Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches

Jasmin Breitenstein*, Jan-Aike Termöhlen*, Daniel Lipinski° and Tim Fingscheidt*

***Abstract*— Automated driving has become a major topic of interest not only in the active research community but also in mainstream media reports. Visual perception of such intelligent vehicles has experienced large progress in the last decade thanks to advances in deep learning techniques but some challenges still remain. One such challenge is the detection of corner cases. They are unexpected and unknown situations that occur while driving. Conventional visual perception methods are often not able to detect them because corner cases have not been witnessed during training. Hence, their detection is highly safety-critical, and detection methods can be applied to vast amounts of collected data to select suitable training data. A reliable detection of corner cases will not only further automate the data selection procedure and increase safety in autonomous driving but can thereby also affect the public acceptance of the new technology in a positive manner. In this work, we continue a previous systematization of corner cases on different levels by an extended set of examples for each level. Moreover, we group detection approaches into different categories and link them with the corner case levels. Hence, we give directions to showcase specific corner cases and basic guidelines on how to technically detect them.**

I. Introduction

Automated driving and its technologies have experienced significant progress in the last years. While this progress has been made and advances in automated driving have received a lot of attention, it still faces some challenges for a safe and reliable application in daily life. Visual perception methods form an important part of the intelligent vehicles. They are expected to detect, and understand their environment. Therefore, there already exists a vast amount of algorithms for visual perception tasks associated with the vehicle's environment including object detection (e.g., [1]), semantic segmentation (e.g., [2], [3]), instance segmentation (e.g., [4]), and many more. However, one crucial factor is the behavior of visual perception methods in unexpected situations that deviate from normal traffic situations. Those situations, so-called *corner cases*, exist in an infinite number of examples. Their dominant and connecting feature is their deviation from what is generally considered normal traffic behavior. Possible corner cases are for example the classical situations everyone fears of encountering while driving, such as a person running onto the street from behind an occlusion, a ghost driver or simply lost cargo on the street.

*Jasmin Breitenstein, Jan-Aike Termöhlen and Tim Fingscheidt are with the Institute for Communications Technology, Technische Universität Braunschweig, Schleinitzstr. 22, 38106 Braunschweig, Germany. Email: {j.breitenstein, j.termoehlen, t.fingscheidt}@tu-bs.de

°Daniel Lipinski is with Volkswagen AG, Berliner Ring 2, 38440 Wolfsburg, Germany. Email: daniel.lipinski@volkswagen.de

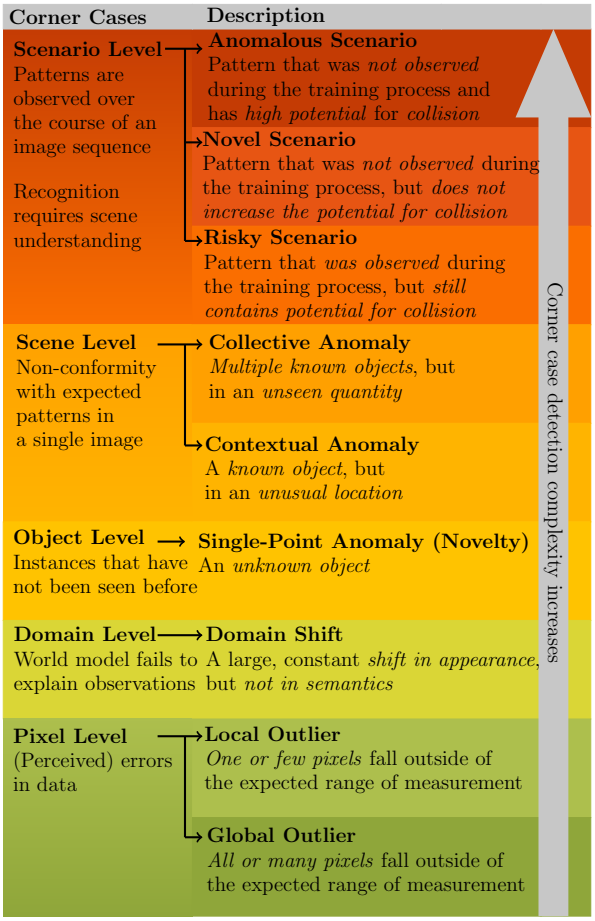

Fig. 1: Systematization of corner cases on different levels as given in [5]. The theoretical complexity of the detection typically increases from the bottom to the top.

A reliable detection of such corner cases is crucial for safety in automated driving as it can reduce the number of accidents of autonomous cars, and hence helping widespread acceptance and application of this technology. It is necessary both online, in the vehicle, and offline applications during development. A stable and confident corner case detection method recognizes critical situations. In the online application, it can be used as a safety monitoring and warning system which identifies situations while they occur. In the offline application, the corner case detector is applied to large amounts of collected data to select suitable training and relevant test data in the development of new visual perception algorithms in the laboratory. While the detection

both online and in the laboratory are related to safety, the offline applications also lead to both monetary and time savings by automatizing the selection process of training data. Hence, there already exists a variety of works dealing with the detection of corner cases in the automotive context such as the detection of obstacles [6], [7], or newly appearing objects [8].

Even though faithful and efficient corner case detection will have a large impact on automated driving, consistent generally accepted definitions and classifications are still missing to describe them. We follow the definition, that corner cases are present when "there is a non-predictable relevant object/class in a relevant location", of Bolte et al. [9]. To facilitate the systematical development of detectors, a categorization was introduced in [5]. A trimmed version of the corner case systematization can be found in Figure 1. It depicts a hierarchy ordered by the theoretical complexity of detection. We consider corner cases on pixel, domain, object, scene, and scenario level, which are described in more detail in Section II.

While this systematization already paves the way for a more methodical development of detection methods, it also poses the question on how to actually detect the specific corner cases of each level. In the context of smart manufacturing, a systematization of possible faults of the system has been established by Lopez et al. [10]. Next to this systematization, the authors propose a categorization of detection methods into feature extraction, regression, knowledge-based, signal model, state estimation, clustering, and classification methods, connecting each method category with specific anomaly categories in smart manufacturing. We follow their example in order to extend the previous systematization of corner cases for visual perception in automated driving by a categorization of detection approaches, and associating them with the previously defined corner case levels. Additionally, we give both specific examples of corner case for each level and provide first guidelines towards basic detection methods.

Due to the omnipresence and success of deep learning methods in visual perception algorithms, we limit this categorization to deep learning methods. They are highly functioning methods with promising results in many visual perception applications and have also shown success in the detection of unusual occurrences. Moreover, we limit this work to purely visual approaches excluding other sensor data such as RADAR and LiDAR, but consider corner cases that can be detected either from single image frames or entire image sequences.

The paper is structured as follows. We briefly review the systematization of corner cases in Figure 1. Then we provide more thorough examples for each corner case level with the purpose of enabling both a more comprehensive understanding of what kind of situation the corner case levels can encompass, and the recording of corner cases by almost mimicking stage directions. Moreover, we extend the previous systematization by categories for detection approaches with their respective related work. Finally, we map detection approaches to corner case levels by providing hints and intuition for the development of new methods.

II. Systematization of Corner Cases

Previously, a systematization of corner cases for visual perception in automated driving has been introduced [5], which we briefly summarize in the following. This systematization can be found in Figure 1 in a reduced version. The corner case levels are ordered by detection complexity. Going from lower to higher complexity of detection, we have corner cases on pixel level which can be divided into **global** and **local outliers**. Examples for those are overexposure and dead pixels, respectively. Then there are domain-level corner cases which are caused by **domain shifts** such as for example a change in location, weather, or the time of the day. On object level, corner cases are **single-point anomalies** or novelties. This can for example be a wild animal, e.g., a lion, appearing on the street, or walking aids such as a rollator, or crutches. For scene-level corner cases, we again distinguish between two types: collective and contextual anomalies. **Contextual anomalies** denote known objects in unusual locations as a tree in the middle of the street. **Collective anomalies** are known objects in unusual quantities such as a demonstration.

The highest complexity of detection have scenario-level corner cases, which are observed over the course of an image sequence. **Risky scenarios** have been observed in a similar fashion but still pose a potential for collision such as overtaking a cyclist. **Novel scenarios** have not been observed but do not increase the potential for collision such as accessing the freeway. **Anomalous scenarios** also have not been observed, but pose a very high potential for collision such as a person suddenly walking onto the street in front of the ego vehicle.

While in the introduction of the systematization [5], the different corner case levels were discussed in detail and suitable datasets and metrics were pointed out, in Section IV we extend this systematization by another dimension, following the approach of Lopez et al. [10]. In this dimension, different detection methods are grouped into broad categories and linked to the respective corner case levels. Moreover, we extend the columns in Figure 1 by a comprehensive list of examples which basically provides a playbook of corner cases.

III. Showcasing Corner Cases

In Table I, we provide examples for the corner case levels in Figure 1. This is supposed to clarify what corner cases can be found on each level of the systematization and serve almost as stage directions for possible corner case recordings. Moreover, it also gives an indication as to what content of datasets is needed to develop and test reliable corner case detectors. Again, Table I sorts the example situations by their respective corner case levels. The situations are described in some detail, so they can be directly translated into directions for data acquisition. Furthermore, the following sections will introduce categories of detection methods to later associate with respective corner case levels and thus, give some

guidelines to detect the example corner cases depicted in Table I.

IV. Concepts of Detection Approaches

We distinguish between five broad concepts of detection approaches: reconstruction, prediction, generative, confidence scores, and feature extraction. We subdivide the confidence score category into learned scores, Bayesian approaches and scores obtained by post-processing.

Reconstruction approaches are typically based on autoencoder-type networks. Most of those methods follow the paradigm that normality can be reconstructed more faithfully than anomalies. This causes reconstruction-based approaches to appear on every level of the corner case hierarchy. Especially, they can be applied to both the corner cases concerning single images and those comprising an entire image sequence. Hasan et al. [11] train an autoencoder both end-to-end and on handcrafted features where the reconstruction error serves as anomaly score. While they consider video sequences, there also exist similar approaches with single images as network input [12]. Some reconstruction approaches rely on prototypical learning. During training, prototypes of the normal samples are learned in the latent space, which during inference lead to a more faithful reconstruction of normal samples compared to anomalous ones [13]. Oza et al. [14] also utilize reconstruction in a class-conditioned autoencoder for open-set recognition, where next to the unsupervised open-set training they perform supervised closed-set training.

Prediction-based approaches can be mostly found on scenario level. Typically, they predict a future frame and later compare it with the true frame to detect any anomalies. Thus, they can be trained in a supervised manner where we assume that all training samples are normal. Such a method has been applied by Bolte et al. [9] specifically for corner case detection in automated driving. Another approach predicts future frames in videos using a generative adversarial network structure while ensuring appearance and motion constraints [15].

Generative and reconstruction-based approaches are closely related, since also this type of method can base its decision on the reconstruction error. Generative approaches, however, also regard the discriminator's decision or the distance between the generated and the training distribution. Moreover, some also borrow simply from related techniques such as adversarial training or perturbations. Lee et al. [16] introduce a confidence loss to enforce low confidence on out-of-distribution samples while also generating boundary out-of-distribution training samples for this task and jointly train the generative objective with a classification one. Adversarial-based training is also used to enforce uniform confidence predictions on noise images, leading simultaneously to decreased confidence for anomalous samples [17]. Based on generating images from masks, Lis et al. [18] identify unknown objects in the data by considering the error between the generated image and the original one. Generative methods, namely variational and adversarial autoencoders, have been applied to collective anomaly detection considering the error between the original and the generated image as anomaly score [19]. Also, domain shifts can be measured by generative methods. For example, representation learning guided by the Wasserstein distance [20] uses a general adversarial network inspired architecture, where a domain-critic network estimates the Wasserstein distance between source and target domain features. Löhdefink et al. [21] apply a generative adversarial network based autoencoder to detect domain shifts based on the earth mover's distance [22].

Methods relying on **confidence scores** are grouped into three categories to provide more clarity for this approach. Those that obtain their scores by post-processing, those that learn their scores, and those that rely on Bayesian approaches.

Confidence scores can be based on applying *post-processing* techniques to the neural networks without interfering with the training process. A baseline approach exists obtaining confidence scores by comparing softmax values against fixed thresholds [23]. Moreover, the scores are, for example, obtained by applying a Kullback-Leibler divergence matching to the softmax outputs during inference to compare with the class-specific templates obtained from a validation set in a multi-class prediction setting [24]. The method is trained in a supervised manner without requiring anomalous examples and gives segmentation maps as output. Another post-processing approach employs temperature scaling to the inputs [25]. It is based on the paradigm that for such modified normal inputs, the network is still able to infer the correct class but not for unknowns. It is also only supervised via normal training samples.

In contrast to obtaining confidence scores by post-processing, they can also be learned during training. In this category of *learned confidence scores*, we also include any method that relies on the training set in general to, e.g., provide a threshold. As an example for thresholds based on the training set, Shu et al. [26] compute three thresholds: one for acceptance of a sample, one for rejection and a distance-based one for when a sample falls in between the two other thresholds. As the threshold-based values are based on the training set, we consider the resulting confidence scores as learned. While the method is trained in a supervised manner, no corner case examples are required during training. The method then outputs a label as either one of the known classes or as unknown. Another way to obtain learned confidence scores is to borrow techniques from multi-task learning and include a second branch into the network to learn confidence scores [27]. Also in this case, training is done in a supervised manner, but only requiring normal samples. While originally used for classification, this method has been adapted to segmentation as well [24]. Learned confidence scores can be obtained by prototypical learning, where scores are based on the distance to normal training samples [28]. The open-max activation also provides learned confidence scores after a supervised training with normal samples with the aim to detect unknowns [29]. Learning to detect geometric transforms applied to the data from

Corner Case Level		Example
Scenario Level	Anomalous Scenario (potentially dangerous, unknown)	• We are driving through a street. On our right side, a large construction trailer is parked at the roadside. While we are driving next to it, suddenly a person steps onto the road. Before appearing on the road in front of the trailer, the person was fully occluded by it. • We are driving through a busy road with multiple lanes. Everyone is driving relatively fast. On the lane to our left, there is another car driving. It is slightly in front of us. Suddenly, the car changes into our lane without giving any indication first and it is causing us to brake heavily in order to avoid a collision. • We are driving in a street and are approaching a crossroad. The traffic signs indicate that we have the right of way, also indicated by stop signs on the other street. Due to the surrounding houses, we cannot see into the other street approaching the crossroad. While we drive onto the crossroad, another traffic participant does not abide to the traffic rules and is not stopping at the stop sign. They drive directly onto the crossroad at the same time as we do, not yielding our right of way.
	Novel Scenario (no potential danger, unknown)	• We drive towards a railroad crossing. The visual perception algorithm has not seen this before. The gates at this crossing are open. Thus, it is safe to cross. • We are driving to another city and want to use the freeway to get there. While the training data contained changing lanes and turnings, it did not contain the specific situation of accessing the freeway. • We are driving through a residential area in an urban environment. It is difficult to find parking spots in this area. We see a small car at the side of the road. Driving towards it, we first believe that it is driving onto the street, but closing in, we realize that it is actually parking orthogonally to the other cars to fit into an even smaller parking spot. This way of parking, we first interpreted as a driving car because of its positioning orthogonal to the street.
	Risky Scenario (potentially dangerous, known)	• We drive through a narrow street. Another car is driving towards us. As the cars pass each other, there is not much space left between them. • We drive through a street. While there is a sidewalk, there is no separate cycling lane. Driving further, there is a cyclist driving on the street. As the car is faster than the cyclist, we overtake them. • We drive through a street with only one lane for each direction. The motorized vehicle in front of us is going considerably slower than the allowed speed limit. When there is no oncoming traffic, we overtake the other vehicle by driving on the lane for the other direction.
Scene Level	Collective Anomaly	• In the city center, there is a demonstration and many people are walking and standing on the street, holding banners and shouting paroles. • It is in the evening and most people have left work and are on their way home. It is rush hour and we incur a major traffic jam. • We are at a crossroad. While it is usually governed by traffic signs, at the moment there is a huge construction site. Next to the usual signs, there are now many more signs to govern the behavior around the construction site.
	Contextual Anomaly	• We are in the city. The night before there was a huge storm and a tree fell on the street. • On a street, oil was spilled. While this is no longer visible, there are still traffic cones on the street, indicating to drive around the spot where it happened. • A car is parking on the sidewalk in a street with only few available actual parking spots.
Object Level	Single-Point Anomaly	• Unexpectedly, in a residential area, there is a bear in the middle of the street. • At a traffic light, a person with a rollator and a person on crutches cross the street. • It is a sunny day and we can see a shadow of a pedestrian approaching our street to cross, but the actual pedestrian is still occluded by a wall or the roadside.
Domain Level	Domain Shift	• It is February and it has snowed heavily the past couple of days. We are driving through the city and at the side of the road there are piles of snow everywhere. • We are going on vacation and drive to Great Britain by car all the way from continental Europe. After passing the Channel Tunnel, everyone is driving on the left. • We usually live in a city and the visual perception system has only been trained on urban data. On a sunny spring day, we decide to take a trip to the rural surrounding areas. • We are driving through a busy street in the city. It is rainy and dark. Moreover, there is a lot of oncoming traffic.
Pixel Level	Local Outlier	• Our camera fell down and now there are dead or broken pixels. • It is a windy day and there appears dirt on the windshield. • It is fall and a leaf has fallen onto our windshield.
	Global Outlier	• We are driving in a tunnel. It is a sunny day and at the end of the tunnel there is overexposure on our camera images because of the sun. • We drive at night and another car is coming towards us. The lights of this car have not been adjusted properly and we are blinded by them. • We are driving through a street during sunset. When we turn into another street leading west, we are blinded by the setting sun.

TABLE I: Example situations on each level of the corner case systematization as shown in Figure 1.

normal training data, also leads to learned confidence scores by assuming that these transforms can be more accurately detected on normal samples [30].

Bayesian confidence scores are usually obtained by an estimation of the model uncertainty, the epistemic uncertainty [31]. A network is trained to output a posterior distribution over its weights. Typical examples of this type of methods include the Monte Carlo dropout technique [31], [32] or deep ensembles [33]. The supervised training relies on normal training data. A current method to obtain model uncertainty scores is for example deterministic uncertainty quantification, which is based on ideas from radial basis networks [34]. In the semantic segmentation setting, Bayesian neural networks provide both class labels and an estimation of the model's uncertainty about it for each pixel. Common measures for this uncertainty include entropy and variance [32]. An example for the application of Monte Carlo dropout for uncertainty estimation in semantic segmentation is introduced by the Bayesian SegNet, which includes dropout units into the network architecture to obtain confidence maps for the model [35]. Pham et al. [36] use a Bayesian framework for instance segmentation in an open set recognition setting. An extension to include an entire time span has been achieved by considering the moving average over multiple frames [37].

Feature extraction approaches employ deep neural networks to extract features from the input data. These features are then either further processed using another technique or they are directly used to provide a classification label. In contrast to confidence scores, feature extraction methods either directly classify the sample as corner case or they use the extracted features in another way to obtain their decision. Confidence scores typically provide the scores next to their decided label. One such approach extracts features which are then fit on a hypersphere during training [38]. Thus, while the approach is unsupervised, it requires the training data to be considered normal because in inference, data is decided as anomalous if the distance on the hypersphere exceeds a threshold. When regarding video sequences, features can also be extracted from single frames and then be considered over a particular time interval to compare the probability distribution in the interval with the one outside [39]. Classification-reconstruction learning for open-set recognition (CROSR) also learns a feature representation for an unknown class detector [40]. The feature representation is composed of latent representations learned from reconstructing each intermediate layer of the network. The class membership is modeled via an extreme-value-theory-based distribution of the distances between extracted features of normal training data and the respective class mean [40]. Standard classification approaches train a network in a supervised manner, using the softmax function as activation on the last layer to obtain a corresponding class for the input sample. Jatzkowski et al. [41] utilize such an approach for overexposure detection. Feature extraction approaches are also found in domain adaptation methods. There, for example, cross-entropy based metrics [42], [43] are minimized in the adaptation process, indicating that they are valid measures of domain mismatch. Bolte et al. [44] consider the mean-squared error of extracted features as a measure of domain shift.

V. Associating Detection Approaches and Corner Case Levels

In this section, we associate the detection approaches of Section IV with the corner case levels in Figure 1, as was similarly done for smart manufacturing [10]. We discussed some examples for each detection concept in Section IV, which already hint at which type of corner case they can be suitably applied to. Moreover, we wish to prompt an idea on how to detect certain corner cases, as for example the ones listed in Table I. A summary of this section can be found in Table II, where we denote, which type of method has been applied to detect which corner case level. Additionally, we point out which approaches we believe to be leading to efficient and reliable future detection methods.

Overall, it can be said that due to the lack of large scale datasets containing all types of corner cases, and the associated open-world problem of corner case detection, unsupervised methods or ones trained only on normal samples currently seem to be the most effective way to obtain corner case detectors. Approaches dependent on anomalous training data need a more complex and specialized training set and run the risk of focusing on the specific corner cases related to its samples, thus turning a blind eye on the possibility of unknown corner cases appearing in inference.

On *pixel level*, to our knowledge only few deep learning approaches exist. But to detect such corner cases, for global outliers, feature extraction approaches provide promising results [41], since we aim to detect a corner case which influences large parts or even the entire image. In this case, the detection can be considered as a binary classification problem and a network is able to extract sufficient features for that task. Supervised training is possible because there is no unexpected amount of diversity for this type of corner case. Due to the lack of datasets for automated driving with labeled global outliers such as overexposure, however, an investigation of methods utilizing few-shot learning or similar techniques could be beneficial. Moreover, we are interested in the detection of multiple global outliers such as, for example, detecting overexposure and underexposure in images jointly. In the case of exiting a tunnel, they can even appear in the same image. This can be investigated in future work by considering joint or multi-task-learning.

Local outliers influence only a small portion of the image, as in the case of dead pixels. Detection of those corner cases can be learned supervisedly, as it can be simulated in the training data. Due to the possibility of simulation, detection could be treated in a semantic segmentation method by including another class. This will also lead to pixel-wise labels which will inform about the location of the dead pixels. We believe detection of local outliers will profit from taking a predictive approach, and thus including a time span. A predicted location of, for example, dead pixels can be compared to the actual location. Ideally, the actual location

Corner Case Level		Prediction	Reconstruction	Generative	Feature Extraction	Confidence Score		
						Post-processing	Bayesian	Learned
Scenario Level	Anomalous Scenario	✓⋆	✓		✓	✓⋆	✓	
	Novel Scenario	✓⋆	✓		✓	✓⋆		
	Risky Scenario	✓⋆	✓		✓	✓⋆	✓	
Scene Level	Collective Anomaly			✓		✓⋆		
	Contextual Anomaly				✓		✓⋆	
Object Level	Single-Point Anomaly			✓⋆	✓	✓⋆	✓⋆	✓⋆
Domain Level	Domain Shift			✓⋆	✓			
Pixel Level	Local Outlier	✓⋆						
	Global Outlier				✓			

TABLE II: Detection approaches are ascribed to corner case levels. A ⋆ denotes the suggested approaches to detect corner cases on that level in Section V.

is in contrast to the predicted one based on the learned optical flow.

To detect *domain-level* corner cases, we do not need to use domain adaptation methods but find suitable measures of domain mismatch. These measures, however, often stem from domain adaptation methods where they are utilized as loss functions. Typically, such measures are considered feature extraction approaches. While the training can require supervision by normal samples from a source domain, data from another domain for training should be explicitly excluded. Methods that employ specific examples of a second domain in training are in danger not to reach the same performance for a third domain. Bolte et al. [44] use the mean-squared error distance to measure the difference between features in the source and target domain in an unsupervised domain adaptation setting. It could also prove advantageous to consider out-of-distribution detection methods that are typically evaluated by considering one dataset as in- and another one as out-of-distribution. Out-of-distribution detectors are often evaluated by distinguishing between a dataset they have been trained on and another one [23], [16]. Those methods could be extended from a classification setting to the automotive visual perception setting, as they only require training supervised via normal samples. For reliable detection of domain-level corner cases, we require trustworthy measures of domain mismatch. To that end, we depend upon an evaluation using more than just one target domain. One such measure applying a generative adversarial network based autoencoder has been introduced previously providing a new domain mismatch metric based on the earth mover's distance [21].

On *object level*, the main goal is to detect *unknown unknowns* [45]. These are instances belonging to a new class not seen before in training. Providing examples of such corner cases during training would lead the network at inference to detect only corner cases similar to those examples, which is self-defeating to our aim. Detection of object-level corner cases falls into the broad area of open-set recognition and the related approaches usually provide some type of confidence score. Ideally, for detection and localization we ask for pixel-wise scores. There also exist reconstruction and generative methods which comply with this idea. However, reconstruction-based approaches tend to provide less meaningful results [18]. We would like to obtain a semantic segmentation mask for the input images where the pixels belonging to unknown objects are associated with an unknown class label or with a high amount of prediction uncertainty. With this goal in mind, pursuing confidence score and generative detection methods seems the most fruitful, and many recent methods comply with this indication [24], [18], [28]. Using Bayesian confidence scores, we ask for a model with high uncertainty associated to those unknown objects. Here, scalable methods for Bayesian deep learning applying Monte-Carlo dropout [32] or deep ensembles [33] provide a first step towards detection. In terms of the definition of those single-point anomalies as unseen instances during training, we conjecture that efficient and reliable detection approaches cannot rely on training samples including corner cases. Here, one has to resort to unsupervised approaches which can merely be trained using samples of normality.

On *scene level*, we aim to detect known classes in either unseen quantities or locations. Chalapathy et al. [19] employ generative methods to detect collective anomalies, which achieves promising results. Moreover, we believe future work should leverage instance segmentation to obtain a group size by counting the number of instances of each class. In this case, a threshold is required for defining a

collection as anomalous. Detecting contextual anomalies can be approached by employing feature extraction approaches [38]. However, in the case of automotive visual perception, feature extraction might not be able to capture the complexity of the entire scene. Hence, many existing methods give confidence scores [35], [33] or reconstruction errors [13] and distinguish between normal and anomalous samples. We propose to investigate how an incorporation of class priors influences the process as those priors might be able to facilitate the detection of misplaced class representatives. In the same light, confidence scores resulting from Bayesian deep learning indicate where the model is uncertain, and hence they can be useful to localize objects appearing in an unusual context. Both corner case types on scene level can be trained supervisedly with normal data since both detect instances of known classes, just either in an unusual location or quantity. However, in contrast to object-level corner cases, in this case while we might want pixel-wise semantic segmentation labels for our visual perception application, we additionally ask for instance-wise labels telling us an object appears in an unusual location, or image-wise labels if there appears an unseen quantity.

Scenario-level corner cases are comprised of patterns that appear over a particular time span and might not seem anomalous in a single frame. Here, prediction-based methods whose decision depends on the comparison between a predicted and the actual frame provide rewarding results [9]. Purely reconstructive methods obtain again less faithful corner case detection scores. Predictive approaches can be trained supervisedly, as they only require normal training samples in order to detect corner cases during inference. This is especially important for novel and anomalous scenarios, where we cannot capture every possibility due to the infinite number and the considerable danger of the corresponding circumstances. Also, including samples could actually prejudice the network to only detect such scenarios. For future work, we need to define adequate metrics for detection of this type of corner case. While we might still want to know the location of the corner case in the image, we also require the point of time when such a corner case happens. To achieve this, we could consider image-wise labels over a certain time span. Next to an investigation of metrics, we suggest the use of a cost function to give higher priority to the detection of vulnerable road users appearing at the verge of the visual field. This could improve, for example, the detection of a person running onto the street from behind an occlusion, as the person could already detected when just a few human pixels appear in the frame. Such an approach also requires a frame-wise mask identifying pixels which have not been included in the previous one because they have been occluded or outside the field of view.

While the detection approaches for all corner case categories have been treated separately, we also need to discuss the concept of a general corner case metric. Considering, there already is an ideal corner case detector available, that we want to apply to select training data for a visual perception module. It takes as input an entire video sequence with all types of corner cases contained in it, the question poses how to report the results. While we, e.g., suggest to report pixel-wise labels on object level but image-wise ones on pixel level, it needs to be clarified that in the end this needs to be combined to a general metric, which expresses if a video sequence contains corner cases, and thus qualifies as necessary training data. Here, one could consider a type of averaging metric similar to common velocity measures.

VI. Conclusions

After reviewing the systematization of corner cases, we introduced a more detailed list of examples meant for deeper understanding of the previously proposed categories and enabling direct application for the acquisition of corner case data. Moreover, we extended the corner case systematization by covering detection approaches and their respective categories. Afterwards, we associated the detection approaches to corner case levels, and additionally provided some basic guidelines on how to detect certain types of corner cases. Hence, we are able to portray specific corner case examples and follow coarse guidelines for a baseline detection approach.

Acknowledgment

The authors gratefully acknowledge support of this work by Volkswagen AG, Wolfsburg, Germany.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks," in *Proc. of NIPS*, Montréal, QC, Canada, Dec. 2015, pp. 91–99.

[2] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. of MICCAI*, Munich, Germany, Oct. 2015, pp. 234–241.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *Proc. of ICCV*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[5] Jasmin Breitenstein, Jan-Aike Termöhlen, Daniel Lipinski, and Tim Fingscheidt, "Systematization of Corner Cases for Visual Perception in Automated Driving," in *Proc. of IV*, Las Vegas, NV, USA, Oct. 2020, pp. 986–993.

[6] 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, Oct. 2016, pp. 1099–1106.

[7] S. Ramos, S. Gehring, P. Pinggera, U. Franke, and C. Rother, "Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling," in *Proc. of IV*, Redondo Beach, CA, USA, June 2017, pp. 1025–1032.

[8] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving," in *Proc. of ICCV - Workshops*, Seoul, South Korea, Oct. 2019, pp. 1–10.

[9] J.-A. Bolte, A. Bär, D. Lipinski, and T. Fingscheidt, "Towards Corner Case Detection for Autonomous Driving," in *Proc. of IV*, Paris, France, June 2019, pp. 438–445.

[10] F. Lopez, M. Saez, Y. Shao, E. C. Balta, J. Moyne, Z. M. Mao, K. Barton, and D. Tilbury, "Categorization of Anomalies in Smart Manufacturing Systems to Support the Selection of Detection Mechanisms," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.

[11] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning Temporal Regularity in Video Sequences," in *Proc. of CVPR*, Las Vegas, NV, USA, June 2016, pp. 733–742.

[12] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning Discriminative Reconstructions for Unsupervised Outlier Removal," in *Proc. of ICCV*, Santiago, Chile, Dec. 2015, pp. 1511–1519.

[13] D. Gong, L. Liu, Vuong Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection," in *Proc. of ICCV*, Seoul, Korea, Oct. 2019, pp. 1705–1714.

[14] P. Oza and V. M. Patel, "C2AE: Class Conditioned Auto-Encoder for Open-set Recognition," in *Proc. of CVPR*, Long Beach, CA, USA, June 2019, pp. 2307–2316.

[15] W. Liu, W. Luo, D. Lian, and S. Gao, "Future Frame Prediction for Anomaly Detection – A New Baseline," in *Proc. of CVPR*, Salt Lake City, UT, USA, June 2018, pp. 6536–6545.

[16] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin, "Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples," in *Proc. of ICLR*, Vancouver, BC, Canada, Apr. 2018, pp. 1–16.

[17] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why ReLU Networks Yield High-Confidence Predictions Far Away From The Training Data And How To Mitigate The Problem," in *Proc. of CVPR*, Long Beach, CA, USA, June 2019, pp. 41–50.

[18] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the Unexpected via Image Resynthesis," in *Proc. of ICCV*, Seoul, Korea, Oct. 2019, pp. 2152–2161.

[19] R. Chalapathy, E. Toth, and S. Chawla, "Group Anomaly Detection Using Deep Generative Models," in *Proc. of ECML PKDD*, Dublin, Ireland, Sept. 2019, pp. 173–189.

[20] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein Distance Guided Representation Learning for Domain Adaptation," in *Proc. of AAAI*, New Orleans, LO, USA, Feb. 2018, pp. 4058–4065.

[21] Jonas Löhdefink, Justin Fehrling, Marvin Klingner, Fabian Hüger, Peter Schlicht, Nico M. Schmidt, and Tim Fingscheidt, "Self-Supervised Domain Mismatch Estimation for Autonomous Perception," in *Proc. of CVPR - Workshops*, Seattle, WA, USA, June 2020, pp. 1–10.

[22] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, Nov. 2000.

[23] Dan Hendrycks and Kevin Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks," in *Proc. of ICLR*, Toulon, France, Apr. 2017, pp. 1–12.

[24] Dan Hendrycks and Thomas Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," in *Proc. of ICLR*, New Orleans, LA, USA, May 2019, pp. 1–15.

[25] S. Liang, Y. Li, and R. Srikant, "Enhancing the Reliability of Out-Of-Distribution Image Detection in Neural Networks," in *Proc. of ICLR*, Vancouver, Canada, Apr. 2018, pp. 1–27.

[26] Yu Shu, Yemin Shi, Yaowei Wang, Yixiong Zou, Qingsheng Yuan, and Yonghong Tian, "ODN: Open Deep Network for Open-Set Action Recognition," in *Proc. of ICME*, San Diego, CA, USA, July 2018, pp. 1–6.

[27] Terrance DeVries and Graham W. Taylor, "Learning Confidence for Out-of-Distribution Detection in Neural Networks," *arXiv preprint arXiv:1802.04865*, Feb. 2018.

[28] Chen Xing, Sercan Arik, Zizhao Zhang, and Tomas Pfister, "Distance-Based Learning from Errors for Confidence Calibration," in *Proc. of ICLR*, Addis Ababa, Ethopia, Apr. 2020, pp. 1–12.

[29] A. Bendale and T. Boult, "Towards Open Set Deep Networks," in *Proc. of CVPR*, Las Vegas, NV, USA, June 2016, pp. 1563–1572.

[30] I. Golan and R. El-Yaniv, "Deep Anomaly Detection Using Geometric Transformations," in *Proc. of NIPS*, Montréal, Canada, Dec. 2018, pp. 9781–9791.

[31] 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, Dec. 2017, pp. 5574–5584.

[32] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. of ICML*, New York, NY, USA, June 2016, pp. 1050–1059.

[33] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles," in *Proc. of NIPS*, Long Beach, CA, USA, Dec. 2017, pp. 6402–6413.

[34] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal, "Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network," *arXiv preprint arXiv:2003.02037*, Mar. 2020.

[35] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding," *arXiv*, , no. 1511.02680, Nov. 2015.

[36] T. Pham, V. B. G. Kumar, T.-T. Do, G. Carneiro, and I. Reid, "Bayesian Semantic Instance Segmentation in Open Set World," in *Proc. of ECCV*, Munich, Germany, Sept. 2018, pp. 3–18.

[37] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, "Efficient Uncertainty Estimation for Semantic Segmentation in Videos," in *Proc. of ECCV*, München, Germany, Sept. 2018, pp. 536–552.

[38] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep One-Class Classification," in *Proc. of ICML*, Stockholm Sweden, July 2018, pp. 4393–4402.

[39] B. Barz, E. Rodner, Y. G. Garcia, and J. Denzler, "Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1088–1101, May 2019.

[40] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-Reconstruction Learning for Open-Set Recognition," in *Proc. of CVPR*, Long Beach, CA, USA, June 2019, pp. 4016–4025.

[41] I. Jatzkowski, D. Wilke, and M. Maurer, "A Deep-Learning Approach for the Detection of Overexposure in Automotive Camera Images," in *Proc. of ITSC*, Maui, HI, USA, Nov. 2018, pp. 2030–2035.

[42] Yuliang Zou, Zelun Luo, and Jia-Bin Huang, "DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency," in *Proc. of ECCV*, Munich, Germany, Sept. 2018, pp. 36–53.

[43] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation," in *Proc. of ECCV*, Munich, Germany, Sept. 2018, pp. 801–818.

[44] Jan-Aike Bolte, Markus Kamp, Antonia Breuer, Silviu Homoceanu, Peter Schlicht, Fabian Hüger, Daniel Lipinski, and Tim Fingscheidt, "Unsupervised Domain Adaptation to Improve Image Segmentation Quality Both in the Source and Target Domain," in *Proc. of CVPR - Workshops*, Long Beach, CA, USA, June 2019, pp. 1404–1413.

[45] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability Models for Open Set Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.