World Academy of Science, Engineering and Technology
International Journal of Mechanical and Materials Engineering
Vol:16, No:9, 2022

A Simulated Environment Approach to Investigate the Effect of Adversarial Perturbations on Traffic Sign for Automotive Software-in-Loop Testing

Sunil Patel, Pallab Maji

Abstract**—**To study the effect of adversarial attack environment must be controlled. Autonomous driving includes mainly 5 phases sense, perceive, map, plan, and drive. Autonomous vehicles sense their surrounding with the help of different sensors like cameras, radars, and lidars. Deep learning techniques are considered Blackbox and found to be vulnerable to adversarial attacks. In this research, we study the effect of the various known adversarial attacks with the help of the Unreal Engine-based, high-fidelity, real-time raytraced simulated environment. The goal of this experiment is to find out if adversarial attacks work in moving vehicles and if an unknown network may be targeted. We discovered that the existing Blackbox and Whitebox attacks have varying effects on different traffic signs. We observed that attacks that impair detection in static scenarios do not have the same effect on moving vehicles. It was found that some adversarial attacks with hardly noticeable perturbations entirely blocked the recognition of certain traffic signs. We observed that the daylight condition has a substantial impact on the model's performance by simulating the interplay of light on traffic signs. Our findings have been found to closely resemble outcomes encountered in the real world.

Keywords**—**Adversarial attack simulation, computer simulation, ray-traced environment, realistic simulation, unreal engine.

I. Introduction

THE autonomous driving software system is becoming increasingly complex day by day. The Auto-Pilot system needs to be tested end to end rigorously before deployment or even before on-road testing. Auto-Pilot systems use a Deep Learning-based network to perceive the environment using information captured by various sensors, like Camera, Radar, and Lidar. The perceived environment and applied information from HD Maps are critical in mapping and planning the drive. Lidar provides the vehicle with the shape and depth of static and dynamic objects in the surrounding environment including pedestrians. Lidar cannot be used alone and hence cameras are used for most decision-making tasks including sign detection, clear sight, lane navigation, traffic light identification, road sign identification and even switching to high and low beams while driving. Radar supplements the camera by providing accurate speed and relative dimensions of the object in shorter ranges. Lidar, radar, and cameras are used all around the vehicle to help with a sufficient level of autonomy. On-road testing is very essential and is imperative in any vehicle testing, however, it is expensive and cannot test all the rare and potentially hazardous scenarios. Thereby, the most viable and practical approach is to pre-screen through simulation-based software-in-loop (SIL) testing. Simulation-based testing requires accurate simulation of sensors, models, environments, traffic scenarios, and vehicles. Simulation testing is very helpful in testing rare and theoretically proven extreme and hazardous scenarios. High Fidelity simulators like NVIDIA DriveSim [1], Carla [2], and AirSim [3] are the most used ones at present times for simulation-based testing. A component of Nvidia Drive - DRIVE AV has a module for perception, mapping, and planning layers, as well as diverse deep neural networks (DNNs) trained on high-quality real-world driving data. Among many high-quality networks, a sign identification network (SignNet) is used for sign detection. Nvidia sign network has a module for the identification of the US and UK road signs. Unreal Engine helps to simulate the realistic behavior of light with ray tracing to create cinematic-quality content rendering in real-time. Unreal Engine version 4.22 supports real-time ray tracing with Nvidia RTX™ GPUs. Video frames generated through the Unreal Engine (UE) are consumed by SignNet.[1]

DNNs are found to be highly vulnerable to adversarial perturbations [4]-[9]. Road signs are also found to be affected by adversarial perturbations [10]-[13].

The adversarial attack can be of two types 1) a Whitebox attack and 2) a Blackbox attack. The attack is Whitebox if the attacker has access to model architecture, parameters, and outputs. Blackbox attack relies on the phenomenon of transferability [15], where an adversarial example generated model trained locally by the attacker remains adversarial for another target model used for a similar application.

In a Blackbox attack, the attacker has no knowledge of the model and its related parameters [17], [18]. The Blackbox attack is more difficult. Although the attacker might not have the access to the actual model, due to the transferable nature of the Blackbox attack network with a similar topology it may be vulnerable to the same kind of attack. Detection and segmentation networks are found to be vulnerable to both Whitebox and Blackbox attacks [19]-[21].

[1] Adversarial Simulator: A simulator with required 3D assets and Models used in the current research are available at: https://github.com/snlpatel001213/AdversarialSimulator

Sunil Patel and Pallab Maji are Data Scientists in Deep Learning, Nvidia Graphics, Karnataka, 560045 India (e-mail: supatel@nvidia.com, pmaji@nvidia.com).

The methods are found to be working on the images taken from a static location. The object as perceived from a static camera and the same object perceived by video frames in relative motion are perceptually different. The attack works well on the detection settings but may or may not work in the detection + localization task. Validating the effect of adversarial perturbation is essential as perceived by the moving vehicle. For example, the adversarial attack like single-pixel variation works when the perturbed image is given as it is to the model. Single-pixel is very insignificant in the entire view so when such an image of a road sign is perceived from a moving vehicle, it might not work. Therefore, the distillation of adversarial attack succeeding while the vehicle is on the move while performing detection & classification tasks should be validated.

II. Environment Setup

Unreal Engine uses the Nvidia PhysX>=3.3 engine for physical simulation calculation in rigid bodies, cloth, and fluid particle systems. The environment was set so that it imitates the real-life specifications, these include the camera, environment, lighting conditions, and physics of the car. The specification of the object used to create an environment for the simulation is given below:

A. Camera Setup

For the recording, the camera's resolution was set at 1920 x 1080. The field of view was fixed at 60 degrees. The camera was placed at the top of the windshield behind the rear-view mirror operating at 60 frames per second used for all the experiments. The simulated camera had additional features including 1) Auto exposure: to render the scene based on the scene brightness and 2) Light propagation volumes: for dynamically generating global illumination.

B. Vehicle Setup

A 4-wheeled vehicle with a limited-slip rear-drive differential configuration was used with the following particulars: 1) Maximum RPM: 7500, 2) Movement of Inertia: 1 kgm^2, 3) Damping rate when full throttle applied (Simulating friction by road): 0.15 kg m^2/s, 4) Damping rate when zero throttle clutch applied: 2 kgm^2/s, 5) Damping rate when zero throttle clutch applied: 2 kg m^2/s. To keep maneuver variability lowest in the environment, we have used UE4 cinematics-based fixed-rate translation. A cinematics-based dashboard was used to control different aspects of experiments such as headlight flux, speed, camera calibration, placement of objects, etc.

C. Realistic Rendering

Cinematic rendering and emulating the property of light are very essential for generating a realistic scene. Deep learning techniques are found to be highly sensitive to the characteristics of train data. A slight insignificant change in light characteristics was found to confuse the model resulting in higher False Positive and False Negatives [14]. In our experiments, instead of baked lighting, dynamic lighting was used and the lightmap resolution was set to 512 while the numbers of indirect light bounces were set to 3. The indirect light quality was set to 5 which also represents the number of photons in the space and greatly improves the quality. Alongside, natural light properties such as Rayleigh scattering, Mie anisotropy, and atmospheric scattering were extensively used in the simulation. Additional parameters used to render the world are given in the supplementary material.

To make the test repeatable and structured, various UE4 tools like fixed material definitions, camera rails, and spline-based cinematic recording are used in the environmental maps.

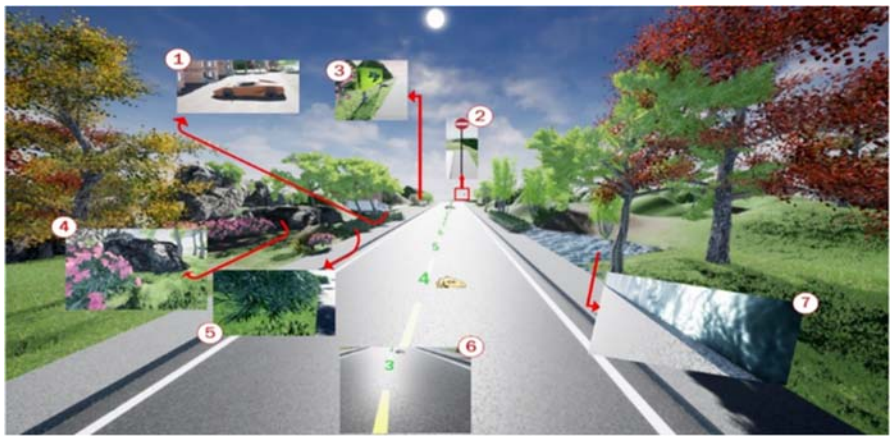


Fig. 1 Map of the test facility developed within Unreal Engine. The diagram shows a test track for one traffic sign. All the traffic signs are independently arranged in parallel in a non-occluding fashion. High fidelity environment with real-time ray tracing simulating shadows and reflection. Fine details like objects, 1) cars and buildings, 2) traffic signs at the end of the road, 3) holdings, 4) detailed rocks and vegetation with shadows, 5) realistic vegetation, 6) tarmac-like material for the road with lane markings and, 7) water reflecting sun rays

D. Test Facility

A Test Facility with eight parallel tracks was developed for the experimentation where each track looks as shown in Fig. 1. Each track has natural elements like vegetation, landscape, rocks, debris, advertisement holdings, and buildings. The test track has tarmac-like material with light reflection behavior like

the tarmac. A traffic sign was placed at the end of the track. A vehicle starts at a distance where the traffic sign is barely visible and approaches it at a constant speed. All the intermediate frames are identified as perceived by the moving vehicle. All eight signs are processed at a single stretch keeping the experimental conditions the same. An aerial view of the test facility with all signs is provided in the supplementary material.

III. Methodology

Traffic signs can be classified as regulatory, warnings, and mandatory. Mandatory signs are critical, and any violation carries a high probability of casualty. We choose eight signs based on criticality and relevance.

Fig. 2 Different traffic signs considered for the experimentation

The unreal engine was used to create a simulated environment and scenario generation. Drive SDK was used for predicting the scenarios created. Various experiments were carried out to evaluate adversarial attacks namely;

1. Signs perturbed with the adversarial algorithms,
2. Variable environmental conditions,
3. Effect of adversarial patches.

All these modifications are tested in combination where applicable.

A. Adversarial Perturbation

A function f classifies the input image $x \in X$ to label $y \in Y$. An adversarial attack is possible by adding a small perturbation to x so that a new image $\acute{x} = x + \epsilon$. When such an adversarial attack is successful, the same function f is applied to $\acute{x}$ then the labels produce from both the images are different such that $f(x) \neq f(\acute{x})$. The function f is commonly referred to as a model in the deep learning fraternity.

One simple example of the Whitebox attach is the Fast Gradient Sign Method (FGSM) [16]. FGSM can be represented by:

$$\acute{x} = x + \epsilon.sign\big(\delta_x j(\theta, x, y)\big) \quad (1)$$

where δ is the gradient of cost function w.r.t x. The objective is to create an image that maximizes the loss, logically FGSM adds the noise (ϵ) whose direction is the same as the gradient of cost function in reference to the image. The magnitude of change is constrained by the max norm and direction is more important than the magnitude. Similarly, all the algorithms working by the Whitebox approach take the help of gradient or any other parameter associated with the model and apply small perturbation so that loss can be maximized.

There are over 24 algorithms that claim to work by the Whitebox approach. To select only potent attacks, all these adversarial algorithms were tested against 18 known models. A potency of attack for any adversarial attack is chartered by its effectiveness at the least value of epsilon (perturbation percentages). Four adversarial algorithms are selected which show the highest potency over all the selected traffic signs at different values of epsilon in the range between 0 to 1, as shown in Fig. 4. These four adversarial algorithms are: 1) Projected Gradient Descent (PGD) [22], 2) Basic Iterative (BA) [23], 3) Fast Gradient (FG) [24], and 4) Repeated Additive Uniform Noise (RAUN). There are many ways to measure the magnitude of gradient vector including L1, L2, and Linf (L-infinity norm), only Linf norm was found to be most effective. RAUN repeatedly samples uniform noise with a fixed L-infinity size. As the model architecture of SignNet is unknown, we have taken the assumption that the network can have architecture like YOLO [25]. YOLO can have various kind of the backbone including ResNet-50 [26], ResNext101x32-8d [27], DenseNet-161 [28], and VGG-19 [29]. All signs are perturbed with the previously selected adversarial algorithms.

To see the effect of adversarial attacks, perturbation was applied over eight signs at six different levels of epsilons for each of four adversarial algorithms over four models. Each experiment generates 800 frames. This experiment collectively generates 6,14,400 frames for analysis.

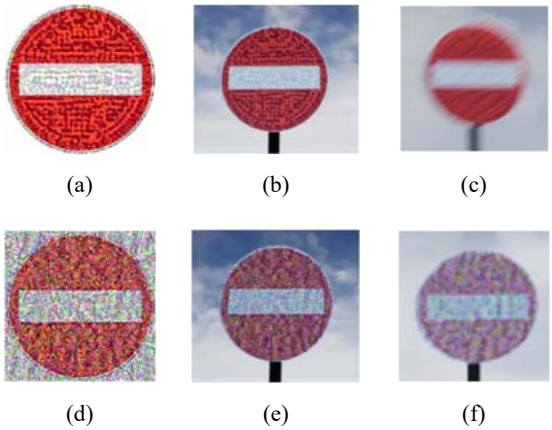


Fig. 3 (a) Static sign when perturbation applied with FG ($ResNet-50, \epsilon = 0.3$), (b) Sign when placed in an environment and perceived from static camera, (c) Sign as perceived from moving vehicle, (d)-(f) a sign with perturbation, static sign in the environment, and perceived through moving vehicle respectively when a basic iterative attack succeeded over ResNet-50 applied with $\epsilon = 0.8$

The experiments are carried out by introducing a pixel-wise perturbation amount (ϵ) in the original image. The larger the epsilon the more noticeable the aberrations. The attempt was made to identify perturbations that are extremely hard to notice and yet make significant deviations. Perturbation is applied within a range of epsilon and attack which one is working best with the least value of epsilon selected. Keeping the environmental condition constant, the experiment was repeated for all combinations of the traffic signs and the different values

of ϵ.

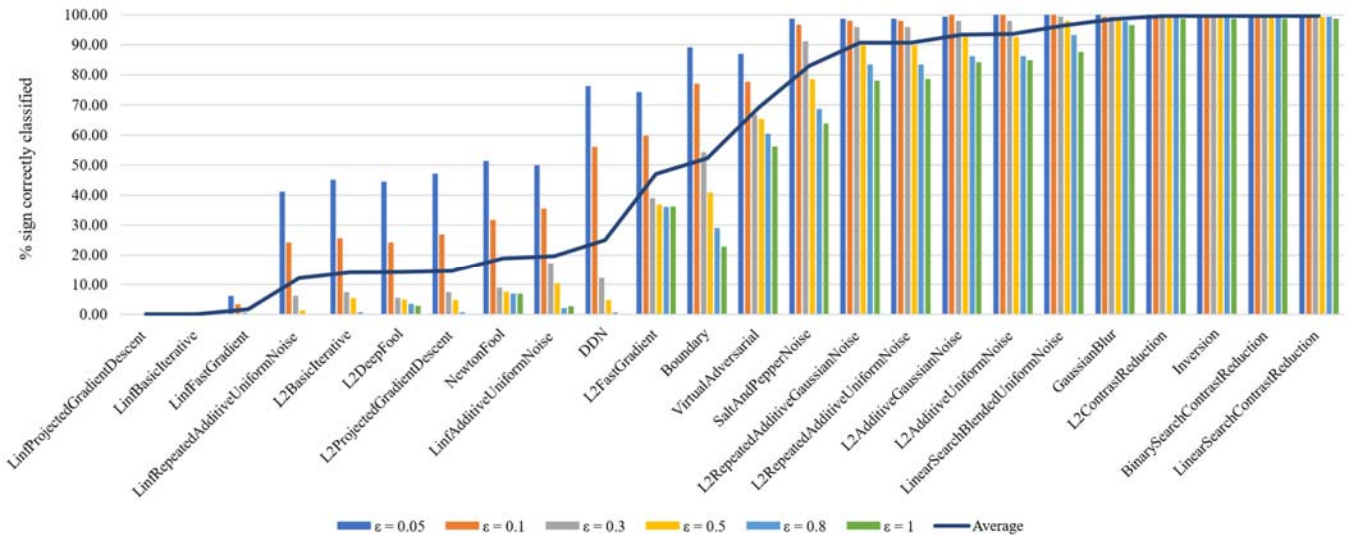
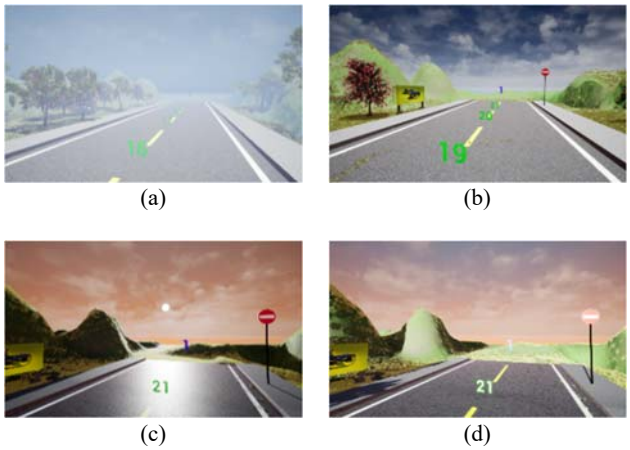


Fig. 4 Percentage of signs correctly classifier after applying a perturbation to Traffic signs. This experiment was performed over 8 traffic signs. Where 24 adversarial algorithms were used to generate perturbation at six different values of epsilons applied to 18 different deep learning models

B. Environmental Variations

Autonomous vehicles are to be driven in various environmental conditions. The following experiments were carried out, to test the effect of environmental variations on traffic sign identification.

1. Morning light (light source in front of the vehicle) with an intensity of 5 flux, 10 flux and 20 flux.
2. Evening light (light source behind the vehicle) with intensity 5 flux, 10 flux and 20 flux.
3. A varying percentage of fog, 10%, 30%, and 50%.

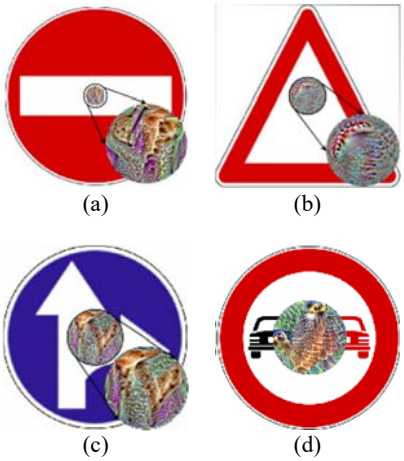


(a) (b) (c) (d)

Fig. 5 (a) Effect of fog, (b) Simulating noon, (c) Sun in front of the vehicle simulating morning Scenario, (d) Simulating evening scenario sun behind the vehicle

C. Adversarial Patch

A Blackbox method - Adversarial patch [30] found to cause a shift in the prediction. Adversarial patches are targeted attacks and hence we generated patches for an object that looks very similar to the traffic signs. We tested the effect of adversarial patches when applied over traffic signs in different sizes. Four different types of patches were used including; 1) balloon, 2) baseball, 3) kite, and 4) toaster. Where patches representing balloons, baseball, and kite were generated, a patch representing a toaster was adapted as published [30] with ensemble models. The patch for a toaster serves as the reference if the balloon, baseball, and kite are generated correctly.



(a) (b) (c) (d)

Fig. 6 Adversarial Patch over different Traffic signs with different sizes: (a) Balloon - 10%, (b) Baseball - 20%, (c) Toaster - 30%, (d) Kite - 40%. Sizes are in a percentage of original traffic signs covered by Adversarial Patch. Enlarged patches showed wherever applicable.

IV. Result and Discussions

Each experiment generates 800 frames. Deviation in prediction per frame was calculated in the following way:

$$X^M = \{x_1, x_2, x_3, \dots, x_{800}\} \quad (2)$$

$$Y^M = f(X) \quad (3)$$

$$count(C^M) = \sum_{i,j=1}^{M} 1, if\ (x_i^M = y_i^M) \quad (4)$$

$$Diff(\%) = \frac{count(C^{std}) - count(C^{perturb})}{count(C^{std})} * 100 \quad (5)$$

Here X is an input set of frames and Y represents prediction by applying model function f on X. C^{std} indicates correct predictions on the standard set of inputs and $C^{perturb}$ indicates a set with perturbation. Such perturbation can be induced with a Whitebox attack, a Blackbox attack, or with different environmental conditions. M is the set (standard or with perturbation) on which prediction is applied. Finally, the deviation (%) between both predictions was calculated. The higher the percentage difference, the more will the impact of an attack on traffic sign identification.

(a) (b) (c)

Fig. 7 (a) Static sign when patch applied to cover 30% area with a balloon as target, (b) Patch when placed in an environment and perceived from static camera, (c) Patch as perceived from moving vehicle

A. Effect of Adversarial Perturbation

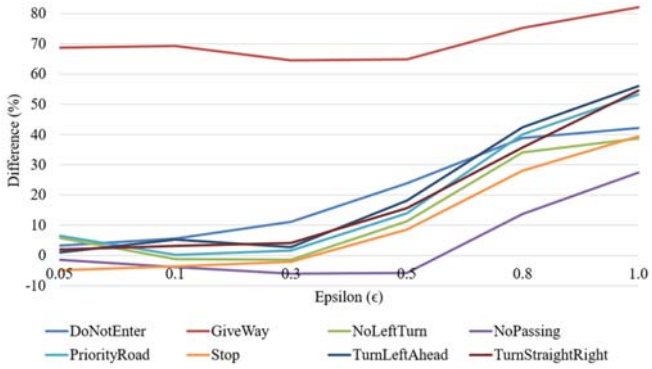


Fig. 8 Mean of the percentage of deviation when measured in relation to the standard environment

It is evident from Fig. 8 that, 1) When compared with a standard experiment at epsilon 0.5, some signs like 'PriorityRoad' and 'DoNotEnter' remain undetected in the 20% of total frames; 2) At an epsilon value of 0.8, almost all the signs are skipped by 40% of the frames except the sign – 'No Passing'; 3) 'Giveaway' is the sign which is most affected at all values of epsilon. The visual difference between adversarial perturbation when applied with regard to different adversarial algorithms is provided in the supplementary material. At ε = 0.5 all the traffic signs under study are visually distinguishable with a small amount of unnoticeable perturbation on them. In moving vehicle conditions such perturbations are almost non-identifiable as shown in Fig. 4. At ε ≥ 0.8 perturbations are visible and some adversarial algorithms (like FG) almost make traffic signs unrecognizable in static conditions as well as in moving conditions as shown in Fig 4. In the case of BA, PGD, and RAUN at ε ≥ 0.8, signs are well identifiable with noticeable perturbations on them.

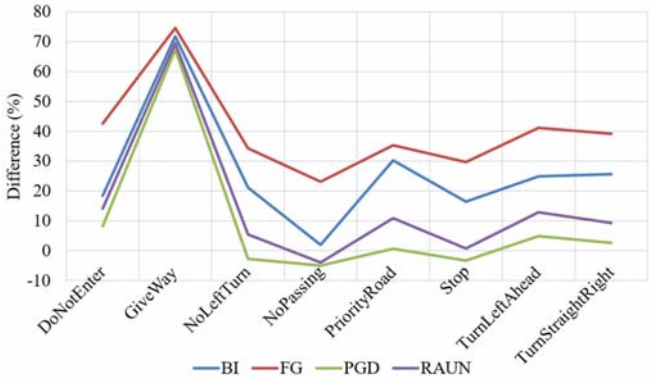


Fig. 9 Effect of the different adversarial algorithms over different traffic signs (Max difference (%) between standard and variations)

Fig. 9 shows that the Fast Gradient attack is the most effective and the Basic Iterative attack was found to be the least effective among the selected. Also, perturbation by Fast Gradient attack is more aggressive and signs are barely identifiable at ε ≥ 0.8.

TABLE I
COUNT OF A SCENARIO WHERE 100% FAILURE TO RECOGNIZE THE TRAFFIC SIGN IS OBSERVED ACROSS ALL THE FRAMES

Signs	Adversarial Attacks			
	BI	FG	PGD	RAUN
DoNotEnter		2 (0.8, 1.0)		
Giveway	1 (1.0)	2 (0.8, 1.0)		1 (0.1)
NoLeftTurn	3 (0.3,0.5,1.0)	2 (0.8, 1.0)		1 (0.8)
NoPassing		2 (0.8, 1.0)		
PriorityRoad	1 (1.0)	2 (0.8, 1.0)		1 (0.05)
Stop		2 (0.8, 1.0)		
TurnLeftAhead	1 (1.0)	2 (0.8, 1.0)	1 (0.1)	
TurnStraightRight	1 (1.0)	2 (0.8, 1.0)		

Values in the parentheses show epsilon at which attack was successful.

As depicted in Table I, FG affects most of the signs due to its coarser/noticeable perturbations where in other algorithms such as BI, PGD, and RAUN successful attacks are resulting in lower epsilon values too.

B. Variable Environmental Condition

Various traffic sign remains undetected in the morning light (sun angle at 20°-40°), and evening light (sun angle at 150°-170°). The probable reason for both of these observations can be that, 1) like the human eye, the camera with auto-exposure gets blinded by a strong beam of light coming almost parallel in direction; and 2) the reflective material of the traffic sign reflects light falling onto it almost at 90° and creates hindrance in the traffic sign detection. A similar effect to evening light is

observed in the night scene where the reflection of a strong beam from the vehicle itself makes signs almost unrecognizable. In the present observation, the model was found to be performing better in a fog-like environment. Fog-like conditions may be adding random noise like one found in the naturally captured data as well. Hence, frames with fog are performing better due to the generalization effect. Also, it is found that with 100% failure, this model is not suitable for the low light or night scenarios. Also, factors like graffiti and patchy roads were found to be affecting the different signs differently.

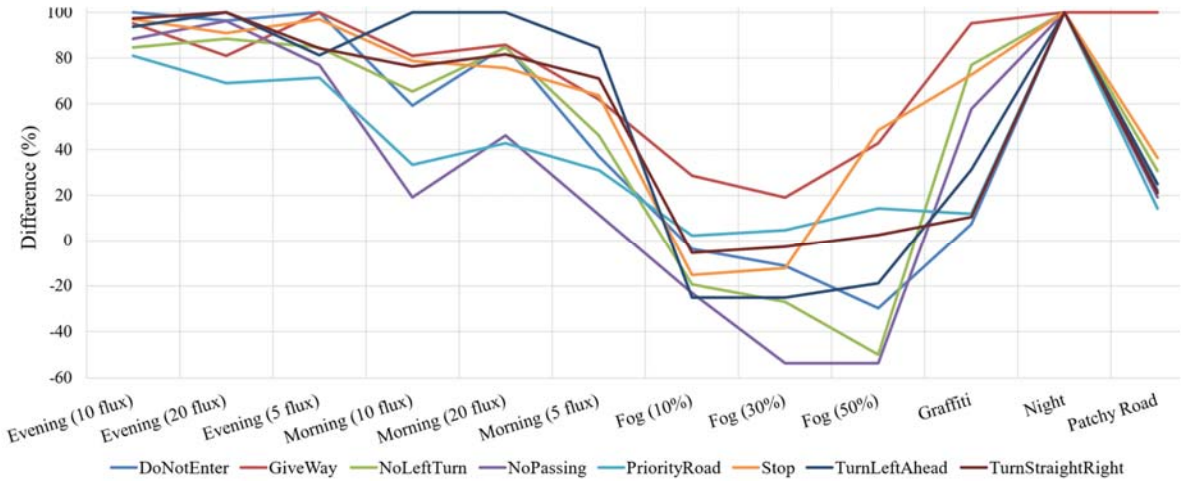


Fig. 10 The traffic sign affected the most (Max difference between standard and variations)

C. Effect of Adversarial Patches

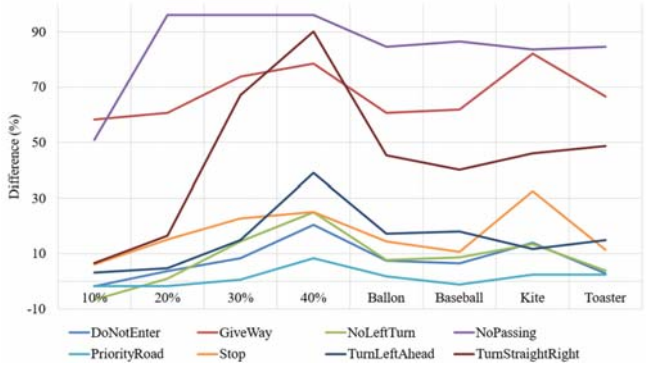


Fig. 11 Effect of the size and type of the adversarial patch on the traffic sign

Two trends are observed based on Fig. 11: 1) Adversarial effect of the patch increases as the relative size of the applied patch over traffic sign increases. This can be due to the increasing visual blockage of the traffic sign; and 2) A variety of adversarial patches affects different traffic signs differently. As reported by researchers [18], [30], we did not observe total misdetection across all the frames due to any of the adversarial patches. The traffic sign – 'NoPassing' is affected the most. If an adversarial patch of size > 20% is placed in the center of the 'NoPassing' sign, it makes it unrecognizable as shown in Fig. 6 (d).

V. Conclusion

We have presented a high fidelity, raytraced, simulated environment that can be used for controlled model safety analysis and to identify its vulnerability towards adversarial attacks in vehicular SIL testing. We also have shown that perturbation-based Whitebox methods affect different signs differently. Some signs are vulnerable and remain 100% misclassified at lower values of epsilon too, whereas some classify correctly at higher values of epsilon too. Perturbation by algorithms like Fast Gradient is noticeable and makes traffic signs misclassified through all the frames whereas unnoticeable perturbation at a low value of epsilon also makes some traffic signs undetectable, which was an interesting observation. Environmental factors such as light conditions play a vital role in traffic sign misdetection. Particularly early morning light and evening light were found to be affecting traffic signs to a greater extent. No traffic signs were detected in the night conditions. The interaction of light rays, their direction, and their intensity plays an important role. This also shows how accurately we simulated the reflective material and interaction of light with it in real-time with the help of Ray Tracing. NVIDIA Drive documentation specifies that the SignNet model is not trained for the low light scenario which is evident from the performance of the network in the low light simulated environments as well. Ensemble black box exists but the synthesis of patches is model and data-dependent. Frames perceived from the moving vehicle are very different from the frames from a static location. It is estimated that motion induces noise in the frames and hence makes smaller detail from the patch almost negligible. These smaller details in the adversarial patch are responsible for its adversarial nature. The multiclass confusion matrix as shown in the supplementary material shows a significant shift in the predicted classes when any of the above-discussed variations are applied when compared to the confusion matrix for traffic signs without any perturbations applied. Ray tracing allows real-time light simulation, light interacts with the world and

objects to generate realistic scenes. The simulator and all the objects developed in the present approach can be used to analyze model vulnerability, with singular or plurality of affecting factors.

REFERENCES

[1] "NVIDIA DRIVE - Software | NVIDIA Developer." https://developer.nvidia.com/drive/drive-software (accessed Oct. 23, 2020).

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "CARLA: An Open Urban Driving Simulator." Accessed: Oct. 23, 2020. (Online).

[3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," 2018, pp. 621–635.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," Dec. 2015, Accessed: Oct. 23, 2020. (Online). Available: https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/.

[5] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018, pp. 36–42, Feb. 2017, Accessed: Oct. 25, 2020. (Online). Available: http://arxiv.org/abs/1702.06832.

[6] B. Li and Y. Vorobeychik, "Feature Cross-Substitution in Adversarial Classification," 2014. Accessed: Oct. 25, 2020. (Online).

[7] B. Li and Y. Vorobeychik, "Scalable Optimization of Randomized Operational Decisions in Adversarial Classification Settings," PMLR, Feb. 2015. Accessed: Oct. 25, 2020. (Online). Available: http://proceedings.mlr.press/v38/li15a.html.

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations." Accessed: Oct. 25, 2020. (Online). Available: https://github.com/.

[9] A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 07-12-June-2015, pp. 427–436, Dec. 2014, Accessed: Oct. 25, 2020. (Online). Available: http://arxiv.org/abs/1412.1897.

[10] K. Eykholt et al., "Robust Physical-World Attacks on Deep Learning Visual Classification," 2018. Accessed: Oct. 25, 2020. (Online). Available: https://iotsecurity.eecs.umich.edu/#roadsigns.

[11] A. Liu et al., "Perceptual-sensitive GAN for generating adversarial patches," in 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Jul. 2019, vol. 33, no. 01, pp. 1028–1035, doi: 10.1609/aaai.v33i01.33011028.

[12] C. Sitawarin et al., "DARTS: Deceiving Autonomous Cars with Toxic Signs," vol. 1, no. 1, 2016. (Online). Available: http://arxiv.org/abs/1712.09665.

[13] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, "SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations," Jul. 2020, Accessed: Oct. 25, 2020. (Online). Available: http://arxiv.org/abs/2007.04137.

[14] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks." Accessed: Oct. 25, 2020. (Online).

[15] C. Szegedy et al., "Intriguing properties of neural networks," Dec. 2014, Accessed: Oct. 23, 2020. (Online). Available: https://arxiv.org/abs/1312.6199v4.

[16] C. Zuo, "Regularization Effect of Fast Gradient Sign Method and its Generalization," Oct. 2018, Accessed: Oct. 25, 2020. (Online). Available: http://arxiv.org/abs/1810.11711.

[17] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The Space of Transferable Adversarial Examples," Apr. 2017, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1704.03453.

[18] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "DPatch: An Adversarial Patch Attack on Object Detectors," CEUR Workshop Proceedings, vol. 2301, Jun. 2018, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1806.02299.

[19] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, Nov. 2016, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1611.02770.

[20] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling Deep Structured Prediction Models," Jul. 2017, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1707.05373.

[21] A. Arnab, O. Miksik, and P. H. S. Torr, "On the Robustness of Semantic Segmentation Models to Adversarial Attacks," Nov. 2017, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1711.09856.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, Jun. 2017, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1706.06083.

[23] A. Kurakin, G. Brain, I. J. G. Openai, and S. Bengio, "Adversarial Examples in the Physical World," 2017. Accessed: Oct. 23, 2020. (Online). Available: https://arxiv.org/abs/1607.02533.

[24] I. Yilmaz, "Practical Fast Gradient Sign Attack against Mammographic Image Classifier." Accessed: Oct. 26, 2020. (Online).

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Cvpr, Jun. 2015, Accessed: Oct. 25, 2020. (Online). Available: http://arxiv.org/abs/1506.02640.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Nov. 2017, vol. 2017-January, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." Accessed: Oct. 25, 2020. (Online). Available: https://github.com/liuzhuang13/DenseNet.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015. Accessed: Oct. 25, 2020. (Online). Available: http://www.robots.ox.ac.uk/.

[30] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial Patch," Dec. 2017, Accessed: Oct. 23, 2020. (Online). Available: http://arxiv.org/abs/1712.09665.