Focus Technique

SiMOOD: Evolutionary Testing Simulation with Out-Of-Distribution Images

Safer
Autonomous
Systems

This project has received funding from the European Union's EU Framework Programme for Research and Innovation Horizon 2020 under Grant Agreement No. 812.788

Raul Sena Ferreira, Joris Guerin,
Jérémie Guiochet, Hélène Waeselynck

13 January 2023

This talk is divided into two parts

- **Scientific context**
  Motivation, solution architecture, results, and limitations

- **Reproducibility instructions**
  Code repository, installing CARLA simulator, installing SimOOD and its dependencies, troubleshooting

Introduction

Deep learning (DL) techniques can be wrong in their predictions even with 100% confidence [1]

=> Potentially leading to hazardous situations in cyber-critical systems

Dependability-ensuring techniques, such as fault tolerance, can be applied
=> Safety monitors (SM) keep the system in a safe state despite hazardous situations [2]

Such monitors aim to detect out-of-distribution (OOD) images at runtime:

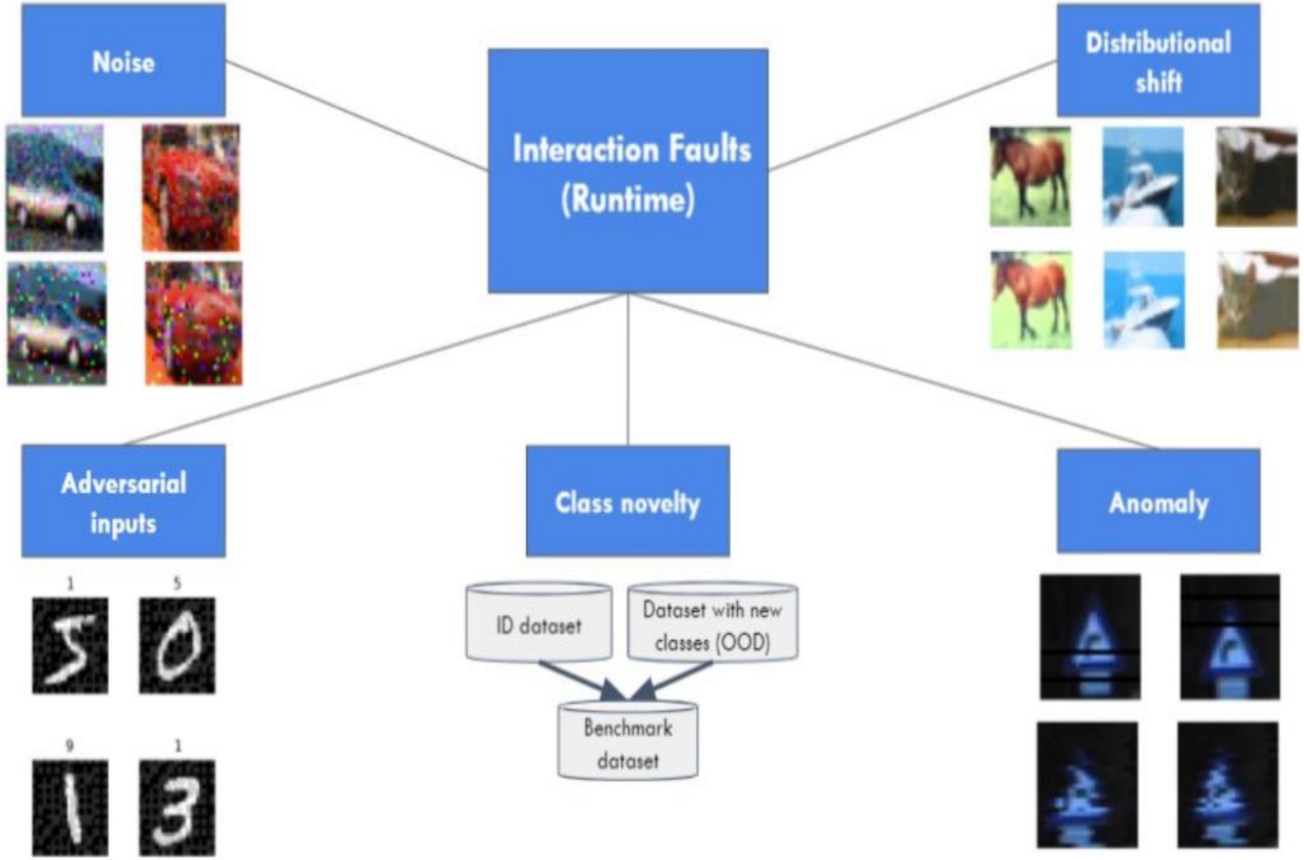
- All data that falls outside of the expect i.i.d* assumption can be considered as OOD data
- OOD data is considered a major threat for image classifiers and object detectors

* **independent and identically distributed data => the same probability distribution as the others and all are mutually independent**

Out-of-distribution data

There are five main types of OOD characteristics that can come on images at runtime

- Noise [9],
- Distributional-shifts [8],
- Novelty classes [6],
- Anomalies [10],
- Adversarial inputs [7]

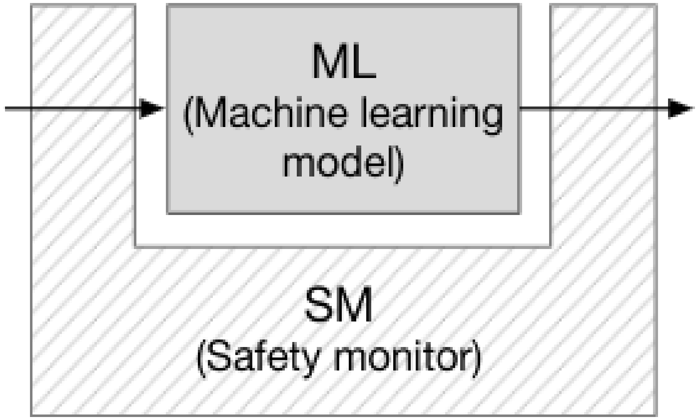


Recent works focus on **data-based** monitors for DL

=> Data-based SM is generally built from the same training data used to build the DL model

Safer Autonomous Systems

Data-based ML monitors



Data-based monitors for DL image classifiers fall in 3 categories:

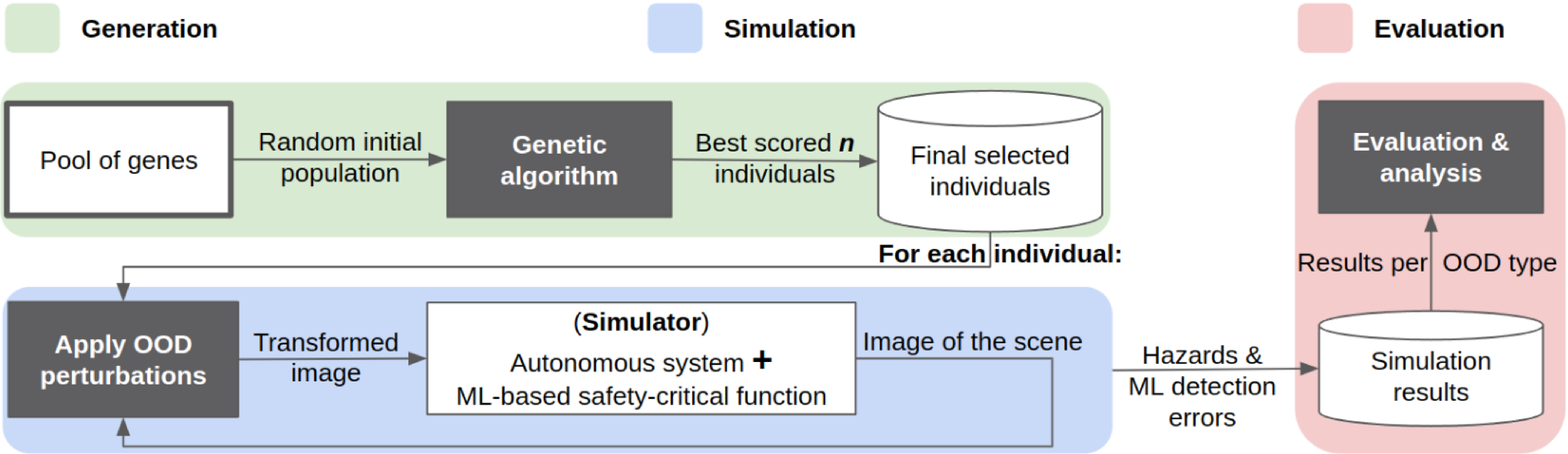
- Observation of the inputs of the DL model [3]
- Observation of the intermediate layers of the DL model [4]
- Observation of the output (decision) from the DL model [5]

Similar to uncertainties inherent to the use of ML, the confidence in such SM is an open issue

- Testing them in a perception system cannot be reduced to measuring ML performances on a dataset but rely on the images captured by the system at runtime
- However, the amount of time spent to generate diverse test cases during a simulation of perception components can grow quickly since it is a combinatorial optimization problem

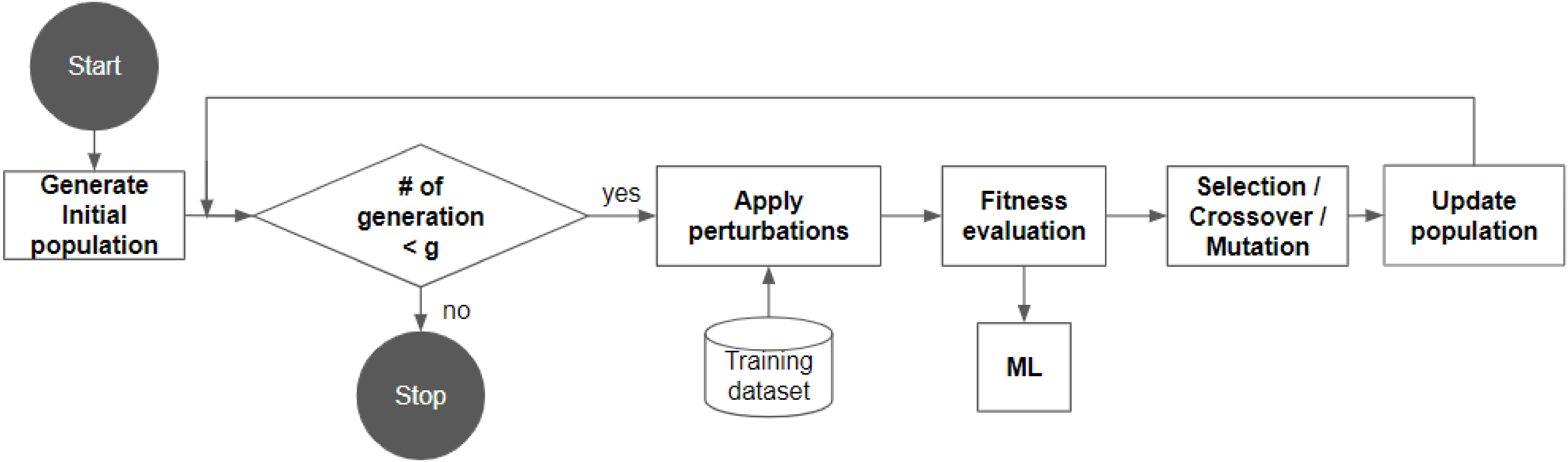
SiMOOD overview

- **Generation**: it performs the task of finding combinations of OOD perturbations with a GA
- **Simulation**: it takes the selected individuals and apply them to each frame of the simulation
- **Evaluation**: it yields processing time, memory, hazards and ML metrics

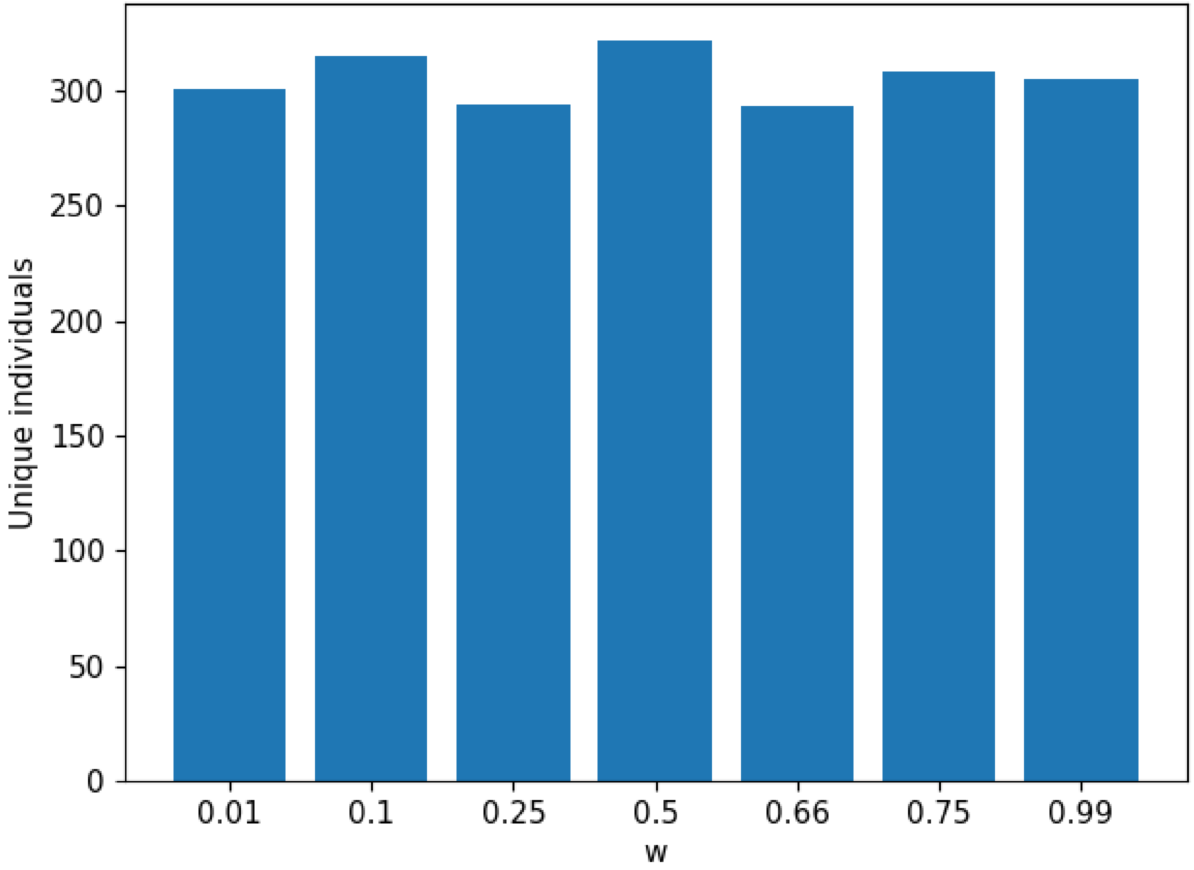


GA approach

- We applied 15 categories of OOD perturbations [15],[16], [17], each one with its own levels of intensity ("no effect" included), totaling 175 different OOD perturbations



Robustness of SiMOOD regarding its parameters



Unique individuals per ω, generated across all variations of generations and population size (440 individuals).

TABLE II: Number of unique genes in the selected population.

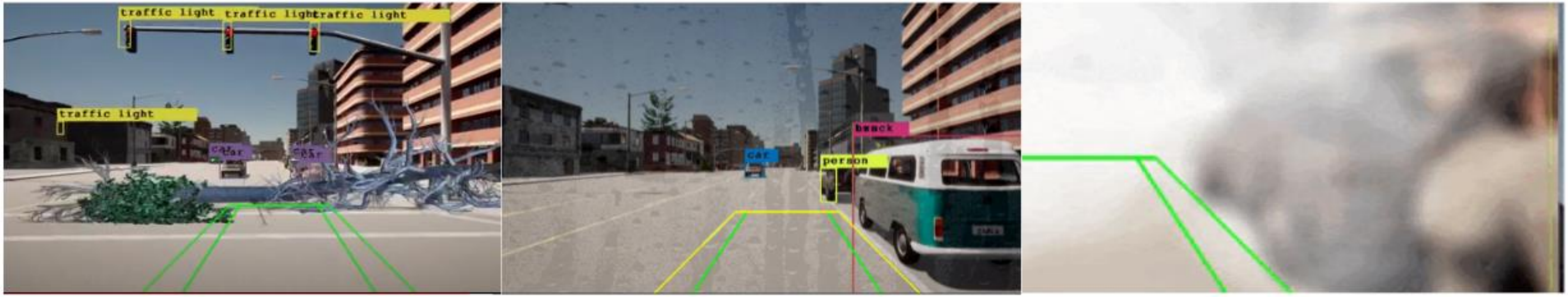
Generations	Population size 10	20	30	50
10	7 (35%)	16 (40%)	34 (57%)*	59 (59%)*
20	8 (40%)*	20 (50%)*	28 (47%)	48 (48%)
30	6 (30%)	5 (12%)	26 (43%)	42 (42%)
50	6 (30%)	7 (17%)	19 (31%)	32 (32%)

TABLE III: Number of hazards.

Generations	Population size 10	20	30	50
10	9 (90%)	11 (55%)	12 (40%)	12 (24%)
20	6 (60%)	20 (100%)*	23 (77%)	21 (42%)
30	10 (100%)*	20 (100%)*	30 (100%)*	25 (50%)
50	10 (100%)*	20 (100%)*	15 (50%)	23 (46%)

Hazards uncovered by applying single OOD perturbations

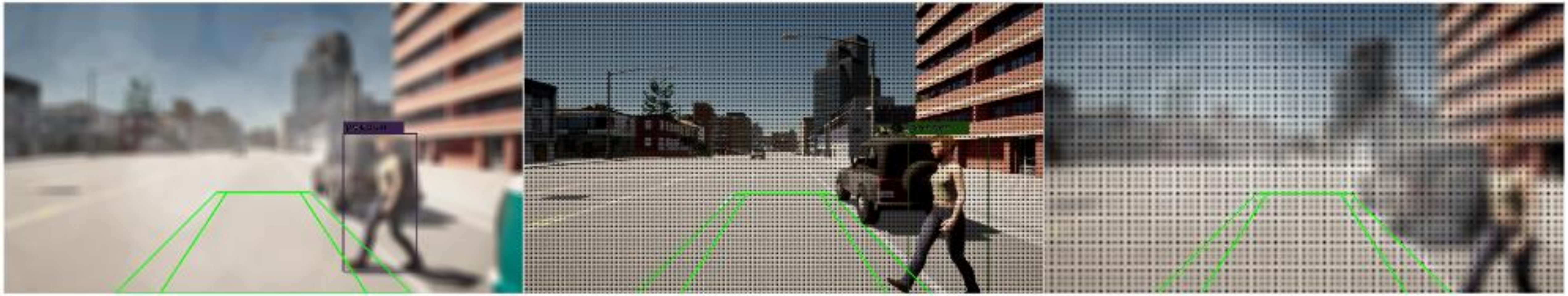
a) Crash due to a fallen tree not detected by the ML model
b) A false detection provoked by condensed water on the camera lens
c) Crash with a pedestrian when exposing the ML model to heavy smoke



a) Novelty class b) Condensed water c) Heavy smoke

Hazards uncovered by combining OOD perturbations

- When one of these combinations happens alone (sub-figures a) and b)), the ML model can correctly detect the pedestrian
- However, the combination of both, even with a lower intensity, can be enough to lead to a hazard

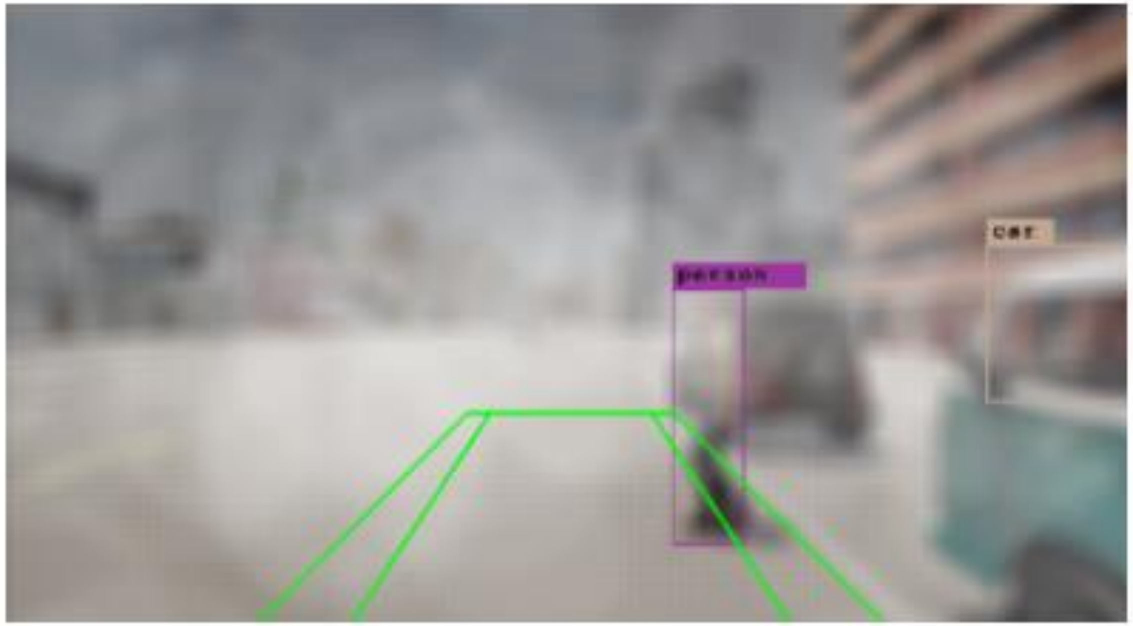


a) Smoke (0.25) b) Grid dropout (2) c) Smoke(0.18) + Grid dropout(2)

OOD perturbations in different order with different outcomes

The order of the perturbations also matters:

- Same perturbations combined in a different order produce subtle differences in the image



(a) Grid dropout (1) + smoke (0.3).

(b) Smoke (0.3) + grid dropout (1).

Processing time and memory

- SiMOOD applies perturbations on high-resolution images (1280x720)
  - it is necessary an extra amount of memory (2.7 GB) to perform the task
- SiMOOD can be optimized to perform better processing and consume less memory
  - By performing parallelization and data compression

TABLE IV: Comparison of processing time and memory.

Time	Time (with SiMOOD)	Overhead
101.94	123.10	20.75%
Memory	**Memory (with SiMOOD)**	**Overhead**
3975.58	6708.09	68.73%

Usage details

Installation is divided in two parts

- CARLA simulator: tested with “carla-0.9.11-py3.7-linux-x86_64.egg” for Linux
- SimOOD: dependencies can be installed with “pip install -r requirements”

SimOOD can be used for two purposes

- Offline: search for OOD perturbations that may lead to hazards during the simulation
- Online: simulate scenarios with specific OOD perturbations or combination of perturbations

Limitations

- We tested just one type of state-of-the-art object detector (YOLO v6)
- At least 16MB of memory available
- Variations over a fixed scenario (case study)

Live interaction time!

https://github.com/raulsenaferreira/SiMOOD

References

[1] Gal, Y and Ghahramani, Z. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”, in International conference on machine learning (ICML), New York, United States, 2016

[2] Machin, M and Guiochet, J and Waeselynck, H and Blanquart, J and Roy, M and Masson, L. “SMOF: A safety monitoring framework for autonomous systems”, in IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018

[3] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, “Adversarially learned one-class classifier for novelty detection,” in CVPR 2018

[4] T. A. Henzinger, A. Lukina, and C. Schilling, “Outside the box: Abstraction-based monitoring of neural networks,” in ECAI 2020

[5] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” in ICLR 2018

[6] Perera, P and Patel, VM, “Deep transfer learning for multiple class novelty detection”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019

References

[7] Kurakin, A and Goodfellow, I and Bengio, S, “Adversarial examples in the physical world”, 2016, accessed in 2021/10 at arXiv preprint arXiv:1607.02533

[8] Ferreira, RS and Zimbrao, G and Alvim, LGM, “AMANDA: Semi-supervised Density-based Adaptive Model for Non-stationary Data with Extreme Verification Latency”, in Information Sciences, pages 219-237, issue 2019

[9] Hendrycks, D and Dietterich, T, “Benchmarking neural network robustness to common corruptions and perturbations”, 2019, accessed 2021/10 at arXiv preprint arXiv:1903.12261

[10] Semiconductor Components Industries, LLC, “Evaluating Functional Safety in Automotive Image Sensors”, 2018, accessed at https://www.onsemi.cn/pub/Collateral/TND6233-D.PDF

[11] T. Dreossi, A. Donze, and S. A. Seshia, “Compositional falsification of cyber-physical systems with machine learning components,” Journal of Automated Reasoning, vol. 63, no. 4, pp. 1031–1053, 2019.

[12] G. Rossolini, F. Nesti, G. D’Amico, S. Nair, A. Biondi, and G. Buttazzo, “On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving,” arXiv preprint arXiv:2201.01850, 2022.

References

[13] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, “Attacking vision-based perception in end-to-end autonomous driving models,” Journal of Systems Architecture, vol. 110, p. 101766, 2020.

[14] H. Fahmy, F. Pastore, and L. Briand, “Hudd: A tool to debug DNNs for safety analysis,” in 2022 IEEE/ACM 44st International Conference on Software Engineering, 2022.

[15] F. Secci and A. Ceccarelli, “On failures of rgb cameras and their effects in autonomous driving applications,” in 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2020, pp. 13–24.

[16] R. S. Ferreira, J. Arlat, J. Guiochet, and H. Waeselynck, “Benchmarking safety monitors for image classifiers with machine learning,” 26th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2021), Perth, Australia, 2021.

[17] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: fast and flexible image augmentations,” Information, vol. 11, no. 2, p. 125, 2020.

Safer Autonomous Systems

Thank you

Email:

rsenaferre@laas.fr

raul.ferreira@continental.com

Project repository:

https://github.com/raulsenaferreira/SiMOOD