Semantic Robustness Testing for Vision-Based Machine Learning Components of Autonomous Cyber-Physical Systems

Attila Ficsor

Abstract: Autonomous cyber-physical systems often utilize vision-based machine learning components. They are frequently part of a safety critical system, requiring special attention during testing. However, testing these systems is an incredibly challenging task, as they need to interact with an immensely complex and continuously changing environment. This makes systematic testing and other safety engineering best practices unfeasible. While some approaches aim to test vision-based machine learning components, these cannot guarantee robustness. In this paper I present a research roadmap of addressing this challenge by (1) proposing a semantic-based robustness testing suite generation approach, (2) determining the minimum level of detail necessary for testing in a simulator, and (3) finding the aspects of the simulation affecting the results. I illustrate my proposed approach on an industrial case study.

Keywords: Computer vision, Automated test generation, Robustness testing

1 Problem and motivation

Testing of computer-vision solutions. Computer vision applications utilizing deep learning algorithms and other artificial intelligence (AI) solutions are widely used in many domains, including critical systems, such as autonomous vehicles and railway systems. However, the reliability and accuracy of AI-based solutions are significantly lower than expected: while SIL 4 level require at most $10^{-8} - 10^{-9}$ probability of failure in one hour (IEC 61508), current vision-based AI applications have at most 99.8% accuracy for a single image in simplified environments[1]. Moreover, the currently used safety engineering best practices and test generation approaches are not applicable to AI solutions directly. This necessitates the development of new AI verification techniques.

Currently, the most widely accepted AI verification technique is testing. One way to test the system is real-world testing. However, this requires a dataset so large, that is impractical or unfeasible in real-world applications. For example, as reported in [1], testing self driving vehicles requires billions of driven kilometers with 100 test cars, just to prove the system's safety in a probabilistic way (compared to human drivers). To make the testing process more scalable, the test cases must be executed in a simulation environment. There are a number of photo-realistic simulators with relatively good performance.

The goal of *metamorphic testing* approaches is to provide valid mutation transformations between input/expected output pairs [2, 3]. In general, those testing techniques aim to construct various **scenes** with the same **logic situation**. Therefore, multiple test cases can be derived from a single scene by applying transformation operations. Thus, the behavior of the AI can be evaluated with the original and mutated scenes to compare their results: if the outcome is different, then at least one issue is detected. It is important to note that metamorphic testing approaches can detect issues *without test oracles or formalized requirements* in this way. If the approach aims to optimize to get the most challenging mutants to a given AI, then it is called *adversarial testing*. There are multiple levels of metamorphic testing in the literature [2]:

1. First, several approaches insert visual glitches (e.g., pixel errors or noise) as mutation.

[1] `https://paperswithcode.com/sota/visual-question-answering-on-clevr`

2. Next, several approaches (like [4, 5]) aim to add filters to images or video streams that are semantically relevant to the given domain, e.g., filters imitating different weather conditions (fog or rain), or lighting effects (lens flare) if the AI aims to operate outdoors.
3. More advanced approaches use domain-specific modifications on images, like recoloring objects or inserting pictures to valid positions (e.g. images of planes to the sky).
4. And finally, in a simulation-based environment, the scene can be manipulated directly by moving objects, changing the type of actors, or adding extra objects to the background. A simple implementation of this principle on two actors is implemented in [6].

Semantic robustness testing. My proposed line of research aims to provide a testing suite for the robustness testing of computer vision-based AI solutions. To test a computer vision AI, we need realistic images depicting diverse situations. Without a large volume of such images, we cannot guarantee that the output of the AI is correct in all situations. On one hand, adversarial machine learning can find inputs that cause malfunction in a machine learning model, but these cannot guarantee coverage of situations. On the other hand, systematic approaches can guarantee coverage by some metric, but are lacking the ability to synthesize realistic images.

Adversarial testing usually involves modifying the pixels of an image directly, resulting in new images, which are often indistinguishable from the original for humans. In contrast, a semantic-based robustness testing suite could provide visually different images, which depict semantically equivalent situation. Using a simulator, the properties of objects can be changed (e.g. texture, 3D mesh resolution), static objects or dynamic agents can be moved around, or even new objects can be placed in various locations, without changing the situation with respect to the requirements of the AI.

Semantic robustness of vision-based AIs can be defined as follows. *If the AI is given two inputs, which are semantically equivalent with regards to the requirements, it provides the same output.*

While semantic adversarial attacks have been attempted by a few approaches, these only focus on a small number of variables, and lack the guarantees provided by systematic approaches. [7] tests object detection by simply varying the camera position whereas [8] tests complete autonomous driving by changing only three parameters, two of which relate to the camera placement. A vision-based AI has to operate with a lot more parameters than we can currently accurately and exhaustively simulate, which brings us to the main research question, I plan to investigate.

Research Question *How to evaluate the semantic robustness of vision-based machine learning components of autonomous cyber-physical systems?*

Challenges in realistic simulation. Achieving complete photorealism in a simulator is practically impossible, so we have to settle for a solution that yields results comparable to the real thing. The simulation has a number of variables with multiple levels of detail, such as the poligon-mesh resolution, or the texture of the models. Determining a level of detail necessary for testing a computer vision solution is not straightforward, and may vary on a case-by-case.

Research Challenge 1. *How to determine the minimum level of detail necessary for testing a computer vision solution?*

Changing the quality of the object we want the computer vision to recognize should affect the performance of the AI. However, changing seemingly unrelated objects around the subject or even the weather conditions may also have a negative effect, as it might see things that are not there, or it might have learned to rely on environmental cues to recognize an object. Removing these cues could unpredictably confuse the AI.

Research Challenge 2. *What aspects of the simulation affect the performance of a computer vision solution?*

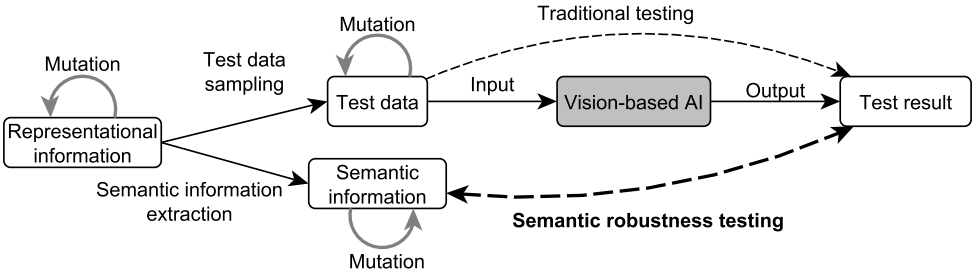

Figure 1: High level conceptual components of the proposed solution

2 Proposed solution

In order to provide a testing workflow (see Figure 1) for computer vision solutions, I propose an approach utilizing a simulator with realistic environment that can accurately represent a real world location. This proposed solution is suitable for testing vision-based AI components. A simple example for this is a number-plate recognition software.

Testing an AI solution requires *Test data* as the input, which is an image for computer vision application. When the AI provides an output, we need to compare the actual result to the expected result. Applying a traditional testing approach would require a way to extract a ground truth from the images to provide the expected result, but without additional information this is impossible. Using this method, we also have no way to describe the input.

Adversarial testing mostly focuses mutating the test data, by introducing pixel-level perturbations without any regard to the semantic meaning. This approach can identify behavioural boundaries by checking checking changes in the output (without knowing the correct output).

In contrast, I introduce a systematic solution for synthesizing the test data, with additional information provided in steps prior to creating an actual image. This way we can identify behavioural boundaries by mutating either the semantic information, or the representational information. The process starts with creating semantic information, which describes scenarios using domain-specific models. This allows us to use an automatic model generator [9].

> **Example:** In our running example of number plate recognition, the semantic information is the positional relation of the camera, the vehicle with the number-plate and objects around the vehicle, and the number plate placed on the vehicle. This step is called *Environment configuration* in Figure 2.

The next step is creating *Representational information* in the form of a three dimensional simulation environment. This is built on a base environment which we need to create beforehand. Here we can choose the level of detail to include in the model, such as the texture or the poligon-mesh of the models. Semantic information is also extracted from the situations.

The third extra step is *Test data sampling*, which includes running the simulation with the defined 3D environment, and extracting images using virtual cameras.

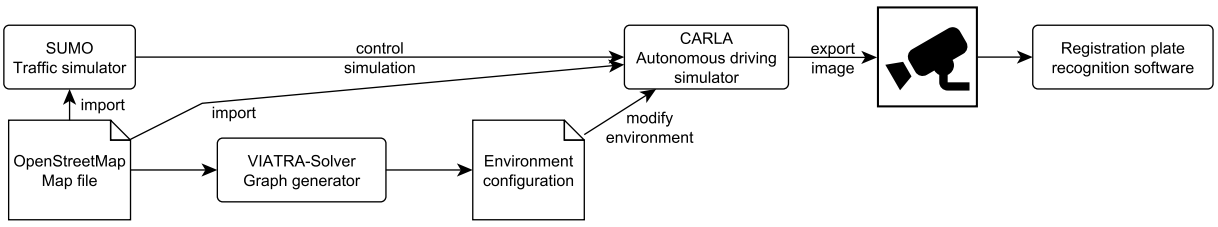

Figure 2: Running example illustrated

(a) Buildings from Google Earth

(b) Buildings from OpenStreetMap

(c) Sunny weather, no obstruction

(d) Rainy weather, with pedestrian

Figure 3: Simulation with various levels of detail, weather conditions and obstructions

Example: In the number-plate recognition example I created a 3D model of an existing intersection in Budapest. I chose two different levels of detail for the building, which can be seen in Figure 3a and Figure 3b. One uses the photorealistic 3D models from Google Earth, while the other, less detailed version uses 3d models based on OpenStreetMap data without any texture. I created a map in CARLA [10] using these models and the map data exported from OpenStreetMap. The semantic information is applied by placing pedestrians, trees, etc. in the field of vision of the camera, according to the relative positions defined in the generated configuration file.

For testing the number-plate recognition software, we need images of vehicles with visible number-plate. To get these, virtual cameras are placed observing the road, while vehicles are driving around (using SUMO [11]). While the positions of the camera and surrounding objects can be fixed or determined by the semantic and representational information.

With these extra steps the testing is performed not on the input data directly, but on the combination of the semantic and representational information. I plan to use this method to generate a testing suite. Then, I will measure the accuracy of vision based AIs on the generated dataset, do determine their semantic robustness.

Acknowledgements This research was partially funded by the EC and NKFIH through the Arrowhead Tools project (EU grant No. 826452, NKFIH grant 2019-2.1.3-NEMZ ECSEL-2019-00003).

References

[1] N. Kalra and S. M. Paddock, "Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?," p. 15.

[2] S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortés, "A survey on metamorphic testing," *IEEE Transactions on software engineering*, vol. 42, no. 9, pp. 805–824, 2016.

[3] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–27, 2018.

[4] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *ICSE*, pp. 303–314, 2018.

[5] S. Segura, D. Towey, Z. Q. Zhou, and T. Y. Chen, "Metamorphic testing: Testing the untestable," *IEEE Software*, vol. 37, no. 3, pp. 46–53, 2018.

[6] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand, and T. Stifter, "Testing autonomous cars for feature interaction failures using many-objective search," in *ASE*, vol. 18, pp. 143–154, Association for Computing Machinery, Inc, sep 2018.

[7] A. Hamdi and B. Ghanem, "Towards analyzing semantic robustness of deep neural networks," in *ECCV Workshops* (A. Bartoli and A. Fusiello, eds.), (Cham), pp. 22–38, 2020.

[8] A. Hamdi, M. Müller, and B. Ghanem, "Sada: Semantic adversarial diagnostic attacks for autonomous applications," in *AAAI*, 2020.

[9] O. Semeráth, A. Babikian, S. Pilarski, and D. Varró, "Viatra solver: A framework for the automated generation of consistent domain-specific models," in *ICSE*, (Montreal, Canada), ACM/IEEE, ACM/IEEE, 2019 2019.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *1st Annual Conference on Robot Learning*, vol. 78, pp. 1–16, PMLR, 13–15 Nov 2017.

[11] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *ITSC*, IEEE, 2018.