

Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 22/06

K. Kowol, S. Bracke, H. Gottschalk

A-Eye: Driving with the Eyes of AI for Corner Case Generation

February 22, 2022

http://www.imacm.uni-wuppertal.de

A-Eye: Driving with the Eyes of AI for Corner Case Generation

Kamil Kowol¹ and Stefan Bracke² and Hanno Gottschalk¹

Abstract—The overall goal of this work is to enrich training data for automated driving with so called corner cases. In road traffic, corner cases are critical, rare and unusual situations that challenge the perception by AI algorithms. For this purpose, we present the design of a test rig to generate synthetic corner cases using a human-in-the-loop approach. For the test rig, a realtime semantic segmentation network is trained and integrated into the driving simulation software CARLA in such a way that a human can drive on the network's prediction. In addition, a second person gets to see the same scene from the original CARLA output and is supposed to intervene with the help of a second control unit as soon as the semantic driver shows dangerous driving behavior. Interventions potentially indicate poor recognition of a critical scene by the segmentation network and then represents a corner case. In our experiments, we show that targeted enrichment of training data with corner cases leads to improvements in pedestrian detection in safety relevant episodes in road traffic.

I. INTRODUCTION

As digitization continues, more and more assistance systems are being developed for automated driving, using AI systems to support them. Since AI systems need to be extensively trained and validated, large amounts of data from the real or virtual world are required. To increase robustness and performance, this data should consist of a large number of clean and diverse scenes³.

In this work, we would like to present an accelerated testing strategy that uses a human-in-the-loop approach to capture corner cases to achieve performance improvement based on safety-critical scenes. In order to obtain many safety-critical corner cases in a short time, we stop training at an early stage so that the network is sufficiently well trained. Nevertheless, the scenes generated in this way are still useful to improve fully trained networks.

For this purpose, a semantic network is trained with synthetic images from the open-source driving simulation software CARLA[1]. In addition, a test rig consisting of 2 control units is connected to CARLA in such a way that the ego vehicle can be controlled with both control units by a human. In this process, the semantic segmentation network is integrated into CARLA in such a way that first the original CARLA image is sent through the network and the prediction is displayed on the screen of one driver. The second driver,

²Stefan Bracke is with Chair of Reliability Engineering and Risk Analytics, University of Wuppertal, IZMD, Gaußstraße 20, Wuppertal, Germany, {bracke}@uni-wuppertal.de in turn, sees the real CARLA image and is supposed to intervene as a safety driver only if he or she feels that a situation is being wrongly assessed by the other driver. We aim to consider situations in which the AI algorithms lead to incorrect evaluations of the scene, which we refer to as safety relevant corner cases, in order to improve performance through targeted data enrichment. This is done by exchanging images from the original dataset with the safety-critical corner cases, thus keeping the total amount of data fixed. We show that the semantic segmentation network that contains safety-critical corner cases in the training data performs better on similar critical situations than the network that does not contain any safety critical situations.

Our approach somehow follows the idea of active learning, where we get feedback on the quality of the prediction by interactively querying the scene. However, unlike in standard active learning we do not leave the query strategy to the learning algorithm, but make use of the human's fine tuned sense of risk to query safety relevant scenes from a large amount of street scenes, leading to enhanced performance in safety critical situations.

The contributions of this work can be summarized as follows:

- An experimental setup, that could also be implemented in the real world, that permits testing the safety of the AI perception separately from the full system safety including the driving policy of an automated vehicle.
- A proof of concept for the retrieval of training data for automated driving which is safety relevant.
- A proof that training on safety relevant situations is beneficial for the recognition of street hazards.

Outline: Section II discusses related work on corner cases, human-in-the-loop and accelerated testing. In Section III we briefly describe the experiment setup used for corner case generation and which data and network was used for our experiments. In Section IV we explain our strategy to generate corner cases. In Section V we demonstrate the beneficial effects of training with corner cases for safety critical situations in automated driving. Finally, we present our conclusions and give an outlook to future directions of research in Section VI.

II. RELATED WORKS

A. Corner Cases

Training data contains few, if any, critical, rare or unusual scenes, so-called corner or edge cases. In the technical field, the term corner case describes special situations that occur outside the normal operating parameters [2].

¹Kamil Kowol and Hanno Gottschalk are with School of Mathematics and Natural Sciences, University of Wuppertal, IZMD, Lise-Meitner-Straße 27-31, Wuppertal, Germany, {kowol, hgottsch}@uni-wuppertal.de

³Andrej Karpathy, keynote speaker at CVPR 2021 Workshop on Autonomous Driving, online, URL: http://cvpr2021.wad.vision/



Fig. 1. View of the semantic driver (top) and the safety driver (bottom).

Fig. 2. Test rig including steering wheels, pedals, seats and screens.

According to [3], a corner case in the field of autonomous driving describes a "non-predictable relevant object/class in relevant location". Based on this definition, a corner case detection framework was presented to calculate a corner case score based on video sequences. The authors of [4] subsequently developed a systematization of corner cases, in which they divide corner cases into different levels and according to the degree of complexity. In addition, examples were given for each corner case level. This was also the basis for a subsequent publication with additional examples [5]. Since the approach in these references is camera-based, a categorization of corner cases at sensor level was adapted in [6], where RADAR and LiDAR sensors were also considered. Furthermore, this reference presents a toolchain for data generation and processing for corner case detection.

Outside normal parameters also includes terms such as anomalies, novelties, or outliers, which, according to [6], correlate strongly with the term corner case. In road traffic, the detection of new and unknown objects, anomalies or obstacles, which must also be evaluated as 'outside the operating parameters', is essential. To measure the performance of methods for detecting such objects, the benchmark suite "SegmentMeIfYouCan" was created [7], [8]. In addition, the authors present two datasets for anomaly and obstacle segmentation to help autonomous vehicles better assess safetycritical situations.

In summary, the term "corner case" can encompass rare and unusual situations that may include anomalies, unknown objects or outliers which are outside of operating parameters. Outside the operating parameters, in the context of machine learning, means that these situations or objects were not part of the training data.

B. Human-In-The-Loop

In machine learning, human-in-the-loop approaches combine human and machine intelligence to develop synergies for solving a specific problem. Efficiency can be increased in this area because machines learn from knowledge of a human [9], [10]. This provide faster and/or more accurate results [11]. As the interest in "human-in-the-loop" and "machine learning" is increasing [12], different approaches with a driving simulator and a test rig have been used and tested in recent years, which will now be briefly summarized.

The authors of [13] propose a real-time Deep Reinforcement Learning (Hug-DRL) method based on human guidance, where a person can intervene in driving situations when the agent makes mistakes. These driving errors can be fed directly back into the agents's training procedure and improve the training performance significantly.

In [14] a realistic test rig including a steering wheel and pedals for data collection was developed. Therefore, thirteen subjects were recruited to drive on different routes while being distracted by static or dynamic objects or by answering messages on their cell phones. By adding nonlinear human behaviors and using realistic driving data, the authors have been able to predict human driving behavior more accurately in testing.

Another driving simulator was presented in [15] to develop and evaluate safety and emergency systems in the first design stage. In addition to a steering wheel, pedals and a gearshift as well as monitors are connected to a computer. As software they use a generic simulator for academic robotics which uses the Modular Open Robots Simulation Engine *MORSE* [16]. They used an experiment with four road users, one human driver and three vehicles driving in pilot mode and forcing two out of 36 collision situations (*a lead vehicle stopped* and *a vehicle changing lanes*) defined by the National Traffic Safety Administration (NHTSA). The impact of a driver assistance system on the driver was one of the factors studied.

In [17] a human-like driving and decision-making framework is introduced, where the lane-changing process is examined. Using the brain emotional learning circuit model (BELCM), a human-like driving model is designed and evaluated with human-in-the-loop experiments.

A decision-making corner case generation method for connected and automated vehicles (CAVs) for test and evaluation purposes is proposed in [18]. For this, the behavioral policy of background vehicles (BV) is learned through reinforcement learning and Markov's decision process, which leads to a more aggressive interaction with the CAV which forces more corner cases under test conditions. The tests take



Fig. 3. Two human subjects can control the ego vehicle. The semantic driver moves the vehicle in compliance with traffic rules in the virtual world and sees only the output of the semantic segmentation network. The safety driver, who sees only the original image, assumes the role of a driving instructor and intervenes in the situation by braking or changing the steering angle as soon as a hazardous situation occurs. Intervening in the current situation indicates poor situation recognition of the segmentation network and represents a corner case. Triggering a corner case ends the acquisition process and a new run can be started.

place on the highway and include lane changes or rear-end collisions.

C. Accelerated Testing

Accelerated testing strategies are intended to reduce workload and, therefore, costs [19], [20]. By imitating the real process parameters, initial findings can be obtained in field tests and, if necessary, optimizations can be made. From this, predictions could be made about product life or performance over time under more moderate conditions of use or design. It is however necessary to carefully investigate the relevance of the data acquired under more challenging conditions for the actual use case [21].

III. EXPERIMENTAL SETUP

A. Driving Simulator

Targeted enrichment of training data with safety-critical driving situations is essential to increase the performance of AI algorithms. Since the generation of corner cases in the real world is not an option for safety reasons, generation remains in the synthetic world, where specific critical driving situations can be simulated and recorded. For this purpose, the autonomous driving simulator CARLA [1] is used. It is an open-source software for data generation and/or testing of AI algorithms. It provides various sensors to describe the scenes such as camera, LiDAR and RADAR and delivers ground truth data. CARLA is based on the Unreal Engine game engine [22], which calculates and displays the behavior of various road users while taking physics into account, thus enabling realistic driving. Furthermore, the world of CARLA

can be modified and adapted to one's own use case with the help of a Python API.

For our work, we used the API to modify the script for manual control from the CARLA repository. In doing so, we added another sensor, the inference sensor, which evaluates the CARLA RGB images in real-time and outputs the neural network semantic prediction on the screen. An example is shown in Figure 1. By connecting a control unit including a steering wheel, pedals and a screen, to CARLA, we make it possible to control a vehicle with 'the eyes of the AI' in the synthetic world of CARLA. We also connected a second control unit with the same components to the simulator, so that it is possible to control the same vehicle with 2 different control units, see Figure 2. The second control unit is thus operated on the basis of CARLA clear image and can intervene at any time. It always has priority and triggers that the past 3 seconds of driving, which are buffered, are written to the dataset on the hard disk. In order for the semantic driver to follow the traffic rules in CARLA, the script had to be modified additionally. The code has been modified to display the current traffic light phase in the upper right corner and the speed in the upper center.

B. Test Rig

The test rig consists of the following components: a workstation with CPU, 3x GPU Quadro RTX 8000, 2 driving seats, 2 control units (steering wheel with pedals), one monitor for each control unit and two monitors for the control center. The driving simulator software used is the open-source software CARLA version 0.9.10.

C. Dataset for Initial Training and Testing

For training, a custom dataset was generated using CARLA 0.9.10, consisting of 85 scenes á 60 frames. In addition, there is a validation dataset with 20 scenes. The dataset was generated on seven maps with one fps and contains the corresponding semantic segmentation image in addition to the rendered synthetic image. The maps include the five standard maps in CARLA and two additional maps that offer a mix of city, highway and rural driving. Various parameters can be set in CARLA, and we focused on the number of Non-Player-Characters (NPC's), including vehicles and pedestrians, and on environment parameters such as sun position, wind and clouds. Depending on the size of the map, the number of NPC's ranged from 50 to 150. The clouds and wind parameters can be set in the range between 0 and 100, with 100 being the highest value. The wind parameter is responsible for the movement of tree limbs and passing clouds and was in the range 0 and 50. The cloud parameter describes the cloudiness, where 0 means that there are no clouds at all, and 100 that the sky is complete covered with clouds. We have chosen values between 0 and 30. The altitude describes the angle of the sun in relation to the horizon of the CARLA world, with values between -90 (midnight) and 90 (midday). Values between 20 and 90 were used for our purpose. The other environmental parameters like rain, wetness, puddles or fog are set to zero. The parameters are chosen so that the scenes reflect everyday situations with a natural scattering of NPC's and in similar good weather. During data generation, the movement of all NPC's was controlled by CARLA. Furthermore, 21 corner case scenes were used as test data, each containing 30 frames. Another test dataset containing 21 standard scenes without corner cases will serve as a comparison, each containing 30 frames.

D. Training

To drive on the predicted semantic mask, a real-time capable network architecture is needed. For these purposes the Fast Segmentation Convolutional Neural Network (Fast-SCNN) model was used [23]. It uses two branches to combine spatial details at high resolution and deep feature extraction at lower resolution with an accuracy of 68.0% mIoU at 123.5 fps on the Cityscapes dataset [24]. The network was implemented in the python package PyTorch [25] and training was done on a NVIDIA Quadro RTX

TABLE I Overview of the datasets used for the training.

no.	comment	mean _{pixels/scene}
1	natural distribution	3583.6
2	pedestrian enriched	6101.1
3	corner case enriched	6215.7

The first dataset contains scenes with a natural spread of NPC's based on daily events. For the second dataset, scenes with a higher number of pedestrians were generated to allow a fair comparison with dataset three, where the training data was enriched with corner case scenes. All datasets contain the same number of training images.

TABLE II

PERFORMANCE MEASUREMENT ON TWO TEST DATASETS.

Traindata		Safety Critical Testdata		Natural Distributed Testdata	
no.	comment	IoU _{pedestrians}	mIoU	IoU _{pedestrians}	mIoU
1	natural distribution	0.4600	0.6954	0.4937	0.761
2	pedestrian enriched	0.5399	0.6911	0.5586	0.7554
3	corner case enriched	0.5683	0.7173	0.5384	0.7517

The comparison shows that the addition of safety-critical scenes in training also improves performance in testing with safety critical scenes.

8000 graphics card. Sixteen of the 23 classes available in CARLA were used for training. Cross entropy was used as the loss function and ADAM as the optimization algorithm. A polynomial decay scheduler was also used to gradually reduce the learning rate.

We intentionally stopped the training after 5 epochs to increase the frequency of perception errors for the network. The resulting network is sufficiently well trained to recognize the road and all road users, although objects further away are poorly recognized. An example is shown in the top of Figure 1.

IV. RETRIEVAL OF CORNER CASES

For the generation of corner cases we have considered the following experimental setup. The scenes are recorded with the help of two test operators in our specially constructed test rig (see Figure 2), where one subject (safety driver) gets to see the original virtual image and the other (semantic driver) the output of the semantic segmentation network (see Figure 1). The test rig is equipped with controls such as steering wheels, pedals and car seats and connected to CARLA to simulate realistic participation in road traffic.

The corner cases were generated as shown in Figure 3. For this purpose, we use a real-time semantic segmentation network from Section III-D where visual perception is limited. We note that autonomous vehicles according to [26] 67619.81 km drive until an accident happens. Using a poorly trained network as a part of our accelerated testing strategy, we were able to generate corner cases after 3.34 km in average between interventions of the safety driver. We note however that the efficiency of the corner cases is evaluated using a fully trained network.

If the safety driver triggers the recording of a corner case the test operators label the corner case with one of four options available (overlooking a walker or a vehicle, disregarding traffic rules, intervening out of boredom) and may leave a comment. Furthermore the kilometers driven and duration of the ride are notated. The operators were told to obey the traffic rules and not to drive faster than 50 km/h during the test drives. After a certain familiarization period, driving errors decreased and sudden braking by the semantic driver was also reduced. The reason for this is that the network partially represents areas as vehicles or pedestrians with fewer pixels. Over time, a learning effect occurred in the drivers to hide such situations because experience showed that there was no object there due to the previous frames.



Fig. 4. Evaluation on corner case test data shows that the model using corner case data in training detects pedestrians better (top left) than the original dataset (bottom left) and the dataset where more pedestrians were detected (top right). Bottom right shows the groundtruth image.

The rides are being tracked and by the intervention of the safety driver the last 3 seconds of the scene are saved. Subsequently, the scenes can be loaded and images saved from the ego vehicle's perspective using the camera and the semantic segmentation sensor. We collect 50 corner cases before retraining from scratch with a mixture of original and corner case images. For each corner case, the last 3 seconds are saved at 10 fps before the intervention by the safety driver. In total, we get 1500 new frames. When using this corner case data for retraining, we delete the same number of frames from the original training dataset.

We selected 50 corner cases in connection with pedestrians. Therefore, the inclusion of corner case scenes into the training dataset significantly increases the average number of pixels with the pedestrian class in the training data. To establish a fair comparison of the efficiency of corner cases as compared to a simple upsampling of the pedestrian class, we created a third dataset that contains approximately the same number of pixels per scene for the pedestrian class as the dataset with the corner cases, see Table I.

V. EVALUATION AND RESULTS

All results in this section are averaged over 5 experiments to obtain a better statistical validity. For testing purposes, we generated 21 additional corner cases for validation. With the same setup as before, we train the Fast-SCNN for 200 epochs on all three datasets and thereby obtain three networks. Table II shows the evaluation of all three models on the class pedestrian for the 21 safety critical test corner cases. We see that adding corner cases to the training data leads to an improvement in pedestrian detection in safety critical situations, which can also be shown by an example in Figure 4. There we see a situation with a pedestrian crossing the road, with a slope directly behind him that seems to end the road at the level of the horizon. Therefore, the networks that did not have corner cases in the training data seem to have problems with this situation, while the model with corner cases detects the human much better. While training the network using naive upsampling of pedestrians does not have any positive effect on the classe's IoU as compared with the original training data, we achieve a gain in the IoU by 2.19% when using the dataset containing corner cases. In addition, the 3 models were tested on a dataset with a natural distribution of pedestrians. Here it can be seen that the model trained with corner cases does not perform as well as the model with the same number of pedestrians. It follows that the model performs better in critical situations, while the models without corner cases perform less well. We therefore demonstrated the benefits of our method to generate corner cases, especially for safety critical situations. We were also able to show that adding safety-critical corner cases improves performance, so future datasets should include such situations.

VI. CONCLUSION

This work presents an experimental setup for human driving with the eyes of AI. We have designed a test rig on which an ego-vehicle can be driven by 2 subjects in the virtual world of CARLA. The semantic driver receives the output of a semantic segmentation network in real-time, based on which she or he is supposed to navigate in the virtual world. The second driver takes the role of the driving instructor and intervenes in dangerous driving situations caused by misjudgements of the AI. We consider driver interventions by the safety driver as safety-critical corner cases which subsequently replaced part of the initial training data. We were able to show that targeted data enrichment with corner cases leads to improved pedestrian detection in critical situations.

Future research projects include the use of some networks of different quality, the world parameters such as the number of pedestrians, vehicles, but also the weather will be changed and accident scenarios are to be provoked so that the number of corner cases can be artificially increased in test operation. The intervention of the safety driver in the driving situation will also be observed. To this end, criteria for measuring human-machine interaction (HMI) will be developed to track, for example, latency, attention, and intervention due to boredom of the drivers.

ACKNOWLEDGMENT

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project "KI Data Tooling – Methoden und Werkzeuge für das Generieren und Veredeln von Trainings-, Validierungs- und Absicherungsdaten für KI-Funktionen autonomer Fahrzeuge" under the grant number 19A200010. The authors thank the consortium for the successful cooperation. We also thank Matthias Rottmann for his productive support and Natalie Grabowsky and Ben Hamscher for driving the streets of CARLA and capturing the corner cases.

References

- A. Dosovitskiy et al., "CARLA: an open urban driving simulator," in 1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings, ser. Proceedings of Machine Learning Research, vol. 78. PMLR, 2017, pp. 1–16. [Online]. Available: http://proceedings.mlr.press/v78/ dosovitskiy17a.html
- [2] U. Chipengo, P. Krenz, and S. Carpenter, "From antenna design to high fidelity, full physics automotive radar sensor corner case simulation," *Modelling and Simulation in Engineering*, vol. 2018, pp. 1–19, 12 2018.
- [3] J.-A. Bolte et al., "Towards Corner Case Detection for Autonomous Driving," in 2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019. IEEE, 2019, pp. 438–445. [Online]. Available: https://doi.org/10.1109/IVS.2019.8813817
- [4] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and Fingscheidt, "Systematization of corner cases for visual perception in automated driving," in *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020.* IEEE, 2020, pp. 1257–1264. [Online]. Available: https://doi.org/10.1109/IV47402. 2020.9304789
- [5] J. Breitenstein *et al.*, "Corner cases for visual perception in automated driving: Some guidance on detection approaches," *CoRR*, vol. abs/2102.05897, 2021. [Online]. Available: https: //arxiv.org/abs/2102.05897
- [6] F. Heidecker, J. Breitenstein, K. Rösch et al., "An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving," in 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 2021.

- [7] R. Chan et al., "SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation," in *Thirty-fifth Conference on Neural Information* Processing Systems (NeurIPS) Datasets and Benchmarks Track, 2021. [Online]. Available: http://arxiv.org/abs/2104.14812
- [8] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5128–5137.
- [9] F. M. Zanzotto, "Viewpoint: Human-in-the-loop artificial intelligence," in Discussion and Doctoral Consortium papers of AI*IA 2019 -18th International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, ser. CEUR Workshop Proceedings, vol. 2495. CEUR-WS.org, 2019, pp. 84–94. [Online]. Available: http://ceur-ws.org/Vol-2495/paper10.pdf
- [10] H. Thiruvengada, A. Tharanathan, and P. Derby, *PerFECT:* An Automated Framework for Training on the Fly. London: Springer London, 2011, pp. 221–238. [Online]. Available: https: //doi.org/10.1007/978-0-85729-883-6_11
- [11] M. Monarch, R. Munro, and R. Monarch, Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI. Manning, 2021.
- [12] X. Wu *et al.*, "A survey of human-in-the-loop for machine learning," *CoRR*, vol. abs/2108.00941, 2021. [Online]. Available: https://arxiv.org/abs/2108.00941
- [13] J. Wu *et al.*, "Human-in-the-loop deep reinforcement learning with application to autonomous driving," *CoRR*, vol. abs/2104.07246, 2021. [Online]. Available: https://arxiv.org/abs/2104.07246
- [14] K. Driggs-Campbell, V. Shia, and R. Bajcsy, "Improved driver modeling for human-in-the-loop vehicular control," in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1654–1661.
- [15] A. E. Gómez *et al.*, "Driving simulator platform for development and evaluation of safety and emergency systems," *CoRR*, vol. abs/1802.04104, 2018. [Online]. Available: http://arxiv.org/abs/1802. 04104
- [16] G. Echeverria *et al.*, "Modular open robots simulation engine: Morse," in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 46–51.
- [17] P. Hang, Y. Zhang, and C. Lv, "Interacting with human drivers: Human-like driving and decision making for autonomous vehicles," 2022.
- [18] H. Sun *et al.*, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transportation Research Record*, vol. 2675, no. 11, pp. 587–600, 2021. [Online]. Available: https://doi.org/10.1177/03611981211018697
- [19] B. Dodson and H. Schwab, Accelerated Testing: A Practitioner's Guide to Accelerated and Reliability Testing, ser. Knovel Library. SAE International, 2006.
- [20] W. Nelson, Accelerated Testing: Statistical Models, Test Plans, and Data Analysis, ser. A Wiley-interscience publication. Wiley, 1990.
- [21] W. Q. Meeker and L. A. Escobar, "A review of recent research and current issues in accelerated testing," *International Statistical Review* / *Revue Internationale de Statistique*, vol. 61, no. 1, pp. 147–168, 1993. [Online]. Available: http://www.jstor.org/stable/1403600
- [22] Epic Games, "Unreal engine." [Online]. Available: https://www. unrealengine.com
- [23] R. P. K. Poudel *et al.*, "Fast-scnn: Fast semantic segmentation network," in *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019.* BMVA Press, 2019, p. 289. [Online]. Available: https://bmvc2019.org/wp-content/uploads/papers/0959-paper.pdf
 [24] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene
- [24] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [25] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/ 9015-pytorch-an-imperative-style-high-performance-deep-learning-library. pdf
- [26] F. M. Favarò *et al.*, "Examining accident reports involving autonomous vehicles in california," *PLOS ONE*, vol. 12, pp. 1–20, 09 2017.
 [Online]. Available: https://doi.org/10.1371/journal.pone.0184952