Focus Technique

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

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
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.
Data-based ML monitors 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

[Diagram showing the process of generation, simulation, and evaluation]
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 $\omega$, generated across all variations of generations and population size (440 individuals).

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

<table>
<thead>
<tr>
<th>Generations</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7 (35%)</td>
<td>16 (40%)</td>
<td>34 (57%)*</td>
<td>59 (59%)*</td>
</tr>
<tr>
<td>20</td>
<td>8 (40%)*</td>
<td>20 (50%)*</td>
<td>28 (47%)</td>
<td>48 (48%)</td>
</tr>
<tr>
<td>30</td>
<td>6 (30%)</td>
<td>5 (12%)</td>
<td>26 (43%)</td>
<td>42 (42%)</td>
</tr>
<tr>
<td>50</td>
<td>6 (30%)</td>
<td>7 (17%)</td>
<td>19 (31%)</td>
<td>32 (32%)</td>
</tr>
</tbody>
</table>

### TABLE III: Number of hazards.

<table>
<thead>
<tr>
<th>Generations</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9 (90%)</td>
<td>11 (55%)</td>
<td>12 (40%)</td>
<td>12 (24%)</td>
</tr>
<tr>
<td>20</td>
<td>6 (60%)*</td>
<td>20 (100%)*</td>
<td>23 (77%)</td>
<td>21 (42%)</td>
</tr>
<tr>
<td>30</td>
<td>10 (100%)*</td>
<td>20 (100%)*</td>
<td>30 (100%)*</td>
<td>25 (50%)</td>
</tr>
<tr>
<td>50</td>
<td>10 (100%)*</td>
<td>20 (100%)*</td>
<td>15 (50%)</td>
<td>23 (46%)</td>
</tr>
</tbody>
</table>
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
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.
The order of the perturbations also matters:
- Same perturbations combined in a different order produce subtle differences in the image.
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.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Time (with SiMOOD)</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>101.94</td>
<td>123.10</td>
<td>20.75%</td>
</tr>
<tr>
<td>Memory</td>
<td>3975.58</td>
<td>6708.09</td>
<td>68.73%</td>
</tr>
</tbody>
</table>
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


References


Thank you

Email:
rsenaferre@laas.fr
raul.ferreira@continental.com

Project repository:
https://github.com/raulsenaferreira/SiMOOD