Data synthetization for verification and validation of Machine Learning based systems

Hamid Ebadi[1], Joakim Rosell[2], Thanh Bui[2], Lukáš Maršík[3], Jack Jensen[4] and Martin Karsberg[1]

(1) INFOTIV, Gothenburg, Sweden, e-mail: {hamid.ebadi, martin.karsberg}@infotiv.se (2) RISE Research Institutes of Sweden, Sweden, e-mail: {joakim.rosell, thanh.bui}@ri.se (3) CAMEA, Brno, Czech, e-mail: l.marsik@camea.cz (4) BERGE, Gothenburg, Sweden, e-mail: jack.jensen@berge.io

Abstract - *The incorporation of machine learning (ML) components presents challenges to the verification and validation (V&V) process due to the inherent opacity of ML systems. This paper introduces a tool-chain for generating synthesize datasets, aimed at facilitating search-based testing on a traffic monitoring system that utilizes machine learning.*

Keywords: *Simulator, machine-learning, verification and validation, traffic scenario, synthetic data*

Introduction

Machine learning (ML) is a key enabler in many applications today, including traffic monitoring systems. Unlike rule-based systems, ML models are data-driven making their verification and validation complex and challenging. In the realm of verification and validation (V&V) methodologies, numerous approaches have been suggested. However, employing graphical simulators appears to be the most advantageous option for testing the requirements of the perception component to identify potential issues early in the production phase. Under certain circumstances, the use of simulators has proven to be advantageous, especially when; real-world testing poses a risk to human safety, when the collection and annotation of data is difficult, the data coverage is constrained, and when the need for reproducibility and scalability is critical.

Since the possible input space when testing automotive systems is practically infinite, attempts to design test cases for comprehensive testing over the space of all possible simulation scenarios are futile. Hence, mutation and search based testing has been advocated as an effective and efficient strategy for generating test scenarios in simulators.

Within the VALU3S project(VALU3S, 2020), a method namely "Scenario based V&V automation using simulator"(VALU3S, 2022) has been proposed to exploit synthetic data generated by graphical simulators for V&V of ML-based systems(Borg, et al., 2023). The method has been demonstrated through an application on the CAMEA traffic control system, with focus on its ML-based License Plate (LP) detection component. The requirements for synthetic V&V data are structured based on the assurance desiderata proposed by (Ashmore, Calinescu, and Paterson, 2022), which are categorized into four key properties: relevant, complete, balanced, and accurate. A tool-chain comprising a graphical simulator (BERGE simulator) and a scenario manipulator has been proposed to generate synthetic data conforming to the four properties mentioned above. Specifically, the BERGE simulator is responsible for ensuring the relevant and accurate properties, whereas the scenario manipulator focuses on the complete and balanced properties.

CAMEA Traffic Control System

CAMEA is a Czech-based company that specializing in the development and production of embedded systems specifically designed for traffic monitoring. Their most advanced and intricate product is the Unicam system (Fig. 1), which is a highly sophisticated and versatile platform that provides high-performance solutions to the challenges encountered in traffic monitoring. Its advanced video processing algorithms, along with its multi-functional and scalable design, enable it to accurately identify critical information and facilitate effective preventive actions. The Unicam system has also practical application in industrial inspection systems.

The Unicam-based traffic monitoring systems are equipped with high-resolution cameras that have advanced image processing capabilities and algorithms, such as the License Plate (LP) detection algorithm that facilitates Optical Character Recognition (OCR). The main objective of the Unicam system is to accurately identify vehicles and extract relevant information, such as speed, LP number, and trajectories, to detect vehicles that are driving in the wrong lane or exceeding speed limits, thereby triggering preventive actions. In this paper, the Unicam system will be referred to as an LP recognition system.

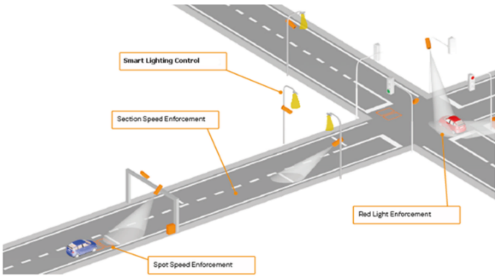

Figure 1: Unicam traffic surveillance system.

Figure 2: Åkareplatsen resecentrum (Bjarnefors, 2017) (top) and its 3D model in BERGE simulator (bottom).

BERGE Simulator

Various simulators have been recently developed to meet the demand for realistic simulation of autonomous driving functions. These simulators include SVL simulator (Rong, et al., 2020), PreScan, and Pro-SiVIC (Hiblot, et al., 2010), as well as those relying on game engines such as BeamNG (BeamNG GmbH, 2022) and CARLA (Dosovitskiy, et al., 2017). However these simulators lack the user friendly interface to quickly and flexibly define a set of test scenarios. Therefore an Unreal Engine-based (Epic Games, 2019) traffic simulator has been developed by one of the research partner, BERGE, in VALUE3S (VALU3S, 2020) project to define realistic traffic scenarios with all required input parameters.

One of the major challenges in using synthetic data from simulator for V&V is to ensure that the simulator accurately represents and models the real-world environment. This involves carefully and thoroughly modeling of the physical, and behavioral characteristics of the real-world environment in the simulator. In the VALU3S project, Åkareplatsen resecentrum (Fig. 2), a real location in the central Gothenburg, Sweden, has been selected for the

Figure 3: Highway and country road scenes.

placement of CAMEA Unicam sensors (radar and camera). The cameras are triggered by events (at regular intervals or triggered by radar-based speed sensors) and capture pictures of the scene. These images are later transferred and processed by an ML-based system.

A Further walk through of the efforts to accurately depict the static and dynamic aspects of the real world includes scenario parameters and how they are implemented in the BERGE simulator.

Diversity of Scenes

BERGE simulator supports three typical traffic scenes: (i) Urban area by modeling Åkareplatsen resecentrum as shown in Fig. 2, (ii) a highway scene, and (iii) a country road as shown in Fig. 3.

Fidelity

Fidelity refers to the degree to which the simulator accurately represents the real-world. There are several parameters that can be used to measure the fidelity, such as the number of polygons, shaders, ray tracing range, and sensor resolution, all of which are indicative of the simulator's rendering capability.

In the Åkareplatsen scene there are 4782 assets with a total of 20.2 million polygons. In the country road scene, there are 12470 assets with 8.8 million polygons. And in the highway scene has 6638 assets with a total of 28.5 million polygons in total. All three scenes run at over 60 FPS (frames per second), with the highway scene for instance running at 68 FPS.

Shaders are typically used to control lighting, shadows, color, texture mapping, and other visual effects. There are in total of 1491 shaders in BERGE simulator. Shaders also have a big impact on the FPS, so it is important to create the shaders for real-time applications like an Unreal Engine scene.

Visual Scenario Editor

A visual editor assists operators to define relevant base scenario using a graphical interface. The visual editor facilitates manipulation of agents (e.g. cars, pedestrians), their trajectories (way points) and their properties (e.g. color, type) using an intuitive graphical interfaces to save time and effort, especially for operators who are less familiar with programming.

For example, for a hypothetical scenario involving

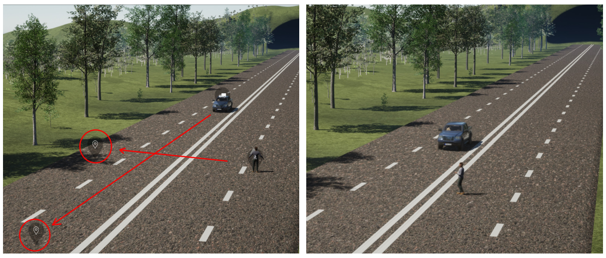

Figure 4: Placing elements in visual scenario editor (top) and execution of the scenario (bottom).

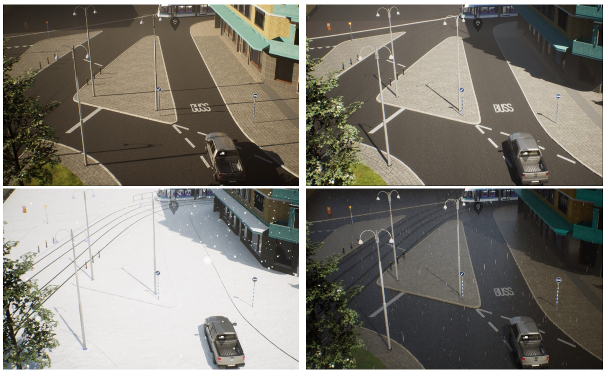

Figure 5: Different weathers and time, evening, sunny, snowy, rainy.

a pedestrian recklessly crossing a highway, a car and a pedestrian, along with waypoints describing their trajectories and associated speeds.

The following types of agents are supported for use as a part of a scenario:

- Pedestrians of different genders and clothing styles
- Vehicles: cars and trucks of different brands, colors and models

Fig. 4, illustrates the addition of way-points and agents as well as the successful execution of the scenario.

The resulting scenario can be saved persistently in a JSON file format and restored from either the visual editor or the command prompt.

Environment Parameters

It is important that simulators support a variety of weather conditions and lightning situations to ensure that the perception component is adequately tested and validated prior to real-world deployment.

The BERGE simulator offers the flexibility to set and customize different weather conditions, including sunny, rainy, and snowy weather, as well as varying intensities of precipitation. In addition, the user has the ability to manipulate the placement of light sources, their intensity, and the time of day, which consequently affects the overall lighting, reflections, and shadows in the scene. Reader can refer to Fig. 5 for a demonstration of this feature.

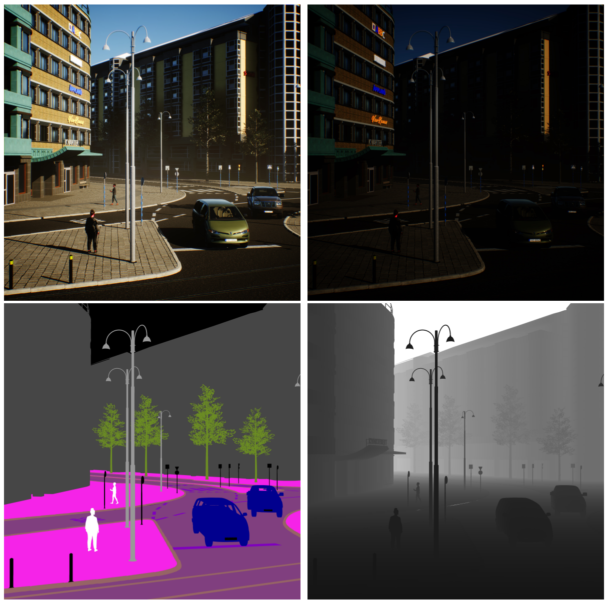

Figure 6: Normal image, distorted image, semantic segmentation, depth sensor.

Sensor models

The accuracy of the synthetic data is determined by the fidelity of the simulator and the accuracy of the sensor models, i.e. the ability to generate sensor responses that are close to the outputs of real sensors. The BERGE simulator offers a diverse set of sensor models, including cameras, cameras with distorted outputs, depth sensors, radars, and semantic segmentation sensors as demonstrated in Fig. 6. The sensor models available in the BERGE simulator offer different resolution and parameter options for their respective outputs.

Scenario Manipulator

One of the key challenges in utilizing synthetic data from simulators for V&V is to guarantee that the simulator accurately captures a comprehensive and well-balanced representation of the real-world environment. This requires that the test scenarios should cover a valid range of inputs parameters, while also adhering to the accurate distribution of inputs with regards to Operational Design Domain (ODD). For instance, Sweden's geographical location introduces unique variations in lighting conditions that has to be captured. Mutation and search based testing are ideal methods for evaluating the quality and robustness of an ML based system. Once a template scenario is specified according to the ODD, the next step is to mutate the scenario parameters to create a set of scenarios according to the predefined input parameter range and distribution that can be fed into the ML model. Scenario Manipulator (Ebadi, 2023) is a tool to generate such set of scenarios from a relevant base scenario for this purpose, thus enabling completeness and balance properties.

Currently, number of input parameters can be defined for each scenario, and the system should be

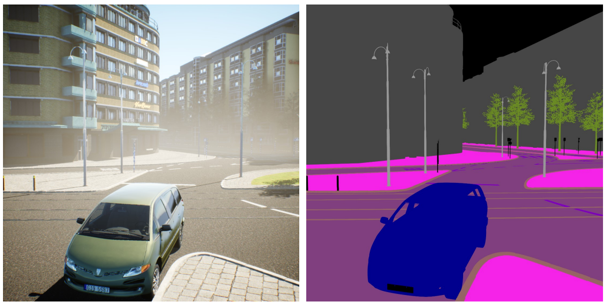

Figure 7: Semantic segmentation of a scene in BERGE simulator.

Figure 8: Semantic segmentation by BERGE simulator and bounding box detection by Unicam LP detection.

robust to their variations within the defined boundaries. For most static objects (waypoints, light sources, sensors) and agents (vehicles and pedestrians), their position (x, y, z) and their rotation (roll, pitch, yaw) are modifiable. For waypoints, the target speed, initial speed and maximum speed and for agents, their type/model, start delay before the agent starts moving, color, gender, body type, and their accessories are modifiable. For cameras, parameters such as aperture, focal length, exposure, shutter speed, ISO, field of view can be modified, and for radars horizontal and vertical ranges and reach can be set. Finally for the environment, you can change the weather conditions and the amount of precipitation, and for the scene, you can change the time (hour, minute) and the number plates of the vehicles.

The acceptable range of change is determined by the experts and is intended to be large enough to test the robustness of the system but at the same time small enough to protect the integrity of the scenario and the ODD.

The parameters specified in the JSON file may have diverse valid ranges or may even be of different data types. For example, the field of view and object coordinates are both real numbers but may belong to different ranges, while the car model is a string belonging to a limited set of supported cars, and the precipitation value is a numeric value.

Similar to ScenarioGenerator (Ebadi, et al., 2021), The Scenario Manipulator tools start with a random noise vector that ranges between -1 and +1. This homogeneous vector of noise values is added to the heterogeneous JSON values to create new valid scenarios. An expert must be consulted to determine the appropriate range of values for a given scenario, in order to ensure it remains within its designated ODD.

After running each of these derived scenarios for multiple frames, the outputs of the sensors (camera, radar) that are placed in the scene are extracted. Fig. 7 shows an example of semantic segmentation of a camera image produced by the BERGE simulator.

In the next phase, each of these frames is passed to the Unicam's ML model which employs state-of-the-art ML-based LP detection algorithms. As it is part of the company secret and considered proprietary, it is not distributed in the form of a library, but runs on a server with authenticated APIs. This allows the V&V project partners to access the full functionality of the detection pipeline in real time. The model scans each image frame in its entirety and performs detection and recognition, thereby retrieving the following information: bounding box coordinates, confidence score for the bounding box coordinates, OCR detected license plate number, and the OCR confidence score.

From the bounding box coordinates, a new mask image is generated, which is used for comparison with the LP segmentation image. To evaluate the performance of the bounding box detection, the Intersection over Union (IoU) and the confidence score from the ML system are used. Fig. 8 displays the ground truth segmentation produced by BERGE simulation and the bounding box detected by the CAMEA LP detection system.

The Levenshtein distance (Levenshtein, 1966) between the predicted license plate and the correct license plate are also calculated as a relevant metric for evaluating the CAMEA OCR ML model.

In order to assess the correctness and accuracy of the LP detector system and to identify the most relevant features, a number of metrics are combined on the images produced during each simulation run:

- The number of licence plates detected during scenario execution with IoU greater than 0.75 and Levenshtein distance of 0
- The total sum of IoUs
- The total sum of Levenshtein distances
- The total sum of the bounding box detection confidences values
- The total sum of the OCR detection confidences

Conclusion

Simulation-based testing offers an exceptionally effective approach for conducting system-level tests of ML based systems and serves as a valuable complement to on-road testing. It allows for early-stage testing, facilitates the replication of critical corner test scenarios, and offers cost-effective testing options. On the other hand, field testing of these systems can be prohibitively expensive, inefficient, and even pose safety risks in certain cases. Simulators have specially proven to be valuable tools for the verification and validation of machine learning-based systems by accurately replicating real-world environments and incorporating relevant operational design domains. They enable a relevant, complete, balanced, and accurate representation of the real environment, serving as a cost-effective and efficient means of testing the reliability of machine learning-based systems.

Acknowledgment

This work was carried out within the VALU3S project funded by the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.

Part of the research has received funding from the Horizon Europe Programme under the SAFEXPLAIN Project (www.safexplain.eu), grant agreement No 101069595.

References

Ashmore, R., Calinescu, R., and Paterson, C., June 2022. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. en. *ACM Computing Surveys*, 54(5), pp. 1–39. ISSN: 0360-0300, 1557-7341. `https://doi.org/10.1145/3453444`.

BeamNG GmbH, June 15, 2022. *BeamNG.tech*. Version 0.25.0.0. Available at: <https://www.beamng.tech/>.

Bjarnefors, S., 2017. *Göteborgs-Posten*. Available at: <https://www.gp.se/nyheter/g%C3%B6teborg/%C3%A5kareplatsen-klar-i-juni-ett-halv%C3%A5r-f%C3%B6rsenad-1.4892803>.

Borg, M., Henriksson, J., Socha, K., Lennartsson, O., Sonnsjö Lönegren, E., Bui, T., Tomaszewski, P., Sathyamoorthy, S. R., Brink, S., and Helali Moghadam, M., Mar. 2023. Ergo, SMIRK is safe: a safety case for a machine learning component in a pedestrian automatic emergency brake system. *Software Quality Journal*. ISSN: 1573-1367. `https://doi.org/10.1007/s11219-022-09613-1`.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V., 2017. CARLA: An Open Urban Driving Simulator. In: *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16.

Ebadi, H., 2023. *GitHub - ebadi/sim2ml — github.com*. `https://github.com/ebadi/sim2ml`.

Ebadi, H., Moghadam, M. H., Borg, M., Gay, G., Fontes, A., and Socha, K., 2021. Efficient and Effective Generation of Test Cases for Pedestrian Detection – Search-based Software Testing of Baidu Apollo in SVL. In: *IEEE AITest 2021*.

Epic Games, Apr. 25, 2019. *Unreal Engine*. Version 4.22.1. Available at: <https://www.unrealengine.com>.

Hiblot, N., Gruyer, D., Barreiro, J.-S., and Monnier, B., 2010. Pro-SiVIC and ROADS, a software suite for sensors simulation and virtual prototyping of ADAS. In.

Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8). Doklady Akademii Nauk SSSR, V163 No4 845-848 1965, pp. 707–710.

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., Agafonov, E., Kim, T. H., Sterner, E., Ushiroda, K., Reyes, M., Zelenkovsky, D., and Kim, S., May 2020. SVL Simulator: A High Fidelity Simulator for Autonomous Driving. *arXiv e-prints*, arXiv:2005.03778, arXiv:2005.03778. arXiv: `2005.03778 [cs.RO]`.

VALU3S, 2020. *VALU3S, Verification and Validation of Automated Systems, Safety and Security*. `https://valu3s.eu/`.

VALU3S, 2022. *Scenario based V&V automation using simulator*. en. Available at: <https://repo.valu3s.eu/method/scenario-based-v-v-automation-using-simulator>.