World Electric Vehicle Journal

Article

The Safety Risks of AI-Driven Solutions in Autonomous Road Vehicles

Farshad Mirzarazi †, Sebelan Danishvar *,† and Alireza Mousavi †

College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK; farshad.mirzarazi@brunel.ac.uk (F.M.); alireza.mousavi@brunel.ac.uk (A.M.)
* Correspondence: sebelan.danishvar@brunel.ac.uk
† These authors contributed equally to this work.

Abstract: At present Deep Neural Networks (DNN) have a dominant role in the AI-driven Autonomous driving approaches. This paper focuses on the potential safety risks of deploying DNN classifiers in Advanced Driver Assistance System (ADAS) systems. In our experience, many theoretically sound AI-driven solutions tested and deployed in ADAS have shown serious safety flaws in practice. A brief review of practice and theory of automotive safety standards and related body of knowledge is presented. It is followed by a comparative analysis between DNN classifiers and safety standards developed in the automotive industry. The output of the study provides advice and recommendations for filling the current gaps within the complex and interrelated factors pertaining to the safety of Autonomous Road Vehicles (ARV). This study may assist ARV's safety, system, and technology providers during the design, development, and implementation life cycle. The contribution of this work is to highlight and link the learning rules enforced by risk factors when DNN classifiers are expected to provide a near real-time safer Vehicle Navigation Solution (VNS).

Keywords: advanced driver assistance systems (ADAS); deep learning classifier; autonomous driving; functional safety; hyperparameters; Safety of the Intended Functionality (SOTIF); ISO 26262; ISO 21448; ISO PAS 8800; autonomous road vehicles (ARV); Vehicle Navigation Solution (VNS)

check for updates

Citation: Mirzarazi, F.; Danishvar, S.; Mousavi, A. The Safety Risks of AI-Driven Solutions in Autonomous Road Vehicles. *World Electr. Veh. J.* **2024**, *15*, 438. https://doi.org/10.3390/wevj15100438

Academic Editors: Yaoji Deng, Xinglong Zhang and Fen Lin

Received: 12 August 2024
Revised: 20 September 2024
Accepted: 24 September 2024
Published: 26 September 2024

1. Introduction

In the automotive, marine, and aviation industries, there is a strong economic and technological imperative to design and apply safe, clean, and sustainable autonomous vehicles. The demand in the automotive industry which is the focus of this work is expected to triple by 2027. Autonomous driving systems, along with the use of machine learning and artificial intelligence in vehicle design, are changing the way which we interpret and interact with modern transportation systems. This has resulted in the requirement and need for robust safety standards that can address the unique challenges introduced by these emerging technologies.

Machine learning (ML) and artificial intelligence (AI) can significantly increase road safety and traffic flow efficiency through the adoption of advanced driver assistance systems (ADAS). These technologies let vehicles learn from volumes of data and make decisions in real time. Hence, this helps them to predict and respond to a wide range of driving scenarios more appropriately. However, with the integration of ML and AI, new safety concerns and challenges arise that have to be considered with due diligence. Standards for safety have to be followed stringently for the proper functioning-reliability and safety-of such systems.

Researchers and practitioners are working on new safety standards that specifically address how ML and AI will be utilized in the automotive industry to overcome these challenges. Our purpose is to establish a framework for analysis and matching the design, engineering, and validation of autonomous driving systems with the help of these emerging standards, as well as providing instructions for making sure they are used objectively to reduce safety risks. The introduction of such standards is crucial in fostering confidence in the safety and reliability of autonomous vehicles and accelerating their widespread adoption. The different aspects of safety within the connection of AI-based autonomous driving systems, the limitations of existing standards, and the proposed solutions to these challenges are presented in the conclusion.

The automotive industry has shifted its research, and investments into offering autonomous driving experiences and applications. Autonomous vehicles are built on ADASs which enhance driver safety, comfort, and convenience, and reduce overall accident risk. There are numerous types of ADAS, including lane-keeping systems, blind spot detection, adaptive cruise control, emergency braking, obstacle detection, and predictive navigation systems [1].

ADAS systems are typically classified into different levels in accordance with guidelines set by the Society of Automotive Engineers (SAE) International Level of Automation Scale [2]. These levels range from level 0, which is no automation, to level 5, which is full autonomy. Automation levels 1 and 2 typically involve driver assistance systems such as Adaptive Cruise Control or Lane Keeping Assist, while levels 3 and 4 are able to take over most of the driving tasks and involve partial automation, and level 5 represents full autonomy (Table 1).

Table 1. SAE’sAutonomous Driving Levels.

Level 0	No Automation; The driver is in complete control of the vehicle and all its functions.
Level 1	Driver Assistance; Certain functions of the vehicle are automated, such as cruise control and lane centering.
Level 2	Partial Automation: The vehicle executes acceleration and braking, while the human driver is responsible for steering.
Level 3	Conditional Automation; The vehicle is capable of performing all dynamic driving tasks, but the human driver must be ready to take control at any time.
Level 4	High Automation: The vehicle performs all dynamic driving tasks without any input from the human driver.
Level 5	Full Automation: The vehicle performs all dynamic driving tasks and monitors the driving environment without any input from the human driver.

The automotive industry is quickly moving towards level 5 automation, and it is expected that by 2030, the automotive industry will move towards level 4 and 5 automation [3]. Table 1 shows SAE’s autonomous driving levels. The challenge is to develop ADAS technologies that are robust and reliable enough to safely handle complex driving scenarios [4]. For highly automated driving vehicles ADAS systems must integrate a large number of intricate sensing components, from radar to optical sensors such as camera and Lidar (Light Detection and Ranging) sensors, with sophisticated algorithms in order to detect and respond to any potential road hazards. This makes the development of ADAS systems challenging for OEMs (Original Equipment Manufacturers) [4]. Furthermore, the development and deployment of ADAS systems require an additional layer of design, namely the Operation Design Domain (ODD), which defines the operational context of the system [5] in terms of boundaries, constraints, and features.

Autonomous vehicles are increasingly vulnerable to cybersecurity threats as they rely more on advanced connectivity and communication technologies. Authors in [6] claim that the most damaging cybersecurity threats originate from autonomous vehicles connecting to the internet, providing onboard Wi-Fi services, communicating with other vehicles and infrastructure, and supporting advanced features like over-the-air firmware updates. Furthermore, as noted by authors in [7], attackers can manipulate the ADAS sensors of an AV to create deceptive scenarios, leading vehicles to misinterpret their environment. For instance, projecting fake obstacles can cause unnecessary braking, while hiding real obstacles can result in collisions, posing serious safety risks.

Further aspects requiring careful consideration when deploying DNN models in autonomous vehicles include explainability, interpretability, and accountability. Explainability refers to the ability of a model to provide a rationale for its outputs that can be easily understood and trusted by humans [8]. Interpretability refers to how well a human can comprehend a model's prediction and decision-making solely by model's design, without additional information [9]. At architectural design phase of DNN models, collaborative review and joint design between development teams become more efficient and outcome-oriented when models are more explainable (see Section 4.1).

Interpretability in DNN models is extremely valuable during the verification and validation (V&V) phase. Test engineers can assess a model's behavior, ensuring that it operates as expected under various driving conditions. On the other hand, a lack of interpretability can lead to insufficient V&V test coverage, leaving residual risks of untested software that could compromise safety and reliability in real-world scenarios (see Section 4.4).

Accountability (i.e., explaining the wrongdoing of AVs) is another challenge in developing and deploying DNN solutions for autonomous driving. Nordhoff, in his paper Resistance towards Autonomous Vehicles (AVs) [10], argues that resistance to AV deployment can be attributed to unresolved legal accountability. To address this challenge and reduce opposition, he suggests governance and regulation with clear legislation and defined stakeholder roles, particularly in accidents, along with greater transparency in data collection. In a different approach, researchers in [11] proposed a decision-table-based tool for legal accountability for automated decision-making.

Similarly, there are important ethical considerations to note. Decision-making in autonomous driving is increasingly being shifted, either partially or entirely, from humans to AI in areas that historically required human interpretation and ethical judgment [12]. Determining clear moral responsibility for harm or injury caused by system behavior is crucial to gaining public trust in autonomous systems [12].

Human oversight and intervention throughout all phases of machine learning development for autonomous driving systems can be an effective way to address the grey zone created by issues such as explainability, accountability, and ethical concerns. However, it is important to note that human oversight can be costly and is not completely free from biases and the risk of error.

Researchers at OpenAI [13] found scalable oversight techniques can allow humans to supervise models in an efficient way. For example, humans ask models to critically evaluate the outputs of other models [14]. However, they also introduced a method called "WEAK-TO-STRONG GENERALIZATION" which takes a different approach by focusing on generalizing beyond human supervision such that models perform well even if human supervision is not reliable.

Autonomous driving is a highly complex topic with multiple facets, each presenting different perspectives and challenges that must be addressed. Beyond security, explainability, accountability, and human oversight, other critical topics include but are not limited to real-time data processing and latency in AI systems, continuous learning and model updates, and collaborative AI for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. A more complex machine learning model might offer better generalization and accuracy, but improvements in latency and run-time processing performance are essential to meet safety requirements [15]. Updating models after deployment is crucial to ensure the system adapts itself to new unseen driving scenarios. By using self-evaluating and self-learning algorithms, incorporating Reinforcement and meta-learning techniques, models can be continuously refined through Flash Over The Air (FOTA) updates, to maintain the AV's ability to respond to evolving conditions [16,17]. Collaborative AI for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication provides machine learning models with more reliable information about the vehicle's surroundings [18], enhancing safety by enabling more coordinated decision-making.

Lane Keeping Assistant (LKA) is a critical ADAS function for highly autonomous driving which uses a complex network of radar sensors, cameras, and algorithms to monitor the vehicle's lane position that proactively maintains proper lane alignment, considering road markings. Therefore, accurate object classification is necessary for the system to respond and effectively to potential risks and hazards, providing timely warnings and taking corrective actions. If the autonomous system inaccurately classifies the drivability of an object or fails to recognize road markings, it can lead to serious safety risks or injuries.

The importance of accurate classification in LKA systems is illustrated in Figure 1. The ego vehicle is situated in lane 1, several hundred meters ahead lies a stationary road closure obstacle. Lane 2 is currently unoccupied, while in lane 3, there is an approaching vehicle. If the system misclassifies the drivability of these objects or fails to recognize the road markings, it could result in incorrect lane-keeping decisions, such as unnecessary lane changes or abrupt braking, potentially leading to hazardous situations. Hence, in this particular driving context, the precision of object classification, coupled with road marking identification, remains essential for the Lane Keeping Assistance (LKA) system to operate safely and efficiently in the presence of both stationary and moving obstacles.

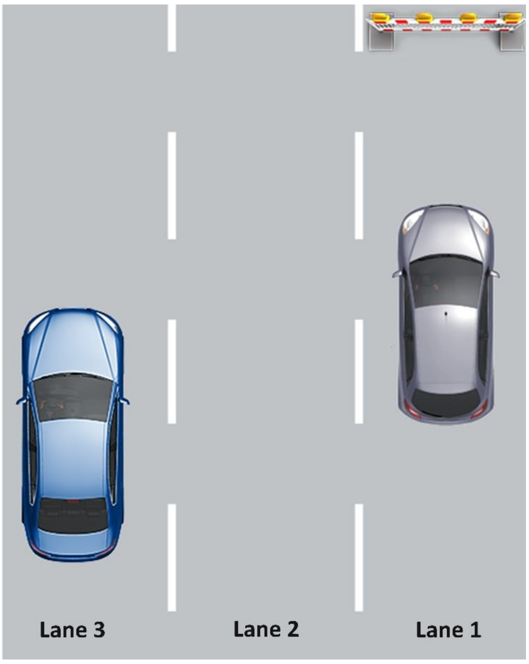

Figure 1. The significance of accurate object classification for ADAS Lane-Keeping Assistant systems.

Car manufacturers struggle with the enormous challenge of designing systems that are feasible both economically and from the point of safety, constantly detecting and responding to potential roadside hazards in real-time and under changing conditions-a task that involves a fast and robust computer vision system, sensor fusion methodology, and machine learning innovations. In addition, risk evaluation, mitigation mapping, and effective response strategies are used to achieve a safe and reliable situation awareness and action life cycle. These works complement the increasing awareness in the industry for the integration of advanced technologies like artificial intelligence and machine learning into ADAS systems. Regardless of these challenges, the importance of ADAS systems has grown in the automotive sector and is a significant step toward full autonomy. While the industry is working on more sophisticated systems, it expects advanced technologies to be integrated. On top of the technological developments, however, automakers also have to clear the regulatory and safety hurdles which ADAS systems have to meet before they can be deployed in production vehicles.

In this paper, we attempt to make a contribution to the literature by performing a systematic and comprehensive comparative analysis between DNN classifiers and established automotive safety standards perspective, which, to the best of our knowledge, is still largely underrepresented in the existing literature. The novelty in our approach lies in fully dislodging the interrelation and complexity of the many factors involved in deploying DNN classifiers in ADAS systems, which provides insight into the emergence and practical implications that are instrumental in refining safety protocols in autonomous driving technologies. We not only consolidate existing knowledge on safety risks but also highlight unaddressed gaps and then provide actionable and specific recommendations which, besides aiming at the standardization of safety methodologies, contribute to informed decision-making in the development and deployment of ADAS systems. The emphasis is on providing a comprehensive guide that is practical and relevant for safety, system, and technology providers of autonomous vehicles throughout the lifecycle of design, development, and implementation. In this work, we emphasize the inclusion of risk factors into learning rules, thus enabling us to develop near real-time safer vehicle navigation solutions by refining DNN techniques.

The organization of the paper is outlined as follows: Section 2 deals with the current and most relevant studies related to secure AI-driven methodology, describing the key findings and contributions of each. Section 3 provides a close review of the main automotive safety standards, focusing on the limitations of ISO 26262 [19] and SOTIF related to AI-driven and automated driving technologies and presenting ISO PAS 8800 [20]. Section 4 is the core of this manuscript, discussing in detail the safety issues related to the architectural setting, training phases, deployment strategies, and validation of deep learning systems for automated driving. This section presents a thorough safety framework that provides efficient countermeasures and alternatives to mitigate these risks, as well as examines several critical validation techniques necessary for the implementation of AI-driven automated driving systems. Section 5 serves as the conclusion, outlining final observations and prospective future avenues in this domain.

2. Related Work

Authors in [21] represent how crucial it is to develop methods for quantifying the risks related to deep neural networks, especially as they become more prevalent in safety-critical applications, such as medical diagnosis systems and autonomous vehicles. They defined a new class of risk metric called “uncertainty example” based on a probabilistic modeling approach and developed a framework that allows quantification of both the likelihood and severity of safety-critical metrics in a computationally effective algorithm. They evaluated the framework on several image classification tasks and demonstrated its effectiveness in identifying safety risks associated with specific neural network architectures and training procedures. They also demonstrate how the framework can be used to guide the design of more robust and reliable neural network systems. Their work has made a significant contribution to quantifying safety risk metrics such as robustness, reachability, and uncertainty metrics in DNNs. However, we believe that Quantitative risk assessment techniques are often difficult to apply to deep neural networks (DNNs) due to their complex architectures and a large variety of implementation algorithms. Therefore, a quantitative computation of safety risk metrics of such DNN networks with a high number of layers seems to be practically not feasible. Estimating the safety risks of these networks can be done more effectively by analyzing the performance of the classifier through the metrics of false positives, true positives, false negatives, and true negatives. Performance measurement can give a repetitive way of examining the exactness of a classifier and spotting likely hazards in an economical and immediate fashion.

The DDE process, proposed in the paper [22] offers a systematic V-Model development process solution to ensure the quality and composition of data sets used for ML. This process is compliant with the System Processes in Automotive Engineering (ASPICE) standards, making it easy to integrate into existing development processes and gain acceptance in automotive engineering. The motivation of the authors to propose the DDE process is to address the challenges of developing high-quality machine learning (ML)-based systems in the industry. The authors recognize that the quality of ML-based systems depends on the quality of the data used for training, verification and validation tests. They proposed the DDE process as a systematic and structured approach to ensure that the generated data sets are of high quality and meet the requirements of the operational design domain (ODD). Despite its advantages, the DDE process does not cover other aspects of ML such as model selection or hyperparameter tuning and does not offer a comprehensive approach to functional safety.

The paper [23] provides an overview of available methods for supporting the safety argumentation of machine learning solutions in safety-critical systems in accordance with the ISO 26262 safety standard. It identifies open challenges in this area and argues that the development and certification of safety-critical software using machine learning (ML) is different from traditional approaches. Since ML models are data-driven and automated their design, verification, and validation require new methods. To address this, the authors suggest that ISO 26262-part 6 processes for software development can be applied, and the main focus must be placed on the requirement engineering, development, verification, and validation parts. Regarding requirement engineering, the authors emphasize that the incorporation of available expert knowledge and experience into the formulation of use-case, system, and function requirements should be expressed in specialized key performance indicators (KPIs). To provide a proper safety argumentation for the design and development part, they recommend that domain experts should reason all general design choices -which are not specified in the requirements- related to NN's model design and the training objectives. To enhance the robustness of the design, authors believe that measures such as regularization and training data preparation might be particularly useful.

The authors cited in [24] note that most traditional neural network models are usually designed based on the assumption of minimizing the training error while without considering the impacts of outliers or mislabeled examples, which is critical. Such an oversight may cause great risk in real-world applications since the loss caused by misclassification may be substantial and informative. With regard to this problem, the authors propose a novel training algorithm that takes into account not only the training error but also the risk associated with every single sample. The proposed algorithm consists of two stages: first, the neural network is trained by using any conventional method which tends to minimize the training error. Then, the risk of misclassification for every sample is calculated and used to update the weights of the neural network. The above mentioned measures help in lowering the chances of the network misclassifying samples that are related to higher risk values. The authors evaluate the proposed algorithm on various datasets, showing it outperforms state-of-the-art neural networks in terms of risk. The algorithm also proved to be resistant against various types of noise and other anomalies in the dataset. This is indeed a new kind of methodology that underlines the possible risks of neural network classifiers and may have far-reaching implications for practical applications where the misclassification costs are high.

Researchers at Bosch [25] highlight the critical importance of safety implications of DNNs in automated driving perception systems. They propose a systematic approach to safety engineering, with a particular focus on the safety of intended functionality (SOTIF) as per the ISO 21448 standard [26]. The authors introduce a structured method to categorize safety concerns into four key areas: operational design domain, data preparation, DNN characteristics, and analysis/evaluation. Additionally, they present a list of fourteen safety concerns specific to the application of Deep Learning in automated driving perception. This categorization aids in identifying relevant stakeholders and clarifying responsibilities among the various engineering teams involved in addressing safety issues within AD systems. According to the paper, a "safety concern" refers to functional insufficiencies in DNNs, such as misclassifying traffic signs due to adversarial inputs, which can lead to hazardous behavior at the vehicle level if not properly addressed. The proposed safety concept focuses on demonstrating the absence of unreasonable risk in these four areas. Some of the key safety concerns and challenges they addressed include:

Distributional Shift Over Time—DNNs assume a stationary data distribution, but in real-world operations, this distribution naturally changes over time (e.g., seasonal weather changes, sensor aging). This shift between training and operational data can degrade model performance, posing a safety concern.

Insufficient Labeling Quality - Poor-quality labels in supervised learning can introduce errors into DNNs, reducing performance and generalization. Inaccurate labeling also risks unreliable evaluation and can result in misjudging the model's capabilities, leading to performance issues during real-world use.

DNN Characteristics—The authors note that Deep Neural Networks (DNNs) are universal function approximators, meaning they can learn and fit any function through training. However, their complexity often prevents human experts from fully predicting or interpreting their behavior, which raises safety concerns.

Brittleness—DNNs can exhibit brittleness, meaning small, non-meaningful changes in input can lead to significant changes in output, like misclassifying an object with added noise or contextual changes. This brittleness can affect predictions over time (e.g., video sequences), leading to inconsistent or unstable results, and creating issues for downstream systems like tracking or fusion in autonomous driving.

3. Automative Safety

In the automotive industry, the safety of passengers, other vehicles, and pedestrians, with respect to safe flow is of utmost importance. To ensure the safety of road users and pedestrians, the International Organization for Standardization (ISO) and the Special Interest Group for Automotive Safety have released safety standards ISO 26262 and ISO 21448-SOTIF- [19]. These standards have been in place for a while now, but they are evolving over time based on the ever-changing consumer, and legislative demands of the automotive industry.

3.1. ISO 26262

ISO 26262 is a normative guideline that provides a framework for the entire development process of electrical and/or electronic systems in road vehicles. It outlines specific approaches to identify and assess hazards related to systematic failures as well as random Hardware (HW) malfunctions, and how to mitigate the impact of the risks within the framework of the product lifecycle. This includes the design, implementation, validation, and verification of safety-related systems. ISO 26262 offers a comprehensive approach to the development of safety-related systems, helping engineers to identify and reduce risks throughout the entire development process.

3.2. ISO 21448-SOTIF

The ISO 21448—Road Vehicles—Safety of the Intended Functionality (SOTIF) standard is a newer automotive safety standard and was released in 2019 as an extension of ISO 26262, aiming to address risks that are not covered by the original standard. It provides guidance on how to ensure the safety of road vehicles when they perform their intended functions without faults [26]. The standard focuses on preventing unforeseeable risks that may arise from functional limitations of the intended functions or performance insufficiency of the system or from predictable misuse by people or any environmental influences [27]. Some key aspects of the SOTIF standard are:

- The definition of intended function and malfunction.
- The identification and assessment of SOTIF scenarios and risks.
- The establishment of requirements and validation strategies for SOTIF.
- The documentation and traceability of the SOTIF process.

SOTIF validation strategy considers two types of hazards: known area hazards and unknown area hazards. Known area refers to the set of scenarios where the system's behavior is well-defined and predictable. Unknown hazard area refers to the set of scenarios where the system's behavior is uncertain or unexpected due to unforeseen situations or limitations [28].

For example, a lane keep assist function may work well in a known area where the road markings are clear and visible, but it may fail in an unknown hazard area where the road markings are faded or covered by snow [28] SOTIF validation aims to identify and mitigate potential hazards in both known and unknown areas by applying various methods such as functional specification, functional hazard analysis, risk assessment, verification and validation [29].

3.3. Inadequacy of ISO 26262 and SOTIF Standards for Automated Driving

In recent years, the automotive industry has seen a tremendous increase in the use of advanced technologies such as machine learning (ML) and artificial intelligence (AI) [30]. However, with this increase in technology comes the need to address safety concerns.

The ISO 26262 and 21448 (SOTIF) standards have been widely adopted in the automotive industry for developing safety-critical systems. However, when it comes to the development of Machine Learning (ML) solutions and autonomous driving, these standards are inadequate.

This inadequacy is due to the fact that these standards are designed for deterministic systems and not for stochastic behavior [31]. Furthermore, due to the unpredictable and complex nature of road-driver-vehicle-pedestrian dynamics (state space) vis-à-vis autonomous driving, it is difficult to ensure safety in these systems using the existing safety standards [32]. Thus, it is important to develop safety standards that are specific to ML and autonomous driving.

3.4. ISO PAS 8800

ISO PAS 8800 Road Vehicles—Safety and Artificial Intelligence [20]-drafted in 2022- is a publicly available specification for the safety assurance of automated driving systems, providing compliance guidance on various aspects of safety assurance for automated driving systems. These include safety requirements, safety management, validation, and verification methods. It is intended to be used in combination with existing safety standards, such as ISO 26262 and SOTIF, to ensure the safety and reliability of automated driving systems.

Overall, the combination of these safety standards is essential to ensure the safety and reliability of ML solutions in the automotive industry. They provide a common framework for the development process of ML solutions, as well as guidance on how to ensure safety and reliability in the development of autonomous driving systems. ISO PAS 8800 is a proposed standard for road vehicles that define safety-related properties and risk factors impacting the insufficient performance and malfunctioning behavior of artificial intelligence (AI) within a road vehicle context [20,33].

The motivation for developing ISO PAS 8800 is to address the challenges and gaps that arise from applying AI systems in road vehicles, such as perception, decision-making, learning, adaptation, etc. ISO PAS 8800 aims to establish a common approach for developing and testing AI systems that are used in road vehicles [34]. It will cover topics such as:

- Derivation of Safety requirements for AI systems
- Conducting Safety analysis methods for AI systems
- Verification and validation methods for AI systems
- Safety assurance cases for AI systems
- Safety management processes for AI systems

4. Identifying Safety Risks in Using Machine Learning Solutions

The development of DNN classifiers is associated with potential risks, as depicted in Figure 2. Immediate risk factors and risks due to design variations are marked. It is crucial to emphasize that the careful selection of these design variations is essential to minimize the risk of frequent misclassifications in real-world applications. We will elaborate on these risks in the following sections and propose methods and solutions to mitigate the risk and comply with mandatory requirements of safety standards.

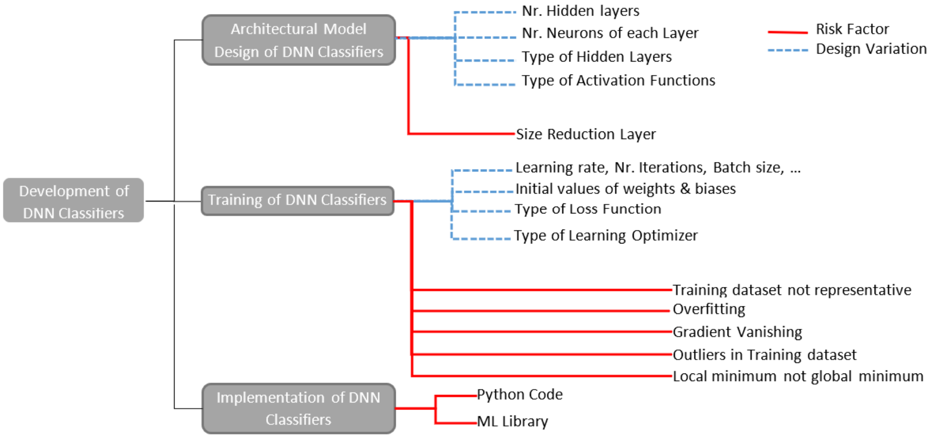
Development of DNN Classifiers
Architectural Model Design of DNN Classifiers
Nr. Hidden layers
Nr. Neurons of each Layer
Type of Hidden Layers
Type of Activation Functions
Size Reduction Layer
Training of DNN Classifiers
Learning rate, Nr. Iterations, Batch size, ...
Initial values of weights & biases
Type of Loss Function
Type of Learning Optimizer
Training dataset not representative
Overfitting
Gradient Vanishing
Outliers in Training dataset
Local minimum not global minimum
Implementation of DNN Classifiers
Python Code
ML Library
Risk Factor
Design Variation

Figure 2. Variations and Risk Factors across Key Steps in DNN Classifier Development.

4.1. Architectural Model Design of Deep Neural Network Classifier

Deep neural networks (DNNs) are a type of machine learning model that can be used to solve a variety of tasks, including object recognition and classification. There are many different types of DNNs, each with its own strengths and weaknesses.

Some of the most common types of DNNs include:

- Convolutional Neural Networks (ConvNets): ConvNets are well-suited for image recognition tasks, such as object classification for autonomous driving. They work by extracting features from images [35] using a series of convolution and pooling layers. Advancements like ResNet [36] have contributed to deeper architectures, enhancing accuracy, while architectures like MobileNet [37] have improved efficiency.
- MLPs (Multi-Layer Perceptrons): MLPs are the most basic form of neural networks. They consist of input layers, one or more hidden layers with densely connected neurons, and an output layer. MLPs are used for a wide range of tasks, including regression and classification, and have been enhanced with innovations in training techniques like Batch Normalization [38] and ReLU activations [39] but they don’t have built-in mechanisms for handling grid-like data like images.
- Transformers: Transformers, originally designed for natural language processing [40], have been adapted for image recognition tasks through advancements such as Vision Transformers [41], enabling them to excel in image tasks.

The network design and architecture should be carefully chosen so that the model does not overfit or underfit the data. It is critical to note that, as none of the approaches offers an ideal solution, DNN architects have to trade-off between the robustness and accuracy of their model and the implementation overheads. Therefore, the selection of the most appropriate design and architecture largely depends on the specific application and the resources available.

In the context of autonomous driving, ConvNets, with their established success in image classification, remain a robust choice. An example of a ConvNets classifier consists of five functional layers: the input, the hidden layer, the activation function, and the output layer as shown in Figure 3.

An example of a neural network classifier consists of five functional layers: the input, the hidden layer, the activation function, and the output layer is shown in Figure 3. A DNN classifier may also utilize a size reduction layer, max pooling, and other elements.

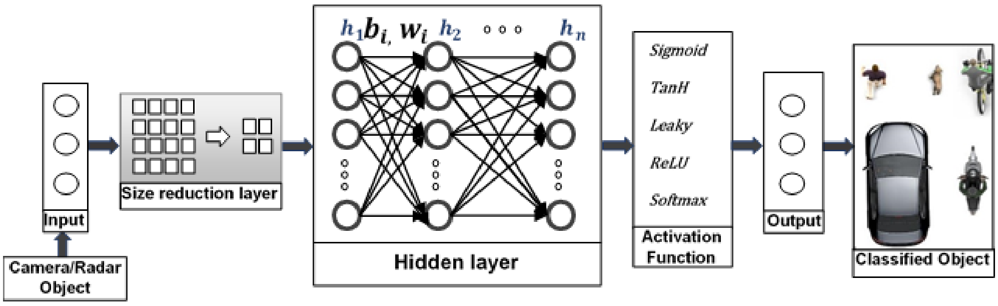

Figure 3. Architectural components of a Deep Neural Network Classifiers.

In a practical implementation for ADAS ECUs, such as Radar ECU or Video ECUs, the data acquisition process gathers image data from a camera or extracts object information (distance, azimuth angle, and radial velocity) from a radar ECU. After processing raw data and conducting related digital signal processing, the software's perception algorithms further refine the acquired information, preparing it for the subsequent classification task.

The network design and architecture should be carefully chosen so that the model does not overfit or underfit the data. It is critical to note that, as none of the approaches offers an ideal solution, DNN architects have to tradeoff between the robustness and accuracy of their model and the implementation overheads. Therefore, the selection of the most appropriate design and architecture largely depends on the specific application and the resources available. An example of a neural network classifier consists of five functional layers: the input, the hidden layer, the activation function, and the output layer. A DNN classifier may also utilize a size reduction layer, max pooling, and other elements.

The input layer in an ADAS system is responsible for taking in sensory data from cameras, Lidar, or radar subsystems [42]. In the case of a camera image, the input layer consists of pixels that are converted into grayscale values. The input layer can also be used to process radar objects, such as obstacles or other vehicles. In this case, the input layer takes the raw data from the radar and performs normalization and feature extraction (the pre-processing steps).

The design decisions of DNN architect at this stage include choosing an appropriate image resolution, color space, and data format. For radar data, the design decisions include choosing the appropriate data format and pre-processing techniques. If the input layer is too small, the model may not be able to feed sufficient information to succeeding layers leading to misclassification. On the other hand, if the input layer is too large, the model may overfit the training data, resulting in poor classification of new data.

The general components of the size reduction layer are: compression, resizing, cropping, and scaling. Compression, in turn, is aimed at reducing the file size by means of removing all extra and unnecessary data, whereas resizing means changing the dimensions of the picture, and the aspect ratio can be changed. Cropping reduces the image's size by means of removing a part of it, and scaling modifies the size of the picture without touching the aspect ratio. While compression techniques can decrease the size of a file, they also degrade accuracy and resolution of the image at the same time. Compression specifically tends to reduce the level of detail and sharpness of an image. Further, resampling and cropping distort or actually blur the resulting image. On the other hand, scaling offers the opportunity to increase the size of an image without sacrificing its accuracy. It is important to consider the accuracy and computational requirements carefully when selecting a size reduction layer.

The hidden layers are responsible for transforming the input data into a more meaningful representation. The neurons in the hidden layers are connected with weights and biases, which determine the strength of the connections between the neurons. Fully connected (FC) and convolutional neural networks (CNN) are among few others the most common types of hidden layers used in deep neural networks. The FC layers are the most basic layers in DNN models and each neuron in the layer is connected to every other neuron in the previous layer. The FC layers are very easy to implement and understand but they can also be used to model complex relationships between input and output. FC layers are prone to overfitting, as they can easily learn redundant features. Additionally, they often require a lot of parameters to model the data, which can lead to longer training times. Convolutional layers, also known as convolutional neural networks (CNNs), are comprised of a series of filters that are applied to an input image or feature map to extract features. Convolutional layers are effective for image processing (also video) tasks, as they can easily extract features from images. They are also much more efficient than FC layers, as they share weights across the input, reducing the number of parameters that need to be tuned. However, convolutional layers are not efficient enough to model relationships between input and output when they are highly complex. This would be a weakness for safety-critical tasks. Additionally, they can require significant amounts of training data to learn useful features.

Activation functions are an essential component of the neural network architecture and allow the neural network to map non-linear input-output relations. There are a variety of activation functions available for use in neural networks, each with its own strengths and weaknesses. ReLU, Leaky ReLU, sigmoid, tanh and softmax are the most widely used activation functions.

ReLU (Rectified Linear Unit) is one of the most widely used activation functions in deep learning. It is a piecewise linear function with a threshold of 0, and a maximum of 1 and can be used in both hidden and output layers of a neural network. ReLU has the advantage of being non-saturating and has a non-zero gradient, which allows for faster training of the network. However, it can suffer from the "dying ReLU" problem where the neuron's output is 0 and can no longer be trained [39].

Leaky ReLU (LReLU) is an extension of ReLU that introduces a negative slope on the left side of the activation function. This helps to alleviate the dying ReLU problem by allowing for a small positive gradient for negative inputs. However, it can suffer from noisy gradients and can be difficult to tune [43].

Sigmoid is another popular activation function and is often used in below binary classification problems. It is a smooth, non-linear, and differentiable function that produces values between 0 and 1. It has the advantage of introducing non-linearity into the network and can be used to approximate an arbitrary function. However, it can suffer from the "vanishing gradient problem" where the gradients become very small, and the network is unable to learn [44].

Tanh (Hyperbolic Tangent) is similar to the sigmoid function with a single difference in that its output ranges between $(-1,1)$. It is smooth and non-linear and has the advantage of allowing for faster training compared to the sigmoid due to its centered output. However, it can suffer from the same vanishing gradient problem as the sigmoid [43].

Architectural Design Solutions

Plausibility checks, Degradation strategy, Fusion:

Architects of DNNs can reduce the risk of wrong classifications in their architecture design by incorporating plausibility checks, degradation, and fusion strategies. Plausibility checks can help detect outliers and false positives, while degradation strategies are needed to ensure that the model still performs safely in the face of various levels of data corruption or perturbations. Fusion of radar and camera sensors, i.e., associating radar and camera information [45] in an Advanced Driver Assistance System (ADAS) decomposes the safety load between available perception sensors and hence can drastically reduce the risk of misclassification. A comprehensive guideline for data processing and fusion methodologies for autonomous driving has been proposed in [46]. By combining the data provided by redundant sensors, the system can take advantage of the complementary information provided by each to reduce the uncertainty of any given classification and better identify objects in the environment.

Software safety analysis, Review, Comprehensive documentation of detailed design decisions:

DNN models are highly complex algorithms, not intuitive for humans, and their output may not be readily interpretable. Software safety analysis ISO 26262 is the recommended method and aims to identify and mitigate potential hazards and risks in software systems. These methods can be applied in a Hazard and Operability (HAZOP) based cause-effect relationship by identifying the root causes of potential safety-critical issues arising from these algorithms and implementing measures to prevent or mitigate their effects. By analyzing the potential causes of software failures and their consequences, safety analysts can develop strategies to improve the safety and reliability of software systems.

Comprehensive documentation of detailed design decisions is a crucial step when designing DNN models for autonomous driving to ensure that the model is reproducible and that its performance can be accurately assessed and tracked. This documentation should include all decisions made during the design process of the model architecture, the dataset used, the hyperparameters, and the training procedure.

4.2. Training of DNN Classifiers

Training of Deep Neural Network classifiers involves a process known as supervised learning, where labeled data are used to train the model. The data is divided into three primary sets: the training set, the validation set, and the test set. This data helps the model learn the relationship between the input features and the target output. The validation set is used during the training process to tune the model's hyperparameters and prevent overfitting. This set provides a "reality check" during training to ensure the model generalises well. The final part of the data is the test set which used only once after the model has been trained and validated. This set provides an unbiased evaluation of the final model, giving us an estimate of how the model would perform on unseen, real-world data. The exact proportions for splitting the data can vary depending on the specific problem and data availability.

Figure 4 shows the two-phase process of training Deep Neural Network (DNN) classifiers. The first phase is the training phase, where a forward and a backward pass are implemented. During the forward pass, input data is fed into the model to generate predictions, while the backward pass involves the backpropagation of errors, which adjusts the weights in the neural network to minimize the difference between the predicted and actual output. Key hyperparameters, such as learning rates, optimizers, and loss functions, need to be specified during this phase. Optimizers help to modify neural network attributes, such as weights and learning rates, to reduce losses. Loss functions measure the discrepancy between the actual and predicted outputs, and learning rates control how much to change the model in response to the estimated error each time the model weights are updated. The second phase involves testing the model on unseen data and evaluating its performance based on the accuracy of its predictions. Here, we provide a summary explanation of the hyperparameters of the DNN learning algorithm.

The Learning rate:

The learning rate is one of the most important indicators of the performance and stability of DNN. It is directly related to the ability of the neural network to learn from the data and converge to an optimal solution. Learning rates determine the size of the steps taken during the optimization process as the model iteratively updates its weights in order to minimize the loss function. A well-chosen learning rate ensures that the model converges efficiently and effectively without overshooting the optimal solution or becoming stuck at local minima [47].

The selection of an appropriate learning rate is essential to achieve the right balance between the speed of convergence and the accuracy of the model. A high learning rate may result in unstable training and potentially poor performance if the model oscillates wildly around the optimal solution. A low learning rate may, in contrast, cause the model to converge very slowly, which consumes considerable computational resources and increases the risk of over-fitting as the model spends more time training on the same data. Therefore, DNN architects must carefully choose and tune the learning rate based on the specific application and available resources. They often employ techniques such as learning rate annealing or adaptive learning rate algorithms to optimize its value throughout the training process [48].

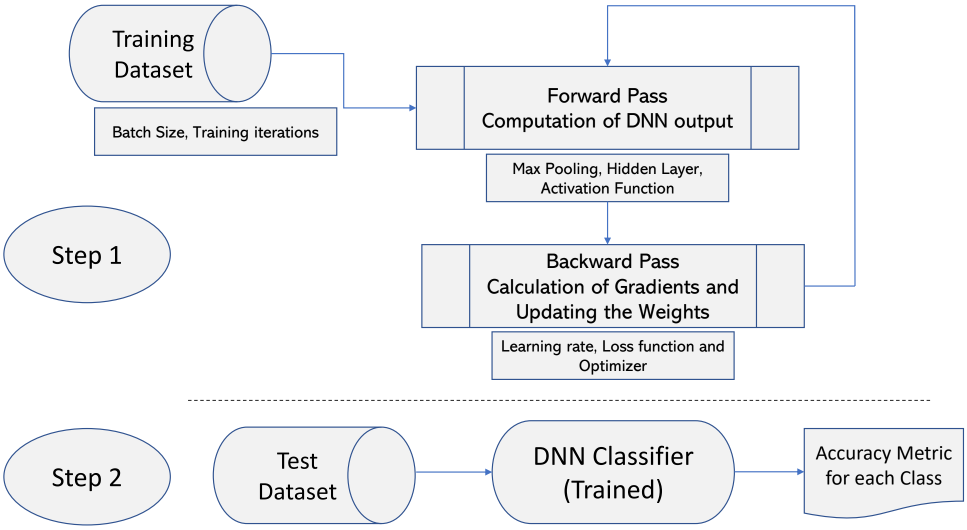
Training Dataset
Batch Size, Training iterations
Forward Pass Computation of DNN output
Max Pooling, Hidden Layer, Activation Function
Step 1
Backward Pass Calculation of Gradients and Updating the Weights
Learning rate, Loss function and Optimizer
Step 2
Test Dataset
DNN Classifier (Trained)
Accuracy Metric for each Class

Figure 4. The training process of a DNN network.

The Optimizer:

Optimizers are responsible for adjusting the weights and biases of the network during the training process and they try to minimize the loss function. Optimizers are capable of fine-tuning these parameters in order to help the DNN learn the best representation for the given data, thereby enabling it to make accurate predictions. Different types of optimizers exist, each with its own strengths and weaknesses. The most commonly used optimizers include Gradient Descent, Stochastic Gradient Descent (SGD), Momentum, AdaGrad, RMSprop, and Adam [49].

Gradient Descent is an optimization algorithm in which the network parameters are adjusted in line with the steepest gradient of the loss function. In spite of its ability to converge to the global minimum, this method is computationally expensive and slow, especially when dealing with large datasets.

Such solutions create long time lags, making them inefficient in autonomous vehicle applications. On the other side, Unlike Gradient Descent, the Stochastic Gradient Descent (SGD) optimization approach is based on a subset of the dataset randomly selected, which speeds up the training process. However, it can be less stable and converge to the global minimum with more fluctuations.

The Momentum optimizer incorporates momentum to accelerate convergence, reduce oscillations, and overcome local minima. It combines the current step's gradient with a fraction of the previous step's gradient to make weight updates smoother and more consistent. The AdaGrad is an adaptive learning rate optimizer that assigns individual learning rates to each parameter. It accelerates convergence in cases where the data is sparse or has varying scales, but it can lead to premature convergence due to its aggressive learning rate decay. The RMSprop addresses the limitations of AdaGrad. It utilizes a moving average of squared gradients to adjust the learning rate, which prevents the aggressive decay of the learning rate and leads to better convergence properties. The Adam optimizer combines elements from both Momentum and RMSprop that maintains separate moving averages for the gradients and squared gradients and it allows for adaptive learning rates and momentum. Adam is known for its fast convergence and robustness to different types of datasets and architectures. The choice of an optimizer is essential for the performance and efficiency of a DNN. Each optimizer has its own set of advantages and disadvantages, and the choice depends on the specific application, dataset, and resources available. If we select an appropriate optimizer, DNN architects can achieve a balance between model accuracy, robustness, and implementation overheads.

The Loss function:

The loss function is the measure of a performance to be optimized within the model training phase. It assigns a number representing how far apart the predicted and true outcomes are. Such measures help in determining the best design and architecture. In addition, it allows DNN architects to fine-tune the models and make good trade-offs among robustness, accuracy, and various implementation overheads.

A loss function can take many forms when it is used in DNNs to address a wide variety of problems and objectives. Some common ones are MSE for regression tasks, Cross-Entropy Loss for classification tasks, and Hinge Loss for support vector machines, and many more. Therefore, careful choice of the appropriate loss function should be made for an application, since it directly influences the learning process of the model and the behavior of the optimization algorithm [50].

The role of the loss function goes beyond simply calculating a model's performance. During the training phase, it acts as a guide for the optimization algorithm, helping it adjust the model's parameters to minimize error. In essence, the optimization process seeks to reduce the loss function's value by locating the model's optimal configuration in the parameter space. This is typically done using gradient-based methods like Stochastic Gradient Descent (SGD) or advanced variants such as Adam and RMSprop [51].

Moreover, the loss function contributes to the prevention of overfitting or underfitting, which are common challenges in DNNs. By incorporating regularization terms, the loss function can penalize complex models that overfit the data or prevent models that are too simplistic and underfit the data. This enables the model to achieve a balance between fitting the training data and generalizing it to new, unseen data.

4.3. Implementation and Integration of DNN Classifiers

Figure 5 illustrates how deep neural network (DNN) models are typically implemented and deployed. Machine-learning developers use open-source machine-learning libraries such as TensorFlow and PyTorch in their code to implement DNN models. Whilst C and C++ are considered to be industry-standard, Python has become increasingly popular for the development of DNN models in automotive applications. To manage the coding complexity, ISO 26262 has several normative orders such as having a specific coding guideline, avoiding dynamic memory allocation, using static analysis tools, and following a consistent coding style. Coding complexity can be measured by metrics such as: cyclomatic complexity, nesting depth, and the number of parameters [52]. Finally, the higher-level aspects of programming are: data structures, concurrency, polymorphism, global variables, and exception handling need to be controlled so that quality code will be delivered. Accordingly, with these sets of rules of coding, a developer will be able to tell if the code is reliable, robust, and efficient. Although machine-learning libraries are used extensively within automotive engineering, one point to be made is that this library is general-purpose and not designed to meet the ISO 26262 criteria.

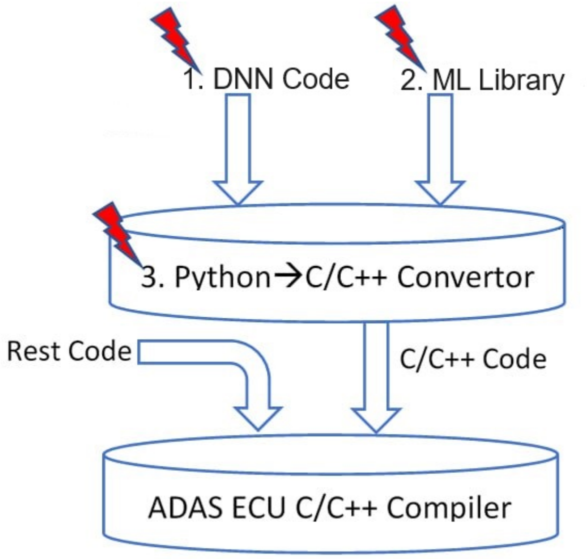

Figure 5. Training phase of a DNN network.

4.4. Verification and Validation Methods for ML-Based Automated Driving Systems

Verification and validation (V&V) are among the most essential approaches in building deep neural networks for autonomous driving applications. It is highly critical in finding inaccuracies in DNN models, flaws in their implementation, or significant performance gaps. Verification ensures the DNN model implemented is correct and meets the design specifications. In contrast, validation establishes confidence in the accuracy and reliability of the predictions from the DNN model. There are many such V&V techniques to determine whether the implementation of DNNs is correct and performing as intended. Such techniques include unit testing, endurance run tests, scenario-based testing, fault injection with white noise on sensor data, adversarial attacks, and benchmarking of False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN) metrics.

Unit Testing:

The unit design testing forms a critical stage for the verification of the functional capability of different units that constitute a DNN model. Through the disintegration of the system into more manageable units, an engineer is then able to carefully test each unit in isolation and verify its functionality against specified requirements. Hardware-in-the-loop simulation provides one efficient approach toward conducting automated, low-cost tests that are reproducible and can highlight possible issues well before the deployed system is used in the real world.

Endurance Run Testing:

This is one of the most common V&V techniques applied to automotive systems. In this type of testing, a DNN model is tested for its ability to process sensor data in a wide range of conditions over a long period of time. Endurance run tests help to validate the robustness of the model and its performance under various conditions.

Scenario-based Testing:

Scenario-based testing focuses on exploring the behavior and functionality of a DNN model when presented with specific scenarios that an autonomous vehicle may encounter in the real world. The model response under these scenarios is analyzed to assess its ability to detect overridable and non-overridable obstacles and take appropriate decisions. This helps to guarantee that the model is able to handle unforeseen circumstances in a safe manner, by accurately detecting and responding to obstacles and other objects in the operational environment (driving state space).

Fault Injection with White Noise (FIWN) on Sensor Data:

FIWN is used to evaluate the performance of a DNN model when it receives noisy or corrupted sensor data. In this type of testing, noise is imposed onto the system and the model robustness is tested to see how sensitive it is to its sensor data quality. This kind of testing helps to identify any weaknesses in the model's ability to accurately process and interpret the data.

Adversarial Attacks:

Adversarial attacks are a type of attack that can be used to fool DNN models and cause them to misclassify data or produce incorrect results. Some examples of Adversarial attacks are: (a) Adversarial Patch Attack involves adding a small patch to an image that causes the DNN model to misclassify the image, (b) Adversarial Perturbation Attack involves adding small perturbations to an image that are not visible to the human eye but cause the DNN model to misclassify the image.

False Positive, False Negative, True Positive, and True Negative benchmarking:

These metrics are used to evaluate the performance of a DNN model when it is tested on known testing data. FP and FN metrics measure the number of incorrect predictions made by the model, while TP and TN metrics measure the number of correct predictions. A confusion matrix is one of the several benchmarking methods that summarize the model's predictions by listing the counts of TPs, TNs, FPs, and FNs. This matrix provides a clear and detailed breakdown of the model's performance. The result must exceed a certain level to ensure safety in autonomous driving systems. In 2019, some car manufacturers and Tier I suppliers released the white paper, SAFETY FIRST FOR AUTOMATED DRIVING [53], which inputs an in-depth analysis of the verification and validation techniques for SAE L3 and L4 automated driving from a practical perspective. They demonstrate the positive risk balance of automated driving solutions compared to the average human driving performance and also provide guidance for potential methods and considerations in the V&V of Level 3 and 4 automated driving systems. However, it is not intended to serve as a final statement or minimum or maximum guideline or standard for automated driving systems.

5. Conclusions and Future Work

An analysis of the safety risks associated with deploying DNN classifiers as the dominant method for guiding autonomous road vehicles was conducted. A range of risks observed in experiments within one of the world's most advanced research centers for autonomous vehicle solution provision revealed a range of safety risks in the design, training, implementation, and deployment of DNN classifiers. Addressing these risks is essential to ensure the safety of AI solutions. To mitigate these risks, we proposed a number of AI measures and their mapping with safety standards, summarized in Table A1. In the near future, we will deploy, test, and validate the improvements we have made in the DNN models and report those findings in a follow-up paper. We hope that autonomous road vehicle solution providers and vehicle manufacturers be able to re-evaluate and upgrade the design process and the existing software applications to meet the ISO 8800 Standard based on the highlighted shortcomings and the future, suggestions we will provide for the DNN adjustment.

Author Contributions: Conceptualization, F.M. and A.M.; methodology, F.M. and S.D.; validation, F.M. and A.M.; investigation, F.M. and S.D.; writing—original draft preparation, F.M. and S.D. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Mapping of Proposed Methods to Automotive Safety Standards ISO 26262 and PAS 8800 Normative Demands.

III.A ARCHITECTURAL MODEL DESIGN	
Selecting suitable AI technology DNN model, Activation function, etc.	ISO/AWI PAS 8800:2023 General requirements 7.4.1–7.4.10
Function degradation Plausibility checks	ISO/AWI PAS 8800:2023 12.5 Measures to ensure the safety of the AI system during operation
System redundancy and fusion strategies	ISO/AWI PAS 8800:2023 7.6.1 Measures for Architectural Redundancy
Safety Analysis	ISO/AWI PAS 8800:2023 10.4 Safety analysis of the AI system
Comprehensive Review	ISO/AWI PAS 8800:2023 11.5 Structuring Assurance Arguments for AI Systems
III.B TRAINING OF DNN CLASSIFIERS	
Hyperparameter tuning	ISO/AWI PAS 8800:2023 7.6.4 Training Safety Measures
Dataset Safety Analysis	ISO/AWI PAS 8800:2023 8.4.3 Dataset Safety Analysis
Adversarial attack testing	ISO/AWI PAS 8800:2023 7.6.4.2 Robust Learning
III.C IMPLEMENTATION AND INTEGRATION OF DNN ALGORITHM CLASSIFIERS	
Qualification of ML libraries	ISO 26262-8:2018 Software tool qualification report
Reinforcement of low complexity Coding guideline	ISO 26262-6:2018 Table 1—modeling and coding guidelines
III.D. VERIFICATION AND VALIDATION METHODS FOR ML-BASED AUTOMATED DRIVING SYSTEMS	
Static code analysis Fault injection test Unit/ scenario-based/endurance testing	ISO 26262-6:2018 Table 10—Methods for verification of software integration
False Negative/Positive benchmarking	ISO/AWI PAS 8800:2023 9.5.5.1 Performance Evaluation Methods

References

1. Shaout, A.; Colella, D.; Awad, S. Advanced Driver Assistance Systems—Past, present and future. In Proceedings of the Seventh International Computer Engineering Conference (ICENCO'2011), Cairo, Egypt, 27–28 December 2011; pp. 72–82. [CrossRef]
2. Society of Automotive Engineers. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *Sae Int.* **2018**, *4970*, 1–5.
3. Research Insights Automotive. 2030. IBM. Available online: https://www.ibm.com/downloads/cas/NWDQPK5B (accessed on 23 September 2024).
4. Belmonte, F.J.; Martín, S.; Sancristobal, E.; Ruipérez-Valiente, J.A.; Castro, M. Overview of Embedded Systems to Build Reliable and Safe ADAS and AD Systems. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 239–250. [CrossRef]
5. Lee, C.W.; Nayeer, N.; Garcia, D.E.; Agrawal, A.; Liu, B. Identifying the Operational Design Domain for an Automated Driving System through Assessed Risk. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1317–1322. [CrossRef]
6. Aurangzeb, S.; Aleem, M.; Khan, M.T.; Anwar, H.; Siddique, M.S. Cybersecurity for autonomous vehicles against malware attacks in smart-cities. *Clust. Comput.* **2024**, *27*, 3363–3378. [CrossRef]

7. Durlik, I.; Miller, T.; Kostecka, E.; Zwierzewicz, Z.; Łobodzińska, A. Cybersecurity in Autonomous Vehicles—Are We Ready for the Challenge? *Electronics* **2024**, *13*, 2654. [CrossRef]
8. Vouros, G.A. Explainable Deep Reinforcement Learning: State of the Art and Challenges. *Acm Comput. Surv.* **2022**, *55*, 1–39. [CrossRef]
9. Baur, L.; Ditschuneit, K.; Schambach, M.; Kaymakci, C.; Wollmann, T.; Sauer, A. Explainability and Interpretability in Electric Load Forecasting Using Machine Learning Techniques—A Review. *Energy AI* **2024**, *16*, 100358. [CrossRef]
10. Nordhoff, S. Resistance towards autonomous vehicles (AVs). *Transp. Res. Interdiscip. Perspect.* **2024**, *26*, 101117. [CrossRef]
11. Judson, S.; Elacqua, M.; Cano, F.; Antonopoulos, T.; Könighofer, B.; Shapiro, S.J.; Piskac, R. Soid: A Tool for Legal Accountability for Automated Decision Making. In *International Conference on Computer Aided Verification*; Springer: Cham, Switzerland, 2024; pp. 233–246.
12. Burton, S.; Habli, I.; Lawton, T.; McDermid, J.; Morgan, P.; Porter, Z. Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective. *Artif. Intell.* **2020**, *279*, 103201. [CrossRef]
13. Burns, C.; Izmailov, P.; Kirchner, J.H.; Baker, B.; Gao, L.; Aschenbrenner, L.; Chen, Y.; Ecoffet, A.; Joglekar, M.; Leike, J.; et al. Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F., Eds.; PMLR: New York, NY, USA, 2024; Volume 235, pp. 4971–5012.
14. Saunders, W.; Yeh, C.; Wu, J.; Bills, S.; Ouyang, L.; Ward, J.; Leike, J. Self-critiquing models for assisting human evaluators. *arXiv* **2022**, arXiv:2206.05802. [CrossRef]
15. Song, Z.; Liu, L.; Jia, F.; Luo, Y.; Jia, C.; Zhang, G.; Yang, L.; Wang, L. Robustness-Aware 3D Object Detection in Autonomous Driving: A Review and Outlook. In Proceedings of the IEEE Transactions on Intelligent Transportation Systems, Genova, Italy, 24–27 September 2024; pp. 1–30. [CrossRef]
16. Yang, S.; Huang, Y.; Li, L.; Feng, S.; Na, X.; Chen, H.; Khajepour, A. How to Guarantee Driving Safety for Autonomous Vehicles in a Real-World Environment: A Perspective on Self-Evolution Mechanisms. *IEEE Intell. Transp. Syst. Mag.* **2024**, *16*, 41–54. [CrossRef]
17. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5149–5169. [CrossRef] [PubMed]
18. Adnan Yusuf, S.; Khan, A.; Souissi, R. Vehicle-to-everything (V2X) in the autonomous vehicles domain—A technical review of communication, sensor, and AI technologies for road user safety. *Transp. Res. Interdiscip. Perspect.* **2024**, *23*, 100980. [CrossRef]
19. International Organization for Standardization (ISO). ISO 26262 Road vehicles—Functional Safety. Available online: https://www.iso.org/standard/68388.html (accessed on 23 September 2024).
20. ISO/AWI PAS 8800—Road Vehicles—Safety and Artificial Intelligence. Available online: https://www.iso.org/standard/83303.html (accessed on 23 September 2024).
21. Xu, P.; Ruan, W.; Huang, X. Towards the Quantification of Safety Risks in Deep Neural Networks. *arXiv* **2020**, arXiv:2009.06114.
22. Zhang, R.; Albrecht, A.; Kausch, J.; Putzer, H.J.; Geipel, T.; Halady, P. DDE process: A requirements engineering approach for machine learning in automated driving. In Proceedings of the IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021; pp. 269–279. [CrossRef]
23. Schwalbe, G.; Schels, M. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. In Proceedings of the 1st European Congress on Embedded Real Time Software and Systems (ERTS 22), Toulouse, France, 11–12 June 2022.
24. Santana, M.A.; Calinescu, R.; Paterson, C. Mitigating Risk in Neural Network Classifiers. In Proceedings of the 48th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA), Gran Canaria, Spain, 31 August–2 September 2022.
25. Abrecht, S.; Hirsch, A.; Raafatnia, S.; Woehrle, M. Deep Learning Safety Concerns in Automated Driving Perception. In *IEEE Transactions on Intelligent Vehicles*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–12. [CrossRef]
26. ISO-ISO/PAS 21448:2022-Road Vehicles—Safety of the Intended Functionality (SOTIF). Available online: https://www.iso.org/standard/77490.html (accessed on 23 September 2024).
27. SOTIF—A New Challenge for Functional Testing | SpringerLink. Available online: https://link.springer.com/article/10.1007/s38314-020-0257-4 (accessed on 23 September 2024).
28. Xu, S.; Ding, H.; Du, A.; Chu, C.; Han, Y.; Li, H.; Zhu, Z. A Review of SOTIF Research for Human-machine Driving Mode Switch of Intelligent Vehicles. In Proceedings of the 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 28–30 October 2022; pp. 1–6. [CrossRef]
29. Putting Safety of Intended Functionality SOTIF into Practice. Available online: https://www.sae.org/publications/technical-papers/content/2021-01-0196 (accessed on 23 September 2024).
30. Borrego-Carazo, J.; Castells-Rufas, D.; Biempica, E.; Carrabina, J. Resource-Constrained Machine Learning for ADAS: A Systematic Review. *IEEE Access* **2020**, *8*, 40573–40598. [CrossRef]
31. Koopman, P.; Wagner, M. Challenges in Autonomous Vehicle Testing and Validation. *SAE Int. J. Trans. Saf.* **2016**, *4*, 15–24. [CrossRef]
32. Henriksson, J.; Borg, M.; Englund, C. Automotive Safety and Machine Learning: Initial Results from a Study on How to Adapt the ISO 26262 Safety Standard. In Proceedings of the IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), Gothenburg, Sweden, 28 May 2018; pp. 47–49.

33. (ISO) ISO PAS 8800 Road Vehicles—Safety and Artificial Intelligence. Available online: https://unece.org/transport/documents/2021/09/informal-documents/iso-iso-pas-8800-road-vehicles-safety-and-artificial (accessed on 23 September 2024).
34. Autonomes Fahren—Auf der sicheren Seite—DE/Safe Intelligence. Available online: https://safe-intelligence.fraunhofer.de/artikel/autonomes-fahren-auf-der-sicheren-seite (accessed on 23 September).
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. *Gradient-Based Learning Applied to Document Recognition*; IEEE: Piscataway, NJ, USA, 1998; pp. 2278–2324.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Howard, A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
38. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), Lille, France, 6–11 July 2015.
39. Nair, Y.; Hinton, G. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
40. Vaswani, A. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
41. Dosovitskiy, A. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
42. González-Saavedra, J.F.; Figueroa, M.; Céspedes, S.; Montejo-Sánchez, S. Survey of Cooperative Advanced Driver Assistance Systems: From a Holistic and Systemic Vision. *Sensors* **2022**, *22*, 3040. [CrossRef]
43. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier reading below neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
44. Hinton, G.E. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 599–619.
45. Dong, X.and Zhuang, B.; Mao, Y.; Liu, L. Radar Camera Fusion via Representation Learning in Autonomous Driving. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Nashville, TN, USA, 20–25 June 2021; pp. 1672–1681.
46. Chen, Z.; Li, Z.; Sun, Y. Radar-Camera Fusion for Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review. *arXiv* **2021**, arXiv:2304.10410.
47. Xue, M.; Li, J.; Luo, Q. Toward Optimal Learning Rate Schedule in Scene Classification Network. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
48. Iiduka, H. Appropriate Learning Rates of Adaptive Learning Rate Optimization Algorithms for Training Deep Neural Networks. *IEEE Trans. Cybern.* **2022**, *52*, 13250–13261. [CrossRef] [PubMed]
49. Kaviani, S.; Sohn, I. Application of complex systems topologies in artificial neural networks optimization: An overview. *Expert Syst. Appl.* **2022**, *180*, 115073. [CrossRef]
50. Li, L.; Doroslovački, M.; Loew, M.H. Approximating the Gradient of Cross-Entropy Loss Function. *IEEE Access* **2020**, *8*, 111626–111635. [CrossRef]
51. Tian, Y.; Su, D.; Lauria, S.; Liu, X. Recent advances on loss functions in deep learning for computer vision. *Neurocomputing* **2022**, *497*, 129–158. [CrossRef]
52. Mokhov, S.B.; Paquet, J.; Debbabi, M. Assessing the Adherence of an Industrial Autonomous Driving Framework to ISO 26262 Software Guidelines. In Proceedings of the IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Xi'an, China, 22–23 April 2019; pp. 1–10.
53. White Paper. Safety First for Automated Driving (SaFAD). Mercedes-Benz, Aptiv, Audi, Baidu, BMW, Continental, Fiat Chrysler Automobiles, HERE, Infineon, Intel and Volkswagen. Available online: https://www.connectedautomateddriving.eu/wp-content/uploads/2019/09/Safety_First_for_Automated_Driving.pdf (accessed on 23 September 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.