Modelling and Study of Adversarial Attack Arising from Deceiving Perception in Car Autopilot

Neel Joshi, Navodit Chandra, Aishwarya Ravi

Carnegie Mellon University
{ndjoshi, navoditc, aravi2}@andrew.cmu.edu

Abstract

Safety of autonomous vehicles has been a matter concern lately. The autopilot system is heavily dependent on AI algorithms, from perception to control. Such systems are vulnerable to adversarial attacks arising from both random as well as intentional practices. This work models a real-life safety-critical scenario, where the moon is perceived as a traffic light, with the help of Digital Twin technology, such that car autopilots can be trained faster using these virtually generated scenarios. We create an end-to-end model starting from sensory perception to generating augmented scenarios based on original scenarios. We also show that majority of such safety-critical scenarios can be easily avoided. For this we demonstrate the efficacy of a real-time solution and a definitive solution.

1 Introduction

Existing work detailing safety-critical scenario generation has been discussed in [1] which aims to provide a detailed taxonomy of literature and classify the algorithms used as data-driven generation, adversarial generation, and knowledge-based generation. The driving scenario could be divided into static and dynamic content forms as per the work done in [2] in which any kind of static objects such as traffic stops and the geometry of the road would constitute a static part of the scenario which would be creating a dynamic interaction with scenarios such as traffic.

Sampling of the data set was implemented by replaying the data through virtual simulations such as in [3] or by random perturbation using lidar models such as in [4]. Another method also exists for sampling which is using clustering [5] which is particularly useful for primitive and traffic representation. For the incorporation of adversarial perturbation in the dataset, static scenarios are generated from object detection and segmentation [6] and the motion of every object for sequential testing, constituting a dynamic scenario using point cloud [6] .

Adversarial generation aimed to test out edge case simulations for multi modal density models to discover any kind of model weakness [7] during adversarial training. Adversarial policies were investigated usually in the form of a Reinforcement learning algorithm [8] with work done in implementing heuristic approaches [9] or implementing piecewise linear models [10] for accelerated evaluation of policies.

2 Motivation

A thought provoking real-life incident which took place in July of 2021 built the inspiration for this project [11]. An autonomous vehicle (AV) owner was driving along a long road in the evening with the autopilot control turned on. As the day progressed, the moon rose on the horizon right in front of the car. The freshly arose moon along the skyline had a yellowish hint to its appearance, which from a distance could be mistakenly related to a bright yellow source in the sky. The autopilot, in fact, did perceive the moon as a yellow traffic light in front of it and started to slow down at once. The car kept slowing down till the moon passed over its perception domain.

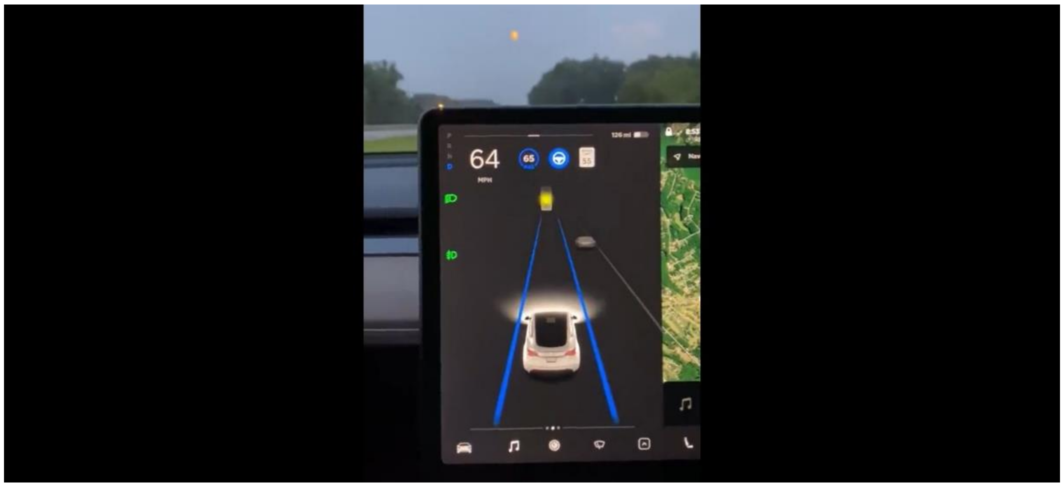


Figure 1: Tesla autopilot detecting the moon as yellow traffic light [11]

The yellow tint of the moon could be related to the wildfire smoke in the atmosphere over parts of the US [12]. So we may assume that such incidents are edge cases happening with car AI. However, the sudden slowdown of the car on a long road could very well have invited safety-critical situations, as an example, a rear-end collision with an unaware driver following behind.

The trustworthiness of driverless vehicles has been a matter of concern due to recent incidents giving rise to safety-critical scenarios, as mentioned above. Our project is about modelling such safety-critical scenarios, which are also rare chances for real-life driving experiences. We resort to Digital Twin technology for the modelling so that we can study the effect of such adverse events on the car AI and suggest viable solution to make the AI more safe and trustworthy.

This work takes into consideration above particular scenario as reported in [11], [12] and tries to reconstruct it for the first time using open source Digital Twin simulation package. We demonstrate how to build a Digital Twin simulation framework from scratch, starting from rare scenario generation, data analytics, to suggesting practically applicable solutions such that a car autopilot could be trained on such scenarios so as to avoid similar events in the future.

3 Scenario Generation

The architecture of a car autopilot is complex in nature, with different modules interacting with other modules in real time. As we did not have the privilege of accessing or replicating a real car autopilot system; we decided to construct our own framework. A schematic of the framework can be referred from Figure 2. A clean observation from the sensor (here, a RGB camera) is input to the perception module, which is modelled using a Convolutional Neural Network (CNN) ResNet18 as a simple image classifier. The output from this module consists of the class index, class name and confidence provided by the CNN, along with a copy of the input image. The output is logged for performing data analysis as per requirement. The image is passed on to the succeeding modules.

Rather than directly feeding the classification information as input to the controller, we insert one more module between this, for generating adversarial images from the clean images. It first overlays the adversarial scenario on the clean image before performing an adversarial attack. The generated adversarial image is fed again to the perception algorithm, same as from the original perception module, but this time the output is logged as well as passed forward to the controller as input arguments. The controller is set on autopilot i.e. the car drives without manual intervention. The controller is programmed such that if the inputs suggest presence of a traffic light, it breaks the car slowly till it stops. As soon as the input shows absence of any traffic light identification, the car accelerates.

We shall cover the sensor module along with Digital Twin generation in a separate section.

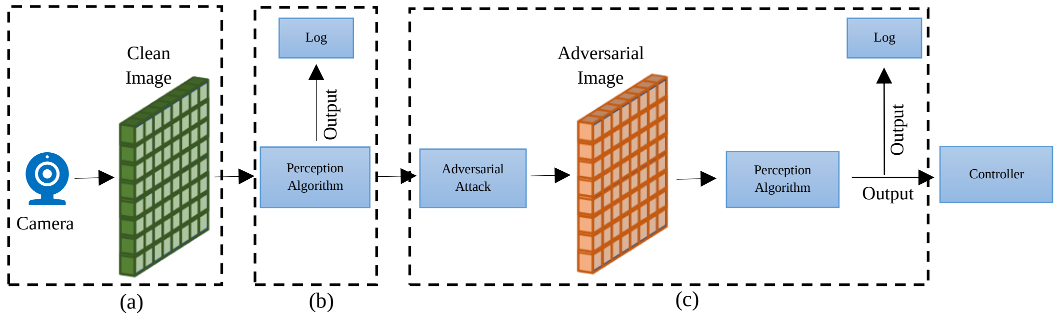


Figure 2: Framework of adversarial scenario generation; **(a)** Sensor module, **(b)** Perception module, **(c)** Adversarial attack module

3.1 Generation of Adversarial Images

There are a number of popular whitebox and blackbox attack algorithms which may be deployed with the adversarial objective of fooling the autopilot system used in self-driving cars. Since this work focuses on perturbing the images captured by the RGB camera mounted on the car so as to make the neural network misclassify the image of the moon as a traffic light, a prudent choice was to make use of a targeted adversarial attack algorithm. An assumption was made that the underlying neural network architecture (ResNet-18) used by the car autopilot system is known to the adversary a priori.

The Projected Gradient Descent (PGD) algorithm which is a well-known whitebox algorithm was used to perturb the captured images. The pseudo-code of the algorithm [13] is described below.

Algorithm 1: Projected Gradient Descent

```
Pick an initial point $x_0 \in Q$
while stopping condition not met do
    Descent direction: pick the descent direction as $-\nabla f(x_k)$
    Step size: pick a step size $\alpha_k$
    Update: $y_{k+1} \leftarrow x_k - \alpha_k \nabla f(x_k)$
    Projection: $x_{k+1} = argmin_{x \in Q} 0.5\|x - y_{k+1}\|_2^2$
```

Here, $Q \in R_n$ denotes the constraint set, the initial point is given by x_0 and $f(.)$ denotes the function to be optimized. The PGD algorithm was instrumental in making the neural network misclassify the image of the moon as a yellow traffic light. The targeted attack was tested using an image in which a yellowish tinge was visible in the evening sky [12] which the neural network misclassified as 'traffic light, traffic signal, stoplight' which has a class label of 920 in the ImageNet dataset [14]. A visual illustration of the PGD attack used for verification can be seen in Figure 3.

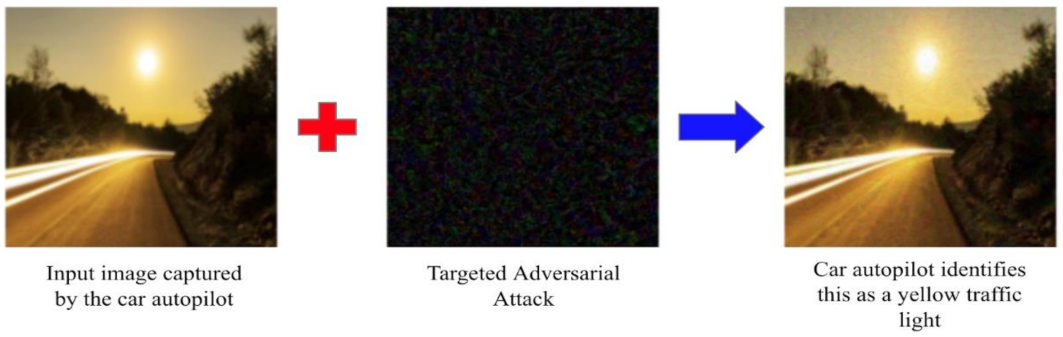
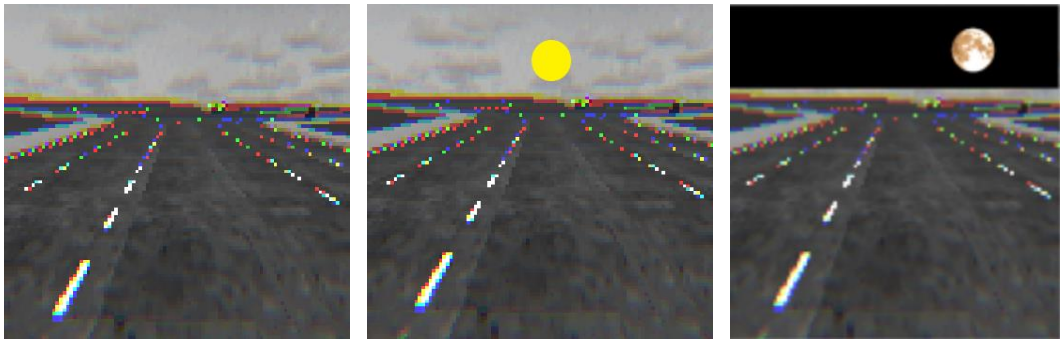

Input image captured by the car autopilot
Targeted Adversarial Attack
Car autopilot identifies this as a yellow traffic light

Figure 3: Test image used to verify whether the PGD attack was influential in making the moon being identified as a yellow traffic light

3.2 Image Processing

To perform image processing for the images captured by the RGB camera mounted on the vehicle, we aim to add a “moon” onto those images to portray the scenario of the car slowing down whilst incorrectly misidentifying the moon for a yellow traffic light. To do that, the first task at hand was to define the space (sky) within which the moon could be added. In the first attempt only a yellow circle/dot of a constant size was placed at random locations within the space defined as “sky” in the image. In order to achieve this OpenCV function of creating a circle was utilized in order to overlay the moon over the image in which the function traversed the image bank and for each image it placed a moon at a random point. The centre point for the dot to be drawn was random however the radius of the moon was to be fixed. One of the generated pictures is as depicted in Figure 4 (b).



(a) (b) (c)

Figure 4: Steps in processing of adversarial images, **(a)** clean image from sensor, **(b)** simplistic graphics, **(c)** realistic graphics

However it was decided to improve upon this further by placing the actual image of a moon. In order to do this, the sky was made to be pitch black by making use of the draw rectangle function. Three types of moon shapes were chosen to be placed, namely, half moon, full moon and crescent moon. An image mask was created by making use of binary thresholding which measures pixel intensity across images and permits for smooth blending. The image mask allows for the cropped image to be placed on the region of interest whilst the dimensions being explicitly specified. The three shapes were placed at random points in the sky region and their sizes were also varied. An image depicting this is shown in Figure 4 (c).

4 Digital Twin

We decided to simulate the adversarial attacks mentioned in the previous section using Digital Twin technology, a state of the art in the AV industry. We used a high-fidelity, compositional simulator called MetaDrive which is capable of generating an infinite number of diverse driving scenarios [15]. One such scenario involving a single agent (i.e. car) is shown in Figure 5. This simulator supports Windows OS and can work on portable computers (with nominal 6-8 GB RAM). MetaDrive makes use of Safety Gym [16], considered a benchmark for measuring progress towards reinforcement learning agents, to create virtual environments. Initial setup for the environment is easily done by passing a configuration dictionary.

For recording observations from the agent, we set a virtual RGB camera with a sensor size of (100, 100) pixels on the top of the car, with the help of provided API. Shown in Figure 4 (a) is a sample image captured by this camera. MetaDrive improves the simulation efficiency at the cost of photorealistic rendering effect, as can be seen from the observation image.

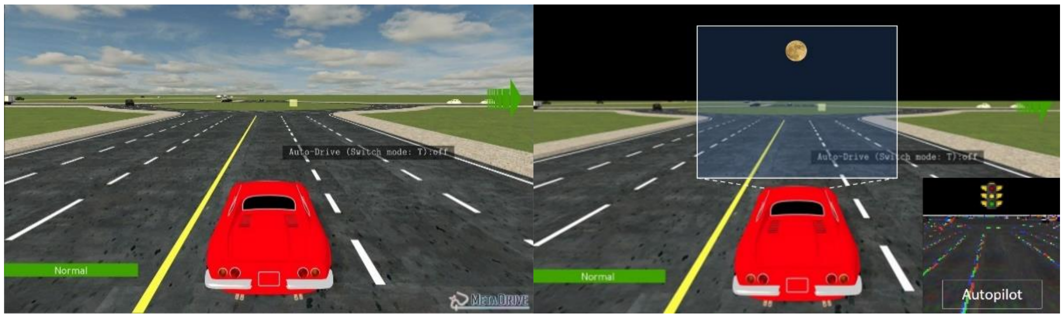


Figure 5: (left) Scenario from MetaDrive simulator, (right) same image modified for depicting how the perception algorithm for a real autopilot works

As the simulation progresses over provided time steps, if at any time step, the car AI (here modelled by ResNet18) detects a traffic light in the observed image; it provides feedback to the car controls to slow down the car until it stops. This feedback is programmed by us to test the adversarial attack where the AI mistakes a far-off object, in this case the moon, for a traffic light. We were able to successfully test our hypothesis using this setup, thus proving the significance of such safety-critical scenarios.

5 Results and Analysis

We use the logged data to extract meaningful information about how the car autopilot responded to adversarial attacks with different parameters (shape of moon, size of moon, color of the moon etc.). By varying the shape of the moon from crescent to full moon, we found that crescent moon of full size and half moon of 50% of full size are most likely to cause adversarial failure, as can be seen in Figure 6.

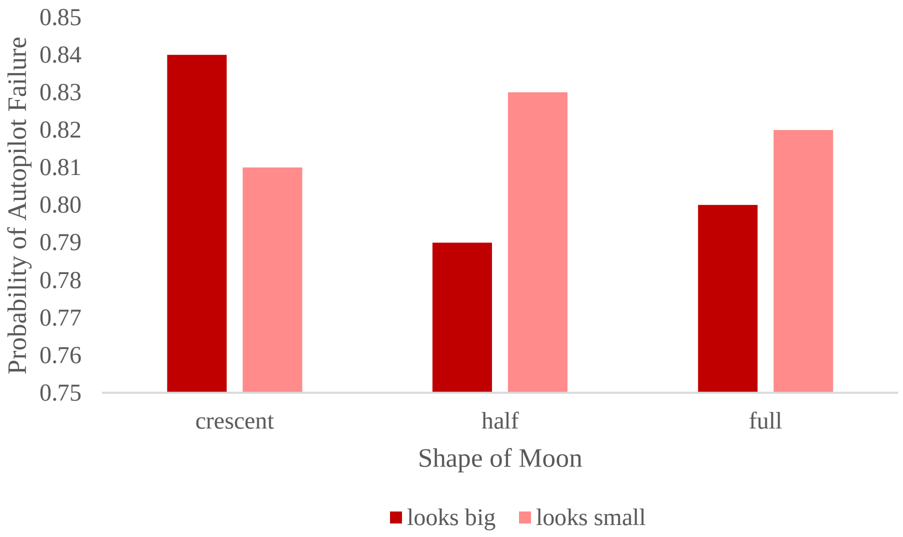


Figure 6: Effect of lunar cycle on probability of autopilot failure. Big corresponds to size of moon when closest to earth and small corresponds to 50% of the full size.

As for the size of the moon parameter, we did not diagnose any particular trend, as can be seen in Figure 7.

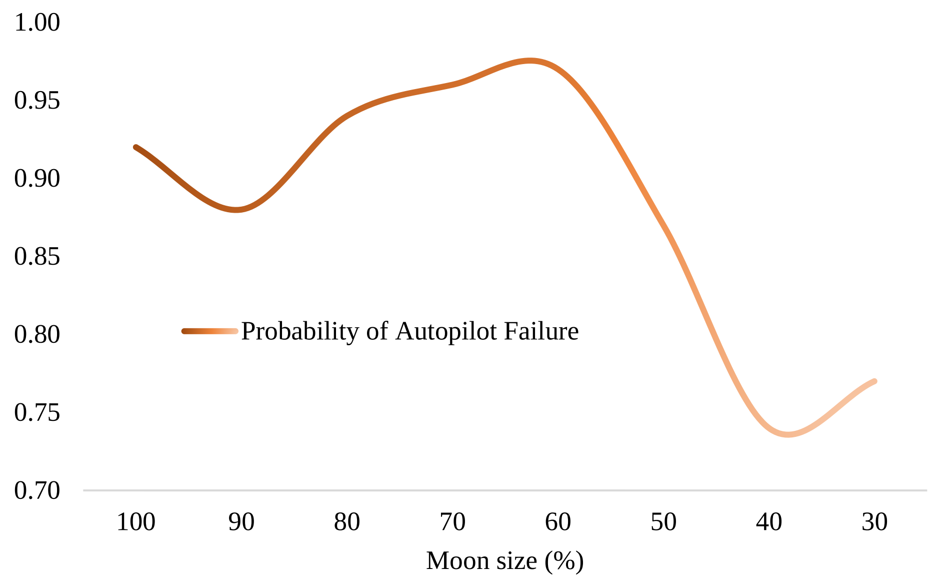


Figure 7: Effect of size of moon as perceived from car sensor on autopilot failure

The use of this extracted data can be made to understand other related safety-critical scenarios which may be avoided beforehand. We studied the effect of change in color of the moon as a design parameter for the simulation and found interesting results.

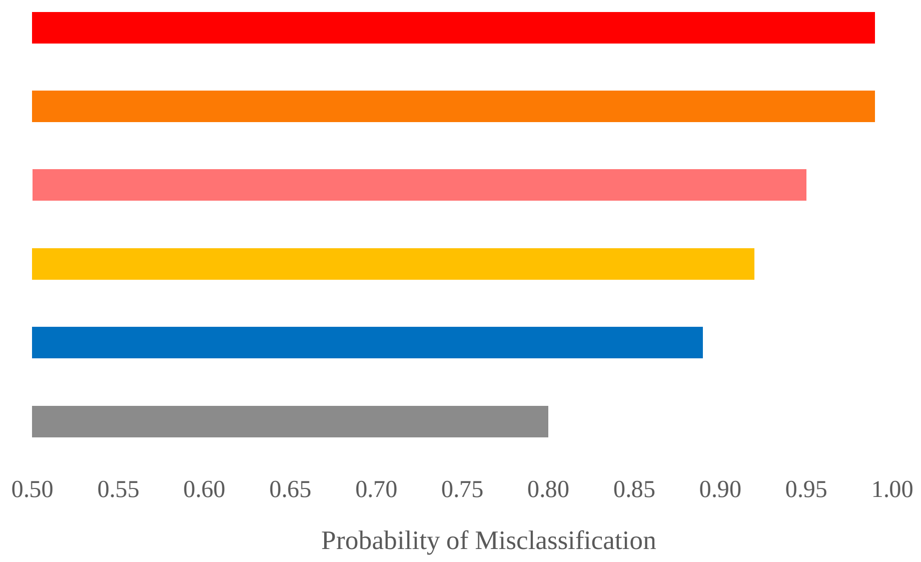


Figure 8: Effect of color of moon on autopilot failure

As we see in Figure 8, red and orange colored moons are most likely to cause autopilot failure. The colors were chosen deliberately. Red is for lunar eclipse, orange and yellow correspond to air pollution, pink is sometimes visible during the month of April and blue happens due to the ashes of volcanic eruptions.

6 Adversarial Defense

After successfully generating the safety-critical scenarios for the AV, our next obvious interest was to research into defensive techniques for car AI so that such situations can be averted.

6.1 Randomized Padding

This defensive technique works well with both black-box and white-box adversarial attacks against deep neural networks [17]. In this technique, the input image X_n goes through a random resizing layer, with certain limits for resizing scale. The resized image X'_n is then passed onto a random padding layer, which pads zeros around the resized image in a random manner. The resulting padded image X''_n is used for classification. This method is computationally inexpensive, simple to apply without need of any special requirements and also preserves the classification of the original image in almost all the cases. This also makes it suitable for real-time applications such as in an AV.

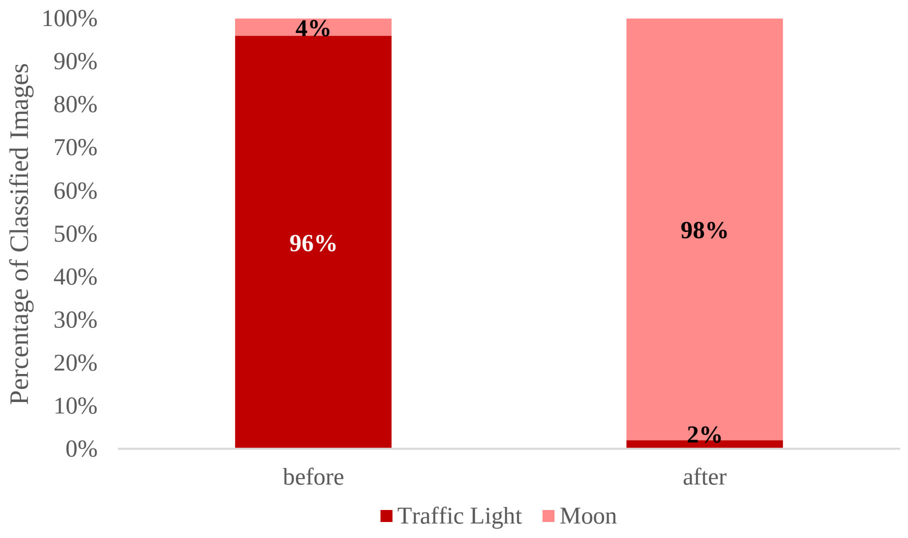


Figure 9: Randomized padding boosts adversarial defense in car autopilot

After using Randomized Padding, we were able to prevent 98% of the adversarial attacks, as can be seen in Figure 9. This real time technique is highly suitable for car autopilots to avoid majority of attacks, even those generated from very efficient algorithms such as PGD, as in this case.

6.2 Adversarial Training

This is the state of the art method for defense against targeted adversarial attacks such as PGD [18]. Unlike random transformations on images such as padding, scaling, cropping etc. it entails training the neural network with both unperturbed and perturbed images with the expectation that after training, the neural network becomes robust to adversarial attacks and is able to distinguish between the two kinds of images.

The ResNet-18 neural network architecture was trained with a total of 384 images which had been captured by the RGB camera mounted on the vehicle used in the simulator. The dataset consisted of an equal proportion of unperturbed and perturbed images. After the model had been trained, it performed exceptionally well and achieved an accuracy of 100% in distinguishing between the unperturbed and perturbed images. A visual representation of training the neural network used by the car autopilot system can be seen in Figure 10.

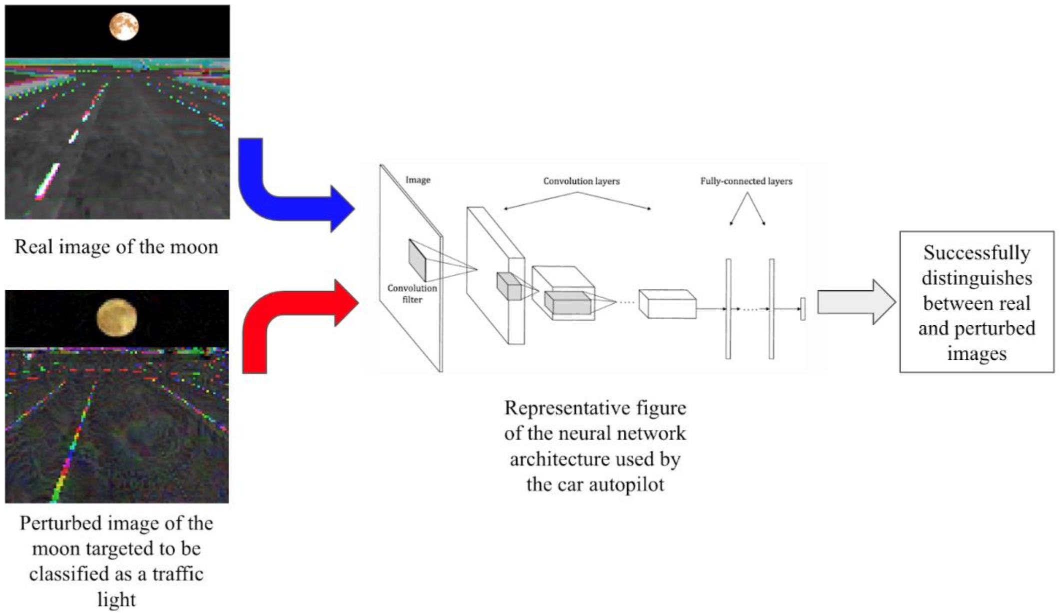

Real image of the moon
Perturbed image of the moon targeted to be classified as a traffic light
Image
Convolution layers
Fully-connected layers
Convolution filter
Representative figure of the neural network architecture used by the car autopilot
Successfully distinguishes between real and perturbed images

Figure 10: Adversarial training used to make the ResNet-18 neural network architecture robust to PGD attack

Acknowledgements

We are grateful to Assistant Professor Ding Zhao and Teaching Assistant for 24-784 (Trustworthy AI Autonomy) Wenhao Ding for their guidance, timely help and encouragement for pursuing this project idea.

References

[1] Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, Ding Zhao, "A Survey on Safety-Critical Driving Scenario Generation – A Methodological Perspective" arXiv:2202.02215v4 [cs.RO] 28 Feb 2022

[2] O. E. L. Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu et al., "Open Ended learning leads to generally capable agents," arXiv preprint arXiv:2107.12808, 2021.

[3] R. van der Made, M. Tideman, U. Lages, R. Katz, and M. Spencer, "Automated generation of virtual driving scenarios from test drive data," in 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, no. 15-0268, 2015.

[4] J. Fang, D. Zhou, F. Yan, T. Zhao, F. Zhang, Y. Ma, L. Wang, and R. Yang, "Augmented lidar simulator for autonomous driving," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1931–1938, 2020.

[5] W. Wang and D. Zhao, "Extracting traffic primitives directly from naturalistically logged data for self-driving applications," IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 1223–1229, 2018

[6] Y. Zhu, C. Miao, F. Hajiaghajani, M. Huai, L. Su, and C. Qiao, "Adversarial attacks against lidar semantic segmentation in autonomous driving," in Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, 2021, pp. 329–342.

[7] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, “Multimodal safety-critical scenarios generation for decision-making algorithms evaluation,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1551–1558, 2021.

[8] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[9] D. Zhao, H. Peng, H. Lam, S. Bao, K. Nobukawa, D. J. LeBlanc, and C. S. Pan, “Accelerated evaluation of automated vehicles in lane change scenarios,” in ASME 2015 Dynamic Systems and Control Conference. American Society of Mechanical Engineers, 2015, pp. V001T17A002—-V001T17A002

[10] Z. Huang, D. Zhao, H. Lam, D. J. LeBlanc, and H. Peng, “Evaluation of automated vehicles in the frontal cut-in scenario — an enhanced approach using piecewise mixture models,” in 2017 IEEE International Conference

[11] Tim Levin, *Tesla's Full Self-Driving tech keeps getting fooled by the moon, billboards, and Burger King signs*, BUS. INSIDER (July 26, 2021).

[12] Jay Ramey, *Tesla FSD Mistakes Moon for Yellow Traffic Light*, Autoweek (July 23, 2021).

[13] Anderson Ang, *Projected Gradient Algorithm* (October 23, 2020)

[14] *ImageNet*, Stanford Vision Lab, Stanford University, Princeton University (March 11, 2021)

[15] Li, Q., Peng, Z., Xue, Z., Zhang, Q. and Zhou, B., 2021. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. arXiv preprint arXiv:2109.12674.

[16] Ray, A., Achiam, J. and Amodei, D., 2019. Benchmarking safe exploration in deep reinforcement learning. arXiv preprint arXiv:1910.01708, 7, p.1.

[17] Xie, C., Wang, J., Zhang, Z., Ren, Z. and Yuille, A., 2017. Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991.

[18] Oscar Knagg, Know your enemy. How you can create and defend against… | by Oscar Knagg | Towards Data Science (January 6, 2019)