Deeper at the SBST 2021 Tool Competition: ADAS Testing Using Multi-Objective Search

Mahshid Helali Moghadam*, Markus Borg*, Seyed Jalaleddin Mousavirad‡
* RISE Research Institutes of Sweden, Sweden
{mahshid.helali.moghadam, markus.borg}@ri.se
† Mälardalen University, Västerås, Sweden
‡ Department of Computer Engineering, Hakim Sabzevari University, Sabzevar, Iran
{sj.mousavirad}@hsu.ac.ir

Abstract—Deeper is a simulation-based test generator that uses an evolutionary process, i.e., an archive-based NSGA-II augmented with a quality population seed, for generating test cases to test a deep neural network-based lane-keeping system. This paper presents Deeper briefly and summarizes the results of Deeper’s participation in the Cyber-physical systems (CPS) testing competition at SBST 2021.

Index Terms—search-based software testing, cyber-physical systems, advanced driver assistance systems, deep learning, automotive simulators

I. INTRODUCTION

There is an increasing trend of leveraging machine learning techniques, in particular deep learning, in a wide range of application areas. Self-driving cars are one of those areas that have benefited considerably from deep learning and its associated techniques such as deep neural networks (DNN). To perform Verification and Validation (V&V) for self-driving cars, there is a need for System-level testing to ensure safety and reliability of these systems before making them available to be used publicly [1].

Simulation-based testing is one of the potential approaches for system-level testing, as a proper complementary solution to field testing. Meanwhile, the approach could capture the entire operational environment efficiently and effectively using thorough and high-fidelity physics-based simulators [2]. Lately, various commercial and open-source simulators have been developed to support the need for realistic simulation of self-driving cars [3–5]. From the literature, several system-level testing techniques, relying on such simulators, have been proposed in recent years. One category of test generation techniques is based on deriving critical test inputs from the input domain using different search-based testing approaches [6–8]. The critical test inputs are those that break, or almost break, the safety requirements of the system under test.

This paper presents the results obtained by our proposed approach, the Deeper test generator tool [9], in the Cyber-physical systems (CPS) testing competition at SBST 2021. Deeper uses an evolutionary process mainly based on Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [10], a multi-objective optimization algorithm, to search the input space and generate critical test inputs for a lane-keeping assistance system (LKAS) in the competition setup. The tool has been integrated into the competition’s testing infrastructure according to the competition’s guideline and executed against the test subject with regard to certain test budgets and pre-defined driving conditions, such as single, flat roads surrounded by green areas and fixed weather conditions.

II. DEEPER

Deeper leverages an initial quality population seed to boost the search process with respect to the fixed test budget. It uses Catmull-Rom cubic splines [12] to represent the roads and relies on the evolution operators implemented by DeepJanus [13]. Throughout the development, the impact of the different initial population seeds has been investigated. For instance, starting from a random population seed like an initial seed generated based on a recursive evaluation algorithm for Catmull-Rom cubic splines [14] could not lead the search to find the desired solutions for high failing thresholds within a reasonable test budget.

III. BENCHMARK RESULTS

Deeper is integrated into the testing infrastructure of the competition, which uses the research-specific BeamNG driving simulator. The test subject is BeamNG.AI, the built-in machine learning-enabled driving agent in the simulator. Two experiments, DEFAULT and SBST21, with different setups, presented in Table 1 were arranged. The
TABLE I
EXPERIMENTS’ SETTINGS

<table>
<thead>
<tr>
<th>Ex. Name</th>
<th>Test Budget</th>
<th>Map Size</th>
<th>Speed Limit</th>
<th>Failing Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>5h</td>
<td>200 × 200</td>
<td>None</td>
<td>0.95</td>
</tr>
<tr>
<td>SBST21</td>
<td>2h</td>
<td>200 × 200</td>
<td>70 Km/h</td>
<td>0.85</td>
</tr>
</tbody>
</table>

experiments were executed on a PC, with a quad-core Intel i7-7700K CPU, 16 GB RAM, and an NVidia GeForce GTX 1080 GPU, that runs Microsoft Windows 10 [15].

The target of the competition was to generate the highest number of diverse test cases leading to failures, i.e., the valid test inputs causing the car to get out of the lane. The quality criteria used in the competition for evaluating the tools are as follows [15]:

**Detected Failures**, which refer to the number of test cases that led the car to drive out of the lane with respect to the pre-defined tolerance threshold. Regarding this metric, Deeper could expose failures with an average rate of 2.6 and 0.5 in all runs on DEFAULT and SBST21 setups, respectively.

**Failure Diversity**, which represents the calculated sparseness between the test inputs leading to the failures. The version of Deeper submitted to the competition did not propose any specific mechanisms for promoting diversity between the generated test roads, thus it showed low sparseness for the generated test cases.

**Test Generation Effectiveness and Efficiency**, which relate to how the test generator tool uses the allocated test budget to generate the test cases, i.e., how many test cases are generated in total, and what fraction of test cases are valid (or invalid). Regarding efficiency and effectiveness, Deeper showed satisfactory performance. The tool generated many test cases within the allocated test budget in both experimental setups, i.e., Deeper ranked second regarding the number of generated test cases. Meanwhile, it generated the highest number of valid test cases among the competing tools, and achieved around 90% generation effectiveness in both setups. Notably, it did neither generate any self-intersecting test roads nor any roads outside the map boundaries [15].

ACKNOWLEDGMENT

This work has been funded by Vinnova through the ITEA3 European IVVES project (https://itea3.org/project/ivves.html). Furthermore, the project received financial support from the SMILE III project financed by Vinnova, FFI, Fordonstrategisk forskning och innovation under the grant number: 2019-05871 and Kompetensfonden at Campus Helsingborg, Lund University, Sweden.

REFERENCES