

An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving

Florian Heidecker¹, Jasmin Breitenstein², Kevin Rösch³, Jonas Löhdefink², Maarten Bieshaar¹, Christoph Stiller⁴, Tim Fingscheidt², and Bernhard Sick¹

Abstract—Systems and functions that rely on machine learning (ML) are the basis of highly automated driving. An essential task of such ML models is to reliably detect and interpret unusual, new, and potentially dangerous situations. The detection of those situations, which we refer to as corner cases, is highly relevant for successfully developing, applying, and validating automotive perception functions in future vehicles where multiple sensor modalities will be used. A complication for the development of corner case detectors is the lack of consistent definitions, terms, and corner case descriptions, especially when taking into account various automotive sensors. In this work, we provide an application-driven view of corner cases in highly automated driving. To achieve this goal, we first consider existing definitions from the general outlier, novelty, anomaly, and out-of-distribution detection to show relations and differences to corner cases. Moreover, we extend an existing camera-focused systematization of corner cases by adding RADAR (radio detection and ranging) and LiDAR (light detection and ranging) sensors. For this, we describe an exemplary toolchain for data acquisition and processing, highlighting the interfaces of the corner case detection. We also define a novel level of corner cases, the *method layer* corner cases, which appear due to uncertainty inherent in the methodology or the data distribution.

I. INTRODUCTION

Perception is a challenging task in highly automated driving. Especially when, as in Figure 1, low winter sun overexposes the camera, the LiDAR provides false information due to reflections from the partly slippery/icy road, and the RADAR has much noise in its data due to the snow. Most importantly, to ensure safety, the environment perception methods need to operate reliably. This is crucial in driving situations deviating from what is considered “normal”, and possibly even dangerous as in Fig. 1. Such situations are generally termed as *corner cases*, and their robust detection is necessary for reliable perception methods, and hence, safe highly automated driving. Next to the online application,



Fig. 1. Multiple corner cases: A winter scene with an icy, slippery, reflective road, combined with low winter sun and people on cross-country skis crossing the road.

detection of corner cases is also required during development of machine learning (ML) perception methods. Here, they provide both the appropriate training, and the crucial test data to successfully develop and validate robust perception methods.

While there exist many terms related to corner cases in literature, as well as many detection approaches, a general definition and description is missing. The lack of agreed technical terms and definitions makes detection of corner cases cumbersome. Previous work on corner case definitions and detection has mostly considered the camera sensor and visual perception. In this work, we consider corner cases of RADAR and LiDAR sensors in addition to the camera sensor.

The authors of [1] formulate a systematization of corner cases for visual perception in highly automated driving, where corner cases are categorized in levels. These levels are based on the type of situation they encompass and ordered by their theoretical detection complexity. We follow this approach and also consider corner cases on *scene* (Fig. 1), *object* (e.g., people on cross-country skis), and *domain level* (e.g., snowy winter), which we summarize into the *content layer*. Additionally, for corner cases at *scenario level*, we define the *temporal layer*, e.g., the unusual movement of a person with cross-country skis compared to a pedestrian. We grouped the corner cases, depending on whether they concern single image frames and point clouds (*content layer*), or multiple consecutive ones (*temporal layer*). As we aim to provide a more comprehensive conceptualization of corner cases, including multi-modal sensor inputs, we distinguish on the lowest theoretical detection complexity between pixel-, and point-cloud-level corner cases, terming this the *sensor*

This paper is submitted to IEEE Intelligent Vehicles Symposium 2021.

¹Florian Heidecker, Maarten Bieshaar and Bernhard Sick are with Intelligent Embedded Systems, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany, {florian.heidecker, mbieshaar, bsick}@uni-kassel.de

²Jasmin Breitenstein, Jonas Löhdefink and Tim Fingscheidt are with Institute for Communications Technology, Technische Universität Braunschweig, Schleinitzstraße 22, 38106 Braunschweig, Germany, {j.breitenstein, j.loehdefink, t.fingscheidt}@tu-bs.de

³Kevin Rösch is with Mobile Perception Systems, FZI Research Center for Information Technology, Schönfeldstraße 8, 76131 Karlsruhe, Germany, kevin.roesch@fzi.de

⁴Christoph Stiller is with Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Engler-Bunte-Ring 21, 76131 Karlsruhe, Germany, stiller@kit.de

layer. We will provide a detailed description of the expanded systematization, and sensor-specific corner cases.

The main contributions of this work are:

- Conceptualization of a sensor-driven categorization of corner cases for camera, LiDAR, and RADAR into different *layers* and *levels*.
- Differentiation of *single*- and *multi-source* corner cases in consideration of different fusion levels.
- Introduction of *method layer* corner cases generated by the method itself and depending on parameters such as topology, design, and deployment.

The remainder of this article is structured as follows: Section II gives an overview about recent work in the field of corner cases and their categorization. The processing toolchain in highly automated driving including camera, LiDAR, and RADAR sensor is presented in Section III. Section IV handles the individual sensor types in more detail and names specific corner cases. Section V particularly introduces the *method layer* for corner cases, while Section VI shows various methods to detect corner cases. Existing datasets which can be used for corner case detection and potential evaluation methods are provided in Section VII. Finally, Section VIII concludes the key message of the paper.

II. DISTINCTION BETWEEN CORNER CASES AND OTHER TERMS USED IN LITERATURE

In literature, various terms are used to describe unexpected, or anomalous situations and events. In this section, we point out their distinctions and relations to corner cases in highly automated driving.

Edge Cases: In the context of software and hardware testing, both terms *corner case* and *edge case* are used. Edge cases are situations or parameters that occur rarely but are already taken into account during development [2]. This also applies to *extreme cases* or *boundary cases*, which are often not explicitly mentioned but are included in the edge case terminology. By identifying and handling edge cases, the situations or parameters lose the edge case status and are considered normal [2]. In contrast, a corner case results from the combination of several normal situations or parameters that coincide simultaneously, thus representing a rare or never considered case or scene [2]. Transferring this to highly automated driving, an example scene could be as shown in Fig. 1.

In automated driving also entirely new situations can occur and are then considered as corner cases, not just combinations of already known ones. Similar to software testing, they can cease to exist once an appropriate number of examples of a particular corner case have been added to the training and validation data of a perception method.

ML viewpoint on corner cases: In ML, we are interested in corner case situations to be able to validate, but more importantly, also to improve our system by re-training. The recorded data of such a situation can be used as part of the training data for the ML system, as the performance of a perception model on corner case data for tasks such as object detection, classification, and prediction is of great

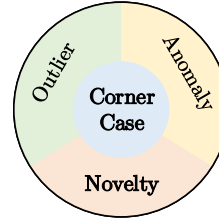


Fig. 2. The terms outlier, novelty, anomaly are often used in machine learning literature. Corner cases are strongly related to them and have an overlapping meaning with each of these terms.

interest. A general ML system can perform poorly on such corner case data because the data contains novel situations, or effects, which are not (sufficiently) present in the system’s training data. The identification of such corner cases is a major challenge, which also requires understanding the performance of an ML model.

In addition to the distinction between the terminology for corner cases in perception for highly automated driving and in software testing, the technical terms *outliers*, *anomalies* and *novelties* from ML literature are related to the meaning of the term *corner case*, see Fig. 2.

Outlier: An outlier is defined as “an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” [3]. In [4], an outlier is described as a legitimate observation of a known process that occurs in an area of low density and, e.g., represents an extreme value. We follow a similar understanding for corner cases, e.g., when the camera is maximally blinded or a LiDAR beam or RADAR impulse is maximally reflected or absorbed by the environment.

Anomalies: For anomalies, multiple definitions can be found in literature (cf. [5], [6], [1]). While there are definitions focusing on anomalies as noisy data occurrences that prompt artifacts which in turn hinder data analysis [6], Jiang et al. [7], e.g., first define normal events by a high frequency of occurrence, where consequently, anomalies are given as the opposite. Other definitions consider anomalies as patterns that are not consistent with learned ones, or with general normal behavior [5], [8]. Moreover, there exists a categorization in contextual, collective, and point anomalies [5]. For corner cases in automated driving, there is a deviation from normality that is manifested in non-conform behavior or patterns. The terms anomaly and corner case are almost used synonymously. Anomalies describe a deviation from normality. Hence, the term appears in the systematization of corner cases [1] as well. For corner cases, other factors also contribute to the terminology and definition, such as the relevance for the driving behavior [9]. Moreover, corner cases can be more complex scenarios that become anomalous only in their entirety but do not consist of anomalous objects, similar again to the application of software testing.

Novelties: Novelties appear as previously unseen instances or objects [5]. In [4] the definition of novelties is somewhat broader, whereby novelties are described as a spatial or

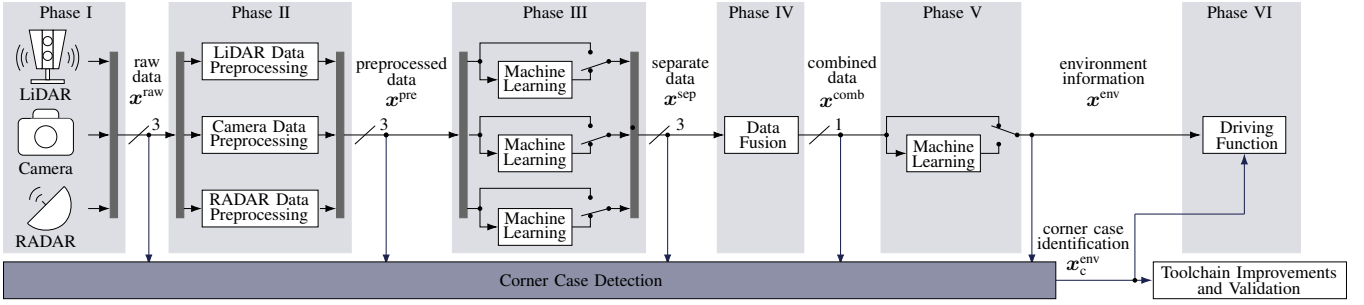


Fig. 3. Toolchain for data processing showing the individual sensors and processing steps with $I = 3$ sensors: LiDAR, camera, and RADAR. The toolchain can be divided into the phases of data acquisition (I), data preprocessing (II), early machine learning (III), data fusion (IV), late machine learning (V), and application (VI).

temporal agglomeration of anomalies or as a change in the distribution of an already known process. The appearance of new situations, objects, and movement patterns is an essential characteristic of corner cases, which does not simplify a clear distinction to novelties.

III. MULTI-SENSOR PROCESSING TOOLCHAIN

Before dealing with the different types of corner cases, we detail the processing toolchain on which our considerations and arguments are based. Fig. 3 shows the different sensors of a prototypical automated vehicle in phase I. For the toolchain, we consider three sensors, namely LiDAR, camera, RADAR. After the acquisition, the raw data $\mathbf{x}^{\text{raw}} = (x_i^{\text{raw}})$ for the different sensors enumerated by $i \in \mathcal{I} = \{1, 2, 3\}$ are first preprocessed in phase II. The connection between phase I and II and between all other phases and blocks are shown with an arrow. For clarity, if necessary, several connections are bundled into one, where the number above the arrow describes the number of connections. As preprocessing depends heavily on the type of sensor, it is done separately for camera images and point clouds from LiDAR and RADAR. Possible preprocessing operations are reshaping, aligning, or filtering.

Afterwards, depending on the type of data fusion, the data $\mathbf{x}^{\text{pre}} = (x_i^{\text{pre}})$, $i \in \mathcal{I}$, first pass through ML algorithms separately in phase III or bypass them and are directly fused into the combined data \mathbf{x}^{comb} in phase IV. The latter is termed early data fusion. The so-called late data fusion is performed when the sensor data x_i^{pre} are first processed separately in ML algorithms, and the outputs $\mathbf{x}^{\text{sep}} = (x_i^{\text{sep}})$, $i \in \mathcal{I}$, are then fused afterward in phase IV. Both processing options are visualized in Fig. 3 by switches. In Section IV, we regard early and late data fusion, and consider the possibilities of machine-learning-based fusion, where the phase IV and V from Fig. 3 become indistinct and can even completely merge. The processing pipeline then outputs all environmental information \mathbf{x}^{env} that are extracted in the pipeline.

Furthermore, Fig. 3 provides an interface description for corner case detection: Corner cases can appear at multiple points in the toolchain. Therefore, if one aims to detect corner cases in the perception pipeline, the detection methods

need to run in parallel to our depicted toolchain. Moreover, we require connections for the corner case detection from each processing phase I to V. This connection enables detecting corner cases in each step of the toolchain, providing all interim data from raw sensor values x_i^{raw} , $i \in \mathcal{I}$, to fully extracted environment information \mathbf{x}^{env} . Thus, we might detect, e.g., unknown objects after late fusion of camera and LiDAR, which deviate from each other, pixel errors directly in the raw data, or uncertainties from the ML module's output. The corner case detection has the second output of the toolchain and provides corner case information $\mathbf{x}_c^{\text{env}}$. The corner case detection output is of the same size as the environment information \mathbf{x}^{env} and provides for each entry a probability for a corner case appearing here. Both, corner case identification and environment information serve in phase VI (see Fig. 3) as input to the driving function, deciding on the appropriate driving behavior. In an offline (laboratory) application, phase VI is excluded, and instead, both \mathbf{x}^{env} and $\mathbf{x}_c^{\text{env}}$ are used to reduce the data at hand intelligently. Note that the formulation of this toolchain can be arbitrarily extended to include more than three sensors by adding them in parallel to the existing ones.

IV. SENSOR-DRIVEN VIEW OF CORNER CASES

The available data or the used algorithms do not only influence the occurrence of corner cases in ML. The application, in this case highly automated driving, the sensor technology, and the installation position have a significant impact. Before discussing corner cases for different types of sensors, we briefly review a hypothetical research vehicle as of today that could be the basis for highly automated vehicles in the future. It is equipped with a stereo camera, LiDAR sensors on the roof, and RADAR sensors on the front and rear. The field of view (FOV) of the different sensors overlaps mostly so that the environment is covered in most parts in each direction with at least two types of sensors. Based on this setup, there are sensor-specific corner cases for each sensor type, which is discussed in more detail below, where we extend a previous systematization [1] of corner cases for visual perception to the LiDAR and RADAR sensor.

The systematization [1] classifies corner cases into different levels, ordered by their theoretical complexity of detection.




	Sensor Layer		Content Layer			Temporal Layer
	Hardware Level	Physical Level	Domain Level	Object Level	Scene Level	Scenario Level
 LiDAR-based corner cases	Laser Error Broken mirror Misaligned actuator	Beam-Based Corner Case Black cars disappear ...	Domain Shift on Single Point Cloud Shape of Road markings	Single-Point Anomaly on Single Point Cloud Dust cloud ...	Contextual/Collective Anomaly on Single Point Cloud Sweeper cleaning the sidewalk	Corner Cases on Multiple Point Clouds and Frames • Person breaks traffic rule • Overtaking a cyclist • Car accident ...
 Camera-based corner cases	Pixel Error Dead pixel Broken lens	Pixel-Based Corner Case Dirt on lens Overexposure	Domain Shift on Single Frame Location (EU-U.S.A.) ...	Single-Point Anomaly on Single Frame Animal ...	Contextual/Collective Anomaly on Single Frame People on a billboard ...	
 RADAR-based corner cases	Impulse Error Low voltage Low temperature	Impulse-Based Corner Case Interference ...	Domain Shift on Single Point Cloud Weather, e.g., snow, rain, etc.	Single-Point Anomaly on Single Point Cloud Lost objects ...	Contextual/Collective Anomaly on Single Point Cloud Demonstration tree on street	

TABLE I. Categorization of camera-, LiDAR-, and RADAR-based *single-source* corner cases. Example situations are given for the individual corner cases on the *sensor layer*, *content layer*, and *temporal layer*. *Method layer* corner cases are not shown in this table.

The highest detection complexity is given for *scenario level* corner cases. These patterns can be observed throughout an image sequence and further subdivided into anomalous, novel, and risky scenarios based on their potential for collision and their observability during training. *Scene level* corner cases are observed on single images and describe known objects in either unseen quantities or locations. On the *object level*, unknown objects are observed in single images, while on the *domain level*, corner cases arise due to the inability of the world model to explain its observations, i.e., a domain shift. The pixel level is the lowest in [1] and includes corner cases resulting from local and global outliers in the camera hardware.

In this work, we now extend and modify the vision-oriented systematization [1] by including other sensors and introducing the additional notion of a layer, see Table I. For clarity, the *sensor layer*, *content layer*, and *temporal layer* are introduced at the top level. The *temporal layer* includes corner cases with a temporal context, thereby corresponding to the *scenario level* in [1]. On the other hand, the *content layer* comprises the *domain*, *object*, and *scene level* from [1] and thus contains corner cases that result (a) from the data at a specific point in time. The four levels *scenario*, *scene*, *object*, and *domain* are existing in signals from LiDAR and RADAR sensors as they exist in camera signals, as such corner cases appear in point clouds and sequences of point clouds in a similar way. However, it is important to consider that corner cases of a certain level can only be transferred from one sensor to another to a limited extent, if at all. A corner case can therefore exist for one sensor but not for a different sensor. Finally, the *sensor layer* describes corner-cases that can be traced back to hardware errors or physical properties. This results in the newly introduced *hardware level* and *physical level*.

Corner cases resulting from a single sensor are called *single-source* corner cases. We have already discussed the categorization of these corner cases into layers and levels. However, the fusion of sensor data can cause *multi-source* corner cases, which we consider as a separate category. Since they can appear more or less in any of the levels, we do not highlight them in Table I.

A. Camera

In the context of highly automated driving, vehicles are often equipped with many different camera systems. Mono and fisheye cameras are often used to cover the areas to the left, to the right, and behind the vehicle. Stereo, or rarely trifocal cameras, are used mostly to cover the area in front of the vehicle [10]. Regardless of the camera system, pixel errors such as dead pixel, a dirty camera lens, or overexposure can lead to a corner case at *hardware level*, or *physical level*, see Table I. Unique camera properties or functions, such as calculating the depth image of a stereo or trifocal camera, can cause corner cases at *hardware level*. This group of corner cases can be summarized with corner cases at the *sensor layer*.

However, many corner cases are found in the image data itself and are often system-independent. Image data have a high information density. This fact is also reflected in various methods, such as object recognition and classification, contour estimation, recognition of the direction of gaze of the person, gesture recognition, traffic sign recognition, extraction of weather conditions and many more. On the other hand, this abundance of information and variability of image content also results in the most diverse corner cases. These corner cases can be divided into *domain level*, e.g., different traffic signs in Germany and the U.S.A., *object level*, such as animals and never-seen-before objects, and *scene level*, where it can be a new situation such as a tree lying on the road. If the depth image is also available, the spectrum is extended by the possibility to determine the position of an object, which results in a further type of *scene level* corner case. However, all these cases belong to the *content layer*.

If a camera captures a scene in a video sequence, it consists of several individual images and the corner case can be represented by either a single frame, or by several consecutive frames. A corner case in a particular frame may require a temporal component as, e.g., in the case of a pedestrian's movement prediction, and is therefore a temporal-based corner case categorized on the *scenario level*, situated at the *temporal layer*.

B. LiDAR

There are different LiDAR systems in practice. Based on the technique, LiDARs are divided into two groups: the LiDAR system consisting of moving parts and the solid-state LiDAR system. In any vehicle, the sensors are exposed to vibrations, which leads to an increased risk of damage for LiDAR sensors with moving parts. In case of such or other hardware defects independent of the used LiDAR technology, the resulting corner cases can be summarized as *hardware level* corner cases, see Table I. In addition to the measuring points, which result in a point cloud, the LiDAR also provides an intensity value for each measuring point. The intensity of the reflected beam can cause corner cases on the *physical level*. For example, the intensity value can be very low for a dark-painted object or extremely high in the case of a mirror or license plate, leading to a wrong interpretation.

The simplest type of LiDAR is the line laser, which can only measure one plane and depends heavily on the alignment. If the LiDAR has several planes, the FOV expands vertically. A 360° LiDAR allows to cover all directions and makes it possible to determine the width, height, and distance to each visible object around the vehicle. In all three LiDAR variants, corner cases occur on the *content layer*. The corner cases can be quite different and corner cases in a LiDAR with several planes do not necessarily have to occur in a 360° LiDAR. On the *domain level*, corner cases can arise, for example, because the road markings that are largely recognizable with a LiDAR in Germany cannot be perceived by a LiDAR in a different country because the markings are made of a different material. Dust or smoke clouds (extreme examples excepted) are a wonderful example for corner cases on *object level*, since in comparison to camera and radar the laser beam of the LiDAR can be reflected by the particles. Corner cases can also arise at the scene level. For example, objects already known to a system that are in an unusual position, e.g., stacked scrap cars in a junkyard or a sweeper on the sidewalk, can cause corner cases.

When a time sequence of measurements or point clouds is processed, corner cases at the *temporal layer* arises. For example, this is expressed in the form of new movement patterns of people (e.g., movement of the extremities) or vehicles (e.g., the inclination of a moving motorcycle). We refer to this phenomenon as *scenario level* corner cases.

C. RADAR

RADAR sensors for highly automated vehicles can be divided into short, mid, and long-range. The FOV becomes narrower with increasing range. For this reason, short and long-range sensors are combined often into one physical sensor system [10]. A RADAR has no moving parts as some LiDAR have, which can be damaged by vibration, or a lens like a camera that can be scratched. Nevertheless, this does not protect against a hardware defect, which can lead to a *hardware level* corner case at the *sensor layer*, see Table I. Corner cases resulting from interferences, e.g., from RADAR sensors of other vehicles, are summarized under the

term *physical level* corner cases. Further corner cases of the *physical level* can be caused by a too small reflective surface of the object, making detection much worse or impossible. *Object level* RADAR corner cases appear in acquired RADAR data and are not caused by poor accuracy or too small reflection surfaces. Multiple reflections can cause, for example, the so-called ghost detection of objects [10]. An example of this is if one side of a truck's tailgate is open and the other one is closed, the RADAR impulse is reflected several times in the trailer until the RADAR sensor receives the impulse again. A major advantage of RADAR sensors is the excellent determination of the relative speed of dynamic objects. On the other hand, different weather conditions affect the data and can lead to corner cases at the *domain level*. For example, corner cases could arise at *scene level* because the RADAR does not sufficiently detect large branches with leaves or trees that have fallen on the road. All these various RADAR corner cases are summarized as *content layer* corner cases.

Corner cases that result from a sequence of RADAR point clouds are summarized under the term *scenario level* corner cases and belong to the *temporal layer*, as previously for camera and LiDAR. As with the other sensors, corner cases are primarily addressed resulting from the trajectory of objects, e.g., movement patterns.

D. Sensor Fusion

While highly automated vehicles will be equipped with various sensors, each sensor has its advantages and limitations. Henceforth, the data fusion module of the perception processing toolchain in Fig. 3 is crucial for the environment perception. Fig. 3 highlights two possibilities for data fusion in phase IV: early and late fusion. In early data fusion, the preprocessed data are directly fused in phase IV, bypassing phase III, i.e., $x^{\text{sep}} = x^{\text{pre}}$. Therefore, a network can learn to extract all information present in the raw data [11].

In late data fusion, information from the preprocessed data x^{pre} are extracted separately in the ML methods in phase III, and their output x^{sep} is combined [11].

The so-called cross fusion [12], or middle fusion [11], fuse the data, e.g., in intermediate layers of neural networks, hence blurring the distinct borders between the phases in Fig. 3, resulting in higher flexibility. ML approaches to data fusion can then also take the raw data as input and directly output environment information [13]. However, there are many options on how and where to fuse the data [11].

If the data from two or more sensors are merged, each sensor provides data that may well overlap in its features. If such an overlap leads to ambiguities that cannot be resolved, we summarize such situations under the term *multi-source* corner cases. A *multi-source* corner case occurs, for example, with a dust cloud: The LiDAR detects the dust cloud on the object level, while the camera and RADAR hardly perceive it or even not at all.

There is the potential that *multi-source* corner cases in the data fusion serve as a corner case detector themselves. For example, during sunset, visual perception of the camera

sensor deteriorates. However, both other sensors, RADAR and LiDAR are not influenced by this corner case for visual perception. If a person crosses the street in front of the ego-vehicle in this situation, ambiguities appear in the data fusion. This leads to the critical question of trustworthiness of the perception methods.

V. METHOD LAYER CORNER CASES

In contrast to the corner cases of the *sensor*, *content*, and *temporal layer*, there exists another corner case layer independent of the respective sensors, and the (actual) temporal and spatial content, which is not shown in Table I, as the respective corner cases are not sensor-, but method-specific. We term this corner case layer the *method layer*, and refer to the respective corner cases as *method level* corner cases. These corner cases are not necessarily perceptible by sensors or a human driver, making this layer more abstract.

Corner cases of the *method layer* are caused by the applied method themselves. In the processing toolchain in Figure 3, this corner case type appears predominantly in all phases involving machine learning methods (i.e., phases III and V). Depending on their topology, design, and deployment, methods can give rise to various corner cases. These might be corner cases due to a lack of knowledge because the ML model has never encountered a similar situation before. In literature, this is also referred to as epistemic uncertainty [14], which can be approximate via Monte Carlo dropout [15], or deep ensembles [16]. As an example for a *method level* corner case, consider a ML method for visual object detection. While we might aim to detect unknown objects by a high epistemic uncertainty [17], also typical “normal” objects can spark a high epistemic uncertainty [14], [18], hence, leading to a *method level* corner case. Epistemic uncertainty is not necessarily limited to ML methods and can appear in various kinds of mathematical models in general. Adversarial samples are another example of *method level* corner cases that are not perceptible by humans (e.g. [19], [20]). Even small changes to the model’s input (e.g., adding a particular noise pattern) can lead to drastic changes in the model’s output. One challenge is to distinguish corner cases of the *sensor*, *content*, and *temporal layer* from corner cases only immanent to the *method layer*.

Depending on the method and model, an inductive bias, i.e. assumptions inherent to the ML method influencing its predictive capability, is introduced to the underlying perception task. This bias is also connected to *method level* corner cases as it introduces certain assumptions about the problem at hand, hence, also manifesting the types of corner cases that can appear.

There is a semantic overlap between *method level* corner cases and the *single-source* corner cases resulting from the above explications. While *method level* corner cases are caused by uncertainty in the methodology, this definition does not exclude other corner case types. If we encounter a bear on the street, this can be considered an *object level* corner case for both RADAR and LiDAR. However, at the same time, it is possible that the appearance of this bear also

results in high epistemic uncertainty [18]. The same holds, e.g., if one encounters a *domain level* corner case through a location change (see Table I), depending on the method, this can also lead to high epistemic uncertainty.

VI. CORNER CASE DETECTION TECHNIQUES

In this section, we give a brief overview of existing methods for corner case detection. Most corner case detection methods in literature regard the camera sensor and aim to detect corner cases in images or image sequences; see the overviews given in [1], [21], we consider methods concerning different sensors, layers of corner cases, and their engagement with the perception toolchain in Figure 3.

To detect corner cases on *sensor layer*, Protopapadakis et al. [22] map maritime RADAR data via stacked autoencoders to a feature space, where they employ density-based clustering to detect outliers. Chakravarthy et al. [23] apply a CNN to raw RADAR data to extract features and classify them using an open-set classification method to detect unknowns in the waveform. As an open-set classifier, they choose SV-Means [24], which is built upon other existing open-set classifiers and formulated for the open-set problem of waveform classification. Both approaches also extend to *content* and *temporal layer* corner cases.

Lis et al. [25] regenerate synthetic images from segmentation masks to identify *content layer* corner cases via the residual error between the real and regenerated image. There are also *content layer* corner case detection methods on LiDAR data, such as Wong et al. [26], who introduce an open-set instance segmentation network on point clouds that identifies unknown points in an embedding space and groups them into unknown instances. Capellier et al. [27] propose a method to detect both known and unknown objects in LiDAR data.

Bolte et al. [9] identify *temporal layer* corner cases for the camera sensor if the residual error between the real and a predicted image, weighted by the criticality of the location in the image, exceeds a threshold.

For the camera sensor, methods from outlier and anomaly detection in more general computer vision can be applied to visual perception in automated driving. Of the three sensors considered in this work, corner case detection methods for RADAR data are the rarest. Adaptations of existing methods for LiDAR point clouds to RADAR data, e.g., the adaptation of PointNets to RADAR [28], can also lead to new corner case detection methods in RADAR data.

Most sensor fusion approaches aim to make the perception of automated vehicles more robust in general. Many focus on detecting obstacles, e.g., using infrared and ultrasonic sensors [29], or RADAR and LiDAR fusion with a Kalman-filter-type algorithm [30]. Yang et al. [31] perform RADAR and LiDAR fusion via a novel neural network architecture leveraging early and late fusion for geometric and velocity data from the RADAR sensor, making object detection more robust.

VII. EVALUATION

In this section, we discuss existing datasets and methods for the evaluation of corner case detectors. Here, we especially elaborate on datasets and metrics to evaluate not only corner cases for visual perception, but those encompassing other sensors as well.

A. Datasets

There exist a multitude of datasets for perception in highly automated driving.

Camera: Many of them provide camera images or sequences (cf. [1]), such as the *Cityscapes* dataset [32], the *BDD100k* dataset [33], the *openDD* dataset [34], or the *Mapillary Vistas* dataset [35], to name a few. All of those are large-scale datasets for visual perception and provide adequate labels for the related tasks such as semantic and instance segmentation and object detection. Moreover, there exist a few datasets specifically for the task of unknown object detection from images, such as the *RoadAnomaly* [25], the *Lost&Found* [36] dataset, and the *Fishyscapes* [37] dataset. Additionally, there exist datasets with a specific underlying task such as VRU detection (*Eurocity Persons*) [38].

LiDAR: Next to those purely camera-based datasets, there exists by now also an increasing amount of multi-modal datasets. One of the earliest of such datasets is the *KITTI* [39] dataset, providing both camera and LiDAR data. The combination of LiDAR and camera data appears in several datasets, such as *ApolloScape* [40], *A2D2* [41], *Waymo Open* [42], *PandaSet* [43], and *KAIST* [44]. Many of them provide 3D bounding boxes as labels [42], [41], [39], some additionally provide point cloud semantic segmentation labeling [41], [40]. The *Canadian Adverse Driving Conditions* [45] dataset specifically provides image and LiDAR data for wintry weather conditions.

RADAR: RADAR data is rare in large-scale datasets for perception in automated driving. To our knowledge, only a few such datasets exist. *nuScenes* [46], *Astyx HiRes2019* [47], *Oxford RobotCar* [48], and *RADIATE* [49] provide multi-modal data from RADAR, camera, and LiDAR, and as far as we know, all of these datasets provide bounding box labels for their RADAR data, except the *Oxford RobotCar* where no ground truth labels are available.

B. Evaluation Methods

A critical aspect of corner case detection is how to evaluate such methods. In contrast to related fields in automotive perception, large-scale benchmarks are missing, and no consistent evaluation rules exist. Here, we aim to provide a brief overview of existing and applied methods to contribute to harmonizing this field of research.

For corner case detection on camera images, often, the area under the receiver operator characteristic (AUC) is used to determine separability between a normal and anomalous corner case class (e.g., [50]). Moreover, the area under the precision-recall curve (AUPRC) determines the separability of one class, e.g., the corner case class [50]. One of few

existing online benchmarks related to corner case detection on automotive camera images, the *Fishyscapes* online benchmark [37], reports results in the average precision (AP) and the false positive rate at 95% true positive rate (FPR₉₅). For corner case detection on LiDAR point clouds, one often employs the task performance metrics average precision (AP) and panoptic quality (PQ) [26]. Wong et al. [26] additionally propose a variation of PQ, the unknown quality (UQ), which emphasizes recall quality over recognition quality.

Next to the detection of corner cases, it is also important not to lose performance on the original perception task [50]. Thus, one should report, e.g., accuracy, IoU (semantic segmentation), AP (object detection), especially when, e.g., including corner case detection and perception in a multi-task learning framework, where a learned corner case detection influences perception and vice versa.

Finally, there exist corner case metrics, which follow specifically proposed corner case definitions. As an example, Bolte et al. [9] propose a prediction-based corner case metric signifying that they obtain a corner case score from the error of an image prediction restricted to relevant objects in relevant locations.

VIII. CONCLUSIONS AND FUTURE WORK

In this work, we introduced a categorization of corner cases for perception in automated driving concerning the camera, LiDAR, and RADAR sensor modalities, highlighting specific definitions and respective examples for the sensors. Additionally, we described a perception toolchain for automated driving, where the interfaces of corner case detection were highlighted in the pipeline. To comply with the multitude of sensors in automated driving, we introduced a *sensor layer* for corner cases, where we distinguish the hardware and the *physical level*. Moreover, *method layer* corner cases were defined, which differ from the other corner case levels by being independent of the sensor modality, but instead are method-specific.

ACKNOWLEDGMENT

This work results from the project KI Data Tooling (19A20001O) funded by German Federal Ministry for Economic Affairs and Energy (BMWi) and the DeCoInt²-project financed by the German Research Foundation (DFG) within the priority program SPP 1835: “Kooperativ interagierende Automobile”, grant number SI 674/11-2.

REFERENCES

- [1] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, “Systematization of Corner Cases for Visual Perception in Automated Driving,” in *Proc. of IV*, Las Vegas, NV, USA, 2020, pp. 986–993.
- [2] P. Koopman, A. Kane, and J. Black, “Credible Autonomy Safety Argumentation,” in *Proc. of the 27th Safety-Critical Systems Symposium*. Bristol, UK: Safety-Critical Systems Club, 2019.
- [3] D. M. Hawkins, *Identification of Outliers*, ser. Monographs on applied probability and statistics. Dordrecht: Springer, 1980.
- [4] C. Gruhl, B. Sick, and S. Tomforde, “Novelty Detection in Continuously Changing Environments,” *Future Generation Computer Systems*, vol. 114, pp. 138–154, 2021.
- [5] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

- [6] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A Review of Novelty Detection," *Signal Processing*, vol. 99, pp. 215–249, 2013.
- [7] F. Jiang, J. Yuan, S. A. Tsafaris, and A. K. Katsaggelos, "Video Anomaly Detection in Spatiotemporal Context," in *Proc. of ICIP*, Hong Kong, Hong Kong, 2010, pp. 705–708.
- [8] O. P. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition – A Review," *IEEE Trans. on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865–878, 2012.
- [9] J.-A. Bolte, A. Bär, D. Lipinski, and T. Fingscheidt, "Towards Corner Case Detection for Autonomous Driving," in *Proc. of IV*, Paris, France, 2019, pp. 438–445.
- [10] H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds., *Handbuch Fahrerassistenzsysteme*. Wiesbaden, Germany: Springer Fachmedien Wiesbaden, 2015.
- [11] 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 Trans. on ITS*, pp. 1–20, 2020.
- [12] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR–Camera Fusion for Road Detection Using Fully Convolutional Neural Networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.
- [13] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection From RGB-D Data," in *Proc. of CVPR*, Salt Lake City, UT, USA, 2018, pp. 918–927.
- [14] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" in *Proc. of NIPS*, Long Beach, CA, USA, 2017, pp. 5574–5584.
- [15] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. of ICML*, New York, NY, USA, 2016, pp. 1050–1059.
- [16] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles," in *Proc. of NIPS*, Long Beach, CA, USA, 2017, pp. 6402–6413.
- [17] F. Heidecker, A. Hannan, M. Bieshaar, and B. Sick, "Towards Corner Case Detection by Modeling the Uncertainty of Instance Segmentation Networks," in *Proc. of ICPR, Workshop*, Milan, Italy, 2021, pp. 1–14.
- [18] J. Mukhoti and Y. Gal, "Evaluating Bayesian Deep Learning Methods for Semantic Segmentation," *arXiv preprint arXiv:1811.12709*, pp. 1–13, 2018.
- [19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing Properties of Neural Networks," in *Proc. of ICLR*, Montréal, QC, Canada, 2014, pp. 1–10.
- [20] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proc. of ICLR*, San Diego, CA, USA, 2015, pp. 1–10.
- [21] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, "Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches," *arXiv preprint arXiv:2102.05897*, 2021.
- [22] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, D. Dres, and M. Bimpas, "Stacked Autoencoders for Outlier Detection in Over-the-Horizon Radar Signals," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–11, 2017.
- [23] R. V. Chakravarthy, H. Liu, and A. M. Pavy, "Open-Set Radar Waveform Classification: Comparison of Different Features and Classifiers," in *Proc. of the IEEE International Radar Conference*, Washington, DC, USA, 2020, pp. 542–547.
- [24] A. M. Pavy and B. D. Rigling, "SV-Means: A Fast SVM-Based Level Set Estimator for Phase-Modulated Radar Waveform Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 191–201, 2018.
- [25] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the Unexpected via Image Resynthesis," in *Proc. of ICCV*, Seoul, Korea, 2019, pp. 2152–2161.
- [26] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying Unknown Instances for Autonomous Driving," in *CoRL*, Osaka, Japan, 2019, pp. 1–10.
- [27] E. Capellier, F. Davoine, V. Cherfaoui, and Y. Li, "Evidential Deep Learning for Arbitrary LIDAR Object Classification in the Context of Autonomous Driving," in *Proc. of IV*, Paris, France, 2019, pp. 1304–1311.
- [28] A. Danzer, T. Griebel, and D. Dietmayer, "2D Car Detection in Radar Data with PointNets," in *Proc. of ITSC*, Auckland, New Zealand, 2019, pp. 61–66.
- [29] G. Farias, E. Fabregas, E. Peralta, H. Vargas, G. Hermosilla, G. Garcia, and S. Dormido, "A Neural Network Approach for Building An Obstacle Detection Model by Fusion of Proximity Sensors Data," *Sensors*, vol. 18, pp. 1–18, 2018.
- [30] H. Hajri and M.-C. Rahal, "Real Time Lidar and Radar High-Level Fusion for Obstacle Detection and Tracking with Evaluation on a Ground Truth," *IJMME*, vol. 12, pp. 821–827, 2018.
- [31] B. Yang, R. Guo, M. Liang, S. Casas, and R. Urtasun, "RadarNet: Exploiting Radar for Robust Perception of Dynamic Objects," in *Proc. of ECCV*, Glasgow, UK, 2020, pp. 1–17.
- [32] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of CVPR*, Las Vegas, NV, USA, 2016, pp. 3213–3223.
- [33] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Video Database With Scalable Annotation Tooling," *arXiv preprint arXiv:1805.04687*, 2018.
- [34] A. Breuer, J.-A. Termöhlen, S. Homoceanu, and T. Fingscheidt, "openDD: A Large-Scale Roundabout Drone Dataset," in *Proc. of ITSC*, Rhodes, Greece, 2020, pp. 1–6.
- [35] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kotschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *Proc. of ICCV*, Venice, Italy, 2017, pp. 4990–4999.
- [36] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, "Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles," in *Proc. of IROS*, Daejeon, South Korea, 2016, pp. 1099–1106.
- [37] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation," *arXiv preprint arXiv:1904.03215*, 2019.
- [38] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes," *IEEE Trans. on PAMI*, vol. 41, no. 8, pp. 1844–1861, 2019.
- [39] A. Geiger, P. Lenz, C. Stillner, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *IJRR*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [40] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The ApolloScape Dataset for Autonomous Driving," in *Proc. of CVPR, Workshop*, Salt Lake City, UT, USA, 2018, pp. 1067–1073.
- [41] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi Autonomous Driving Dataset," *arXiv preprint arXiv:2004.06320*, 2020. [Online]. Available: <https://www.a2d2.audi>
- [42] "Waymo Open Dataset: An Autonomous Driving Dataset," 2019, accessed 22-Jan.-2021. [Online]. Available: <https://www.waymo.com/open>
- [43] "PandaSet: Public Large-scale Dataset for Autonomous Driving," 2019, accessed 22-Jan.-2021. [Online]. Available: <https://scale.com/open-datasets/pandaset>
- [44] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments," *IJRR*, vol. 38, no. 6, pp. 642–657, 2019.
- [45] M. Pitropov, D. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian Adverse Driving Conditions Dataset," *IJRR*, pp. 1–10, 2020.
- [46] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *Proc. of CVPR*, Seattle, WA, USA, 2020, pp. 11 618–11 628.
- [47] M. Meyer and G. Kuschik, "Automotive Radar Dataset for Deep Learning Based 3D Object Detection," in *Proc. of EuRAD*, Paris, France, 2019, pp. 129–132.
- [48] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *IJRR*, vol. 36, no. 1, pp. 3–15, 2017.
- [49] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "RADIAE: A Radar Dataset for Automotive Perception," *arXiv preprint arXiv:2010.09076*, pp. 1–15, 2020.
- [50] R. Chan, M. Rottman, and H. Gottschalk, "Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation," *arXiv preprint arXiv:2012.06575*, 2020.