nevertheless, it is still worth mentioning as we believe this projection can be improved for the sake of precision.

7 Conclusions

This paper demonstrates our study on two defenses against 3D adversarial object. Even though this attack originally aims to fool both LiDARs and cameras, we focus on defending camera model since a robust camera model leads to a robust perception system in general. Our study shows that feature squeezing methods such as color depth reduction alleviates the attack efficacy, however, it

increases the risk of model cannot perform well on other objects. If we also leverage original images, the results are promising. In terms of dimensionality reduction technique, we find it ineffective in our study. Turning to multiple-camera setup, this paper shows that using multiple cameras with overlapping FoVs is more robust compare to the single-camera setup. Furthermore, this setup is also budget-friendly, unlike LiDARs, which are prohibitively expensive. Leveraging color depth reduction and multiple-camera setup at the same time tremendously diminishes attack success rate, from 78% down to only 13%, according to our experiments. Considering the safety of AV perception models, we hope our contributions pave the way for the development of effective and economical defenses.

Acknowledgements. This research was supported by Desay SV Automotive Singapore, as part of NTU-Desay Collaboration project.

References

1. Abdelfattah, M., Yuan, K., Wang, Z.J., Ward, R.: Adversarial attacks on camera-lidar models for 3D car detection (2021). <https://doi.org/10.48550/ARXIV.2103.09448>, <https://arxiv.org/abs/2103.09448>
2. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: International Conference on Machine Learning, pp. 284–293. PMLR (2018)
3. Baidu: Apollo: open source autonomous driving. <https://github.com/ApolloAuto/apollo>
4. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial patch (2017). <https://doi.org/10.48550/ARXIV.1712.09665>, <https://arxiv.org/abs/1712.09665>
5. Caesar, H., et al.: nuScenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
6. Cao, Y., et al.: Invisible for both camera and LiDAR: security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 176–194 (2021). <https://doi.org/10.1109/SP40001.2021.00076>
7. Chattopadhyay, N., Chatterjee, S., Chattopadhyay, A.: Robustness against adversarial attacks using dimensionality. In: Batina, L., Picek, S., Mondal, M. (eds.) SPACE 2021. LNCS, vol. 13162, pp. 226–241. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95085-9_12
8. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018). <http://www.blender.org>
9. De, K., Pedersen, M.: Impact of colour on robustness of deep neural networks. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 21–30 (2021). <https://doi.org/10.1109/ICCVW54120.2021.00009>
10. Eykholt, K., et al.: Robust physical-world attacks on deep learning models (2017). <https://doi.org/10.48550/ARXIV.1707.08945>, <https://arxiv.org/abs/1707.08945>
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics: the KITTI dataset. *Int. J. Robot. Res. (IJRR)* **32**(11), 1231–1237 (2013)
12. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). <https://doi.org/10.1109/ICCV.2015.169>

13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014). <https://doi.org/10.1109/CVPR.2014.81>
14. Hallyburton, R.S., Liu, Y., Cao, Y., Mao, Z.M., Pajic, M.: Security analysis of camera-lidar fusion against black-box attacks on autonomous vehicles. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 1903–1920. USENIX Association, Boston, MA (2022). <https://www.usenix.org/conference/usenixsecurity22/presentation/hallyburton>
15. Ingle, S., Phute, M.: Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *Int. Res. J. Eng. Technol.* **3**(9), 369–372 (2016)
16. Kato, S., et al.: Autoware on board: enabling autonomous vehicles with embedded systems. In: Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2018, pp. 287–296. IEEE Press (2018). <https://doi.org/10.1109/ICCPS.2018.00035>, <https://doi.org/remotexs.ntu.edu.sg/10.1109/ICCPS.2018.00035>,
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, pp. 1097–1105. NIPS 2012, Curran Associates Inc., Red Hook, NY, USA (2012)
18. Lin, T.Y., et al.: Microsoft COCO: common objects in context (2014). <https://doi.org/10.48550/ARXIV.1405.0312>, <https://arxiv.org/abs/1405.0312>
19. Liu, J., Yan, C., Xu, W.: Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicles. DEF CON (2016). <https://doi.org/10.5446/36252> Accessed 22 Mar 2022
20. Lu, J., Sibai, H., Fabry, E., Forsyth, D.A.: No need to worry about adversarial examples in object detection in autonomous vehicles. CoRR abs/1707.03501 (2017). <http://arxiv.org/abs/1707.03501>
21. Petit, J., Stottelaar, B., Feiri, M., Kargl, F.: Remote attacks on automated vehicles sensors: experiments on camera and lidar. In: Black Hat Europe (2015). <https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp1.pdf>
22. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. CoRR abs/1804.02767 (2018). <http://arxiv.org/abs/1804.02767>
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, pp. 91–99. NIPS 2015, MIT Press, Cambridge, MA, USA (2015)
24. Rong, G., et al.: LGSVL simulator: a high fidelity simulator for autonomous driving. CoRR abs/2005.03778 (2020). <https://arxiv.org/abs/2005.03778>
25. Sun, J., Cao, Y., Chen, Q.A., Mao, Z.M.: Towards robust lidar-based perception in autonomous driving: general black-box adversarial sensor attack and countermeasures. In: 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, pp. 877–894 (2020). <https://www.usenix.org/conference/usenixsecurity20/presentation/sun>
26. Sun, P., et al.: Scalability in perception for autonomous driving: WAYMO open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
27. Szegedy, C., et al.: Intriguing properties of neural networks (2013). <https://doi.org/10.48550/ARXIV.1312.6199>, <https://arxiv.org/abs/1312.6199>

28. Wilson, B., et al.: Argoverse 2: next generation datasets for self-driving perception and forecasting. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021) (2021)
29. Xu, W., Evans, D., Qi, Y.: Feature squeezing: detecting adversarial examples in deep neural networks. In: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, pp. 18–21. The Internet Society (2018). http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018.03A-4_Xu-paper.pdf