Università degli Studi di Salerno
Dipartimento di Informatica

Tesi di Laurea di I livello in
Informatica

Adversarial Attacks on Vision-based Deep Neural Networks in Autonomous Driving Vehicles

Relatore
Giuseppe Scanniello
Correlatore
Dott. Nome Cognome

Candidato
Michelangelo Esposito

Academic Year 2021-2022

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

List of Figures

List of Acronyms and Abbreviations

Chapter 1

Introduction

Things to add:

- where different types of coverage are useful
- Add formulas for coverage criteria

Chapter 2

Problem formulation

2.1 Autonomous Driving Vehicles

2.1.1 Introduction

2.1.2 History of autonomous vehicles

Magnetic wire following cars . . . The **Automatic Land Vehicle in Neural Networl**, ALVINN, was the first self-driving car ever proposed, in 1988; it was based on neural networks responsible to detect lines, segment the environment, and drive the car. While the general principles on which it was based worked well, the limited computed capabilities at the time, didn't make the solution take off. Furthermore, the absence of data meant that it was extremely difficult to gather the necessary samples and datasets on which to base the models that would manage vision and driving.

2.1.3 Sensors and hardware

LiDAR stands for Light Detection And Ranging. It's a method to measure distance of object by firing a focused laser beam and measuring the time it takes for it to bounce back at the source after being reflected by something. Compared to traditional camera images, LiDAR data can provide additional information about the surrounding scene...

A self-driving vehicle cannot rely on LiDAR alone, however. While this technology works great in most environments, including dark areas, if the scene surrounding the car becomes more noisy, due to rain or fog for example, the LiDAR sensor can become imprecise or fail.

For this reason, a third sensor is usually employed, the RADAR, which scans the surrounding area based on radio waves. . .

2.2 Object detection

Reliable object detection is a crucial requirement in realizing autonomous driving [1]; being aware of its surrounding environment is necessary to make an autonomous vehicle avoid accidents that may be life-threatening.

IoU: intersection of union

The complexity of the object detection problem varies according to different input modalities:

- Video cameras are the cheapest solution
- LiDAR data

- Camera images + LiDAR data. Most modern autonomous driving solutions employ both LiDAR and RGB camera sensors for perception. These multi-modal perception systems can be classified into two broad categories: cascaded models which use each modality independently, and fusion models which learn from different modalities simultaneously [2].

To approach the problem, a detection pipeline must be established. Typically, the following steps are required:

- Preprocessing.
- Region of Interest (ROT) extraction.
- Object classification.
- Verification.

Today's state of the art approaches for object detection employ Convolutional Neural Networks, CNNs [3] [4]. A problem with Deep CNNs with large receptive fields is that local information is extracted in the early layers, while higher-level represented in deeper layers [1], making the precise localization of the object more difficult. A solution to this problem was proposed by Girshick et al. [5] with Region Based Convolutional Neural Networks, R-CNNs. This scalable approach solves the localization problem by generating many region proposals using selective search [6] to extract a fixed-length feature vector for each proposed region using a CNN and classifying each region with a linear SVM.

CNNs can capture different patterns

2.3 3D Object detection

2.3.1 3D object detection from 2D images

2.3.2 3D object detection from 3D point clouds

The goal of 3D object detection is to predict a three-dimensional bounding box around each object of interest.

Similarly to what happens with 2D detection, IoU can be used to evaluate the performance of a model.

Dataset

- Street signs: German Traffic Sign Recognition Benchmark
- Pedestrians, vehicles and

Formal definition . . . In the case of self-driving vehicles specifically, we can assume a value of zero for roll and pitch since the car is "glued" to the road. Frustum...

, is a more scalable alternative to traditional CNNS. This approach, originally proposed by R. Girshick et al. in 2014. . .

CNNs have been successfully employed on point cloud data as well; PointRCNN

Only image: Monocular 3D Object Detection for Autonomous Driving and Geometru-based Distance Decompositon for Monocular 3D Object Detection or Pseudo-LiDAR

Only LiDAR:

Both:

2.4 Adversarial attacks on neural networks

Most vision-based recognition software on ADS is based on CNNS; often, CNN-based deep learning models are vulnerable to the so called adversarial,

There can be small, pixel-level changes to an image that will cause the AI model to incorrectly interpret it or, on the other hand, a completely new image can be used to trick to model into thinking it is something else. The first kind is particularly dangerous, since such changes can be invisible to the human eye, and thus harder to detect.

There are mainly two categories of methods to achieve adversarial attacks, namely, optimization-based methods and fast gradient step method (FGSM)-based approach.

In [7], Zhang et al. propose and end-to-end evaluation framework for assessing the safety of a self driving deep learning model.

study on street signs: K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018, pp. 1625–1634.

In general, adversarial attacks are organized in three categories: evasion, poisoning, and extraction attacks:

- Evasion attacks modify the input to a classifier such that it is misclassified, while keeping the modification as small as possible. Evasion attacks can be black-box or white-box: in the white-box case, the attacker has full access to the architecture and parameters of the classifier. For a black-box attack, clearly this is not the case.

- In poisoning attacks, attackers have the opportunity of manipulating the training data to significantly decrease the overall performance, cause targeted misclassification or bad behavior, and insert backdoors and neural trojans

- Extraction attacks aim to develop a new model, starting from a proprietary black-box model, that emulate the behavior of the original model.

Chapter 3

Literature

In this chapter we provide a general overview of the literature concerning 3D object detection in self-driving vehicles, as well as the solutions employed to deal with adversarial attacks on their deep learning models

Example algorithm

Algorithm 1: MOSA

input : $U = \{u_1, ..., u_m\}$ the set of coverage targets of a program
Population size M
output : A test suite T

begin
    $t \leftarrow 0$
    $P_t \leftarrow$ RANDOM-POPULATION(M)
    archive $\leftarrow$ UPDATE-ARCHIVE(P_t, Ø)
    while *not*(*search_budget_consumed*) **do**
        $Q_t \leftarrow$ GENERATE-OFFSPRING(P_t)
        archive $\leftarrow$ UPDATE-ARCHIVE(Q_t, *archive*)
        $R_t \leftarrow P_t \cup Q_t$
        $F \leftarrow$ PREFERENCE-SORTING(R_t)
        $P_{t+1} \leftarrow 0$
        $d \leftarrow 0$
        while $(|P_{t+1}| + |F_d| \leqslant M)$ **do**
            CROWDING-DISTANCE-ASSIGNMENT(F_d)
            $P_{t+1} \leftarrow P_{t+1} \cup F_d$
            $d \leftarrow d + 1$
        Sort(F_d) // according to the crowding distance
        $P_{t+1} \leftarrow P_{t+1} \cup F_d[1 : (M - |P_{t+1}|)]$
        $t \leftarrow t + 1$
    $T \leftarrow$ *archive*

Chapter 4

Conclusions

Bibliography

[1] Joel Janai et al. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”. In: *Found. Trends Comput. Graph. Vis.* 12.1-3 (2020), pp. 1–308. DOI: 10.1561/0600000079. URL: https://doi.org/10.1561/0600000079.

[2] Mazen Abdelfattah et al. “Adversarial Attacks on Camera-LiDAR Models for 3D Car Detection”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*. IEEE, 2021, pp. 2189–2194. DOI: 10.1109/IROS51168.2021.9636638. URL: https://doi.org/10.1109/IROS51168.2021.9636638.

[3] Zhaowei Cai et al. “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection”. In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*. Ed. by Bastian Leibe et al. Vol. 9908. Lecture Notes in Computer Science. Springer, 2016, pp. 354–370. DOI: 10.1007/978-3-319-46493-0_22. URL: https://doi.org/10.1007/978-3-319-46493-0%5C_22.

[4] Xiaozhi Chen et al. “3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.5 (2018), pp. 1259–1272. DOI: 10.1109/TPAMI.2017.2706685. URL: https://doi.org/10.1109/TPAMI.2017.2706685.

[5] Ross B. Girshick et al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81. URL: https://doi.org/10.1109/CVPR.2014.81.

[6] Jasper R. R. Uijlings et al. “Selective Search for Object Recognition”. In: *Int. J. Comput. Vis.* 104.2 (2013), pp. 154–171. DOI: 10.1007/s11263-013-0620-5. URL: https://doi.org/10.1007/s11263-013-0620-5.

[7] Jindi Zhang et al. “Evaluating Adversarial Attacks on Driving Safety in Vision-Based Autonomous Vehicles”. In: *IEEE Internet Things J.* 9.5 (2022), pp. 3443–3456. DOI: 10.1109/JIOT.2021.3099164. URL: https://doi.org/10.1109/JIOT.2021.3099164.

[8] Vincenzo Riccio et al. “Testing machine learning based systems: a systematic mapping”. In: vol. 25. 6. 2020, pp. 5193–5254. DOI: 10.1007/s10664-020-09881-0. URL: https://doi.org/10.1007/s10664-020-09881-0.

[9] Dmytro Humeniuk, Foutse Khomh, and Giuliano Antoniol. “A search-based framework for automatic generation of testing environments for cyber-physical systems”. In: *Inf. Softw. Technol.* 149 (2022), p. 106936. DOI: 10.1016/j.infsof.2022.106936. URL: https://doi.org/10.1016/j.infsof.2022.106936.

[10] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[11] Edmund K. Burke and Graham Kendall. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer US, 2014, pp. 403–449.

[12] Stefan Edelkamp and Stefan Schrodl. *Heuristic Search: Theory and Applications*. Morgan Kaufmann, 2008, pp. 403–449.

[13] Walid M. Taha, Abd-Elhamid M. Taha, and Johan Thunberg. *Cyber-Physical Systems: A Model-Based Approach*. Springer US, 2021.

[14] *Check: framework for unit testing in C*. URL: https://libcheck.github.io/check/.

[15] *JMetal Java framework*. URL: http://jmetal.sourceforge.net/.

[16] A. Author and A. Author. *Book reference example*. Publisher, 2099.

[17] A. Author. “Article title”. In: *Journal name* (2099).

[18] *Example*. URL: https://www.isislab.it.

[19] A. Author. “Tesi di esempio ISISLab”. 2099.