Introspection of DNN-based Perception Functions in Automated Driving Systems: State-of-the-Art and Open Research Challenges

Hakan Yekta Yatbaz, *Student Member, IEEE*, Mehrdad Dianati, *Senior Member, IEEE*, and Roger Woodman

Abstract

Automated driving systems (ADSs) aim to improve the safety, efficiency and comfort of future vehicles. To achieve this, ADSs use sensors to collect raw data from their environment. This data is then processed by a perception subsystem to create semantic knowledge of the world around the vehicle. State-of-the-art ADSs' perception systems often use deep neural networks for object detection and classification, thanks to their superior performance compared to classical computer vision techniques. However, deep neural network-based perception systems are susceptible to errors, e.g., failing to correctly detect other road users such as pedestrians. For a safety-critical system such as ADS, these errors can result in accidents leading to injury or even death to occupants and road users. Introspection of perception systems in ADS refers to detecting such perception errors to avoid system failures and accidents. Such safety mechanisms are crucial for ensuring the trustworthiness of ADSs. Motivated by the growing importance of the subject in the field of autonomous and automated vehicles, this paper provides a comprehensive review of the techniques that have been proposed in the literature as potential solutions for the introspection of perception errors in ADSs. We classify such techniques based on their main focus, e.g., on object detection, classification and localisation problems. Furthermore, this paper discusses the pros and cons of existing methods while identifying the research gaps and potential future research directions.

Index Terms

introspection, automated driving systems, perception errors, safety, deep learning.

I. Introduction

BETWEEN 2019 and 2020, 115,584 road casualties were reported in the UK by the Department of Transport. Overall, 23,529 of these resulted in death or serious injuries [1]. Automated driving systems (ADSs) promise to mitigate these casualties by significantly improving the safety and efficiency of future transport systems, such as passenger cars. To fulfil this promise, an ADS must understand its environment correctly, which is achieved by the perception subsystem in an ADS. This subsystem is responsible for detecting and classifying relevant objects in the environment, e.g., other vehicles or pedestrians. Perception is safety-critical because serious implications can occur in the case of a fault in perception. For example, in 2018, an automated vehicle belonging to Uber was unable to detect and track a pedestrian with a bicycle correctly, which resulted in the pedestrian being killed [2]. Similarly, Tesla's Autopilot deployed on an SUV failed to detect lane markings momentarily, resulting in a crash and the death of its owner [3]. Due to the safety-critical nature of the application, any such system must be designed to be safe and resilient against any fault/error that may occur despite the designers' best efforts. This expectation is also known as the fail-safe property of the system and requires an underlying mechanism to detect such failures and errors, a.k.a. fault detection.

The most promising way of realising the perception systems in ADSs is through the use of deep neural networks (DNN), such as Tesla Autopilot's HydraNet for visual perception [4]. Despite their popularity and performance, DNN-based perception models require training data to learn from, which is unlikely to cover all cases in the operation environment [5], [6]. Also, they have not reached the cognition capacity of a human, which constantly operates, consciously and unconsciously, using a multitude of senses. Hence, they cannot guarantee safety in operation time and require an additional monitoring mechanism for safe operation [7]. Furthermore, detecting faults in such systems is fundamentally different from those in traditional systems. Perception can fail if a similar input is not seen during training, i.e., novel input or out-of-distribution (OOD), while faults in other components, such as sensors, are caused by a hardware malfunction or external conditions such as weather. For this reason, fault-detection in DNN-based systems is often defined by a different term, *introspection*, which is the main focus of this paper.

*This research is supported by the Centre for Doctoral Training (CDT) to Advance the Deployment of Future Mobility Technologies at the University of Warwick.

H.Y. Yatbaz, and R. Woodman are with WMG, University of Warwick, Coventry, CV4 7AL UK e-mail: {hakan.yatbaz, m.dianati, r.woodman}@warwick.ac.uk

M. Dianati holds part-time professorial posts at the School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University of Belfast, Belfast, BT7 1NN, (m.dianati@qub.ac.uk) and WMG at the University of Warwick (m.dianati@warwick.ac.uk).

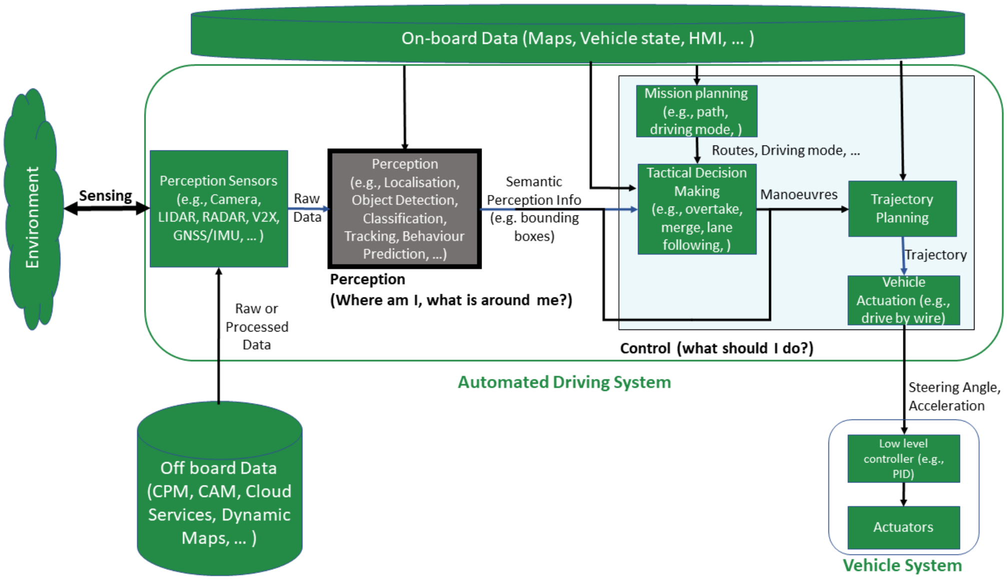
On-board Data (Maps, Vehicle state, HMI, ...)
Environment
Sensing
Perception Sensors (e.g., Camera, LIDAR, RADAR, V2X, GNSS/IMU, ...)
Raw Data
Perception (e.g., Localisation, Object Detection, Classification, Tracking, Behaviour Prediction, ...)
Perception
(Where am I, what is around me?)
Semantic Perception Info (e.g. bounding boxes)
Mission planning (e.g., path, driving mode,)
Routes, Driving mode, ...
Tactical Decision Making (e.g., overtake, merge, lane following,)
Manoeuvres
Trajectory Planning
Trajectory
Vehicle Actuation (e.g., drive by wire)
Control (what should I do?)
Raw or Processed Data
Automated Driving System
Off board Data (CPM, CAM, Cloud Services, Dynamic Maps, ...)
Steering Angle, Acceleration
Low level controller (e.g., PID)
Actuators
Vehicle System

Fig. 1. Logical functional architecture for automated driving systems

Introspection is a mechanism for ADS perception that continuously monitors the system to detect erroneous decisions in run-time in order to provide operational safety. Despite significant advancements in ADS safety through methods such as verification [8], testing [9], and incorporating additional datasets and mechanisms [10], [11], as well as addressing corner cases [12], [13], [14], introspection has received limited attention. However, it is crucial for ensuring the reliable operation of machine learning-based systems, and as such, it warrants further investigation in the field [15]. In the context of ADS, various review studies are available for ADS perception. To illustrate, [10] and [11] review the dataset and methods available for semantic segmentation and object detection for better perception. In [16], [17], the advancements in panoramic imaging for better scene understanding and applications of fish-eye cameras in the ADS domain are reviewed. Introspection, however, is not reviewed as thoroughly as other safety approaches in the literature and is fragmented under different titles and fields. In the literature, the safety and trustworthiness of deep learning models from various perspectives such as verification, testing and interpretability are reviewed in [18]. In [19], methods for finding anomalous samples, i.e., unknown or adversarial samples, for DNN-based systems are summarised. Authors categorised the methods based on the availability of ground truth labels for the anomaly cases. Rahman *et al.* [20] reviewed the available introspection methods for machine learning (ML) for robot perception. They categorise the literature by how introspection is performed and where it operates in the perception pipeline. Similarly, a survey on anomaly detection methods for ADS perception is presented in [21]. They presented a categorisation centred around sensor modalities and also presented available datasets and simulations. Alternatively, the factors, metrics, and datasets are reviewed for their effect on drivability in [22]. The authors highlighted that DNN-based state-of-the-art methods are supervised models that require sufficient training data and provide a comparison among datasets for this purpose.

In recent studies, there has been an interest in reviewing the introspection of deep neural networks (DNNs) for the purpose of safety. However, these studies have tended to focus on introspection methods without considering the specific perception functions that are being monitored. This approach is problematic as different perception functions, such as semantic segmentation and object detection, aim to solve different problems and require different DNN-related operations or functions to improve their performance. For instance, a DNN-based model for semantic segmentation classifies each pixel in an image without taking into account the objects present in the input [23], whereas an object detector aims to locate and categorise each object. Similarly, an object detector such as Faster R-CNN [24] uses a region proposal network structure along with non-maximum suppression to handle multiple objects, while a segmentation model such as DeepLabV3+ [25] uses upsampling and skip connections to preserve spatial information. This difference in the nature of the tasks has led to the development of different techniques to enhance the performance of each model. Hence, it is essential to examine introspection methods in the context of their main focus rather than treating them as a general concept.

This paper provides a more focused and in-depth review of introspection methods suitable for ADSs by analysing them in the context of their target perception functions. It is also important to highlight that this paper includes introspection mechanisms from different domains. These studies were included for their relevance and applicability to ADSs, their importance in the literature and their enhancement or modification to provide ADS solutions. For the perception functions, we have selected three fundamental perception functions utilised in ADS perception. These are classification, object detection, and semantic segmentation. To avoid confusion, it is worth mentioning that the object detection function covered in this paper includes both localisation and classification of the objects, unlike its definition in the ADSs domain, which includes only the localisation of the objects. Introspection of other functions, such as localisation or pose estimation, is also presented in this paper. Since the study focuses on the perception subsystem of ADSs, this article considers the studies where the modular ADS architecture is utilised in studies [26]. Introspection of end-to-end ADS [27], therefore, is not considered in this paper. Similarly, as the focus is on run-time error detection of DNN-based perception functions, detecting hazardous events that can cause perception errors, such as unexpected obstacles, is not considered in the scope of this study. It shall be noted that we also categorise the state-of-the-art introspection methods based on how the introspection system operates. The proposed categorisation in this review is necessary since the design parameters and architectures change significantly between the models for different perception functions, as well as the introspection system design parameters. To the best of our knowledge, this article is the first systematic review of state-of-the-art introspection methods' from the perspective of the perception functions monitored by introspection. Additionally, although all the studies presented in this paper are related to introspection, it is important to emphasise again that some of these studies focus more on the introspection of DNNs regardless of their application domain. However, it is still possible to utilise them for ADS applications.

The rest of this paper is organised as follows. Section II provides an overview of the related background knowledge. Analysis of various introspection methods in the context of the specific perception tasks is given in Section III. Open research challenges and gaps are identified and discussed in Section IV. Finally, concluding remarks are provided in Section V.

II. Background

This section presents an overview of ADSs and the basics of introspection. It first presents a logical functional model for ADS architecture and discusses how the overall system works. Then, it focuses on the integration of introspection into the perception subsystem. Additionally, two critical design choices while developing an introspection mechanism are described in this section.

A. Automated Driving Systems

ADSs perform three main functions called sensing, perception, and control, as presented in Figure 1. In sensing, ADSs collect raw sensor data from their various sensors, such as cameras, LIDARs, and IMUs. It can also obtain data from offboard sources, such as cloud services. This raw sensor data is then fed into the perception subsystem, where it is transformed into meaningful semantic information to answer two questions: "what is the position of the ego vehicle?" and "what are the other objects around the ego vehicle, including their location?". For this purpose, the perception subsystem should: 1) detect the surrounding objects 2) determine their type, i.e., object classification, 3) find its position in the environment using absolute or relative localisation techniques. To achieve perception and answer the relevant questions, ADS perception utilises three basic perception functions presented in Figure 2. The first function is classification, which aims to assign a category to the object in the environment. In semantic segmentation, however, each pixel is classified rather than the given input as a whole. Lastly, in object detection, the aim is to localise and classify the objects. The semantic information extracted is then passed to the control subsystem, which implements key functions such as, mission and path planning, tactical decision-making, trajectory planning and computation of the reference signals for the low-level controllers of the vehicle. The reference signals are fed into the vehicle system through a drive-by-wire system. We don't consider the driver-by-wire or the rest of the vehicle system in this paper, as those are not considered to be conceptually related to the implementation of the ADS.

B. Introspection

Introspection in the human mind refers to paying attention to and examining one's own thoughts. As highlighted in [15], introspection encapsulates the run-time monitoring of both out-of-distribution detection and errors generated due to the model's uncertainty. This differs from traditional fault detection approaches, which are often used in automated systems, for several reasons. Firstly, many intelligent transportation systems are based on deep neural networks (DNNs) that are trained using supervised learning techniques. As a result, their performance in run-time may be limited by the quality and diversity of the training data. In addition, DNN-based perception systems may be subject to errors caused by factors such as malformed input or the inherent limitations of the learning-based model. Furthermore, certain hardware components within intelligent transportation systems may have constraints on their output or behaviour, which can also lead to errors. Therefore, the faults that may occur in DNN-based perception systems are fundamentally different from those in other automotive components in ADSs. The concept of an introspection system is depicted in Figure 3.

In ADSs, introspection models can be developed in different ways by considering two main design choices. One of these choices is the input source for the introspection system. There are four options for input sources: (1) raw sensor data, (2) intermediate outputs from the main perception function, (3) the output of the perception function, or (4) a combination of the first three sources. These input sources can be used to identify potential perception errors in ADSs, such as those that may occur due to the input quality changes between day and night or in adverse weather conditions [28], [29], [30]. Additionally, the presence of transparent objects in the scene can also lead to errors, and can be monitored for error detection [31].

The second design choice is to select how the introspection will operate on the selected input source. Once the input and how introspection will process the input are defined, it is possible to develop various introspection models to detect and generate

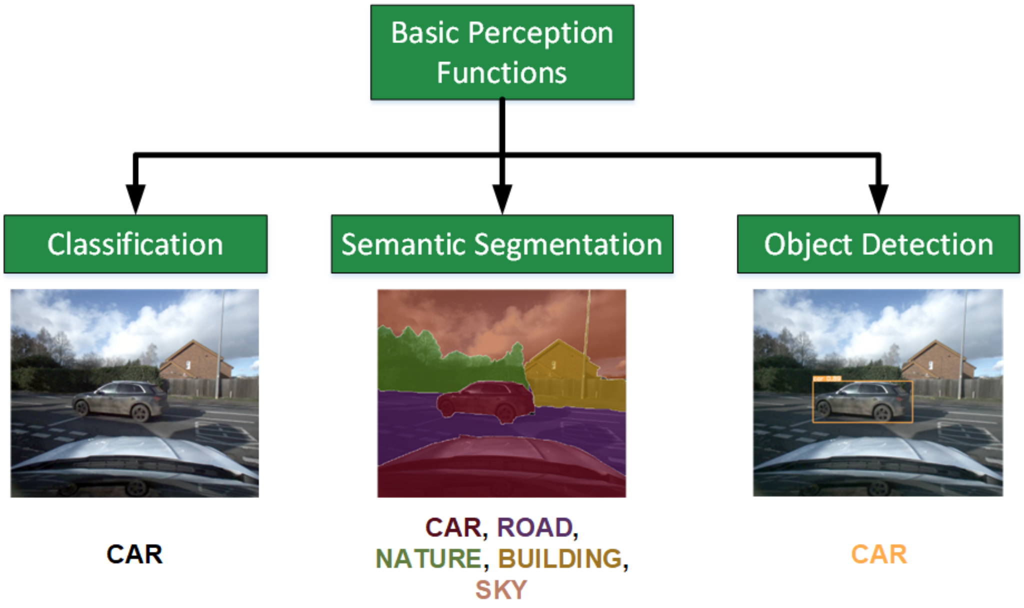

Fig. 2. Examples for each perception function considered in this study. The colours indicate the segmented regions, or predicted bounding box (best viewed in color).

alerts in case of fault. Once perception errors are detected by the introspection system, it can be used in various ways by the system designers. For example, the introspection system can generate an alert which can be used to trigger a minimum risk manoeuvre in Level 4 Automation, or it can be used to hand over the control of the system back to a human operator in Level 3 Automation [32] as presented in Figure 3.

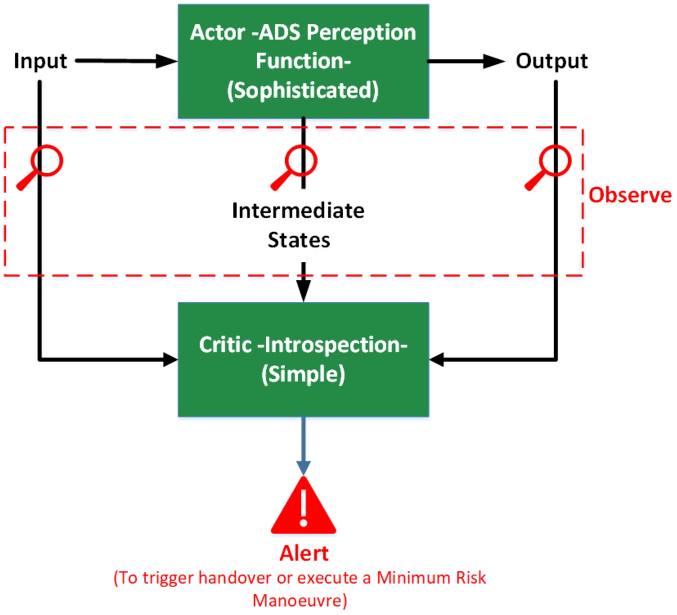
Input
Actor -ADS Perception Function- (Sophisticated)
Output
Observe
Intermediate States
Critic -Introspection- (Simple)
Alert
(To trigger handover or execute a Minimum Risk Manoeuvre)

Fig. 3. Actor-critic architecture for introspecting DNN-based ADS perception: Introspection can monitor perception input, intermediate model outputs, or the final output of the main system (or combinations of them). In case of a fault, it should provide an alert to take further action such as handover or minimum risk manoeuvre. [7]

III. State-of-the-Art of Introspection Methods

In this review article, we presented a two-level classification of the literature using the target perception task and how the introspection system operates. We first consider three perception functions for categorising: classification, object detection, and semantic segmentation. Additionally, we have included introspection of other perception functions, such as localisation.

To analyse the introspection model design trends, studies are further categorised based on how they perform introspection on their corresponding perception function. This two-level categorisation provides a deeper understanding of how introspection is designed and performed for task-specific DNN architectures, such as object detectors. By categorising the literature based on the target perception task, the classification allows for a more targeted and specific analysis of the introspection methods used for each task. In addition, further categorising the studies based on how the introspection system operates allows for a deeper understanding of the different design trends and approaches used for introspection. The proposed categorisation in this study allows for a more detailed analysis of the specific methods and techniques used for introspection in each task and how they are implemented in different DNN architectures, such as object detectors. This can provide insights into the strengths and weaknesses of different introspection methods and how they perform on specific perception functions.

For the second level categorisation, four approaches are used. These approaches are confidence/uncertainty value-based [33], performance metric-based [34], inconsistency-based [35] and past experience-based introspection [36].

Confidence is a value generated by the DNN model, indicating how confident the model is about its decision. Similarly, uncertainty is another way to represent the model's confidence. The more uncertain the model is, the less confidence it has in its decision. These confidence values provide a probabilistic idea for the model's decision. Therefore, it is possible to use them to infer if the decision is good or bad.

Although using confidence/uncertainty values to detect faults appears to be convenient, the value generated may not be realistic [33]. To tackle this issue, researchers focus more on how to provide realistic confidence representations rather than developing introspection. By achieving this, they simply aim to monitor these realistic confidence/uncertainty values to detect errors.

The studies classified under this category can perform introspection in one of three ways. The first way is to calibrate the confidence scores, i.e., the "confidence calibration," [37], provided by the model, so that these scores can represent confidence realistically. This entails adding additional processing to the output without changing the DNN architecture. Alternatively, other studies use an ensemble of models [38] or run the same model multiple times to sample a set of predictions for generating realistic confidence values [39]. Lastly, the model's input, states, or output to estimate realistic confidence or uncertainty values is used as another approach in the literature [33]. Having generated realistic confidence values, such systems can monitor the confidence values using a threshold to generate alert messages. In our review article, studies are put into this category if they use one of the given methods by itself or in combination with another method.

Depending on the target perception tasks, the most commonly used perception models in ADSs are mostly tested using performance metrics such as mean average precision, accuracy, and F1 score. As a result, developing mechanisms that can detect specific patterns indicating a drop in the selected performance metric is another way to look for faults in perception systems. However, because there is no ground truth data available at run-time, performance metrics or drops must be estimated [34], [40], [41], [42], [43]. Such mechanisms can then be integrated into perception to alert the system if a performance drop is detected or the estimated value falls below a certain threshold. This category includes studies that develop an estimator or detection system based on the base system's performance metrics for introspection.

As most humans naturally detect unfamiliar cases around them by recognising the irregularities, looking for inconsistencies in the scene is another possible way to find unreliable decisions. In the scope of ADSs, this requires monitoring multiple perception sensors or algorithms to identify inconsistencies. Consequently, there is an effort to detect fault cases based on inconsistencies, such as when the results of different algorithms, such as object detection and object tracking, don't match [35]. However, this method assumes that there will be a sufficient number of correct decisions to detect unreliable ones, which may not be true all the time.

Similarly, as humans, we have a cumulative knowledge and experience base to reflect on. Inspired by this concept, researchers try to introspect perception systems using the past experiences of the system under similar, or the same conditions [44]. This kind of introspection method commonly introduces a way to store certain features or encodings of the percepts as the prior experience of the system so that, in run-time operation, the decisions or the inputs can be evaluated using this prior knowledge base. The studies introducing a way to encode, store, and query the system's observations for introspection are put under this category in the scope of this review article. Additionally, studies that propose abstraction or encoding over neural network patterns for querying are also considered under this category [45], [46], [47].

A. Classification Task

Classification is one of the fundamental problems that deep learning tackles. DNN-based classification is well established in the literature with various models such as ResNets and datasets. These datasets and architectures made introspection of classification is widely investigated in the literature, which is presented in Table I. It shows that most of the studies in the introspection of classification utilise confidence values. Among these studies, calibration and estimating confidence values are preferred more for classification. In performance metric-based introspection studies, estimating accuracy metric or trying to estimate the hardness of the given input is used to detect faults. Alternatively, reconstructing the input image to calculate a dissimilarity score as an indicator of inconsistency is performed in this category. To generate a past experience base, abstracting neural networks or their patterns are also used. Then such methods can query to introspect the decisions of the classification model.

TABLE I
SUMMARY OF THE INTROSPECTION METHODS FOR CLASSIFICATION TASK

Intropsection Category	Method	Studies	Method Summary	Properties
Confidence / Uncertainty	Confidence Calibration	[37], [48], [49], [50], [51], [52], [53], [54]	Use of confidence outputs, or logits from neural network to process them further.	No change is required for the base network. Easy to deploy.
	Sampling	[55], [38], [39], [56]	Use of a network multiple times or multiple networks for a single input. The aim is to calculate the average confidence with variance to have a better representation.	Statistically sound. No change is needed on the base network. Can be computationally costly.
	Confidence Estimation	[33], [57], [58], [59], [60], [61], [62], [63], [64]	Use of an auxiliary NN, or additional output neuron, to generate, estimate or find drops in confidence/uncertainty values.	Requires change in the network, or increases computational complexity. Can work well with out-of-distribution samples. Older studies use certain functions to generate new values, while recent ones utilise NNs.
Performance Metric	Accuracy Monitoring	[65], [40]	Training DNN as introspection model(s) with output of the perception model, and accuracy of the prediction. Then, use the model to estimate or detect the accuracy drop for each sample.	Easy to deploy, but requires additional training process and increases complexity. Can provide better results when used with MCD.
	Hardness Prediction	[66]	Finding "hard" samples using an auxiliary network, i.e., hardness predictor (HP-Net). A base system should be trained jointly with HP-Net. To introspect, thresholding on HP-Net output is used in inference.	No change in the base network. Since trained jointly, it also improves the base classifier. Introduces complexity.
Inconsistency	Dissimilarity	[59]	Calculating dissimilarity score using the input image and a reconstructed image with generative adversarial network (GAN). To introspect, the score is used as an anomaly score.	Introduces new network architectures, which means additional complexity.
Past Experience	NN Activation Pattern Monitoring	[46], [47], [45], [67]	Encoding & storing the neural network activation patterns, or calculated quality measures for known and unknown classes. Alternatively, [45] defines a region using activation patterns and checks if the activation falls into the region (correct) or not (fault).	Computationally complex. No change is required in the base system. Once established, easy to alter and enhance.

1) Confidence/Uncertainty-based Introspection: Confidence and uncertainty are the common metrics to indicate how confident is the DNN-based model about its decisions. In classification models, such values are represented by the latent output of the model, a vector containing confidence values. Since obtaining these values is trivial, introspection of classification includes a vast number of confidence value-based approaches. These approaches can be categorised under three main categories, which are confidence value calibration, sampling-based approaches and confidence value estimation.

In [48], confidence score processing methods, non-parametrised sigmoid, Platt scaling [49], isotonic regression, and Gaussian processes are tested with different classifiers to see their performance on classifying pedestrians in detected objects. The study aims to compare methods and provide which achieves realistic confidence scores for introspection. Qiu and Miikkulainen [50] proposed a Gaussian process-based confidence calibration method to detect misclassifications of a classifier. In [51], confidence calibration and detecting failures combined, and Mahalanobis distance-based confidence score generation is proposed. They evaluated their proposed solution with both adversarial and out-of-distribution samples to show its efficiency in different problematic cases. Alternatively, in [54], the softmax function is replaced with maximum likelihood and maximum-a-posteriori functions for better probabilistic confidence indication. A simple out-of-distribution detection method, ODIN, is proposed in [52]. The proposed method introduces adversarial cases to improve the model's robustness by adding small perturbations to the input. To calibrate the confidence, it uses temperature scaling [37]. In the last stage, a threshold-based out-of-distribution (OOD) detector decides whether the given image is an OOD sample.

Vyas *et al.* try to use ensembles of neural networks to enhance the detection of the OOD samples [55]. Their method uses an ensemble of K classifiers. To train these K classifiers, they first partition their training dataset into K parts. Then, an OOD sample set and an in-distribution training set are generated by extracting one part of K parts as an OOD set and the remaining K-1 parts as training data. Then, they train K different classifiers on these sets, where each classifier sees a different part of the whole dataset. In inference time, their proposed system calculates an OOD score using softmax vectors from each model to detect faults. Another ensemble-based confidence value prediction is proposed in [38]. In this study, Lakshminarayanan *et al.* used an ensemble of networks and averaged the predicted scores generated by each network to provide better confidence scores. In addition, they suggested that using adversarial training improves the overall system's performance. In adversarial training, perception inputs are augmented with small perturbations, which causes the neural network to fail with high confidence. Similar to the ensemble of neural networks, Gal and Ghahramani [39] proposed a method using a dropout layer, which randomly masks some of the neurons with a given ratio multiple times in inference time to create better confidence and uncertainty indicators. They claim that the dropout operation can be interpreted as a Bayesian approximation of the Gaussian process. Hence, each run with dropout in inference results in a different set of selected neurons, i.e., a different neural network. Their proposed method, Monte Carlo Dropout (MCD), is to run the model M times with dropout rate r and use results to produce better uncertainty and confidence scores. Another well-known confidence-based introspective classification method is presented in [56]. In this study, a new score, Trust Score, is presented, which considers a set of neighbours with a parameter k and a density α to calculate a score using the information about the relative positions of the samples. It is also shown that misclassifications can be found using the Trust Score and a thresholding system.

In a different approach, Corbière *et al.* [33] focused on predicting the confidence of a classifier model by utilising a novel neural network to process extracted features by the convolutional neural network part of the classifier. The novel network is designed and trained for predicting the confidence score, named True Class Probability (TCP), using the extracted features from the classifier. Additionally, TCP affects the behaviour of the main system, i.e., the classifier model, when used instead of the common softmax function, i.e., maximum class probability (MCP). Similar to [33], an additional neural network-based introspection system is developed for both classification and regression tasks in [57]. The proposed model in this work is integrated and trained with the base classification model to provide a binary output of whether to use (accept) or discard (reject) the decision. In [68], authors show that softmax activation loses discriminating information from the logits, i.e., the last layer outputs of the classifier. To avoid this problem, a novel neural network architecture to predict the confidence of the classification is proposed. Mohseni *et al.* propose a self-supervised OOD detection algorithm by adding an auxiliary head to the base classifier [58]. This work introduces a set of reject classes (OOD classes) to the model and performs two-step training to optimise the overall network and OOD detection head. The base classifier is trained with full-supervised learning in the first stage with a training set (in-distribution training set). Then, the auxiliary head is trained using a mixture of OOD training and an in-distribution training set. In the inference time, they take the sum of auxiliary head predictions as the OOD-detection score to determine if the given input is OOD. In [59], a variational auto-encoder-based verifier network is proposed to find anomalies in predictions, i.e., OOD and adversarial samples. Alternatively, Nitsch *et al.* propose a mechanism to find OOD samples without requiring an additional OOD set during training. They propose a Generative Adversarial Network (GAN)-based mechanism to create such samples. Their mechanism also estimates a class-conditioned Gaussian distribution over the network's weights of the bottleneck layer as post-hoc statistics [64]. Ranjbar *et al.* use an additional CNN to map the input images into feature space for estimating von Mises-Fisher Distribution in [62]. Using the distribution, their proposed system first maps the new input into a feature vector and then calculates the likelihood as an indicator of novelty. They evaluated the system on generic image classification and driving scenario-based semantic segmentation data. Alternatively, in [63], a method inspired by dual-process theory for the human mind is proposed for monitoring neural networks. According to this theory, human decision-making includes two systems, one works unconsciously and fast, while the other works slowly and consciously. The authors proposed a similar two-level architecture. The first level focuses on the joint distribution of DNN's input, output and explanation for outputs, while the second level focuses on a broader context. The authors evaluated the first layer in image classification and the second layer in object detection. More specifically, the first layer is a Gaussian model estimated using each class's features. In the second layer, the authors utilise a graph Markov NN to learn the objects and their relations to detect the OOD ones.

Even though the classification task is used as a part of other computer vision tasks, such as object detection, certain introspective classification applications focusing specifically on automated systems are also proposed. In one of the early studies [60], researchers define an introspective capacity for machine learning models where this capacity represents the ability of the model to provide more realistic confidence scores. In other words, the ability to be less confident about the inputs they fail. To assess the capacity, they used normalised entropy and best-versus-second-best heuristic [69]. They compare support vector machines (SVM), Gaussian Process (GPC), and LogistBoost classifiers regarding introspective capacity. For GPC and SVM models, authors also use different kernel functions to see their effect on introspective capacity. Later, they extended their work in [61] with a new model called Informative Vector Machine (IVM) and experimented on commonly used classification frameworks. A simple decision-making mechanism that decides whether to operate or stop based on a loss function is also introduced. In [70], authors extend the work in [61] to propose a model with life-long active learning and forgetting scheme on top of IVM. For active learning, the model requests the label from the human operator during training and tries to minimise the number of questions it asks based on a certain threshold and the calculated performance score. In addition, as the model asks for a label from an operator, it may ask indefinitely and store the information, which will create a storage problem for the ADSs. For this purpose, they introduce a forgetting scheme that bounds the memory requirement for the training set to a fixed size. Despite its suitability of lifelong learning and introspective capacity with the IVM model, this study still requires a human operator to decide whether the given state is a failure based on the model's confidence. Lastly, in [53], Laplace approximation (LA) is proposed for robotic introspection with Bayesian optimisation. The authors show how Bayesian optimisation can mitigate the LA method's under-fitting issue and test some of the well-known deep learning models to show the capability of the proposed method for providing realistic confidence scores.

2) Performance Metric-based Introspection: Due to the extensive literature available on classification models, and their performance, there are various metrics to evaluate the performance of a classifier, such as accuracy, f1-score, recall, and precision. This availability made it possible for introspection to exploit these metrics to detect faults in DNN-based classification. One of the simplest performance indicators for the classifier is the accuracy score, which is the ratio of correctly classified samples to all samples in the given set. This indicator is mainly used for introspection. Alternatively, the "hardness" of the input is also used to detect faults in such models.

In [65], a classifier model is monitored with an additional neural network. This monitoring network is developed in two phases. In the first phase, the model is pre-trained using a dataset related to the target domain (such as the KITTI Dataset [71] for ADSs). In this process, the classifier's last layer output is fed to introspection as input, while the correctness of the prediction is used as the label for introspection. In the second phase, transfer learning on an annotated subset of the target dataset is utilised for training. However, only the last two layers of the monitoring network are updated during this process. Lastly, the final model is deployed to detect the given images' accuracy to find lower-performance cases. To make the monitoring process more robust, MCD [39] is also used with the proposed model. In [40], features and perception inputs are utilised to estimate different performance metrics for object detection and classification. They use a neural network-based model and a gradient boosting (XGBoost [72]) model to estimate the selected performance metrics. They estimate accuracy, F1-score, recall, and precision. The authors also show that their method can be used for model selection, device-server offloading, or dataset shift. Another study emphasises that each classifier and certain inputs have different difficulties in classifying, i.e., hardness [66]. Hence, they propose a model that can learn and estimate the hardness of the selected classifier without explicit supervision. To do so, they alternately train the base classifier and the proposed hardness detector, HP-Net, so that HP-Net can tune itself with the low-performing decisions, and the classifier can improve its decisions based on the provided hardness score. In inference, a hardness score is used to reject decisions for specific samples using thresholding.

3) Inconsistency-based Introspection: Although inconsistencies are not as popular as confidence-based introspection, a deep verifier network is proposed for introspecting image classification in [59]. The proposed method uses a generative adversarial (GAN) structure to reconstruct the input image using the character-wise label embedding and features from the encoder. Using the dissimilarity between the input and reconstructed images, they define an anomaly score for OOD detection.

4) Past Experience-based Introspection: Another method to introspect classification models is to build a knowledge base of prior experience. Such knowledge bases can then be used to check if the model was successful or not in the past. The main approach to building the knowledge base is to check neural network activation patterns and encode them for storing in the system as performance records. Such models, then, query similar scenarios to decide if there is a fault or not.

Cheng *et al.* introduce a monitoring system for the activation pattern of a classifier [46]. They extract patterns from the network and create binary codes with training data. In operation time, the Hamming distance is calculated between the binary code of the classified input and the patterns saved in the system for the assigned class. The prediction is accepted based on the hamming distance and a threshold. They also indicated that the same system could be directly applied to the YOLO detector object detection task. In [47], Khalifa *et al.* introduced a method for introspecting neural networks in safety-critical applications by extending the work in [46]. Their work employs activations of multiple layers to represent certain training inputs to enhance the accuracy of the introspection method. Instead of using all the activations, [45] creates a box abstraction using the activations of the neural network, where "a box" represents the area in feature space for a specific class. Using these boxes, the introspection system only performs membership tests between the incoming input vector and the box region for the assigned class. Alternatively, in [67], authors reconstruct images and compute a quality measure using the difference between the constructed image and the original. They perform this operation for training and storing the values. They claim different domains will have different distributions on these quality measures. In inference time, they use the stored quality measure and compare it against the calculated quality measure for the run-time input.

B. Object Detection Task

Object detection in computer vision refers to the task of localising and classifying object(s) in a given scene. Since it consists of multiple tasks, it is significantly challenging to develop and introspect such perception functions. As presented in Table II, all four different approaches are used for introspecting object detection. Similar to the classification task, sampling, confidence calibration and sampling-based methods are modified for object detection function to be able to introspect with confidence values. In performance metric-based methods, however, studies are split into two categories, focusing on introspecting either the given input(s) or the individual objects in the given input. To check inconsistencies in object detection, studies focus more on different sensor modalities and their results. Alternatively, in past experience-based methods, false negative samples and the performance of the model at the same or visually similar locations are used to build the knowledge base for object detection [81].

TABLE II
SUMMARY OF THE INTROSPECTION METHODS FOR OBJECT DETECTION TASK

Intropsection Category	Method	Studies	Method Summary	Properties
Confidence / Uncertainty	Sampling	[73], [74]	Utilising MCD for object detection.	No change is required on the base network. Easy to deploy.
	Confidence Calibration	[54]	Use of confidence outputs, or logits from neural network to process them further. Maximum likelihood and maximum-a-posteriori are used in [54].	No change is required in the base system. Can work faster since the modified process is within the main pipeline.
	Confidence Estimation	[75], [76], [77], [78], [79], [80], [81]	Generating and estimating confidence values to introspect the model. Examples: Gaussian Mixture Models (GMM) are used in [75], IoU value is estimated in [76], [77] outputs a heatmap for object visibility as confidence.	Introduces complexity to process, and estimate. Once setup, it might be easy to integrate with different models.
Performance Metric	Input-level Introspection	[34], [41], [82], [40]	Extracting and using features from input instance(s) to detect fault cases utilising NN architectures.	Can be fine-tuned with different functions and models.
	Object-level Introspection	[83]	Training a DNN using false-negative objects, i.e., missed objects, of a detector, to develop a false-negative detector.	Can work with any object detector. Requires a false-negative dataset for training. Compared to similar systems, introduces additional complexity, since the predictor works similarly to an object detector.
Inconsistency	Inconsistency Between Methods or Sensors	[35], [84], [85]	Aims to check the output of N similar methods or sensors, such as object detection & tracking.	Utilises existing structure to introspect, which means smaller overhead. Fails if all the methods fail to perform the task.
Past Experience	False-Negative Samples as Experience	[86]	Extracting and storing false-negative samples during data collection (where ground truth is available), and updates the model using experienced knowledge.	Provides constant updates. However, the introspects are not provided in real-time.
	Location-based Experience	[44], [36]	Encodes the performance based on location and checks the model's capability in proximity to the encoded location.	Requires additional storage. Location performance might change based on external conditions, such as weather, construction etc., As the introspection task is querying the knowledge base, and fast decision making.
	Visual Similarity-based Experience	[36]	Encodes visual features to introspect the performance of visually similar cases	More robust than location-based experience. Has similar benefits to the location-based experience.

1) Confidence/Uncertainty-based Introspection: As mentioned, the output of the object detection task is two-fold, the location and the class of the objects. For introspecting the classification part, it is possible to use methods presented in Section III-A. However, evaluating the model's confidence in locating an object is a more complex task since the detector tries to estimate multiple values for localising each object.

For this purpose, a Bayesian object detector, BayesOD, is proposed by Harakeh *et al.* [73]. They use MCD [39] to generate multiple detections and calculate the per-anchor bounding box mean and covariance matrix to represent uncertainty. The common post-processing technique, non-maximum suppression, is also replaced with Bayesian inference for more reliable object detection. Similarly, in [74], Miller *et al.* extend the work in [39] for 2D object detector and process all the predictions generated by MCD. They generated uncertainty values for both label and bounding box predictions. They used them to improve an object detector, SSD [87], by eliminating unreliable predictions in open-set conditions, where the operating environment is not restricted to the categories presented in the training set. In [54], utilising maximum likelihood and maximum-a-posteriori functions rather than softmax and sigmoid is introduced. They evaluated the effect of the functions on the confidence representation of object detection. A new method called GMM-Det [75], which extracts uncertainty of the decisions to establish a find and reject mechanism for open-set faults is proposed. They use class-specific Gaussian Mixture Models to calculate the uncertainty measure for deciding whether to reject the sample. In [76], a post-processing algorithm, MetaDetect, is introduced to provide better uncertainty estimates for object detection. The proposed algorithm uses the output of the object detector and tries to solve two tasks, regression and classification. In the regression task, the value of the intersection over union (IoU) metric is estimated, while in the second one, the correctness of the detection based on IoU is predicted as a binary classification problem. In other words, for classification, they estimate if the IoU between the predicted box and the ground truth is larger than a threshold. They use these estimations as an indicator of the quality of the decision. They also introduced metrics for uncertainty estimation of the predictions along with MCD and showed that the proposed metrics correlate with ground truth IoU. In [77], authors proposed a model to find areas with undetected objects. They indicate that the object detectors may not recognise certain regions/pixels due to external effects such as fog and glare. Their method generates a heatmap like a map to indicate the regions where there is a possibility of missing objects, which can be used as a cost or confidence map. Alternatively, in [78], a spatio-temporal unknown distillation (STUD) mechanism is proposed. This mechanism extracts unknown objects, such as billboards, traffic cones etc. from videos and regularises the model's decision boundary accordingly. In other words, their aim is to identify OOD samples in object detection. In [79], Wilson *et al.* propose another mechanism for OOD detection. Their mechanism leverages activation maps extracted from "OOD sensitive features", output of the detector, and object level features and generates a single vector for learning OOD samples.

Although the majority of the work focuses more on 2D object detection, Feng *et al.* proposes practical mechanisms to identify misdetections by modelling uncertainty [80]. In [88], a feature extraction mechanism is proposed for 3D object detection using LIDAR data. The authors adapted five well-known confidence/uncertainty-based mechanisms and tested them with different datasets. Another work by Cen *et al.* propose a metric learning and unsupervised clustering on the point cloud data for open-set 3D object detection. They utilise uncertainty values depending on Euclidean distance sum and indicated that it is a better score compared to the common softmax probability.

2) Performance Metric-based Introspection: Since the object detection task focuses on identifying and localising multiple objects, different metrics are used to evaluate object detectors. One of the most commonly used metrics is mean average precision (mAP), which indicates the model's overall ability to find objects from different classes by calculating average precision. To determine the quality of each estimated object location and calculate mAP, the intersection over union value is calculated between the estimated and ground truth bounding boxes. The studies in this category try to find mAP drops in the given image(s). However, there is an additional effort to make introspection more granular by introspecting each detected object if there is a missed or incorrectly detected object.

In [34], and [41], object detectors are monitored with neural network-based monitoring systems focusing on detecting performance drops on mAP. In [34], the output of the backbone CNN is used as input to extract features using mean, max, and standard deviation functions with global pooling. The resulting vectors of each function are concatenated and fed into a multi-layer perceptron to decide whether the object detector's mAP is sufficient or not with the given image. In [41], authors extend their method in two ways. First, instead of using a single frame for detecting low-performance situations as in [34], they monitor multiple frames in each iteration. Additionally, they propose a cascaded neural network for extracting features from the backbone CNN of the object detector and deciding whether the performance is dropping. A monitoring system for missed objects, i.e., false negatives, is proposed for traffic sign detection in [82]. The proposed method extracts activation maps from the object detector's backbone and processes them into a one-dimensional vector for false-negative detection. In addition to classification, the method proposed in [40] is tested for object detection using the IoU metric. Yang *et. al.* propose an introspection model to predict false-negative samples for given images. Their method does not use information from the base object detector and tries to extract common features for false-negative samples from the perception input during training [83].

3) Inconsistency-based Introspection: Due to the multi-modality of ADS perception, inconsistency-based failure detection is a good candidate for introspection. In [35], an inconsistency-based fault is defined using 2D object detector's and object tracking algorithm's decisions. Their motivation is the object tracker's capability to find objects when object detection fails or vice versa. They have used stereo, and temporal cues, where the first determines the disparity between two images obtained from left and right cameras, and temporal cues use object tracker and object detector outputs (see Figure 4). In [84], Antonante *et al.* proposed a diagnostic graph which is a directed graph where each vertex represents a processor (RADAR, camera), and each edge represents a consistency test between the vertices. These consistency tests between processors simultaneously or at different times enable the system to identify faults with minimum overhead correctly. They have tested their method for object detection and vehicle localisation. Additionally, in [85], they have extended their idea in [84], and utilise a Graph Neural Network (GNN) for detecting inconsistencies, i.e. faults.

4) Past Experience: Studies in this category aim to store and query the performance of the perception system under similar or challenging conditions. For this purpose, these conditions should be identified first.

Hawke *et al.* introduced an introspection method using past experience-based faults called the experience-based classification mechanism. The method retrains the network with false-negative samples extracted by using scene filters for better detection [86]. Similarly, in [44], a location-specific introspection method is proposed to offer autonomy only when the robot is reliable. They introduced a way to store prior experience, called performance records, where they provide probabilistic performance values for specific locations. Their system assumes the robot works in a restricted environment. Additionally, they introduce a decision-making system to offer or deny autonomy for a robot. In [36], they extend their location-based method with visual similarity-based experience in addition to the performance records in [44].

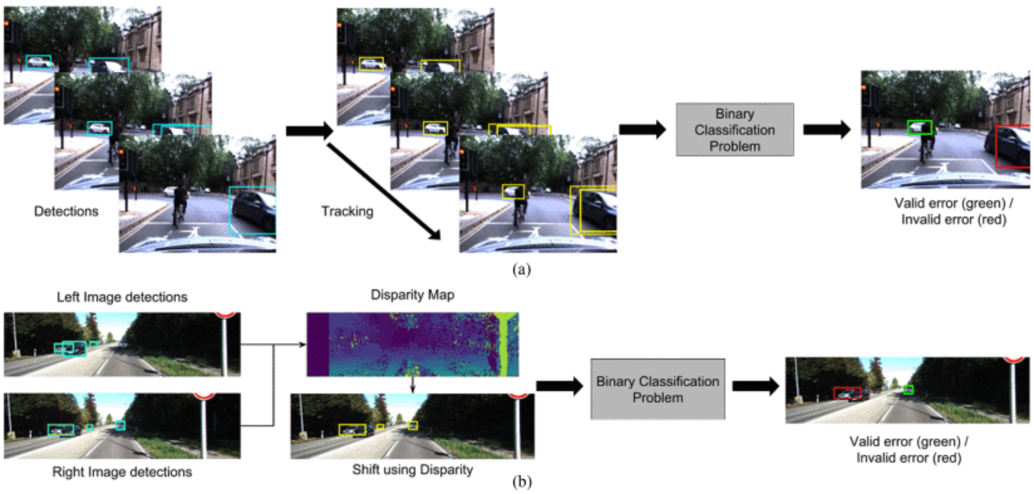
Detections
Tracking
Binary Classification Problem
Valid error (green) / Invalid error (red)
(a)
Left Image detections
Disparity Map
Right Image detections
Shift using Disparity
Binary Classification Problem
Valid error (green) / Invalid error (red)
(b)

Fig. 4. Figure taken from [35]: Top row shows inconsistency checking using object detector and tracker. The bottom row shows inconsistency checking using left and right camera images.

C. Semantic Segmentation Task

TABLE III
SUMMARY OF THE INTROSPECTION METHODS FOR SEMANTIC SEGMENTATION TASK

Introspection Category	Method	Studies	Method Summary	Properties
Confidence / Uncertainty	Sampling	[89]	An additional segmenter is trained to segment images for fault and non-fault cases using MCD. The model aims to detect which pixels are likely to be misclassified.	Introduces the complexity of an additional segmentation model.
	Confidence Estimation	[54], [90], [91], [92], [93], [94]	A CNN is utilised to map input to feature space for estimating von Mises-Fisher Distribution. In testing, the likelihood is used as a novelty score.	Reduces the complexity required for storing all sample vectors. Converts past experience-based introspection to confidence-based.
Performance Metric	Input Validation	[95]	Uses the input directly to determine if the model is capable of segmenting the given input.	Does not depend on the model. Cannot detect model's internal faults.
	Pixel-level Estimation	[43], [42], [96]	Using additional model(s) to label each pixel as fault or non-fault.	Requires additional complex DNN's, which can affect the main perception pipeline.
Inconsistency	Ground Truth Reconstruction	[97], [96], [98]	Reconstructs the input image from segmentation, and calculating similarity measures for introspection.	Assumes if the image is reconstructed sufficiently, it is segmented correctly.
	Inconsistency Between Methods	[99]	Compares segmentation with road segments extracted via LIDAR sensor.	Similar to other inconsistency values, assumes the segments extracted with LIDAR are correct.
Past Experience	Using Past Predictions	[100]	Utilises a classifier trained with prior predictions of the model, assuming that the predictions are ground truth. Then, this classifier is deployed for introspection.	Not used for ADSs, but in another safety-critical area. Has strong assumption on using predictions as ground truth.

Semantic segmentation is a classification task at the pixel-level, where each pixel is categorised. For this reason, the approaches selected to introspect segmentation tasks are similar to the ones in classification. However, the approach to implementing introspection varies depending on changes in the main perception function, specifically in system output and network architecture. Since the output from the semantic segmentation model is an input-sized mask with classes of each pixel, the introspection methods in this category commonly aim to generate an error map where each pixel of the map contains either a probability or a label to indicate fault. The studies on this category are summarised in Table III. The table indicates that although similar methods are used with confidence-based introspection, in the sampling methods, generating the error map is preferred over a single output. Similarly, in performance metric-based methods, error map generation is utilised without utilising confidence/uncertainty measures, along with input validation, which checks if the given input can be sufficiently classified using a selected performance metric. In inconsistency-based methods, reconstructing the input image from the prediction is tested. Similar to the other categories, inconsistency checking between models with different input data is also used for semantic segmentation. For past experience-based introspection, however, the approaches are similar to the other categories.

1) Confidence/Uncertainty-based Introspection: In addition to detecting the objects in the environment, ADSs aim to classify regions into specific categories to detect safe paths or non-drivable areas. Like the other tasks, a fault in classifying a region can make ADSs act irrationally and cause accidents. For introspecting segmentation tasks with confidence, similar approaches introduced in the classification section can be used due to the nature of the problem. In [89], an encoder-decoder architecture is used for segmentation with MCD to provide an uncertainty-based error map as well as a better segmentation output. The authors applied MC Dropout to produce multiple predictions. They use the mean of those predictions as the resulting segmentation mask and the variance as the error map, where higher variance indicates higher uncertainty. Similarly, in [94], two common approaches, reconstruction of the image for extracting disparities, and quantifying uncertainty from the network output, are combined to generate an Error Map-like output for pixel-level error detection. Alternatively, as presented in previous sections, a CNN-based novelty score estimation model is proposed with von Mises-Fisher Distribution in [62]. They estimate the distribution to avoid storing each data point in the system, which changes the original past experience-based introspection into a confidence or quality estimation. They perform pixel-level novelty estimation using Berkeley Deep Drive Dataset (BDD) [101]. Besnier *et al.* propose a mechanism that performs pixel-level error detection and filters the generated error map using instance segmentation [91]. In other words, they focus their pixel-level error detections for each instance (object) for better OOD detection. In [90], GAN architecture is replaced with a new mechanism called normalising flow for a more robust training segmentation mechanism. Authors also propose using JS divergence to identify OOD detections and claim the value they generate can be a competitive replacement for ad-hoc scoring functions. Similarly, in [93], a mechanism to virtually create outliers in training to enhance the model's capability is proposed. An energy score-based OOD detection mechanism is also introduced after the model is trained. Another study [92], proposes a novel re-training approach and a meta-classifier on top of semantic segmentation with softmax entropy thresholding for OOD detection.

2) Performance Metric-based Introspection: Performance metrics for segmentation are similar to the classification as the task is a kind of classification task. However, as the task considers pixel-level classification, the metrics are more focused on the correctly classified areas.

One of the first methods for introspecting segmentation [95] proposes introspecting inputs causing low performance to detect failures. This study explores different tasks, such as semantic segmentation, vanishing point estimation, and image memorability. As the proposed system can work only with a given input, no modification is needed for the base system. They use the base system to predict and label training samples as faults or successes using its performance. In [43], the authors used the segmentation model itself to propose a model that can predict pixel-wise failure or success for segmentation. They use state-of-the-art segmentation architecture DeepLabV3+ [25] as both the segmentation and introspection model. In other words, the proposed model provides a binary output map indicating which pixels are misclassified. Alternatively, in [42], a mechanism utilising the segmentation input, encoding, and output is proposed. The proposed system takes each fault detection input and feeds them into different encoder networks. Then, the generated encodings are combined and decoded for pixel-level error map generation. In [96], the pixel-to-pixel translation conditional GAN model is utilised to reconstruct the original image. A lightweight Siamese network is utilised to estimate per-class IoU estimation using the generated input.

3) Inconsistency-based Introspection: Although it is not popular to utilise inconsistency for semantic segmentation, there are few studies that try to develop inconsistency-based introspection for semantic segmentation.

Haldimann *et al.* proposed a method using a conditional GAN architecture [97] that obtains the semantically segmented output and generates the original image. The generated image is compared with the original input, and a dissimilarity map is generated to indicate where the segmentation model failed. Alternatively, in [99], a pipeline is proposed to validate the segmentation network's result for an autonomous vehicle. Validation is done using a LIDAR sensor for extracting road segments and comparing them with the segmentation model's result. Additionally, the mechanism presented in [96], also utilises the network they propose as a comparison module to identify failures in the classification along with their IoU estimation. Another interesting mechanism is presented in [98]. Authors claim that when the base image is reconstructed, the areas with unknown classes will have poor reconstruction. They utilise this idea to propose a road reconstruction network, and identify unknown areas, i.e. missed detections using the reconstructed image.

4) Past Experience: Similar to inconsistency-based introspection, past experience is also not as popular as others in the literature for introspecting segmentation. However, in [100], the quality of the semantic segmentation is evaluated for medical imaging. Their method introduces a Reverse Classifier, which is trained with predicted segmentation to act as a ground truth. Then evaluating the trained classifier with a reference database where the ground truth is available, the authors indicated that it is possible to check whether the provided prediction is reliable or not.

D. Other Tasks

The perception subsystem can also provide functions other than those presented in Figure 2. Some examples of such functions are ego vehicle localisation, traffic actor pose estimation, and ghost object detection. Although the main scope of this survey

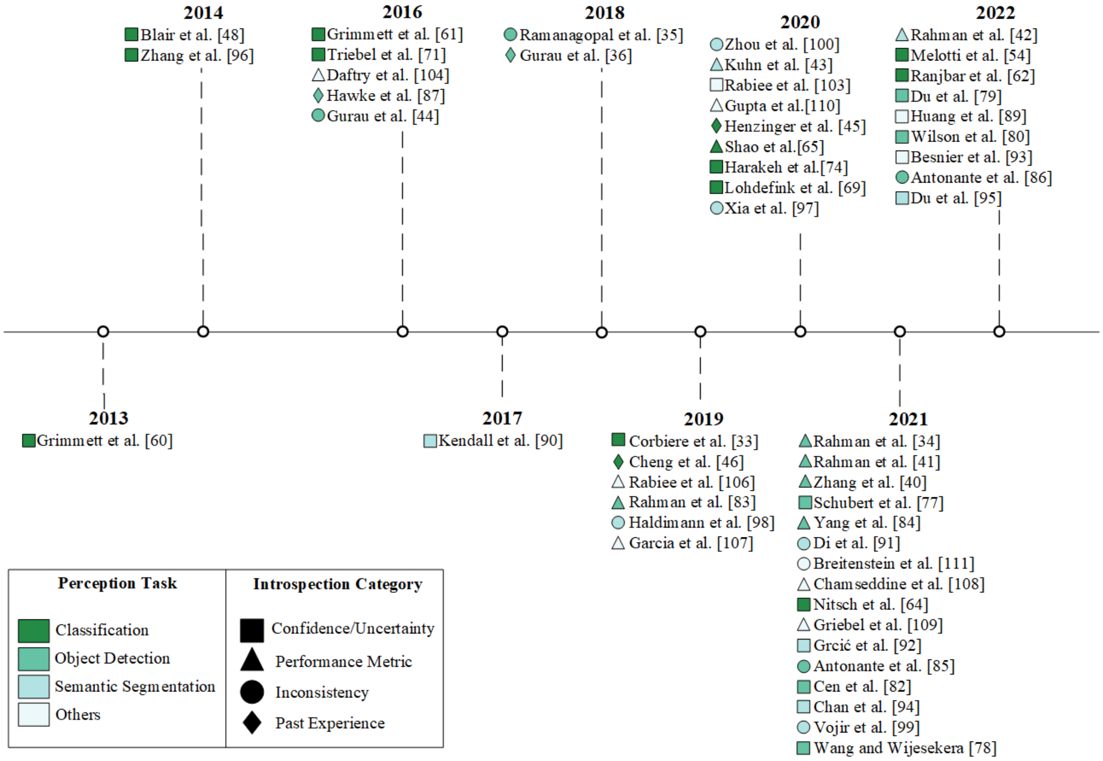

Fig. 5. Timeline of the ADS-related studies: Each study is categorised with a color and a shape to the trends and the two-level categorisation presented in the study. (Best viewed in color)

is the introspection of the basic perception functions presented in Figure 2, we have included the introspection of such related functions, where applicable. It is important to highlight that functions such as radar-based object detection which may focus more on localisation of objects than classification, or instance-segmentation function which combines semantic segmentation with object detection are included in this category as they do not directly fall into the basic functionalities provided. All in all, this section presents the studies do not directly fall into the introspection of basic perception functions provided previously.

One of the important functionalities in perception is localisation, where the vehicle aims to localise itself in the world. To do so, it is possible to utilise global navigation satellite system (GNSS). In [102], an extensive review for introspection of GNSS-based positioning systems is presented. Alternatively, there is significant literature on visual-based localisation. In [103], an introspection function is proposed to generate an uncertainty map using the input image. This map aims to prevent the feature extraction module from extracting unreliable features for the visual simultaneous localisation and mapping (V-SLAM) algorithm. Similarly, in [104], a failure detection mechanism for 2D laser-based SLAM methods is proposed. Their mechanism utilise raw sensory information and extracts features for failure detection. In addition, the localisation errors due to misalignment between varying sensor measurements have been investigated in the literature. In [105] authors investigated the effect of geometric instability on alignment and proposed a novel learning-based approach to predict misalignment. A comparative evaluation of several misaligned point cloud detection methods for point cloud registration problem, where multiple point clouds need to be aligned or merged together is presented in [106]. Similarly, in [107], researchers proposed a novel system to detect alignment errors in point cloud registration. They later extended their work to incorporate RADAR in [108]. Alternatively, in [109], a scene-aware error model is proposed for LIDAR and visual-based odometry and localisation fusion. Alternatively, researchers utilised Markov random fields with fully-connected latent variables, highlighting that the connections enable their model to consider the entire relation, and aim to identify misalignment, and localisation errors due to misalignment in [110]. They later extended and tested their mechanism for 3D LIDAR-based localisation in automated driving systems in [111]. Identifying obstacles and moving objects without focusing on their classes is another functionality provided by perception subsystem. To this end, introspecting vision-based obstacle avoidance is proposed in [112]. The method first splits the input image into patches. Then, it tries to estimate whether the region is true-positive (obstacle detected), false-positive (obstacle but not detected), true-negative (no obstacle), and false-positive (no obstacle but perception indicated there is an obstacle). They create a dataset using sensors with depth information such as LIDAR to label the region as an obstacle. Alternatively, for finding ghost-moving objects on RADAR-based object localisation, an encoder-decoder structure using an occupancy grid map is proposed in [113]. Similarly, a point-based solution is proposed for detecting ghost targets in RADAR-based perception [114]. In [115], a novel grouping algorithm utilising popular DNN-based feature extraction architecture on points set is also proposed for anomaly detection on RADAR-based detection.

Alternatively, creating an experience base is also examined in the scope of instance segmentation where both semantic segmentation and object detection are performed [116]. In this study, the authors generated a histogram with the number of instances per class using ground truth data. In inference, they create the instance histograms for the samples and use Earth Mover's Distance (EMD) to indicate errors.

Lastly, as other traffic actor's movements are crucial for understanding the environment for acting accordingly, a perception subsystem may provide a function to estimate their poses. For such a function, Gupta and Carlone proposed a monitoring system for pedestrian pose estimation using deep neural networks. The proposed method tries to estimate different types of errors to decide whether the predicted poses are incorrect or not [117]. For training the model, they use the input image, 2D projection of the image, extracted 3D Joint information, and camera parameters.

TABLE IV
INTROSPECTION METHOD COMPARISON

	Advantages	**Disadvantages**
Confidence	Availability to use model's confidence scores directly. Low computational cost if not using sampling-based mechanism.	Open to adversarial attacks. Depends on the model's capability to represent confidence.
Performance	Flexible to focus on different aspects of the task.	Dependent on selected metric. Higher computational cost compared to others.
Inconsistency	Uses different sources already available without adding too much overhead. Robust to bias since multiple sources are employed.	Assumes there are enough fault-free recordings to detect inconsistency.
Past Experience	Efficient if the operating environment is limited. Computationally less expensive.	Limits system to a known prior. Requires an efficient methodology for querying past experiences.

E. Discussion

In this section, we summarise the trends observed in selected perception tasks and introspection mechanisms. Additionally, we examine the advantages and disadvantages of the presented introspection methods, focusing on their complexity in automated driving systems. To this end, Figure 5 presents a timeline of studies conducted between 2012 and 2023, specifically developed for ADS or using ADS-related datasets in experimentation. In addition, the advantages and disadvantages of each introspection category are presented in Table IV, taking into account the studies reviewed in this paper.

1) Trends on Introspection of ADS Perception Functions: The timeline presented in this study illustrates that, in parallel with research in the machine learning safety domain, earlier studies focused primarily on classification. However, more recent research has shifted towards introspection of detection and semantic segmentation tasks, reflecting the increased use of these tasks in ADSs. Similarly, it highlights that there is an imbalance between methods, such as confidence-based methods, which are widely used in contrast to past experience-based methods. The reason for this trend is that machine learning safety research tends to focus more on confidence-based mechanisms. However, past experience and inconsistency-based mechanisms are better suited for the ADSs domain, as they are both lightweight and directly feasible for ADSs, given the multi-modality of sensors and perception functions available in the mechanism. Additionally, it is notable that the use of performance metrics is also gaining attention in the field of introspection for ADSs. The reason for this trend may be that many of the DNN-based perception mechanisms are developed and evaluated based on performance metrics, which can provide a benchmark for determining sufficient operation. Furthermore, it is clear that confidence/uncertainty-based mechanisms are a popular choice across various tasks, as they have been widely studied in various domains and are particularly preferred for specific problems such as out-of-distribution detection.

2) Advantages and Disadvantages of Different Introspection Methods: Although preferred and applied differently for changing perception tasks, each method has certain generic benefits and limitations, which are summarised in Table IV. As given in Table IV and Figure 5, confidence/uncertainty-based introspection is already widely used in various domains, such as machine learning safety and robotics, due to the availability of such indicators in existing models. However, it has been shown that confidence values can provide inconsistent results even with small perturbations or in extreme cases. Furthermore, they allow for the development of introspection without modifying the base system and avoid computational overhead if sampling-based methods are not used (MCD [39] or deep ensembles[38]). They are also popular for identifying unknown objects or categories that are not provided in the training set, i.e., out-of-distribution samples, which is a growing research area for ADS introspection. This is an important area of research for introspection in ADSs, as OOD detection can help identify and prevent unsafe situations. However, it is important to note that run-time machine learning safety encompasses multiple challenges, and OOD detection alone may not be sufficient to introspect in the case of misdetected in-distribution samples. Despite ongoing research, these values alone are not yet sufficient for introspection. Therefore, the main focus of current research in this area is on estimating, calibrating, or providing more realistic confidence indicators.

Furthermore, using performance metrics for introspection provides flexibility in monitoring multiple scenarios. As various metrics assess different aspects of systems, these introspection methods can monitor the system from different perspectives. Furthermore, because performance mechanisms are commonly used to assess a model's capability on a given task, they are intuitive to use. However, they require ground truth information to calculate actual metric values. Without ground truth information during operation, these systems can only estimate the metric value, which increases complexity and introduces additional uncertainty. It should also be noted that there is a lack of safety-specific performance metrics for evaluation in the literature.

It is beneficial and practical to consider inconsistencies within the perception system to detect fault cases since it employs already available hardware and modules. This enables the system to find inconsistencies in the case of a sensor fault or set of sensor faults. The main advantage of this method is that it reduces complexity by using existing systems. In addition, it allows for scaling up the systems with more sensors, more algorithms, or with the new advancements in technology without changing a great deal in the introspection. However, such introspection systems assume that a sufficient set of fault-free inputs is available to detect failure, which might not be the case for all operational environments.

In the field of ADSs, using a prior knowledge base to determine the validity of a decision can introduce inflexibility in unfamiliar or dynamic operational environments. While this approach may be effective in limited scenarios, it may not be suitable for more complex or variable conditions. To mitigate this limitation, a system must have a diverse and comprehensive prior knowledge base to account for a wider range of operational scenarios, which is not realistic and feasible. Despite this limitation, such systems can make decisions more quickly than those that rely on past failure records, as they primarily rely on querying and comparing stored data. To optimise performance, these systems must prioritise efficient storage strategies and fast query processes for effective introspection.

All in all, this section highlights the trends and advantages of selected perception tasks and introspection mechanisms in the field of automated driving systems. Studies indicate that introspection mechanisms can vary depending on the specific perception task being performed. This supports the idea that introspection should be approached on a task-by-task basis for improved fault detection. The timeline of reviewed studies, specifically developed for ADS or using ADS-related datasets in experimentation, illustrates that research interest has shifted towards introspection of object detection and semantic segmentation tasks. Additionally, it is clear that the use of performance metrics is gaining attention in the field of introspection for ADSs and that confidence/uncertainty-based mechanisms are a popular choice across various tasks. Additionally, Although each method has its specific benefits and limitations, it is important to note that confidence/uncertainty-based introspection is widely used in various domains due to the availability of such indicators in existing models. However, it is important to note that these values alone are not yet sufficient for introspection in automated driving systems, and more research is needed to develop more robust and accurate introspection methods and to achieve higher levels of safety and autonomy.

IV. Open Research Challenges

The run-time safety of DNN-based mechanisms has been investigated under various keywords, such as open-set recognition, anomaly detection, out-of-distribution detection, and corner case detection. Although studies with these keywords aim to enhance the safety of machine learning models, they may not specifically target ADSs. However, due to the complexity of the environment ADSs operate, introspection specific to the ADS domain is crucial for enabling higher levels of automation. In this section, we highlight the opportunities and open research challenges within the scope of ADSs.

1) **Low computational complexity:** Unlike introspection of other safety-critical systems, such as medical imaging, the computational resources and time to act are significantly lower for ADSs. In other words, the introspection system should not occupy the resources of the main system (see Figure 1), or require too much time to operate. This indicates a vital limitation on ADSs introspection as some of the approaches introduce significant overhead. However, the reviewed work shows that only a few studies mention the complexity of their model. Additionally, there is no mentioned key performance indicator (KPI) or a baseline for the complexity of the methods available. In this regard, investigating the feasibility in terms of time and memory complexity and developing a standard is needed.
2) **Utilisation of diverse algorithms and sensor modalities:** ADSs commonly equipped with various mechanisms and sensor suites to enable them to understand the environment. Although the mentioned inconsistency-based studies try to utilise them in combination, the use of this diversity in the system is underexplored (see Figure 5). However, the diversity available in the system may enable introspection to provide a lightweight, efficient and interpretable solution and hence requires further investigation.
3) **Utilisation of different input sources:** Another diversity available within ADSs is different input sources available in the perception pipeline for introspection, as discussed in Section II. Our review shows that most methods utilise a single part of the perception pipeline. For example, challenging inputs due to the day and night discrepancy [28], [29], unexpected or small road objects [118], [119], adverse weather conditions [30], or corner cases [14], [116] are investigated to identify errors related to the input. However, utilising different parts in the pipeline together has not been explored to enrich the information for introspection.
4) **Cooperative introspection:** In the field of ADSs, the use of a multi-agent environment has been gaining attention as a way to improve perception performance. Cooperation between agents has been shown to be a beneficial approach for

addressing limitations such as occlusion and restricted perception horizon [120]. The idea of cooperation is also another opportunity for researchers and not investigated thoroughly for ADSs introspection.

5) **Multi-class introspection:** Errors in DNN-based systems can be caused by varying factors, as mentioned in previous sections. Although detecting them is within the scope of introspection, it is also important to identify the category of the faults for interpretability and further diagnosis in such systems. Currently, most of the studies propose an introspection model to decide whether there is an error or not. However, there might be different levels or types of where, where some might not require a safety response or be mitigated by other measures. Additionally, in certain situations, perception quality is of paramount importance for other safety-relate mechanisms, particularly in the event of a potential accident, hazardous situation [121], [122] or an irregular driver behaviour [123]. These scenarios warrant further examination, which can be achieved through introspection. One relevant study in this area presents a mechanism, and dataset [124], that focuses on more stabilised segmentation in accidental scenes.
6) **Safety-related evaluation metrics:** To assess the introspection models, well-known classification metrics are used in the literature. However, as introspection is directly related to the safety of the overall system and the actors in the environment, safety-centred metrics and key performance indicators are needed for better introspection.
7) **Benchmark for introspection methods:** Studies in the literature employ publicly available datasets or simulation tools to extract fault cases. However, it may not be possible to reproduce the studies in the literature easily. Thus, the field of introspection in perception systems would greatly benefit from a comprehensive framework and benchmark that take into account both methodology and target perception function. This would allow for a fairer comparison and a deeper understanding of the current state-of-the-art. Recent efforts in this direction include the creation of benchmarks for pixel-wise uncertainty estimation [125], unknown or unseen object segmentation [126], and out-of-distribution detection [127] for semantic segmentation tasks. A similar effort is needed for other DNN-based ADSs perception functions to enable higher levels of autonomy and safety.
8) **Introspection of regression:** Although a significant proportion of perception functions in ADS involve classification, some crucial perception functions, such as object localisation, require regression. This suggests that these regression-oriented functions can be individually analysed and refined. However, despite the potential utilisation of task-agnostic techniques, such as ensemble methods, uncertainty estimation, or the employment of Bayesian neural networks [128], there are certain distinctions between regression and classification that can affect their respective introspection processes. One notable distinction lies in the nature of the outputs: classification results in a discrete class label, whereas regression produces a continuous value. This difference fundamentally alters the approach to error detection. For regression, defining specific boundaries becomes essential to recognise errors. Therefore, exploration of alternative methods for regression introspection could contribute to the introspection of ADS perception functions.
9) **Adaptation to new DNN-architectures:** The landscape of deep neural network (DNN) architectures is continually evolving. Recently, a new model, vision-transformer, has shown exceptional results in image recognition task [129]. Since then, these models have been evolving and continuing to excel in other computer vision tasks [130]. However, although we expect such models to be adapted and utilised in ADS domain, they have not yet been extensively employed. On the other hand, introspection on these models have already started gaining interest. To illustrate, in [131], researchers highlighted that error detection is needed even with the state-of-the-art vision-transformer models for semantic segmentation task in videos. Hence, it is essential for further research to investigate introspection mechanisms of the new perception functions, such as vision-transformer-based models, in ADS domain.

V. Conclusion

Despite the impressive performance of DNN-based perception models on collected datasets, existing perception models are insufficient for safety-critical applications. For ADSs, this insufficiency can result in accidents or risk passengers' safety directly. Researchers from various fields are tackling the problem of ensuring the safety and reliability of autonomous transportation systems by approaching it from three main aspects: validation and verification, robustness and enhancement, and run-time monitoring. Introspection is the mechanism needed to make perception models resilient and find the cases they fail run-time. It aims to improve safety by enabling the system to take appropriate actions when faults are detected, which is essential for level four or higher automation. Although introspection can be developed considering only the DNN architecture, it is important to consider the function that DNNs perform as the task affects how DNN-based systems are designed. This study reviews the existing introspection methods considering which perception functions they are deployed on and how they perform introspection. This survey is expected to serve as a starting point for researchers interested in the introspection of different perception tasks for ADSs. Additionally, although introspection of perception tasks is being investigated, the computational complexity aspect is often overlooked. For real-time operation, the introspection mechanism must be as lightweight as possible to minimise the allocation of resources needed for primary ADS functionalities. Therefore, further research is needed to make introspection more lightweight without losing its performance.

References

[1] U.K Department for Transport, "Reported road casualties great britain, annual report: 2020," 2022. [Online]. Available: https://www.gov.uk/government/statistics/reported-road-casualties-great-britain-annual-report-2020/reported-road-casualties-great-britain-annual-report-2020

[2] U.S National Transportation Safety Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian," www.ntsb.gov/investigations/AccidentReports/Reports/HAR1903.pdf, Nov 2019.

[3] ——, "Collision between a sport utility vehicle operating with partial driving automation and a crash attenuator," www.ntsb.gov/investigations/AccidentReports/Reports/HAR2001.pdf, Feb 2020.

[4] A. Karpathy. (2020) Ai for full-self driving at tesla. Matroid. [Online]. Available: https://youtu.be/hx7BXih7zx8

[5] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, pp. 90–96, 2017.

[6] O. Willers, S. Sudholt, S. Raafatnia, and S. Abrecht, "Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2020, pp. 336–350.

[7] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[8] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, "Software verification and validation of safe autonomous cars: A systematic literature review," *IEEE Access*, vol. 9, pp. 4797–4819, 2021.

[9] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1555–1562.

[10] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[11] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3d lidar data for semantic segmentation? a survey of datasets and methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6063–6081, 2022.

[12] D. Bogdoll, J. Breitenstein, F. Heidecker, M. Bieshaar, B. Sick, T. Fingscheidt, and J. M. Zollner, "Description of corner cases in automated driving: Goals and challenges," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE Computer Society, 2021, pp. 1023–1028.

[13] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, "Systematization of corner cases for visual perception in automated driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1257–1264.

[14] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick, "An application-driven conceptualization of corner cases for perception in highly automated driving," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 644–651.

[15] S. Mohseni, H. Wang, C. Xiao, Z. Yu, Z. Wang, and J. Yadawa, "Taxonomy of machine learning safety: A survey and primer," *ACM Comput. Surv.*, vol. 55, no. 8, dec 2022. [Online]. Available: https://doi.org/10.1145/3551385

[16] S. Gao, K. Yang, H. Shi, K. Wang, and J. Bai, "Review on panoramic imaging and its applications in scene understanding," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–34, 2022.

[17] Y. Qian, M. Yang, and J. M. Dolan, "Survey on fish-eye cameras and their applications in intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 755–22 771, 2022.

[18] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020. [Online]. Available: https://doi.org/10.1016/j.cosrev.2020.100270

[19] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous example detection in deep learning: A survey," *IEEE Access*, vol. 8, pp. 132 330–132 347, 2020.

[20] Q. M. Rahman, P. Corke, and F. Dayoub, "Run-time monitoring of machine learning for robotic perception: A survey of emerging trends," *IEEE Access*, vol. 9, pp. 20 067–20 075, 2021.

[21] D. Bogdoll, M. Nitsche, and J. M. Zöllner, "Anomaly detection in autonomous driving: A survey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2022, pp. 4488–4499.

[22] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3135–3151, 2020.

[23] Q. Guo, Y. Qian, X. Liang, Y. She, D. Li, and J. Liang, "Logic could be learned from images," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 12, pp. 3397–3414, 2021.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[25] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[26] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[27] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2022.

[28] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 769–15 778.

[29] E. Romera, L. M. Bergasa, K. Yang, J. M. Alvarez, and R. Barea, "Bridging the day and night domain gap for semantic segmentation," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1312–1318.

[30] C. Sakaridis, D. Dai, and L. Van Gool, "Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 765–10 775.

[31] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19 173–19 186, 2022.

[32] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." [Online]. Available: https://doi.org/10.4271/j3016_202104

[33] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Per-frame map prediction for continuous performance monitoring of object detection during deployment," in *2021 IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2021, pp. 152–160.

[35] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 2018.

[36] C. Gurău, D. Rao, C. H. Tong, and I. Posner, "Learn from experience: Probabilistic prediction of perception performance to avoid failure," *The International Journal of Robotics Research*, vol. 37, no. 9, pp. 981–995, 2018.

[37] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[38] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 2017-December, 2017, pp. 6403–6414.

[39] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059.

[40] X. Zhang, S. Oymak, and J. Chen, "Post-hoc models for performance estimation of machine learning inference," *arXiv preprint arXiv:2110.02459*, 2021.

[41] Q. M. Rahman, N. Sünderhauf, and F. Dayoub, "Online monitoring of object detection performance during deployment," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4839–4845.

[42] Q. M. Rahman, N. Sünderhauf, P. Corke, and F. Dayoub, "Fsnet: A failure detection framework for semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3030–3037, 2022.

[43] C. B. Kuhn, M. Hofbauer, S. Lee, G. Petrovic, and E. Steinbach, "Introspective failure prediction for semantic image segmentation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, 2020.

[44] C. Gurău, C. H. Tong, and I. Posner, "Fit for purpose? predicting perception performance based on past experience," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 454–464.

[45] T. A. Henzinger, A. Lukina, and C. Schilling, *Outside the box: Abstraction-based monitoring of neural networks*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, vol. 325.

[46] C.-H. Cheng, G. Nührenberg, and H. Yasuoka, "Runtime monitoring neuron activation patterns," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 300–303.

[47] K. Khalifa, M. Safar, and M. W. El-Kharashi, "Verification of neural networks for safety critical applications," in *2020 32nd International Conference on Microelectronics (ICM)*, 2020, pp. 1–4.

[48] C. G. Blair, J. Thompson, and N. M. Robertson, "Introspective classification for pedestrian detection," in *2014 Sensor Signal Processing for Defence (SSPD)*, 2014, pp. 1–5.

[49] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[50] X. Qiu and R. Miikkulainen, "Detecting misclassification errors in neural networks with a gaussian process model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 8017–8027.

[51] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, vol. 2018-December, 2018, pp. 7167–7177.

[52] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[53] M. Humt, J. Lee, and R. Triebel, "Bayesian optimization meets laplace approximation for robotic introspection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020*, 2020.

[54] G. Melotti, C. Premebida, J. J. Bird, D. R. Faria, and N. Gonçalves, "Reducing overconfidence predictions in autonomous driving perception," *IEEE Access*, vol. 10, pp. 54 805–54 821, 2022.

[55] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 560–574.

[56] H. Jiang, B. Kim, M. Y. Guan, and M. R. Gupta, "To trust or not to trust a classifier." in *NeurIPS*, 2018, pp. 5546–5557.

[57] Y. Geifman and R. El-Yaniv, "SelectiveNet: A deep neural network with an integrated reject option," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2151–2159. [Online]. Available: http://proceedings.mlr.press/v97/geifman19a.html

[58] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5216–5223.

[59] T. Che, X. Liu, S. Li, Y. Ge, R. Zhang, C. Xiong, and Y. Bengio, "Deep verifier networks: Verification of deep discriminative models with deep generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 7002–7010.

[60] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing when we don't know: Introspective classification for mission-critical decision making," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4531–4538.

[61] H. Grimmett, R. Triebel, R. Paul, and I. Posner, "Introspective classification for robot perception," *International Journal of Robotics Research*, vol. 35, no. 7, pp. 743–762, 2016.

[62] A. Ranjbar, S. Hornauer, J. Fredriksson, S. Yu, and C.-Y. Chan, "Safety monitoring of neural networks using unsupervised feature learning and novelty estimation," *IEEE Transactions on Intelligent Vehicles*, 2022.

[63] A. Roy, A. Cobb, N. D. Bastian, B. Jalaian, and S. Jha, "Runtime monitoring of deep neural networks using top-down context models inspired by predictive processing and dual process theory." AAAI, 2022.

[64] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena, "Out-of-distribution detection for automotive perception," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2938–2943.

[65] Z. Shao and J. Yang, "Increasing the trustworthiness of deep neural networks via accuracy monitoring," in *Workshop on Artificial Intelligence Safety 2020 (co-located with IJCAI-PRICAI 2020)*, 2020.

[66] P. Wang and N. Vasconcelos, "Towards realistic predictors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[67] J. Lohdefink, J. Fehrling, M. Klingner, F. Huger, P. Schlicht, N. M. Schmidt, and T. Fingscheidt, "Self-supervised domain mismatch estimation for autonomous perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 334–335.

[68] J. Aigrain and M. Detyniecki, "Detecting adversarial examples and other misclassifications in neural networks by introspection," *arXiv preprint arXiv:1905.09186*, 2019.

[69] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2372–2379.

[70] R. Triebel, H. Grimmett, R. Paul, and I. Posner, "Driven learning for driving: How introspection improves semantic mapping," in *Robotics Research*. Springer, 2016, pp. 449–465.

[71] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[72] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[73] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2020.

[74] D. Miller, L. Nicholson, F. Dayoub, and N. Sunderhauf, "Dropout sampling for robust object detection in open-set conditions," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2018, pp. 3243–3249.

[75] D. Miller, N. Sünderhauf, M. Milford, and F. Dayoub, "Uncertainty for identifying open-set errors in visual object detection," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 215–222, 2021.

[76] M. Schubert, K. Kahl, and M. Rottmann, "Metadetect: Uncertainty quantification and prediction quality estimates for object detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–10.

[77] Y. Wang. and D. Wijesekera., "Pixel invisibility: Detect object unseen in color domain," in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS,*, INSTICC. SciTePress, 2021, pp. 201–210.

[78] X. Du, X. Wang, G. Gozum, and Y. Li, "Unknown-aware object detection: Learning what you don't know from videos in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 678–13 688.

[79] S. Wilson, T. Fischer, F. Dayoub, D. Miller, and N. Sunderhauf, "Safe: Sensitivity-aware features for out-of-distribution object detection," 2022.

[80] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 3266–3273.

[81] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu, "Open-set 3d object detection," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 869–878.

[82] Q. M. Rahman, N. Sunderhauf, and F. Dayoub, "Did you miss the sign? a false negative alarm system for traffic sign detectors," in *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 3748–3753.

[83] Q. Yang, H. Chen, Z. Chen, and J. Su, "Introspective false negative prediction for black-box object detectors in autonomous driving," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2819

[84] P. Antonante, D. I. Spivak, and L. Carlone, "Monitoring and diagnosability of perception systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 168–175.

[85] P. Antonante, H. Nilsen, and L. Carlone, "Monitoring of perception systems: Deterministic, probabilistic, and learning-based fault detection and identification," *arXiv preprint arXiv:2205.10906*, 2022.

[86] J. Hawke, C. Gurău, C. H. Tong, and I. Posner, "Wrong today, right tomorrow: Experience-based classification for robot perception," in *Field and Service Robotics*. Springer, 2016, pp. 173–186.

[87] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[88] C. Huang, V. D. Nguyen, V. Abdelzad, C. G. Mannes, L. Rowe, B. Therien, R. Salay, and K. Czarnecki, "Out-of-distribution detection for lidar-based 3d object detection," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 4265–4271.

[89] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *British Machine Vision Conference 2017, BMVC 2017*, 2017.

[90] M. Grcić, P. Bevandić, and S. Šegvić, "Dense anomaly detection by robust learning on synthetic negative data," *arXiv preprint arXiv:2112.12833*, 2021.

[91] V. Besnier, A. Bursuc, D. Picard, and A. Briot, "Instance-aware observer network for out-of-distribution object segmentation," *arXiv preprint arXiv:2207.08782*, 2022.

[92] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *Proceedings of the ieee/cvf international conference on computer vision*, 2021, pp. 5128–5137.

[93] X. Du, Z. Wang, M. Cai, and Y. Li, "Vos: Learning what you don't know by virtual outlier synthesis," *Proceedings of the International Conference on Learning Representations*, 2022.

[94] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, "Pixel-wise anomaly detection in complex driving scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 918–16 927.

[95] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3566–3573.

[96] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. L. Yuille, "Synthesize then compare: Detecting failures and anomalies for semantic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 145–161.

[97] D. Haldimann, H. Blum, R. Siegwart, and C. Cadena, "This is not what i imagined: Error detection for semantic segmentation through visual dissimilarity," *arXiv preprint arXiv:1909.00676*, 2019.

[98] T. Vojir, T. Šipka, R. Aljundi, N. Chumerin, D. O. Reino, and J. Matas, "Road anomaly detection by partial image reconstruction with segmentation coupling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 651–15 660.

[99] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1951–1963, 2020.

[100] V. V. Valindria, I. Lavdas, W. Bai, K. Kamnitsas, E. O. Aboagye, A. G. Rockall, D. Rueckert, and B. Glocker, "Reverse classification accuracy: Predicting segmentation performance in the absence of ground truth," *IEEE Transactions on Medical Imaging*, vol. 36, no. 8, pp. 1597–1606, 2017.

[101] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[102] H. Jing, Y. Gao, S. Shahbeigi, and M. Dianati, "Integrity monitoring of gnss/ins based positioning systems for autonomous vehicles: State-of-the-art and open challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[103] S. Rabiee and J. Biswas, "Iv-slam: Introspective vision for simultaneous localization and mapping," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 16–18 Nov 2021, pp. 1100–1109.

[104] Z. Alsayed, G. Bresson, A. Verroust-Blondet, and F. Nashashibi, "Failure detection for laser-based slam in urban and peri-urban environments," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–7.

[105] S. Nobili, G. Tinchev, and M. Fallon, "Predicting alignment risk to prevent localization failure," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1003–1010.

[106] H. Almqvist, M. Magnusson, T. P. Kucner, and A. J. Lilienthal, "Learning to detect misaligned point clouds," *Journal of Field Robotics*, vol. 35, no. 5, pp. 662–677, 2018.

[107] D. Adolfsson, M. Magnusson, Q. Liao, A. J. Lilienthal, and H. Andreasson, "Corai–are the point clouds correctly aligned?" in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.

[108] D. Adolfsson, M. Castellano-Quero, M. Magnusson, A. J. Lilienthal, and H. Andreasson, "Coral: Introspection for robust radar and lidar perception in diverse environments using differential entropy," *Robotics and Autonomous Systems*, vol. 155, p. 104136, 2022.

[109] X. Ju, D. Xu, and H. Zhao, "Scene-aware error modeling of lidar/visual odometry for fusion-based vehicle localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6480–6494, 2021.

[110] N. Akai, L. Y. Morales, T. Hirayama, and H. Murase, "Misalignment recognition using markov random fields with fully connected latent variables for detecting localization failures," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3955–3962, 2019.

[111] N. Akai, Y. Akagi, T. Hirayama, T. Morikawa, and H. Murase, "Detection of localization failures using markov random fields with fully connected latent variables for safe lidar-based automated driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 130–17 142, 2022.

[112] S. Rabiee and J. Biswas, "Ivoa: Introspective vision for obstacle avoidance," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1230–1235.

[113] J. M. Garcia, R. Prophet, J. C. F. Michel, R. Ebelt, M. Vossiek, and I. Weber, "Identification of ghost moving detections in automotive scenarios with deep learning," in *2019 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, 2019, pp. 1–4.

[114] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller, "Ghost target detection in 3d radar data using point cloud based deep neural network," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 398–10 403.

[115] T. Griebel, D. Authaler, M. Horn, M. Henning, M. Buchholz, and K. Dietmayer, "Anomaly detection in radar data using pointnets," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2667–2673.

[116] J. Breitenstein, A. Bär, D. Lipinski, and T. Fingscheidt, "Detection of collective anomalies in images for automated driving using an earth mover's deviation (emdev) measure," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 90–97.

[117] A. Gupta and L. Carlone, "Online monitoring for neural network based monocular pedestrian pose estimation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, 2020.

[118] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for rgb-d semantic segmentation incorporating unexpected obstacle detection for road-driving images," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5558–5565, 2020.

[119] S. Geisler, C. Cunha, and R. K. Satzoda, "Better, faster small hazard detection: Instance-aware techniques, metrics and benchmarking," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[120] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Network*, vol. 35, no. 3, pp. 96–101, 2021.

[121] D. Xiao, W. G. Geiger, H. Y. Yatbaz, M. Dianati, and R. Woodman, "Detecting hazardous events: A framework for automated vehicle safety systems," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 641–646.

[122] D. Xiao, M. Dianati, W. G. Geiger, and R. Woodman, "Review of graph-based hazardous event detection methods for autonomous driving systems," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[123] A. Roitberg, K. Peng, D. Schneider, K. Yang, M. Koulakis, M. Martinez, and R. Stiefelhagen, "Is my driver observation model overconfident? input-guided calibration networks for reliable and interpretable confidence estimates," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 271–25 286, 2022.

[124] J. Zhang, K. Yang, and R. Stiefelhagen, "Exploring event-driven dynamic context for accident scene segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2606–2622, 2022.

[125] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "The fishyscapes benchmark: Measuring blind spots in semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3119–3135, 2021.

[126] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, P. Fua, M. Salzmann, and M. Rottmann, "Segmentmeifyoucan: A benchmark for anomaly segmentation," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., vol. 1. Curran, 2021.

[127] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song, "Scaling out-of-distribution detection for real-world settings," in *Proceedings of Machine Learning Research*, vol. 162, 2022, p. 8759 – 8773.

[128] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021.

[129] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[130] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[131] W. Jia, L. Yang, Z. Jia, W. Zhao, Y. Zhou, and Q. Song, "Tive: A toolbox for identifying video instance segmentation errors," *Neurocomputing*, p. 126321, 2023.

Hakan Yekta Yatbaz (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from Middle East Technical University Northern Cyprus Campus. He is currently pursuing a Ph.D. degree with WMG at University of Warwick. His research interests include machine learning, computer vision, and intelligent systems.

Mehrdad Dianati (Senior Member, IEEE) is a part-time professor of autonomous and connected systems at the University of Warwick, U.K. He concurrently holds a professorial post at the School of Electronics, Electrical Engineering and Computer Science at the Queen's University of Belfast. His research focuses on the application of digital technologies (information and communication technologies and artificial intelligence) for the development of future mobility and transport systems. He has over 30 years of combined industrial and academic experience, with over 20 years in various leadership roles of multi-disciplinary collaborative research and development projects. He works closely with the automotive and ICT industries as the primary application domains of his research. Previously, he was the Director of the Centre for Doctoral Training on Future Mobility Technologies at the University of Warwick. He has served as an Editor for the IEEE Transactions on Vehicular Technology and several other international journals, including IET Communications. He is also the Field Chief Editor of Frontiers Journal in Future Transportation.

Roger Woodman is an Assistant Professor and Human Factors research lead at WMG, University of Warwick. He received his PhD from Bristol Robotics Laboratory and has more than 20 years of experience working in industry and academia. Among his research interests, are trust and acceptance of ne w technology with a focus on self-driving vehicles, shared mobility, and human-machine interfaces. He has several scientific papers published in the field of connected and autonomous vehicles. He lectures on the topic of Human Factors of Future Mobility and is the Co-director of the Centre for Doctoral Training, training doctoral researchers in the areas of intelligent and electrified mobility systems.