Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio

doi: 10.3850/978-981-18-8071-1_P414-cd

Ensuring Safety in Highly Automated Vehicles: A Review of Testing and Validation Methods for Robustness and Reliability

Carmen Frischknecht-Gruber, Benjamin Ricchiuto, Monika Reif, Joanna Weng, Christoph Senn

Institute of Applied Mathematics and Physics, Zurich University of Applied Sciences, Switzerland.
E-mail: carmen.frischknecht-gruber@zhaw.ch, benjamin.ricchiuto@zhaw.ch, monika.reif@zhaw.ch, joanna.weng@zhaw.ch, christoph.senn@zhaw.ch

The future of mobility is set to be reformed as the rapidly increasing use of driver assistance systems and highly automated vehicles (HAVs) show their great potential. The use of deep neural networks in autonomous driving systems has led to significant progress in this area. However, the increase in accidents involving HAVs highlights the need for effective testing and validation methods to increase the overall safety of these vehicles. With many technology companies and manufacturers aiming to put Level 4 and 5 vehicles into operation soon, the safety of HAVs remains a major concern. Rigorous testing and validation against potential failures and misbehaviour are required to ensure the reliability and robustness of these systems. This paper provides an overview of state of the art in testing and evaluation methods for machine learning-based HAVs. A literature review on these topics is provided to give valuable insights to researchers, practitioners and policymakers. As such, the review describes different types of validation, verification and testing methods, including real-world testing, simulation testing, hardware-in-the-loop testing, adversarial robustness, and methods used for explainability and interpretability in AI. The advantages and limitations are discussed and current challenges are highlighted. Finally, open research questions and future directions in the field are identified.

Keywords: Highly automated vehicles, autonomous vehicles, robustness, reliability, testing, validation, machine learning, safety

1. Introduction

Highly automated vehicles (HAVs) have become a topic of great interest in recent years, and various companies are making large investments in this technology. These vehicles require a variety of sensors and algorithms to perform their specific tasks. However, the challenges we face with HAVs are complex. These challenges include perceiving the environment and dealing with partial, incomplete or erroneous information. It is crucial that HAVs understand the environment to interact with it safely. HAVs perceive the real environment with the help of sensors by constructing a representation of it. For example, to navigate and interact with an unstructured and changing environment, obstacle detection or semantic classification can be performed. Navigation is based on localisation, trajectory planning and the use of sensors such as GPS, range sensors, ultrasonic sensors, high-resolution RGB cameras (Tesla, 2021), depth sensors, sonar, radar and 3D LiDAR (Waypoint, 2022). A combination of different sensor types provides a more reliable picture of the environment, as each type has its own limitations (Yeong et al., 2021). Sensor data can provide insufficient information due to inherent limitations, system errors, volatile changes in environmental conditions (such as weather) and unpredictable behaviour of other actors involved. HAVs require a high degree of robustness in terms of their perception system and trajectory planning. They cannot afford severe failures in any real-world scenario, as these kinds of failures can lead to injuries or even death and damage to the environment. Their application must meet higher requirements than human drivers, as social acceptance and trust in the technology will not be given otherwise (Hutson, 2017). In addition to the use of various sensors in the perception system, stable mapping procedures and reliable trajectory planning are essential for the safe operation of HAVs. In order to support their development and deployment, a variety of testing methods must be performed. These tests should take into account HAV functionalities, including sensor usage, ML algorithms, and complex decision-making processes. In this context, researchers and industry have developed a variety of testing methods, such as simulated test environments, closed-loop testing or on-road testing. Each of these approaches has its own advantages and disadvantages, and their suitability depends on the specific needs and objectives of the test program. Along with these testing methods, other evaluation methods are needed to increase the robustness and explainability of the algorithms to provide sufficient confidence in the application. This paper provides a comprehensive overview of the state of the art in testing and evaluation methods for ML-based HAVs. It also identifies open questions and future directions in the field to guide further research.

2. Highly Automated Vehicles

HAVs are becoming increasingly common in modern transportation, with potential to reduce human error, improve driving practices and prevent road accidents (European Automobile Manufacturers Association, 2019). However, ensuring the safety of these systems requires rigorous testing and validation methods, the creation of and compliance with standards and certification procedures, and consideration of potential environmental impacts.

2.1. *SAE Levels of Driving Automation*™

The Society of Automotive Engineers (SAE) has introduced a classification system for autonomous vehicles (SAE International, 2021), ranging from Level 0 to Level 5, based on the degree of automation. Level 0 represents no automation while a human is in full control over the vehicle. Level 1 and Level 2 vehicles have some automated functions, such as adaptive cruise control, lane departure warning, lane centering, or self-parking, but require a human driver to be ready to take over at any moment. Level 3 to Level 5 vehicles can drive autonomously with minimal to full automation. The higher the level, the more tasks the vehicle must be able to execute under several conditions. These tasks and conditions include driving on a highway or in a designated area, or operating on any road and under any conditions.

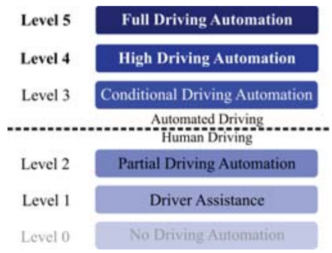

Fig. 1. SAE Levels of Driving Automation™. The threshold between level 2 and level 3 separates them into the categories "Human Driving" and "Automated Driving" (SAE International, 2021).

Automated driving can be described as a process of sensing the environment (sense), thinking about the appropriate action (plan/think), and executing it through actuation (act).

2.2. *Sense*

Sensing is a key aspect of HAVs that relies on sophisticated sensors and cameras to detect and interpret the environment, including the identification of other vehicles, pedestrians, road signs, and traffic signals, as well as environmental conditions. Computer vision is a critical task, as it allows the vehicle to visually represent the environment. Perception involves using high-resolution cameras (Tesla, 2021), LiDAR (Waypoint, 2022), and other sensors to detect and classify objects in the environment. Techniques such as Convolutional Neural Networks (CNNs) for object detection and semantic segmentation (Girshick et al., 2014; Ronneberger et al., 2015; Ren et al., 2015) are commonly used for this purpose. Algorithms such as You Only Look Once (YOLO) (Redmon et al., 2016), Faster Region-based Convolutional Neural Networks (R-CNN (Ren et al., 2015), and Single Shot MultiBox Detector (SSD) (Liu et al., 2016) are highly used to detect, analyse and identify objects. YOLO, for instance, is a real-time object detection algorithm that performs detection and classification in a single pass through the network. It has the advantage of being fast and efficient, allowing it to be used in real-time applications. Nonetheless, earlier versions might encounter difficulties in identifying small objects, performing less accurately in this regard compared to other methods (Yeong et al., 2021). Faster R-CNN is a two-stage object detection algorithm that uses a Region Proposal Network (RPN) to generate potential object locations followed by a classification network to detect and classify objects (Ren et al., 2015). It is slower and less efficient than YOLO but has the benefit of being more accurate and is also able to detect smaller objects. SSD (Liu et al., 2016) is a one-stage object detection algorithm that uses a single convolutional network to detect and classify objects in a single pass. It is comparably fast and efficient as YOLO but has higher accuracy and is also able to detect smaller objects. However, it may struggle with accurately detecting objects at varying scales and aspect ratios. LiDAR point cloud processing is another critical feature for HAVs. These algorithms detect objects and estimate their position and orientation based on the data collected by the LiDAR sensors. Multi-sensor fusion can be used for more accurate perception, combining information from multiple sensors (Yeong et al., 2021). Algorithms such as Kalman filter (Sasiadek and Hartana, 2000), Particle Filter (Jain et al., 2011), and Conditional Random Fields (CRF) (Xiao et al., 2015) are used for this purpose. Finally, HAVs apply a combination of mapping and localisation, subsequently using planning algorithms to navigate.

2.3. *Plan/ Think*

Depending on the scene representation from the previous steps, navigation inputs and traffic rules, specific path restrictions are adopted and taken into account for path planning. Since the position is altered constantly while moving, the trajectory is recalculated and adjusted accordingly. Several trajectory planning algorithms can be used individually or in combination, such as A* algorithm, Rapidly-Exploring Random Trees (RRT) (Zhou et al., 2021), Model Predictive Control (MPC) (Ji et al., 2017), Dynamic-Window Approach (Fox et al., 1951), and Probabilistic Roadmap (PRM). These algorithms plan a collision-free and feasible trajectory, optimising objectives such as minimising travel time or energy consumption while considering the vehicle's dynamics and constraints. Decision-making in HAVs is typically accomplished through a combination of software and hardware components. The software component involves the use of ML models that utilise input data from sensors and other sources to calculate accurate predictions, enabling the vehicle to make informed decisions. The hardware component includes sensors, processors, and actuators that allow the vehicle to perceive its environment, process information, and take physical actions such as steering, accelerating, and braking.

2.4. *Act*

During the "act" phase, appropriate actions are executed by various types of actuators to control the movement of the vehicle based on the decisions made in the "plan" phase. HAVs are designed to perform complex manoeuvres, including lane changes and merging onto highways while ensuring the safety of passengers and other road users. Communication technologies such as Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) can be established, enabling sharing information like speed, direction, road conditions, or traffic congestion. This exchange of information allows for better-informed operations and enhances the decision-making and finally acting capabilities of HAVs.

3. Challenges

The accurate perception of HAVs, which includes mapping, detection, and sensor technology, heavily depends on environmental factors. Nevertheless, unpredictable changes, such as weather conditions, incomplete or incorrect data, can impede accurate perception (Zhang et al., 2023). The quality of the perceived data and the accuracy of predictions made by ML algorithms highly impact the safety of the HAVs (Kim et al., 2023). Overfitting issues, the lack of transparency and understanding regarding ML models, and occurring model drifts while learning online must be mitigated. Decision-making in real-time is a non-trivial task, and even small flaws in this concept can lead to fatal accidents (Okumura, James, Kanzawa, Derry, Sakai, Nishi, and Prokhorov, Okumura et al.). Further, the interaction with other actors (e.g., pedestrians, cyclists, other vehicles) and the environment, which both can show unpredictable or unknown behaviours and conditions, must function appropriately to enable HAVs to respond correctly. Given the open-world characteristics of HAV applications, it is not possible to ensure complete test coverage of all systems. Moreover, HAVs are vulnerable to cyber-attacks (Petit and Shladover, 2014), which compromise safety and security further. Besides the technical considerations, societal implications must be addressed since HAVs are still a relatively new technology. Societal acceptance and trust-building are critical, including concerns about ethics, fairness, data privacy, liability questions and overall robustness. This is why regulations, standards, and certification processes are needed and should be put in place soon (Frischknecht-Gruber et al., 2022). A new emerging problem will be the human-out-of-the-loop once fully autonomous systems are in place, even though human drivers are already failing to act as a fallback in many cases (U.S. Department of Transportation, 2022) (see Figure 2).

Fig. 2. Challenges of automated driving.

3.1. *Safety Standards and Certification*

Regulations, standards, and certification are vital to ensure the safety and acceptance of HAVs and autonomous systems. However, current safety standards, such as ISO 26262, only focus on functional safety and do not cover operational safety of automated driving functions comprehensively. Hence, the standard ISO 21448 Road vehicles - Safety of the intended functionality (SOTIF) was introduced. Due to the infinite number of unknown and unsafe scenarios that HAVs encounter and the central role that AI components play in their operation, new standards are currently being developed. ISO and IEC are collaborating on standardisation in AI through their Joint Technical Committee (JTC 1) in subcommittee 42 (SC 42), which includes standards like ISO/IEC TR 24029-1 "Assessment of the Robustness of Neural Network" and ISO/IEC CD TR 5469 "Functional Safety and AI Systems". IEEE SA is developing standards for AI systems, covering transparency, data privacy, algorithmic bias, and ethics. Their Ethically Aligned Design (EAD) guides the design, production, and use of autonomous and intelligent systems, including standards, certifications, and regulations.

4. Testing

Testing is essential in developing HAVs as infinite scenarios may occur on-road with the emergence of unexpected cases. Accidents of leading manufacturers such as Tesla (U.S. Department of Transportation, 2022) emphasize the need to find these edge cases beforehand. Testing in a specified and/or controlled environment allows developers to run or simulate various scenarios that help identify and fix potential problems early in development. It is also worth mentioning the importance of risk-based methods in selecting test scenarios (Gelder et al., 2019) since they play a significant role in optimising the testing process for HAVs. All of this contributes significantly to fostering trust in the technology.

4.1. *Verification & Validation*

Verification aims to answer the question, "Are we building the system right?" by checking whether the system adheres to the intended design and requirements. Verification activities typically involve code inspection, unit, integration, and system testing to detect system errors, defects, or anomalies. The goal is to identify and address any issues before moving on to validation. Validation ensures that autonomous driving systems perform as intended and meet safety standards through testing and evaluation in various scenarios and conditions on the completed real system.

4.2. *Simulation*

Simulation testing is valuable for developing HAVs, enabling scalability, shorter analysis times, and reduced risks. Two notable open source simulation platforms are CARLA (Dosovitskiy et al., 2017) and AirSim (Shah et al., 2017). CARLA is built on Unreal Engine and provides a robust API, scalable client-server architecture and realistic core physics. It offers a wide range of vehicles, objects, weather conditions, and sensors, making it suitable for comprehensive end-to-end testing. CARLA integrates seamlessly with Robot Operating Systems (ROS) and Autoware, providing flexible APIs and advanced simulation control. It is particularly well suited for testing perception, mapping, localization, and vehicle control algorithms, including traffic scenarios. AirSim, developed by Microsoft Research, is another open-source simulation platform based on Unreal Engine. It specializes in high-fidelity robotics and vehicle simulation and is suitable for both software-in-the-loop (SIL) and hardware-in-the-loop (HIL) simulations. Airsim offers flight and driving simulations and supports ROS. It provides numerous APIs, and sensors, as well as realistic vehicle dynamics and environments. These platforms enable testing of autonomous driving and perception algorithms in various environments. Simulation testing is preferred for dangerous driving scenarios like collisions due to safety concerns, but limitations remain. Realistic system dynamics require dedicated SIL and HIL techniques, while high-level algorithms need to be tested in appropriate environments like common game engines.

4.3. *Software-in-the-Loop Testing*

SIL testing is an essential component of HAV testing. SIL involves testing a system's software in a simulated environment, where the hardware components, vehicle dynamics, and environment are simulated in real-time. Also, SIL testing allows for testing architectures, validating perception systems, and planning and decision-making of HAVs. It provides an efficient and cost-effective way to test and validate the software of HAVs without the risks, limitations and disadvantages associated with real-world testing.

4.4. *Hardware-in-the-Loop Testing*

When conducting HIL testing, physical hardware components are integrated with simulated ones in a closed-loop system. This method is particularly useful in autonomous and HAVs, as it enables researchers to evaluate how the vehicle's hardware components (such as sensors, actuators, and electronic control units) interact with the software algorithms that control them. HIL testing connects hardware components to a simulation environment, generating signals that simulate real-world conditions. The hardware components' outputs are fed back into the simulation environment to generate additional input. This closed-loop system allows for testing of the complete system under realistic conditions, without the need for actual physical testing on the road. Thus, this method is particularly useful for testing safety-critical systems. Also, automated HIL testing enables the efficient analysis of large amounts of data.

4.5. *Closed-course Testing*

In closed-course testing, vehicles are assessed in a restricted setting, such as private test tracks, to ensure controlled conditions. This approach allows engineers to test the vehicle's performance in a safe and manageable environment without the risk of endangering human lives or the environment. It can be used to analyse a range of HAVs functions, such as perception, trajectory planning, decision-making, and control. However, closed-course testing shows limitations in terms of complexity and realism of the testing environment. It cannot cover the diversity of scenarios that might be encountered in real-world driving conditions. Moreover, some limitations regarding obstacles and scenarios that can be tested, as well as restrictions on the level of complexity that can be introduced to the testing environment, might be due to cost constraints. Despite these limitations, closed-course testing remains an important tool for evaluating autonomous vehicles' basic functionality and safety before testing on-road. Additionally, closed-course testing can be used in conjunction with other testing methods, such as simulation and field testing, to provide a more comprehensive and realistic evaluation of the vehicle's performance.

4.6. *On-Road Testing*

On-road testing, while suitable, is time and cost-intensive and requires numerous safety precautions. Companies such as Tesla and Waymo have conducted millions of test kilometres by themselves or received from their customers showing some issues (Victor et al., 2023; U.S. Department of Transportation, 2022). However, it is worth mentioning that these accidents are relatively rare compared to the number of kilometres driven by the systems (Waymo, 2021). So, although numerous test kilometres have been accumulated, it still cannot be ensured that all possible hazardous edge cases have been discovered and mitigated in advance. From on-road testing, a next step could be digitalising the collected real-world data to create a surrogate of the real world, allowing for a simulation testing environment to repeat scenarios and add or generate altered surrogate datasets. Thus, test drives are the final link in the validation chain to evaluate the system's performance in a real-world physical infrastructure setting (Waymo, 2021).

4.7. *Verification of AI Components*

4.7.1. *Adversarial Robustness*

In the field of ML, the importance of adversarial robustness has become increasingly apparent. Adversarial attacks are a type of threat that can compromise the security and integrity of ML models, particularly neural network-based classifiers, by exploiting existing vulnerabilities. (Goodfellow et al., 2015). Whereas adversarial testing is a powerful technique to verify and validate the robustness and resilience of HAVs that are based on ML algorithms. This technique involves exposing the system to unexpected and adversarial scenarios. The goal is to identify potential vulnerabilities, test the system's response and develop mitigation strategies. It includes simulating scenarios such as sudden obstacles appearing on the road, cyber-attacks on the vehicle's communication and control systems (Petit and Shladover, 2014), unexpected weather conditions or dedicated noise, alterations to sensor inputs (Goodfellow et al., 2015). Insights are gained into the system's behaviour, and areas for required improvement are identified. Adversarial testing provides a way to address the uncountable scenarios that can occur in the real world and edge cases that may not have been anticipated during the design and development phase.

4.7.2. *Explainable and Interpretable AI*

ML algorithms used in HAVs, often operate as black boxes, meaning their decision-making processes and outputs are difficult to interpret or explain. This lack of interpretability and explainability can be problematic in understanding how and why a system came up with a particular decision or action. This must be addressed to decide on accountability and liability issues and to build trust. With a lack of understanding, improving models and systems by identifying and addressing any biases, errors, or limitations correctly (Ribeiro et al., 2016) is difficult. From a regulatory and legal perspective, it becomes apparent that HAVs and fully autonomous systems need to be transparent and explainable to ensure reliability, safety, and fairness. Understanding the decision-making process is a crucial aspect of ML, and the use of interpretable models such as decision trees has been suggested as a potential solution to enhance interpretability. Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016), Shapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) or saliency maps (Simonyan et al., 2017) have emerged as promising techniques for explaining the output of these models.

4.7.3. *Test-Case Generation Using AI Approaches*

There exists considerable research on the use of ML algorithms to create suitable test cases. Especially, the creation of particularly critical, hazardous test environments has to be taken into account. There are various possibilities for creating edge cases. One possibility involves using generative models such as generative adversarial network architectures (GANs) to synthetically create further test scenarios from an existing data set of critical cases. Further possibilities are offered by variational autoencoders (VAEs), which can construct similar cases from the latent-space representation of existing data. Additional interesting approaches include KING, which generates safety-critical driving scenarios via kinematics gradients (Hanselmann et al., 2022), Anti-CARLA, which is a framework that automatically generates adversarial test cases or also from the field of game-level generation (Ramakrishna et al., 2022), such as the creation of Mario game levels in the latent space of DC-GANs based on an evolutionary strategy (Volz et al., 2018). Once generated, these test scenarios can be used in a suitable simulation environment to test corresponding agents. Other possibilities in terms of generation include, for example Loiacono et al. (2011); González-Duque et al. (2022); Chen et al. (2021).

5. Discussion and Outlook

This review provides an overview of testing and validation methods for robustness and reliability in current HAVs and autonomous vehicles. Handling the complex and open-world nature of HAVs in real-world applications is a major challenge. We highlight the importance of using simulation environments to identify corner cases and conduct thorough testing, reducing time and costs. Scalable simulation tests, such as those shown in AirSim and CARLA, supplement real-world testing, allowing the design of complex scenarios not feasible in reality. We emphasize the need for diverse testing methods, including SIL and HIL tests, to create a complete picture. Moreover, we highlight the critical role of ML algorithms as essential components within HAV systems, illustrating potential issues that may arise while presenting effective methods for mitigating such challenges. The development and deployment of HAVs and autonomous driving systems are expected to show many benefits, including convenience, improved traffic conditions, and overall safety improvement. However, further research is needed to identify complex scenarios, develop appropriate tools that allow for scenario creation, shorten testing time, enhance algorithm interpretability, and improve system robustness. Despite substantial investments and testing efforts, additional research is required to address challenges and concerns. Future work should explore more suitable algorithms for creating test case scenarios, investigate approaches for achieving full coverage test rates, and delve into accelerated testing methods.

References

Chen, X., J. Xu, R. Zhou, W. Chen, J. Fang, and C. Liu (2021). TrajVAE: A Variational AutoEncoder model for trajectory generation. *Neurocomputing 428*.

Dosovitskiy, A., G. Ros, F. Codevilla, A. Lopez, and V. Koltun (2017). CARLA: An Open Urban Driving Simulator. PMLR.

European Automobile Manufacturers Association (2019). Road Safety.

Fox, D., W. Burgard, and S. Thrun (1951). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine 4*(1).

Frischknecht-Gruber, M.-L., C. W. Senn, and M. U. Reif (2022). Global Efforts Towards Establishing Safety Directives for Intelligent Systems: Review. Dublin. Research Publishing.

Gelder, E., J.-P. Paardekooper, O. Camp, and B. De Schutter (2019). Safety assessment of automated vehicles: how to determine whether we have collected enough field data? *Traffic Injury Prevention 20*.

Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR.

González-Duque, M., R. B. Palm, S. Hauberg, and S. Risi (2022). Mario Plays on a Manifold: Generating Functional Content in Latent Space through Differential Geometry.

Goodfellow, I. J., J. Shlens, and C. Szegedy (2015). Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572*.

Hanselmann, N., K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger (2022). KING: Generating Safety-Critical Driving Scenarios for Robust Imitation via Kinematics Gradients. arXiv:2204.13683.

Hutson, M. (2017). People don't trust driverless cars. *Science 365*.

Jain, S., S. Nandy, R. Ray, and S. N. Shome (2011). Application of Particle Filtering Technique for sensor fusion in mobile robotics. *2011 IEEE ICMA*.

Ji, J., A. Khajepour, W. W. Melek, and Y. Huang (2017). Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control with Multiconstraints Computation. *IEEE TVT 66*(2).

Kim, K., S. Lee, and H. Kim (2023). Quality Assurance of Autonomous Vehicle's Sensor Data based on Classifying Level of Difficulty.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2016). SSD: Single Shot MultiBox Detector. Volume 9905.

Loiacono, D., L. Cardamone, and P. L. Lanzi (2011). Automatic Track Generation for High-End Racing Games Using Evolutionary Computation. *IEEE T-CIAIG 3*(3).

Lundberg, S. and S.-I. Lee (2017). A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874.

Okumura, B., M. James, Y. Kanzawa, M. Derry, K. Sakai, T. Nishi, and D. Prokhorov. Challenges in Perception and Decision Making for Intelligent Automotive Vehicles. *IEEE T-IV 1*(1).

Petit, J. and S. Shladover (2014). Potential Cyberattacks on Automated Vehicles. *IEEE T-ITS 16*.

Ramakrishna, S., B. Luo, C. Kuhn, G. Karsai, and A. Dubey (2022). ANTI-CARLA: An Adversarial Testing Framework for Autonomous Vehicles in CARLA. arXiv:2208.06309.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640.

Ren, S., K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in neural information processing systems 28*.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938.

Ronneberger, O., P. Fischer, and T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. MICCAI*.

SAE International (2021). J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.

Sasiadek, J. and P. Hartana (2000). Sensor data fusion using Kalman filter. In *Proc. FUSION*, Volume 2.

Shah, S., D. Dey, C. Lovett, and A. Kapoor (2017). AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.

Simonyan, K., A. Vedaldi, and A. Zisserman (2017). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *IEEE TVT 66*(2).

Tesla (2021). Tesla Vision Update: Replacing Ultrasonic Sensors with Tesla Vision.

U.S. Department of Transportation (2022). Summary Report: Standing General Order on Crash Reporting for Level 2 Advanced Driver Assistance Systems. *NHTSA*.

Victor, T., K. Kusano, T. Gode, R. Chen, and M. Schwall (2023). Safety Performance of the Waymo Rider-Only Automated Driving System at One Million Miles.

Volz, V., J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi (2018). Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. In *Proc. GECCO*.

Waymo (2021). Waymo Safety Report.

Waypoint (2022). Informing smarter lidar solutions for the future. *The official Waymo blog*.

Xiao, L., B. Dai, D. Liu, T. Hu, and T. Wu (2015). CRF based road detection with multi-sensor fusion. In *IEEE IV*.

Yeong, D. J., G. Velasco-Hernandez, J. Barry, and J. Walsh (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors 21*(6).

Zhang, Y., A. Carballo, H. Yang, and K. Takeda (2023). Perception and sensing for autonomous vehicles under adverse weather conditions. *ISPRS 196*.

Zhou, C., B. Huang, and P. Fränti (2021). A review of motion planning algorithms for intelligent robotics. *Intel Manuf 33*.