Passau University

Master Thesis

AUTOMATIC TEST CASE GENERATION TO EXPLORE PARAMETERS OF CRASH SCENARIOS IN SELF DRIVING CARS

Author:
Name Sameed Tariq

Supervisor:
Prof. Dr.-Ing. Gordon Fraser

Advisor:
Alessio Gambi, Ph.D.

Friday 12th April, 2019

Abstract

The automotive industry is investing billion of dollars in self-driving cars that enable them to perceive the environment and drive the million of miles with range of driving scenarios. The promise of autonomous vehicles is to ensure the safety of roads, pedestrian and reduce the traffic congestion on roads. However, no rules are established yet for the certifying of self-driving car but the Governments are playing supportive role for the encouragement of autonomous car technologies. The intelligent system (car controlling software) must be able to handle the critical scenarios and foresee the environmental cars to avoid any misadventure before the position the cars in the public roads. Virtualization of the test cases in the simulated city (crash scenario) have been introduced to test the car in infinite number of scenarios to unveil the safety bugs. The search based testing techniques to explore parameters for finite cases shows promising result. The intention of the research is generate test cases for simulation of the car crash from Police Report. The direction of the work is to explore meaningful parameter in input scenarios and observed the behaviour of the car to restrict the test cases of crash scenario to interesting and relevant settings.

1 Introduction and Motivation

Self-Driving Cars are becoming one of the most profound technologies and the Auto Makers compete for the dominance of the technology[1]. Autonomous cars are becoming more intelligent and expected to replace the human drivers and increase the safety of roads by avoiding traffic accidents. The sales of autonomous cars is expected to reach 90$ Billion in 2030[2][3].There are several ongoing projects within this domain (e.g. Waymo [4]), most of them are advance enough to deploy on roads and under testing in everyday life. However, the development of fatal crashes in autonomous cars raises questions about the testing of controlling software and decision making in critical scenarios [5]. The testing of autonomous cars on public road is necessary for the development and enhancement of self driving technology because autonomous cars does not know how to behave with the unfamiliar situation or scenarios. Recent development in the autonomy of machines, Data, Machine Learning and Sensor Fusion technologies to imitate a human behavior in real life situations. Testing of the autonomous cars on real road for number of days to find the flaws is time consuming and tremendously challenging task.

The traditional way of testing the self-driving cars for naturalistic driving scenarios is Field Operational Tests (FOT). The downside of FOT that it won't be able to cover all the interesting scenarios and need to cover million of miles to test erroneous behavior and damage (failure) because of collision with car, pedestrian etc [6]. The relevant instance of autonomous car crashes are Waymo, Uber and Tesla Car crash. The Uber self-driving car [7] [8] failed to recognize an object such as women which caused it to ignore a pedestrian walking and classified as "false positives," that wouldn't be an issue for the vehicle, like a plastic bag or piece of paper. On the other side, Google self-driving car [9] hit the bus while changing the lane on the series of assumption of Google's program. The car anticipated that the bus will allow slow down and allow the car to cut in. Hence, it shows that the self-driving car are far away from handling the critical scenarios and sustainable testing system is required for the testing of autonomous cars to prevent the accidents and enhance the safety of roads and society.

Another possibility of testing autonomous cars is the virtual testing, where the car controlling software [10] is simulated with different testing parameters and environmental conditions [11][12]. The distinct input in the self-driving cars with realistic scenarios in virtual testing will reduce the cost of physical testing. Indeed, the virtual testing can't replace the real world testing but it narrow the gap in the testing of autonomous cars [13].

The NHTSA (National Highway Traffic Safety Administration) [14] formulated to save lives, increase safety, and reduce the autonomous car crashes through research and enforcement activity. The administration of NHTSA is enforcing the safety equipment in motor vehicles and emerging Automated Driving System (ADS). However, the Automotive industry does not established any rules and regulation to certify autonomous cars. Therefore, NHTSA has a formed a regulative body for the evaluation of new technologies and test the self-driving cars in critical scenarios before the deployment on public roads.

In this thesis, the motive is to extract the data from semi-structured crash reports as input. The system will apply Natural language Processing (NLP) and Ontology concepts to extract the key features about the vehicle, environment, car dynamics and generate the configuration file. The configuration file will be used to generate test cases to reproduce the original simulation with different parameters. The accuracy of the original and derived simulation from BeamNG[15] will be evaluated from the Canny Edge detection techniques. The Structured Similarity Index Matching (SSIM) and Mean Square Error (mse) will be computed on the each frame after the impact/accident of car. After the series of simulation, the Decision Tree will be generated to select the best Test case that reflect the original simulation.

1.1 Problem Statement

Our aim is to create the simulation of the autonomous vehicle to vehicle crash as close to the description of the accident in the police crash report. The simulation is intended to reduce the simulation time, testing time of the user and produce naturalistic driving scenarios. The hyper-parameters of the simulation will be explored with the help of exploration and exploitation techniques to generate test cases and evaluate the self-driving car situation that reflects the original driving scenario describe in the crash report and improve the autonomy of self-driving cars.

2 Background and State of Art

Lot of work is being done to enhance the automation of self driving cars. Here i will describe the related work to specific areas that focus on the goal of my thesis.

2.1 Natural Language Processing and Ontology

Part of Speech (POS Tag)[16] is a technique relies on Natural Language Processing [17] to extract the key features [18] such as Noun Phrase, Verb Phrase to understand the context and meaning of the sentence. Guarino et al [19] illustrate ontology's as "explicit specification of a shared conceptualization." The ontology's [20] consist of names, definition, properties and inter-relationship of the particular domain like automotive application and weather application. Armand et al [21] depict that how ontology's can be applied to Advanced Driving Assistance System (ADAS) to model the spatial-temporal relationship between the cars. The classified entities in ontology's to understand map, vehicle (Mobile Entity), track (Static Entity) and dynamic state (Context Parameter) of the ego vehicle. Hummel el al[22] enlightened the description logic to find the intersection and geometry of roads. Similar approaches to scene understanding for ADAS by Zhao et al [23] to obtain information about traffic, vehicle and and street map.

2.2 Open Street Map (OSM)

Godoy et al[24] come up with procedure to identify the road intersections and junctions in OpenStreetMap [25]. The path were extracted from the Open Street Map based on the key attribute such as road type, surrounding area and scenic beauty. Guillaume et al [26] surveyed the challenges in Simultaneous Localization And Mapping (SLAM) for the localization of driving cars in Enhanced Maps to resist the environment variability (weather, season). The paper provide an overview of the challenges in large scale experiments of autonomous cars.

2.3 Automatic Test Case Generation

In Erbsmehl [27], the author focus on the simulation of advanced safety systems on real world crashes based on German In-Depth Accident Study database. The comparative and analysis study is done on simulated real world accident scenario and a predicted accident scenario on virtual prototype of safety system to estimate the benefit of safety system in vehicles. The GIDAS pre crash matrix contains all the relevant information such as the geometrical information of the accident (driving lane of each participants, lane borders, lane markers, view obstacles) and the dynamic behaviour of the participants. CarSim with Matlab is used for the simulation each participant and vehicle dynamic with additional car parameter, physical parameters is calculated at every 0.001 seconds. Squillant et al[28] focus on the modeling of accident scenarios with missing data in the databases and the design of safety systems related to the accident. Zhou et al[29], the author proposes a Gaussian Process Classification algorithm to efficiently determine the non-convex boundaries based on the limited number of simulations. The modelling of the safety boundary search consist of a test case ”yaw” with 3 dimensional inputs such as longitudinal inter-vehicle distance,the longitudinal velocity difference between ego vehicle and cut-in vehicle, cut-in angle yaw of the cut-in vehicle and and binary outputs (Collision or not Collision).

Arrieta et al[30] studied the traditional testing techniques, Model based testing are not feasible to apply to Cyber Physical Systems because it is not feasible to evaluate all the possible state and values, So the simulation models with search-based approach are developing to execute large number of reactive test cases with prioritization to effectively execute test cases with critical scenarios and replication of safety critical functions. The fitness function with four objectives functions are analyzed with crossover and mutation operator incorporating five multi-objective search algorithms is evaluated in case studies to find the optimal solution. Furthermore, the modules of 3D simulation to demonstrate the Scene Graph Model and validate the trajectory and autonomous component. The number of traffic scenarios can be evaluated by model parameterization to increase the coverage of testing space[31].

2.4 Edge Detection

Xiaohui et al[32] present an approach to find the moving object in the frame and detect the object with reference to edge detection and frame difference algorithm. It split the image into small parts and find the disparity between each block to speed up the process of edge detection. Li et al[33] extract keyframe from videos and present the summary based on the content of the video and eliminate redundant frames. The comparison of the extracted images is measured with Structural Similarity Index Measure whether there is a match for image processing of content retrieval[34].

2.5 Decision Tree

The series of simulation output can be visualized and explicitly shown on Decision Tree for decision analysis[35]. The decisions tress express the next parameter input in the complex multidimensional input parameter space. The test cases generated with complex scenario is used to categorize the critical scenario in Advanced Driving Assistance System (ADAS). It will further help in reducing the parameter space and choose the best parameter to identify parameter and environmental conditions.

3 Proposed Method

The procedure of automated test cases generation to explore the parameters of the car accident scenario is summarized in the Figure 1.

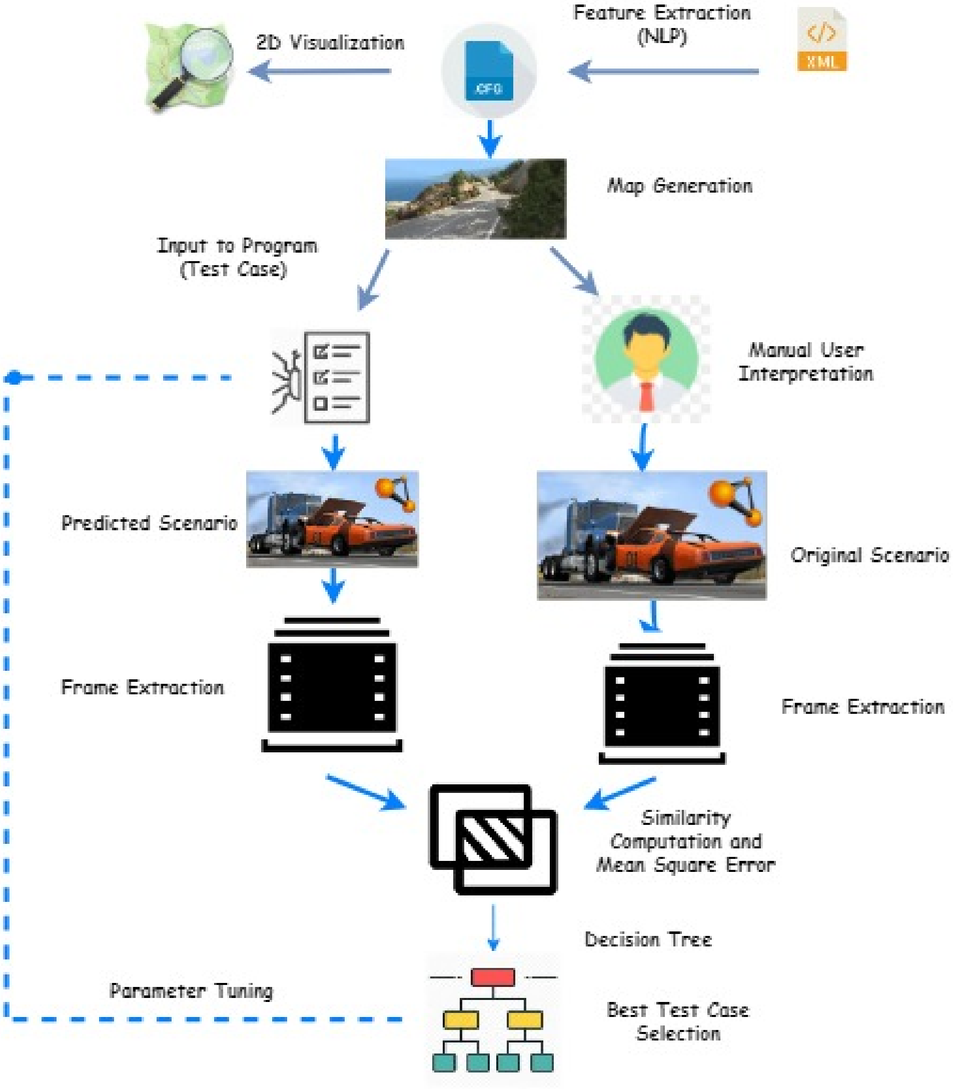

Figure 1: Architecture Diagram

3.1 Natural Language Processing

In the growing world of internet and online services, the data is available in gigabytes in open source platforms and communicated in the form of text. The extracted text is available in the form of tabular data, unstructured data, and raw data in XML. Our target is to apply the Natural Processing algorithms to understand the human language that what are they saying about the environment and accident of autonomous cars.

3.1.1 Text Analysis and Features Extraction

To generate the trajectory based on the car crash report, the following features need to be extracted using Natural language processing Algorithms.

- Automotive Type
- Weather conditions
- Road conditions
- Environmental conditions
- Positions of the car relative to each other
- Accident Type
- Car behavior (Sequences)

All these features extraction is based on the type of information available in the semi-structured data.

3.1.2 Natural Language Processing Algorithm

These individual steps are required to convert text into the machine-readable format for further processing. The Natural language toolkit or Stanford language parse can be used to execute all the steps.

Tokenization Paragraph The initial step of analyzing the text content is to break down the paragraph into smaller sentences. Later, the sentences are break down into words to identify keywords, phrases, and symbols.

Part of Speech Tagging Part of Speech tagging is useful for identifying the grammatical group of a given keyword based on the contextual information. The word tags can be a Noun, Pronoun, Adjective, Verb, Adverb based on the preceding word and avoid disambiguation.

Chunking (Shallow Parsing Part of Speech tagging does not give enough information to retrieve the information about the sentence and waste time to parse the tree of the sentence. A shallow parsing is a technique that gives us a bunch of words related to our application-oriented processing such as Noun Phrase and Verb Phrase to get the semantic information.

Dependency Tree Parsing In each sentence, there is a root words and all other words are directly or indirectly linked to root node. The dependency-based grammars is used to identify each word relationship with root node to get structural and semantic information about the sentence. The dependent words information help us to extract and generate the sequence of action describe in natural language. The Spacy,Penn Tree Bank and Stanford parser are the most useful libraries for dependency tree parsing.

Co-reference resolution (anaphora resolution) In natural language, it is hard to identify the Pronouns,Noun Phrases and individual token/word with simple text mining techniques that point toward the same entity in the real world. Anaphora resolution find the right word that are connected mentioned entity in the sentence.

Stops Word In sentence, there are unnecessary words/token that don't carry any meaning and does not determine and semantic relation between the words are called "Stops Word".The words like "the", "a", "on", "is", "all" are usually removed from the text to carry out further processing.

Stemming and Lemmatization The aim of the stemming and lemmatization to reduce the word to base form considering the relationship of the words with each other. The stemming and lemmatization of the word make it easier to find the root of the entity of the word like environment, vehicle and road etc. These words will be further processing in entity recognition using Ontology.

3.1.3 Ontology-based Entity Recognition

The ontology can be defined as the object that share the relationship with each other and belong to the same concept (category). Ontology is composed of Taxonomy, Relationship, Class Attributes and Rules/Axioms. As an example, putting the element "BMW" in category like animal/mineral/vehicle. The element concept help us in determining the variables available for the reconstruction of crash scenario in self driving cars.

The other way of doing it is using the Named Entity recognition of Stanford parser but it is not applicable to our case because the domain of knowledge of Stanford parser is restricted to only 7 variables such as TIME, DATE, ORGANIZATION etc.

All the information extracted is stored in the XML/JSON file, it will be further processed to create scenarios in Open Street Map and BeamNG.

3.2 Open Street Map (OSM)

The Open Street Map (OSM) is an open source platform to extract the data about roads and nearby characteristic such as building in the given region of interest. Open Street Map allow users to edit map and use data free of cost. Our objective is to extract the data from the open street map according to the description given in the natural language processing. The roads can highways, urban road, rural road, junctions. However, the location is not specified in the description so we will restrict our region of interest to one city and nearby highway lanes. The car trajectory with different colors for each vehicle will be shown on the Open Street Map for visualizations.

3.2.1 Data Extraction from Open Street Map

The Open Street Map[25] data can be extracted from Xapi Api. After obtaining the response from API request, the XML object is parsed to extract the roads and link roads according to the tagged provided in the smart query. The road can be filtered from large data-set and small visualization can be drawn on the Open Street Map.

3.3 Generation Test Cases Scenarios in BeamNG

The BeamNG is open source tool for studying and developing vehicle accidents of self-driving cars with an integrated option of Artificial Intelligence. The BeamNG provide the realistic figures of car dynamics, car damage, environment and roads etc. It provides us the features to extract the data from the simulation, record videos for analysis and provide mechanism to set up speed, start, stop, steering angle, direction of car in the scenarios. Additionally, the BeamNG (3D) will be used to simulate the scenario demonstrated in Open Street Map (2D) with an access to unlimited research version of BeamNG.

3.3.1 Environment Reconstruction

The simulation of self driving cars requires to create similar environment such as cars, roads and obstacles in BeamNG research using Python. The Configuration file in Json/XML format is generated from the natural language processing will used as an input to generate test scenarios. Furthermore, the Grid Map in BeamNG will be used to plan the trajectory of the self-driving cars/ vehicles that encloses traffic dynamics, road network, surround environment such as trees, pedestrian etc.

As the accurate data is not available for the simulation of self-driving cars, we assume the best trajectory of the vehicles and pass it through the Way-points on the reconstructed scenario. It is not necessary that the assumption actually reflect the original scenario describe in Vehicle Crash report.

3.3.2 Parameter Exploration and Exploitation (Automating Test Cases)

As mentioned, our objective is to explore the parameters for self-driving cars by executing the number of test cases that reflect the original scenario.Since, the region space is quite large and contained the large number of values.Hence, it will increase the simulation time of the test case of the given scenario until we find an optimal solution. There are two ways to explore the parameters.
Parameter Exploration The visiting of those regions in the parameter space that is not explored or visited yet.
Parameter Exploitation The visiting of those regions in the parameter space that provides the high probability of success.

Our target is to focus on parameters like *initial position*, *velocity*, *acceleration*, *steering angle* and *the point of impact (Crash)*. The exploration of these parameter heavily relies on the Aftermath of road accident acceleration. The Aftermath analysis of the accident and accuracy of the simulation will help us to choose the next parameters for the test cases to produce simulation nearest to the ground truth.

3.3.3 Modification of Parameters in Test Cases

The modification of Test Case parameter depend upon the search space and the accuracy of the reconstructed simulation of the test case. We can restrict the regions of the search space based on the feedback from the simulation and reduce the time of simulation and computation cost. Since, our evaluation is dependent upon the matching of frames of each simulation and choosing the best test case of the simulation from the Decision tree/ Binary Search Tree. For Example,In the initial scenario the ego car crash with the leading car with a speed/velocity of 20m/s and stays in the exact same position. However we have to explore the velocity at which the ego car crash with the leading car and go away from the impact point in the given direction. It means that we have to collide car with more momentum and speed to reproduce the desired scenario.

3.3.4 Frame Extraction and Video Analysis (Computer Vision)

For video analysis of Aftermath of car crash to calculate the accuracy of the simulation between the original simulation and predicted simulation. The Frames will be extracted from the BeamNG video and the comparison of each frame will be done by using Structural Similarity Index Matching (SSIM) algorithm in OpenCV[36]. The computation cost can be high as each pixel is going to compared in each frame and Mean Square Error (mse). The computation cost can be decreased by extracting the Region of Interest (ROI).
In our case, the region of interest is to detect the edges of the object such as cars present on a picture using the Canny edge detection algorithm. The algorithm does not work on the colored images. Hence, I have to convert the frames to gray-scale pictures and apply the next step of image processing to get the edges and Structural Similarity Index Matching. The Canny edge algorithm consist of 5 different steps such as Noise reduction, Gradient calculation, Non-maximum suppression, Double threshold and Edge Tracking.

4 Planned Evaluation

The primary focus of my research is to focus on the parameters of the car crash and analyze the behaviours of the self-driving car after the car crash. Hence, i have to run a lot of simulations and calculate the Structural Similarity Index Matching (SSIM) and Mean Square Error (mse) at each simulation. After gathering the data on each simulation, the Decision Tree will be constructed for the classification of results and uses tree-like model of decisions. In decision analysis, it has a good advantage because i can easily demonstrate my result to a non technical audience and produce good results with incomplete data. The trajectory result of each simulation will be constructed on the Decision Tree to choose the best possible Test case.

The Decision Tree variables are

- SSIM
- MSE
- Velocity
- Steering Angle

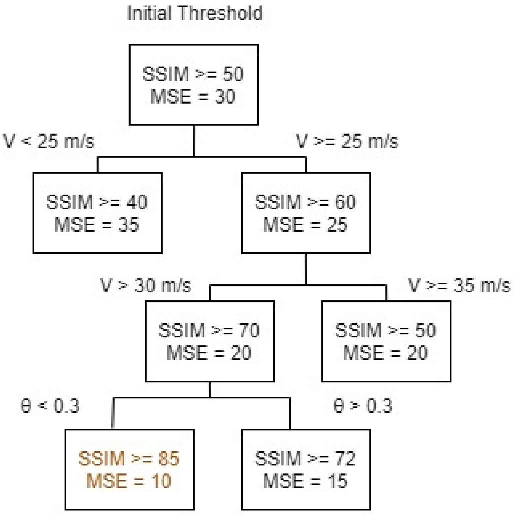

Figure 2: Decision Tree

5 Schedule

I would like to start my work on thesis in September 2019 and follow up-to next 6 months. The milestone of each task and time duration is summarized in given schedule.

Start Date : 1st September 2019
End Date : 1st March 2020

The schedule of my 6 months is divided into 24 weeks and given in the Table 1.

Table 1: Thesis Schedule

Weeks	Task
4	Understanding NHTSA Accident Summary and Generating configuration file
4	Open Street Map and BeamNG Simulation
4	Generation Test Case for Parameter Exploration
4	Calculating Similarity and Efficiency of Test Cases
2	Decision Tree Classification and result evaluation
6	Thesis Writing

6 Success criteria

The research work of my master thesis should contain the following features to evaluate the end results and completion of all these features to be considered as completed or successful.

Table 2: Summary of the Expected Thesis Features

Feature	Must-Have	May-Have	Must-Not Have
Crash Report Analysis (NLP)	✓	-	-
Configuration File for BeamNG	✓	-	-
Open Street Map visualization	✓	-	-
Road Geometry	✓	-	-
Original BeamNG Scenario of Crash Report	✓	-	-
Predicted BeamNG Simulation using Test Cases	✓	-	-
Parameters of Test Cases for Simulation	✓	-	-
Key Frame Extraction and Video Analysis	✓	-	-
Structural Similarity Index Matching (SSIM)	✓	-	-
Mean Square Error (MSE)	-	✓	-
Decision Tree Classifier (DT)	✓	-	-
Minimum 5 Crash Report	✓	-	-
Artificial Intelligence support for BeamNG	-	✓	-

References

[1] PA Hancock, Illah Nourbakhsh, and Jack Stewart. On the future of transportation in an era of automated and autonomous vehicles. *Proceedings of the National Academy of Sciences*, page 201805770, 2019.

[2] Michele Bertoncello and Dominik Wee. Ten ways autonomous driving could redefine the automotive world. *McKinsey & Company*, 6, 2015.

[3] Tasha Keeney. Mobility-as-a-service: Why self-driving cars could change everything. *Ark Invest*, 1(3), 2017.

[4] Eric R Teoh and David G Kidd. Rage against the machine? google's self-driving cars versus human drivers. *Journal of safety research*, 63:57–60, 2017.

[5] Jinwei Zhou and Luigi del Re. Identification of critical cases of adas safety by fot based parameterization of a catalogue. In *2017 11th Asian Control Conference (ASCC)*, pages 453–458. IEEE, 2017.

[6] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[7] Sam Levin. Uber crash shows' catastrophic failure'of self-driving technology, experts say. *The Guardian*, 22, 2018.

[8] Puneet Kohli and Anjali Chadha. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash. In *Future of Information and Communication Conference*, pages 261–279. Springer, 2019.

[9] H Muir. Google self-driving car caught on video colliding with bus. *The Guardian https://www. theguardian. com/technology/2016/mar/09/google-self-driving-car-crash-video-acci dent-bus*, 2016.

[10] Chi Zhang, Yuehu Liu, Danchen Zhao, and Yuanqi Su. Roadview: A traffic scene simulator for autonomous vehicle simulation testing. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1160–1165. IEEE, 2014.

[11] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, pages 303–314. ACM, 2018.

[12] Raja Ben Abdessalem, Shiva Nejati, Lionel C Briand, and Thomas Stifter. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pages 63–74.

[13] Christoph Sippl, Florian Bock, David Wittmann, Harald Altinger, and Reinhard German. From simulation data to test cases for fully automated driving and adas. In *IFIP International Conference on Testing Software and Systems*, pages 191–206. Springer, 2016.

[14] Adam Stocker and Susan Shaheen. Shared automated vehicle (sav) pilots and automated vehicle policy in the us: Current and future developments. In *Road Vehicle Automation 5*, pages 131–147. Springer, 2019.

[15] BeamNG GmbH. BeamNG.research.

[16] Lluís Màrquez and Horacio Rodríguez. Part-of-speech tagging using decision trees. In *European Conference on Machine Learning*, pages 25–36. Springer, 1998.

[17] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[18] Ruslan Mitkov, Richard Evans, Constantin Orăsan, Viktor Pekar, et al. Anaphora resolution: To what extent does it help nlp applications? In *Discourse Anaphora and Anaphor Resolution Colloquium*, pages 179–190. Springer, 2007.

[19] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[20] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[21] Alexandre Armand, David Filliat, and Javier Ibañez-Guzman. Ontology-based context awareness for driving assistance systems. In *2014 IEEE intelligent vehicles symposium proceedings*, pages 227–233. IEEE, 2014.

[22] Britta Hummel, Werner Thiemann, and Irina Lulcheva. Scene understanding of urban road intersections with description logic. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fr Informatik, 2008.

[23] Lihua Zhao, Ryutaro Ichise, Tatsuya Yoshikawa, Takeshi Naito, Toshiaki Kakinami, and Yutaka Sasaki. Ontology-based decision making on uncontrolled intersections and narrow roads. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 83–88. IEEE, 2015.

[24] Jorge Godoy, Antonio Artuñedo, and Jorge Villagra. Self-generated osm-based driving corridors. *IEEE Access*, 7:20113–20125, 2019.

[25] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[26] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[27] Christian Erbsmehl. Simulation of real crashes as a method for estimating the potential benefits of advanced safety technologies. In *21st International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety AdministrationBundesministerium fuer Verkehr, Bau und Stadtentwicklung-BMVBS-(Federal Ministry of Transport, Building and Urban Affairs) Bundesanstalt fuer Strassenwesen (BASt)*, number 09-0162, 2009.

[28] Reinaldo Squillante Jr, Diolino J Santos Fo, Newton Maruyama, Fabrício Junqueira, Lucas A Moscato, Francisco Y Nakamoto, Paulo E Miyagi, and Jun Okamoto Jr. Modeling accident scenarios from databases with missing data: A probabilistic approach for safety-related systems design. *Safety science*, 104:119–134, 2018.

[29] Jinwei Zhou and Luigi del Re. Safety verification of adas by collision-free boundary searching of a parameterized catalog. In *2018 Annual American Control Conference (ACC)*, pages 4790–4795. IEEE, 2018.

[30] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. Employing multi-objective search to enhance reactive test case generation and prioritization for testing industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 14(3):1055–1066, 2018.

[31] Marc René Zofka, Sebastian Klemm, Florian Kuhnt, Thomas Schamm, and J Marius Zöllner. Testing and validating high level components for automated driving: simulation framework for traffic scenarios. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 144–150. IEEE, 2016.

[32] Chaohui Zhan, Xiaohui Duan, Shuoyu Xu, Zheng Song, and Min Luo. An improved moving object detection algorithm based on frame difference and edge detection. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 519–523. IEEE, 2007.

[33] Xuelong Li, Bin Zhao, and Xiaoqiang Lu. Key frame extraction in the summary space. *IEEE transactions on cybernetics*, 48(6):1923–1934, 2018.

[34] Prashan Premaratne and Malin Premaratne. New structural similarity measure for image comparison. In *ICIC*, 2012.

[35] Raja Ben Abdessalem, Shiva Nejati, Lionel C Briand, and Thomas Stifter. Testing vision-based control systems using learnable evolutionary algorithms. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 1016–1026. IEEE, 2018.

[36] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.