Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms, ICSE′18

M Parvez Rashid
NC State University
mrashid4@ncsu.edu

***Abstract*— Vision-based control systems are key enablers of many autonomous vehicular systems, including self-driving cars. Testing such systems is complicated by complex and multidimensional input spaces.In this poster, paper [1] has been discussed, an automated testing algorithm that builds on learnable evolutionary algorithms. These algorithms rely on machine learning or a combination of machine learning and Darwinian genetic operators to guide the generation of new solutions. This approach combines multiobjective populationbased search algorithms and decision tree classification models to find critical test scenarios.**

I. INTRODUCTION

On-road testing of autonomous cars are typically restricted to a small number of vehicles driven by professional safety drivers during specific hours on some designated roads with specific speed limits. Such testing is often expensive and time-consuming. It is further impractical to perform a full-fledged on-road vehicle-level testing after every change to self-driving software systems. To ensure safety of self-driving technologies, vehicle-level testing alone is neither enough nor practical. Therefore, it needs to be complemented by testing methods performed on computer software simulators. This paper this paper focuses on simulation-based testing of visionbased control systems. In the automotive domain, they are referred to as Advanced Driver Assistance Systems (ADAS), and are main enablers of self-driving cars. Examples of ADAS include automatic parking, night vision and collision avoidance systems. Simulation platforms for ADAS [2] allow engineers to run a much larger number of test scenarios compared to vehicle-level testing without being limited by conditions enforced during on-road testing.The main difficulty with simulation-based testing of ADAS is that the space of test input scenarios is complex and multi-dimensional. Engineers require techniques that allow them to explore complex test input spaces and to identify critical test scenarios (i.e., failurerevealing test scenarios). It is argued that for testing at the system level, search-based techniques provide effective and flexible guidance for test generation. Evolutionary algorithms work by iteratively sampling the input space, selecting the fittest scenarios (critical test scenarios in our work), and evolving the fittest using genetic search operators to generate new scenarios. The scenarios are expected to eventually move towards the fittest regions in the input space.

This paper proposes a system testing algorithm that combines evolutionary search algorithms and decision tree classification models. First, classification models guide the search-based generation of tests faster towards critical test scenarios. Second, search algorithms refine classification models so that the models can accurately characterize critical regions.

II. ADVANCED DRIVER ASSISTANCE SYSTEM (ADAS)

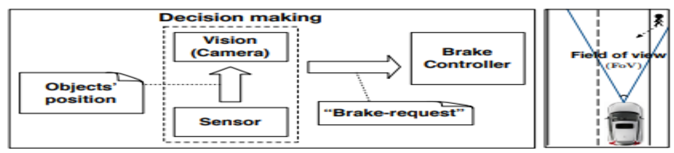

Fig. 1: An example of a vision-based control system: Automated Emergency Braking (AEB) system.

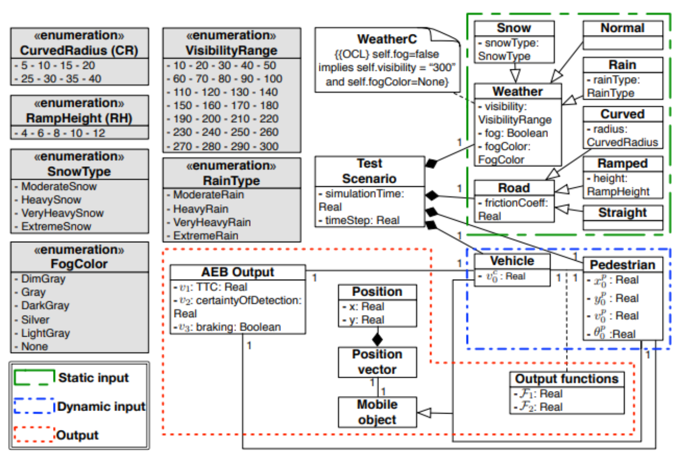

Fig. 2: The AEB domain model.

Curved road	Ramped road	Straight road
Range θ_0^P = [120..250]	Range θ_0^P = [40..160]	Range θ_0^P = [40 ..160]
Range x_0^P= [32..50]	Range x_0^P= [60..95]	Range x_0^P= [30 ..85]
Range y_0^P= [50..76]	Range y_0^P= [2..16]	Range y_0^P= [24 ..36]

Fig. 3: The ranges of the pedestrian position (x_0^P) and orientation $(\theta_0^P)$for different road topologies.

III. ADAS FORMALIZATION

$$CB(U, V) = (F_1(u_1, u_2) < 50cm) \wedge (v_2 > 0.5) \wedge (F_2(u_1, u_2) > 30km/h) \quad (1)$$

IV. SEARCH GUIDED BY CLASSIFIERS

ADAS critical test scenarios as a multi-objective search optimization problem where the ADAS outputs specifying its critical behaviors act as the search fitness functions. This paper use the Non-dominated Sorting Genetic Algorithm version 2 (NSGAII)[3]. The NSGAII algorithm generates a set of solutions forming a Pareto nondominated front

Fitness Functions: maximize F_2 and v_2, and minimize F_1

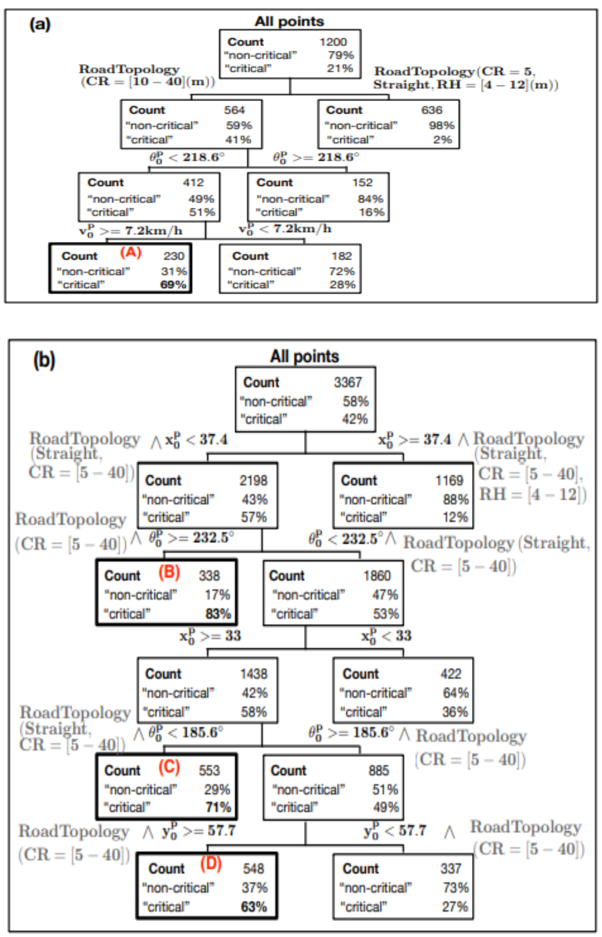

Fig. 4: Decision trees generated for the AEB system: (a) An initial decision tree, and (b) A decision tree obtained after some iterations of the NSGAII-DT algorithm.

V. CONCLUSIONS

A simulation-based testing algorithm is proposed for vision-based control systems such as ADAS. This algorithm builds on learnable evolution models and uses classification decision trees to guide the generation of new test scenarios within complex and multidimensional input spaces. The results indicate that our classification-guided search algorithm outperforms a baseline evolutionary search algorithm and generates 78% more distinct, critical test scenarios compared to the baseline algorithm.

Algorithm 1: NSGAII-DT

Input: - (S, O, I, D, C): An ADAS specification
- $F_1, \ldots, F_l$: Search fitness functions
- *label*: A boolean function to label scenarios as critical/non-critical
- g: Number of search iterations to be applied at each critical leaf

Result: - *criticalScenarios*: A set of critical test scenarios
- $R_1, \ldots, R_k (\subseteq D_1 \times \ldots \times D_n \times D'_1 \times \ldots \times D'_m)$: A set of critical regions

begin
  Select an initial population set P randomly.
  /*Each $p \in P$ is a vector of values for $(s_1, \ldots, s_n, i_1, \ldots, i_m)$ */
  $k \leftarrow 1$; $Best \leftarrow \emptyset$
  $R_1 \leftarrow D_1 \times \ldots \times D_n \times D'_1 \times \ldots \times D'_m$
  /*R_1 is the entire search space and includes all elements in P*/
  repeat
    for $i = 1$ **to** k **do**
      $Q \leftarrow P \cap R_i$
      $B, Q' \leftarrow$ NSGAII$(g, Q, F_1, \ldots, F_l, R_i, C)$
      /*Inputs passed to NSGAII:
      g: the number of search iterations applied to each critical leaf;
      Q: the set of scenarios used as the initial population of NSGAII;
      $F_1, \ldots, F_l$: search fitness functions;
      R_i: the critical leaf in which we want to run NSGAII; and
      C: the ADAS constraints.
      Outputs received from NSGAII:
      Q': all the solutions generated during search; and
      B: best solutions generated by NSGAII.*/
      $P \leftarrow P \cup Q'$
      $Best \leftarrow B \cup Best$
    $rank_1, \ldots, rank_r \leftarrow ComputeRanks(Best)$
    $criticalScenarios \leftarrow rank_1$
    $(P^+, P^-) \leftarrow ComputeLabel(label, P)$ /*P^+ : *non-critical*, P^- : *critical**/
    Build a decision tree *Tree* based on (P^+, P^-)
    Let $R_1, \ldots, R_k$ characterize the leaves of *Tree* where P^- has a higher probability than P^+
    /* For each region $R_i = d_1 \times \ldots \times d_n \times d'_1 \times \ldots \times d'_m$, we have: $\forall j \in \{1, \ldots, n\} \Rightarrow d_j \subseteq D_j$, and $\forall j \in \{1, \ldots, m\}$, $\exists min \in D'_j, \exists max \in D'_j$ s.t. $min < max \wedge d'_j = [min..max]$*/
  until *search time has run out*

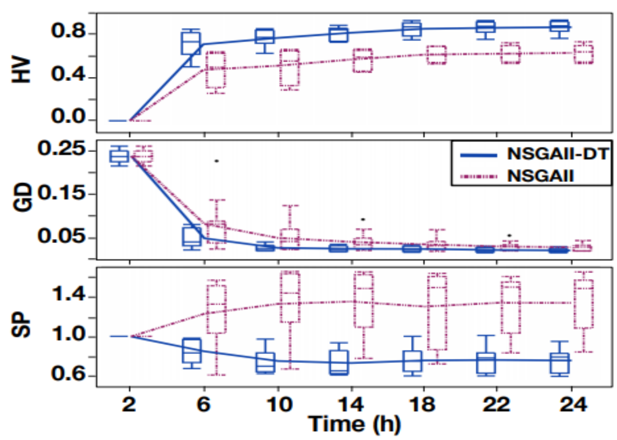

Fig. 5: Comparing HV, GD and SP values obtained by NSGAII and NSGAII-DT.

REFERENCES

[1] Raja Ben Abdessalem , Shiva Nejati , Lionel C. Briand , Thomas Stifter."Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms." Proceedings of 40th International Conference on Software Engineering (ICSE), 2018.

[2] TASS International. 2017. PreScan simulation of ADAS and active safety. (Aug. 2017). Retrieved August 24, 2017 from https://www.tassinternational.com/prescan

[3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 2 (2002), 182197

[4] Sean Luke. 2013. Essentials of Metaheuristics (second ed.). Lulu, Fairfax, Virginie, USA. Available for free at http://cs.gmu.edu/sean/book/metaheuristics/.