Introspective Failure Prediction for Autonomous Driving Using Late Fusion of State and Camera Information

Christopher B. Kuhn, *Graduate Student Member, IEEE*, Markus Hofbauer, *Graduate Student Member, IEEE*, Goran Petrovic, and Eckehard Steinbach, *Fellow, IEEE*

***Abstract*— We present an introspective failure prediction approach for autonomous vehicles. In autonomous driving, complex or unknown scenarios can cause a disengagement of the self-driving system. Disengagements can be triggered either by automatic safety measures or by human intervention. We propose to use recorded disengagement sequences from test drives as training data to learn to predict future failures. The system then learns introspectively from its own previous mistakes. In order to predict failures as early as possible, we propose a machine learning approach where sequences of sensor data are classified as either failure or success. The car itself is treated as a black box. Our method combines two sensor modalities that contain different types of information. An image-based model learns to detect generally challenging situations such as crowded intersections accurately multiple seconds in advance. A state data based model allows to detect fast changes immediately before a failure, such as sudden braking or swerving. The outcome of the individual models is fused by averaging the individual failure probabilities. We evaluate our approach on a data set provided by the BMW Group containing 14 hours of autonomous driving. The proposed late fusion approach allows for predicting failures at an accuracy of more than 85% seven seconds in advance, at a false positive rate of 20%. The proposed method outperforms state-of-the-art failure prediction by more than 15% while being a flexible framework that allows for straightforward addition of further sensor modalities.**

***Index Terms*— Failure prediction, machine learning, autonomous driving, introspection.**

I. Introduction

AUTONOMOUS systems such as drones and, more recently, cars have become increasingly popular over the last years. Most research effort has been focused on improving the performance of such systems. Achieving error-free autonomous operation is still a highly challenging task. In applications such as driving on public roads, errors can have serious consequences. The safety of autonomous driving has therefore received significant attention [1]. However, failures of autonomous vehicles remain inevitable. A possible solution is teleoperation, where an expert human driver takes control of the car remotely when needed [2]. This requires a system to determine when a human should be back in the loop. Such a system needs to automatically detect failures of the autonomously driving car. After a failure is detected, the car requests a human to take over control. The human operator can then determine the right course of action such that a potentially disastrous failure can be avoided.

There are some existing approaches for determining whether machine learning based models can be trusted or whether control should be handed over to a human driver. For example, the uncertainty of a model's output can be estimated [3], [4]. However, this only allows estimating the uncertainty of one given model. Autonomous vehicles can consist of many different components. A disengagement of the vehicle might not be traceable to a single component. We are therefore interested in designing a method that treats the car as a black box and does not make any assumptions about the underlying system. Such a method could be applied to any kind of autonomous system regardless of the design.

Another important property of automatic failure detection in dynamic systems such as cars is that it is predictive as well. Detecting failures reactively might not allow for enough time to ensure a safe takeover by a human operator. Most existing works that predict failures for autonomous systems focus on the next one or two seconds [5]. We argue that starting to look for failures even earlier is a reasonable approach. Surroundings that are likely to be challenging such as crowded intersections can often be anticipated many seconds in advance. In this paper, we investigate how failures can be predicted in advance using concepts from deep learning. In the context of autonomous driving, we define failures to be the disengagements of the autonomous mode of the car. We follow the idea of introspection as introduced by Daftry *et al.* [6], using previous failures to learn to predict future ones. The car is treated as a black box, meaning the underlying system can be arbitrarily complex. Only the input and the resulting output, i.e., whether the car is driving successfully or not, is required. No changes to the vehicle need to be made. To the best of our knowledge, our approach is the first to

Manuscript received July 9, 2020; revised October 8, 2020 and November 30, 2020; accepted December 10, 2020. Date of publication December 29, 2020; date of current version May 3, 2022. The Associate Editor for this article was A. Y. Lam. *(Corresponding author: Christopher B. Kuhn.)*

Christopher B. Kuhn is with the BMW Group, 80788 München, Germany, and also with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 München, Germany (e-mail: christopher.kuhn@bmw.de).

Markus Hofbauer and Eckehard Steinbach are with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 München, Germany (e-mail: markus.hofbauer@tum.de; eckehard.steinbach@tum.de).

Goran Petrovic is with the BMW Group, 80788 München, Germany (e-mail: goran.petrovic@bmw.de).

Digital Object Identifier 10.1109/TITS.2020.3044813

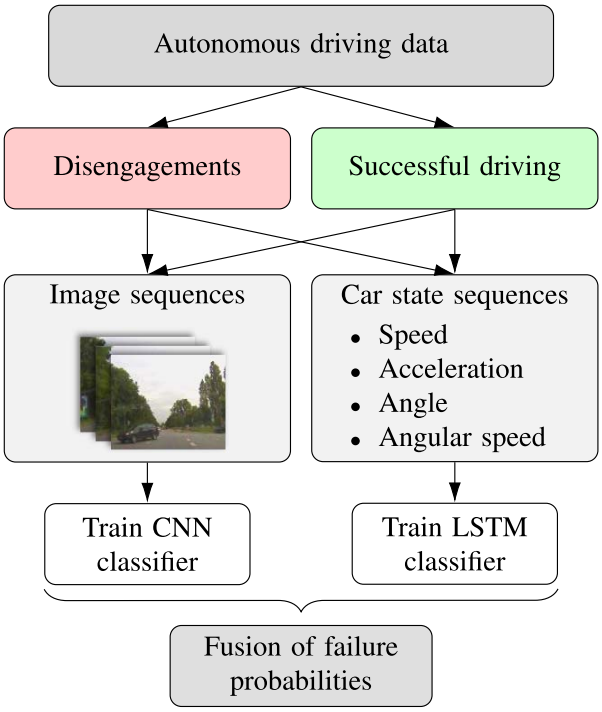

Fig. 1. Overview of the proposed approach. A state-based and an image-based model are trained to differentiate between successful driving and disengagements. The individual failure probabilities are then fused to compute the final failure probability.

use disengagements from autonomous cars driving on public roads to learn explicitly to predict failures.

In a previous work, we showed a first proof of concept of this idea [7]. We used sequences of state data to predict disengagements of an autonomous car up to seven seconds in advance at an accuracy of around 80%. Here, we further develop this idea and design a framework that significantly outperforms our previous work. In addition to state data, we analyze how to incorporate image sequences as another sensor modality. Then, we propose to exploit the failure information provided by both sensor types by fusing the output of the individual models. We implement and test our framework with driving data from public roads. For this, we use data collected over more than six months by BMW research vehicles, driving in their autonomous mode. In Figure 1, we summarize the overall process of the proposed failure prediction approach.

First, test drives from autonomous cars are recorded. From those recordings, all driving sequences ending in disengagements are extracted. An equal number of undisrupted driving sequences is sampled from the remainder of the data. Then, different data sets are constructed. In this work, we design two separate models, one using image sequences as input, one using sequences of car state data. After training a convolutional neural network (CNN) classifier with image sequences and a Long Short-Term Memory (LSTM) model with state sequences, we combine the individual results by averaging the predicted failure probabilities. The resulting model predicts failures at almost 90% accuracy at a false positive rate of 20%. This outperforms our approach from [7] by almost 10% and previous state-of-the-art failure prediction techniques by over 15%.

In summary, our main contributions are:

- A thorough discussion of how introspection can be used to predict failures in autonomous driving. We discuss what the term "failure" can mean and why introspection is a promising approach for achieving accurate failure prediction.
- We significantly extend our work from [7] by thoroughly investigating how images can be used as another sensor modality. We explore how sequences of images can be efficiently used and compare several architectures to exploit the rich information present in images.
- As a second extension of [7], we combine the individual models using late fusion to allow for a flexible and modular framework. We show that images are key to assess the general complexity of a scene, whereas state data allows detecting fast changes immediately before a failure. We train separate models for each sensor modality. We show that using both state and image data improves the accuracy of our previous work from [7] by up to 10%.
- Compared to [7], we evaluate our approach on a significantly larger and thus more challenging data set containing more unique failure cases, increasing the number of disengagements from over 1000 to over 2500.

The rest of the paper is organized as follows. Section II summarizes related work in the field of failure prediction. We discuss the theoretical framework of introspective failure prediction in Section III. Next, we present the design of different failure prediction models based on different sensor modalities in Section IV. In Section V, we discuss the experiments we performed to validate our method and show the results. Section VI concludes the paper.

II. Related Work

To the best of our knowledge, there are no other works that predict failures of real autonomous cars driving on public roads. In this section, we present relevant approaches from the general field of failure prediction. In the literature, the terms prediction and detection are sometimes used interchangeably. In this paper, we use prediction in the temporal sense, meaning an impending failure is detected seconds before it happens. We divide related work into two main categories. First, we present approaches that only detect failure for the current input. Then, we summarize existing methods that focus on predicting critical events in advance.

A. Failure Detection

First, we present existing works focused on the detection of failures. A relevant research field for failure detection is uncertainty estimation. In the context of autonomous driving, a highly uncertain vehicle should request a human to take over even if it currently is not making mistakes yet. A common distinction is between aleatory and epistemic uncertainty [8]. Aleatory uncertainty is inherent to the input and cannot be

reduced by a better model, for example due to noisy input. Epistemic uncertainty is uncertainty caused by lack of training data or an imperfect model. Our approach does not distinguish between sources of uncertainty and learns from failures caused by both types of uncertainty.

1) Uncertainty for Neural Networks: Most existing works about uncertainty estimation are focused on neural networks. Several studies examined how to integrate Bayesian theory into neural networks [9]–[11]. An influential idea in this field is Monte Carlo (MC) dropout introduced by Gal and Z. Ghahramani [3]. They generated multiple predictions from the same neural network using different dropout masks. The variance of the resulting predictions can be used as an uncertainty estimation. While MC dropout has since been used for tasks relevant to autonomous driving such as semantic segmentation [12] and object detection [13], it is computationally expensive as many forward passes are required for each input. Gurau *et al.* [14] used distillation [15] to distill the uncertainty estimation obtained from many MC dropout samples into a student model designed to estimate uncertainty. During testing, only the student model needs to be evaluated for uncertainty estimation. Another relevant Bayesian approach is *Bayes by backprop* [16], [17]. The idea is to learn a distribution for each weight instead of a singular value. Lakshminarayanan *et al.* [4] proposed using the variance of deep ensembles, outperforming MC dropout in their experiments. Wang et al. integrated uncertainty estimation into the training process by training a hardness predictor alongside the model [11]. These approaches cannot be added retroactively to an existing complex system such as an autonomous vehicle. They can also only be used for specific neural network architectures. In contrast, our approach can be used for any kind of system.

2) Out-of-Distribution Detection: Failure detection can also be approached using out-of-distribution detection. Input that differs significantly from the training data is likely to cause failures. A common issue of existing novelty detection approaches is that learning a typical distribution of high-dimensional data such as images is challenging [18]. This makes it difficult to use for autonomous driving. Chalapathy and Chawla [19] proposed to address this issue by integrating anomaly detection into the training process. This again requires the system to be one model, making it unsuitable for a complex pipeline such as an autonomous car.

B. Failure Prediction

Besides automatically detecting failures, we are interested in predicting them as early as possible. There are several related approaches to predicting future events.

1) Human Driving Prediction: Anticipating events before they occur is an important part of successful driving. Methods from deep learning have shown great potential for this task. Convolutional layers can be extended to 3D to also learn patterns over time, allowing spatio-temporal predictions as done in [20]. Saxena and Cao [21] successfully used Generative Adversarial Nets (GANs) to perform spatio-temporal predictions. Another option is to use recurrent neural networks (RNNs) such as LSTMs for time-series predictions [22]. Instead of feeding an input sequence to a model directly, Fridman *et al.* [23] created an image composed of three difference images as a compact representation of an image sequence. Kuefler *et al.* [24] used imitation learning to learn from recorded human driving. Their GAN model achieves realistic predictions of future human actions in a simulation. Brain4cars [25] used an LSTM approach to predict human maneuvers seconds in advance, allowing them to alert drivers of impending problems. Hallac *et al.* [26] trained an LSTM to predict the entire state of the car. While those works show that predicting the future state of the car is possible, they allow to predict failures only implicitly by predicting critical events such as hard breaking.

2) Crash Prediction: Some papers approach failure prediction more explicitly by predicting crashes. Tian *et al.* [27] used crash videos from public video sources such as the news to detect accidents. Similarly, Huang *et al.* [28] created a near-accident data set to predict dangerous situations. Both use object tracking and detection to monitor other traffic participants. Finally, Chan *et al.* [29] used dashcam accident videos to train an RNN that predicts accidents before they happen. References [30] and [31] further improved the prediction of traffic accidents using recurrent neural networks. However, we are interested in predicting accidents for the ego vehicle rather than for other cars.

3) End-to-End Failure Prediction: Some works proposed to predict the failures of end-to-end driving models. Huang *et al.* [32] predicted failures of a deep learning trajectory predictor trained with human driving data by observing differences to a physics-based predictor. While they used fusion of state data and images to train their trajectory predictor, we use it to directly predict failures. Hecker *et al.* [5] trained an end-to-end network predicting steering angles as a simulated autonomous driving system. They monitored disagreements with human driving data and used the resulting data to train a dedicated failure prediction model to predict large steering angle deviations. Similar to our work, they also use state and image data as input for their failure prediction model. Both those works define failures as deviations of the steering angle from human driving data. However, a differing steering angle is not necessarily a failure. Additionally, these approaches require human driving data as a reference. Our approach does not perform a comparison to human driving, but is entirely based on data generated with the autonomous vehicle itself. It can therefore be applied to any kind of autonomous car, requires no human expert data and poses no restrictions on the type of failure. Fridman *et al.* [23] monitored the disagreements between an end-to-end network predicting the steering angle and a Tesla car in autopilot to predict disengagements on highways up to five seconds in advance. Michelmore *et al.* [33] monitored the uncertainty of an end-to-end network and observed that the uncertainty spikes several seconds before a crash occurs in a driving simulator. Since those two approaches do not require human driving data, we use them as reference approaches.

4) Introspection: The idea of introspection as introduced by Daftry *et al.* [6] allows failure prediction while treating the underlying system as a black box. They used previous

failures of a system to train a separate failure prediction model. They predicted whether the intended trajectory of an autonomous drone will lead to a crash by training a spatio-temporal CNN with image sequences from previous crashes. This allowed them to fly with fewer collisions. However, this idea has only been used for drones or for specific tasks such as obstacle detection and semantic segmentation [34]–[36]. In our previous work [7], we transferred the idea to autonomous driving and showed the concept can be applied to highly complex systems as well. In [7], we used state data to predict disengagements at 80% accuracy up to seven seconds in advance. In this paper, we build and expand on this approach.

C. Failure Data Sets

An important requirement for machine learning based failure prediction approaches is the availability of suitable data. Little data from autonomous test drives is made public. Waymo has released a data set of driving in autonomous mode [37], but they only show successful sequences, not the failures. Some public data sets containing driving failures exist, including the Dashcam Video data set [29], a crash data set based on news videos from [27] or a near-accident data set from [28]. Those data sets consist of failures caused by humans, not by autonomous vehicles. Additionally, the crashes are caused by many different drivers with different behaviors. Since we want to learn to predict the failures of one specific type of autonomous vehicle, we require failures from the same source. One of the few existing works predicting disengagements generated their own data by recording highway driving in a Tesla in autopilot mode [23]. However, this is restricted to highway driving and assisted driving functions. To the best of our knowledge, no data set of recorded failures of fully autonomous vehicles is publicly available. In this work, we will address this issue by creating our own data set from data recorded with BMW test vehicles driving in autonomous mode.

III. Introspection for Autonomous Driving

In this section, we discuss the framework of introspection and how it can be applied to autonomous driving. In general, the idea of introspection can be used for any kind of autonomous vehicle. In the literature, introspection has been used for drones [6] and rovers [35] as well as cars [7]. Since the task of autonomous driving is highly complex, specific conditions need to be considered. We therefore first discuss the concept of introspection with regards to our problem setting. Then, we discuss what the term "failure" means in this context and how we approach the goal of predicting it. Finally, we present the data used in this work. Since autonomous driving is closely connected to competing companies where little data is made public, having sufficient training data is a challenge. For this work, we were supplied with test data recorded with BMW research vehicles.

A. Concept of Introspection

The concept of introspection was first introduced by Daftry *et al.* [6]. Their goal was for a system to "know when it does not know". This can be done by observing a system and recording which inputs led to failure and which led to successful behavior. After a sufficient number of failures has been observed, an introspective model can be trained to learn where or when those failures occurred. In their work, they train a CNN that learns from both temporal and spatial data. Their model learns common causes of failure, for example that a drone is more likely to crash after sudden illumination changes in the surroundings. More generally, Daftry et al. argue that training a separate failure prediction model can be more useful than deriving the uncertainty from the model itself. A separate model can learn different features and can predict failures in the cases where the baseline model is overconfident, but wrong. The assessment of the introspective model can still be combined with model-based confidence measures.

Besides this theoretical advantage of introspection over model-specific confidence metrics, the idea of using previous failures as training data is a powerful concept. It allows to generate training data automatically without additional effort. The evaluation of previous test trials as either successes or failures implicitly generates labels for the corresponding set of inputs. The core requirement of introspection is therefore that previous trials of the system are available. It is challenging to use this to improve a system in its initial phase. For complex systems such as autonomous driving, an iterative testing phase is a necessary part of development. Introspection therefore allows to make use of the evaluation that takes place anyway. If the fail cases of a system change over time, the introspective model can be adjusted by retraining with the most recent failure examples. No assumptions on the underlying system are made. Once failures of a given system have been observed, the goal is to find patterns among those failures that can be detected in future scenarios as well.

Since the main requirement of introspection is to obtain failures of a system, we first discuss what the term failure can mean in the context of autonomous driving.

B. Failures in Autonomous Driving

Depending on the domain, a failure of a system can mean different things. In general, any metric measuring the success of a system can be used to determine what a failure is. Next, we discuss how we define failure in this work.

1) Definition of Failure: In driving, one of the most relevant failures that needs to be avoided is crashing. In testing of autonomous driving, crashes occur very rarely. In an analysis of reported data of autonomous driving test drives from 2016 [38], there were a total of twelve crashes in a time period of over a year. Crashes are therefore not a suitable failure definition in autonomous driving with regards to collecting failures as training data. A more relevant event is the disengagement of the autonomous system. Currently, there are safety drivers present in most autonomously driving cars, meaning the car can give up control as often as necessary to maintain safety standards. Even without a driver in the vehicle, teleoperation still allows a car to request human intervention if a situation becomes too challenging. A disengagement therefore indicates that the car cannot guarantee safe driving anymore. While

an actual crash would only occur later, being able to predict disengagements is a useful approximation of being able to predict potential crashes.

As outlined in [38], disengagements are significantly more common than crashes, ranging from 1 to 100 for every 1000 test miles driven. In this work, we therefore define failure as the moment the autonomous mode of a car disengages. There are two types of disengagements to consider. First, the car itself triggers a request for a human driver to take control. Second, a human safety driver can intervene despite the car not requesting a takeover. Both types of disengagements are discussed next.

2) Automatic Disengagement: When exactly a car disengages depends highly on how the autonomous system is designed. Potential disengagement causes can be issues with perception such as unrecognized objects or impending violations of traffic rules. In reports about disengagements, specific details of which part of the autonomous system failed are usually not disclosed. The black box assumption of introspection as proposed in this work means we do not need this information. Introspection can be applied regardless of why the car decided to disengage. We therefore focus on detecting that the car will disengage, not why. With data obtained from automatic disengagements, the introspective model implicitly attempts to learn the rules of the existing safety systems. However, it does so using only the inputs of the car. If the existing safety system makes a mistake, the introspection model can still recognize that the situation resembles previously challenging scenarios and predict a failure. The current assessment of the system is not considered, only its previous performance.

3) Manual Disengagement: Besides learning to detect common patterns among automatic disengagements, introspection can also learn from human expert knowledge. During current tests on public roads, vehicles are always monitored either by a safety driver in the car or remotely as done by some test vehicles of Waymo [39]. In practice, model-intrinsic confidence measures cannot detect all problems. An expert human supervisor guarantees that almost no dangerous situation is missed. Usually, human labeling of data is an expensive additional cost. In autonomous driving, a human is monitoring the car anyway during test drives. Each manual takeover is effectively a manual labeling of the current situation as a failure scenario. Introspection allows exploiting this necessary safety feature of having a safety driver to obtain labeled data. In this work, we aim to predict that driving will become unsafe as indicated by a disengagement. We want to find patterns among all dangerous situations and therefore do not distinguish between the source of the disengagement. Differentiating between automatic and manual disengagements could be an interesting direction for future work.

4) Autonomous Driving Failure Data: A key challenge of using introspection to predict disengagements is the availability of disengagement data to learn from. In this work, we are interested in analyzing fully autonomous driving. Since no suitable data set is publicly available, we cooperated with the BMW Group that is actively testing their autonomous driving research on highway and urban roads near Munich, Germany. Working with data from real hardware tested on public roads is a valuable component of researching and improving autonomous driving. We were supplied with the data from autonomous test drives from over six months. A safety driver was always monitoring the car and took over control whenever they deemed the situation to be dangerous. Driving took place in various road, weather, and daytime settings. Construction sites were also regularly present. This makes the data set a challenging, yet realistic representation of driving. A summary of the data used in this work is given in Table I.

TABLE I
OVERVIEW OF THE DATA USED IN THIS WORK. THE DATA SET CONSISTS OF OVER 14 HOURS OF DRIVING IN AUTONOMOUS MODE

Number of disengagement sequences	2549
Number of undisrupted sequences	2549
Sequence length	10 seconds
Sensor frequency	10 Hz
Number of image frames	5 098 000
Selected state variables	Speed v Frontal acceleration a_x Lateral acceleration a_y Steering angle θ Angular speed ω

The bottleneck of obtaining data for introspection is the number of failure cases. In our previous work [7], we had around 1000 disengagements to learn from. In the meantime, continued test drives led to encountering more challenging scenarios. In total, over 2500 disengagements have been recorded. The increased number of unique failure cases makes it a more realistic test scenario for evaluating our approach. The fact that we could increase the data set size by 150% over a few months highlights the advantage of using introspection as proposed here. Test drives are performed during research anyway and generate this data as a side product. We can effectively make use of month-long human supervision and the resulting labeling of dangerous situations.

In our previous work, we have used state data of the car to successfully predict failures [7]. In this work, we consider both the state and the image input. The state consists of the speed, the angle, the acceleration and the angular velocity. We choose a frequency of 10 Hz for all sensors following works such as [5].

We use sequences of 10 seconds before each disengagement as training data. Additionally, we evenly sample the same number of 10 second sequences from the successful driving scenes. We argue that in 10 seconds, the environment can already change significantly. A car traveling at 40km/h moves over 100 meters in this time, for instance. We therefore do not start looking for failures any earlier. With over 2500 failure and 2500 success sequences, our data set contains 14 hours of driving resulting in over 5 million image frames.

Having access to hardware implementations of autonomous driving in a fleet of test vehicles is highly useful for obtaining large amounts of realistic data. While the data set we used is not public, such data can be generated for any autonomous

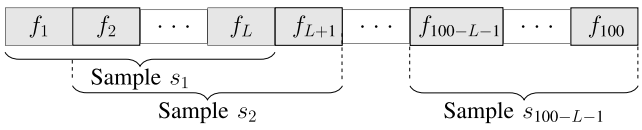

Fig. 2. Generation of training samples from one sequence of 100 feature vectors from [7]. Each sample consists of L feature vectors which corresponds to $L/10$ seconds of driving.

system where test runs are available. The only requirements are common sensors such as cameras and measurements of the system state. With a large, diverse and real-life data set of failures available, the next step is to discuss how models can be designed that allow predicting failures seconds in advance.

IV. Model Design for Failure Prediction

In this section, we discuss how failure prediction models can be designed. As outlined in Figure 1, we implement both a state-based and an image-based approach using concepts from machine learning. Then, we propose a late fusion strategy to combine the strengths of both sensor modalities.

A. State-Based Failure Prediction

Here, we summarize the state-based failure prediction approach first introduced in our previous work [7]. First, we discuss how we preprocess the raw recording data to generate a training set. Then, we present the architecture of our proposed failure prediction model.

1) Data Preprocessing: Our data consist of 10 second sequences with 10 readings per second, for a total of 100 sensor readings. Here, we only use the state information of the car. In one sequence, we therefore have 100 feature vectors f_i given by

$$f_i = [v_i \quad \theta_i \quad a_{x,i} \quad a_{y,i} \quad \omega_i], \qquad i \in \{1, 100\} \tag{1}$$

where v_i denotes the car's speed, θ_i the steering angle, $a_{x,i}$ the frontal acceleration, $a_{y,i}$ the lateral acceleration and the angular speed ω_i.

For an introspective model to learn temporal patterns, we need to use sequential data. Training the model directly with the entire 10 second sequences would make it learn very long term patterns. We are interested in the model detecting fast changes leading to failures as well. We therefore split the 10 second sequence into many shorter samples of length L. Figure 2 visualizes how one sequence is structured to obtain multiple samples.

The value of L decides the temporal length of the patterns the model can learn. In [7], we evaluated different sample lengths, with three second samples leading to the best performance. At 10 Hz, the length of one sample is then $L = 30$. From a sequence of length 100, we can therefore extract up to 71 samples. After observing N_F disengagements, we obtain N_F failure sequences that we divide into 71 failure samples $s_{n_F,i}$ as shown in Figure 2. We evenly sample N_F success sequences from all recordings of undisrupted driving, each one 10 seconds long and resulting in 71 success samples $s_{n_S,j}$. The balanced data set D_{state} is then given by

$$D_{\text{state}} = \{s_{n_F,i}, s_{n_S,j}\}, \quad i, j \in \{1, 71\}, \; n_{F,S} \in \{1, N_F\} \tag{2}$$

Each sample from a failure sequence is labeled as "failure" and each sample from a success sample is labeled as "success". Next, we can train a model with our state-based data set D_{state}. For all models and experiments, we randomly split each data set into 80% training, 10% validation and 10% testing sequences.

2) Model Design: We approach introspective failure prediction as a binary classification task. We train a model to classify each sample as belonging to either a failure or a success sequence. We therefore need a classifier C_{state} that assigns a given input sample s_t a failure probability $p_{\text{state},t}$:

$$p_{\text{state},t} = C_{\text{state}}(s_t), \;\; p_{\text{state},t} \in [0, 1], \; s_t \in \mathbb{R}^{5\times L} \tag{3}$$

The classifier should be able to learn temporal patterns. There is a range of approaches for this task in the literature, with the most common choice in the field of failure prediction being LSTM networks [5], [7], [23], [26]. We also employ an LSTM architecture. In our previous work [7], we have shown that a simple bidirectional model with only a few layers can already show strong classification performance. Here, we use an architecture similar to [7] as shown in Figure 3. It consists of two bidirectional LSTM layers with 40 nodes each. They are followed by two fully connected (FC) layers with 40 and 20 neurons each. Further increasing the number of layers or nodes did not improve performance. Decreasing the size of the second FC layer allowed to decrease model complexity while not reducing accuracy. Finally, a third FC layer with a softmax activation function performs the classification. At time t, the input consists of the current state data feature vector f_t as well as the previous $L-1$ feature vectors. This sequence is fed into the network, which outputs the failure probability $p_{\text{state},t}$.

For training the model, we used the adam optimizer with a learning rate of 0.001. We trained for a total of 10 epochs until the network had converged. Our focus lies on demonstrating that this approach can be successfully used for failure prediction even with a straightforward architecture. We leave designing more elaborate models that optimize classification accuracy to future work.

3) Output Filtering: At 10 predictions per second, the unprocessed prediction value p_t is prone to incorrect outliers. To address this, we add a smoothing filter on the temporal output sequence. We choose a moving average filter since it is both easy to implement and well suited for temporal outlier removal [40]. While more complex filters could be investigated, we want to keep the added complexity and computational cost to a minimum. We therefore use a straightforward filter with a horizon H given as

$$p_{\text{state},t} = \frac{1}{\min(t, H)} \sum_{k=1}^{\min(t,H)} p_{\text{state},t+1-k}. \tag{4}$$

We set $H = 30$, the same as the sample length L. A disengagement is therefore only predicted if the model outputs an average failure probability of 0.5 over three seconds. In our testing scenario, we start 10 seconds before each disengagement. For $t < 30$, we thus average over the last t predictions instead of the entire length of one sample.

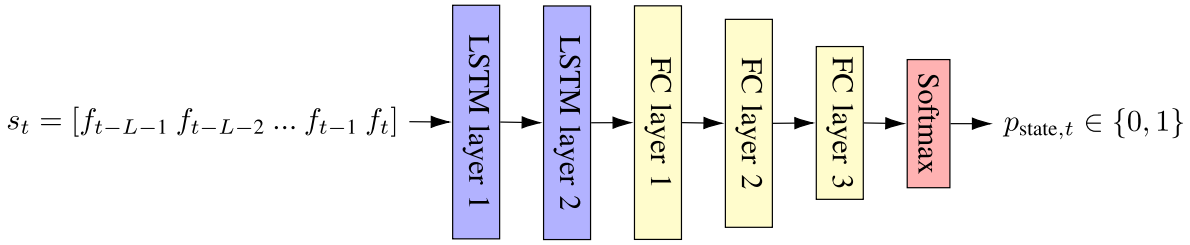

Fig. 3. The schematic architecture of the state-based failure prediction model. The sequential input of length L is fed into two bidirectional LSTM layers followed by two fully connected (FC) layers. A final FC layer performs binary classification and assigns a failure probability p_t to the input sample s_t through a softmax activation.

B. Image-Based Failure Prediction

While state-based failure prediction achieved useful results in [7], images are a much richer source of information about the environment. Cameras are a key sensor in most related failure prediction approaches for autonomous driving [5], [6], [23]. Learning features from sequences of images can be done in different ways. In the work that introduced introspection, the authors used difference images to represent the temporal flow [6]. In one of the few other works predicting disengagements of an autonomous car, the authors again used difference images to learn patterns from temporal sequences [23]. Using difference images to describe an image sequence allows to formulate the failure prediction problem as a classification task of a single image. This allows us to utilize well-researched architectures from the field of image classification. Next, we summarize how we process the raw camera images to obtain dynamic difference images.

1) Data Preprocessing: We adopt the idea from [23] to efficiently capture temporal patterns in images. The authors of [23] create gray-scale difference images from three time intervals and combine the result into a single three-channel dynamic difference image. As for the state-based approach, we consider the previous three seconds of sensor readings. For each image, we therefore also use the image from one, two and three seconds before. The dynamic difference image $I_{\text{dd},t}$ at time t is then defined as

$$I_{\text{dd},t} = \{I_t - I_{t-10}, I_{t-10} - I_{t-20}, I_{t-20} - I_{t-30}\}. \quad (5)$$

This way, each dynamic difference image contains information from four images taken across three seconds. Subtracting the images highlights dynamic objects which are often the cause of disengagements, such as pedestrians or turning cars. It also normalizes the image input. An example of the construction of a dynamic difference image out of four images from a time span of three seconds is shown in Figure 4.

Each image is cropped to obtain 224×224 square images since most image classification architectures require such square inputs. In Figure 4, the critical development over time is the car cutting in from the right. This movement is captured and highlighted by the three differential images. The approaching traffic light, which indicates an approaching intersection, is highlighted as well.

From a 10 second sequence with 100 images, we can generate 70 dynamic difference images this way. We use the same N_F failure sequences and the same evenly sampled successful

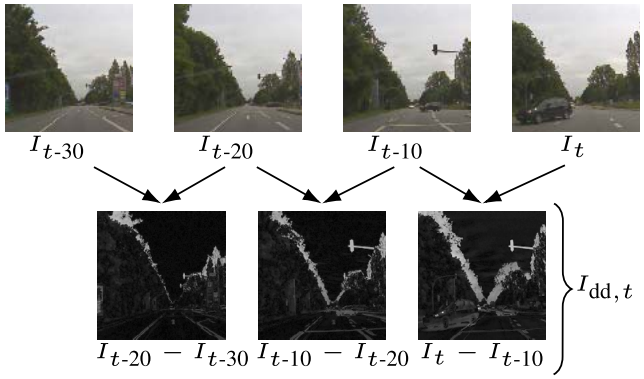

Fig. 4. An example image I_t at the moment of disengagement and the images one, two, and three seconds earlier. The dynamic difference image $I_{\text{dd},t}$ is constructed from the resulting three difference images.

sequences as before. The failure and success sequences are then divided into 70 dynamic difference images $I_{\text{dd},n_F,i}$ and $I_{\text{dd},n_s,j}$, respectively. This results in the dynamic difference image data set D_{img} given as

$$D_{\text{img}} = \{I_{\text{dd},n_F,i}, I_{\text{dd},n_S,j}\}, \quad i, j \in \{1, 70\},\ n_{F,S} \in \{1, N_F\}. \quad (6)$$

All sequences are split into the same 80% for training, 10% for validation and 10% for testing as before. Next, we discuss which models we train with our data set D_{img}.

2) Model Design: Each dynamic difference image is labeled as either success or failure. We again approach failure prediction as a binary classification task where a classifier C_{img} assigns a failure probability $p_{\text{img},t}$ to a dynamic difference image $I_{\text{dd},t}$:

$$p_{\text{img},t} = C_{\text{img}}(I_{\text{dd},t}),\ p_{\text{img},t} \in [0, 1],\ I_{\text{dd},t} \in \mathbb{R}^{224\times224\times3} \quad (7)$$

Next, we design the classifier C_{img}. Due to the preprocessing step that turns a sequence of images into a single three-channel image, we can use established image classification architectures to perform failure prediction for a given image sequence. Since our images contain natural objects in natural scenes, we can use existing models pretrained on ImageNet [41]. Such models can be used to extract the most relevant low-level features. We then use the dynamic difference images to finetune the later layers of the network.

Daftry *et al.* [6] used the popular AlexNet architecture [42] to extract both temporal and spatial features that they used for their introspective failure prediction. Subsequent

TABLE II
OVERVIEW OF THE SELECTED CLASSIFICATION ARCHITECTURES AND FINETUNING SETTINGS

Architecture	Layers	Trainable Layers
MobileNetV2	53	20
AlexNet	8	5
ResNet18	18	10
ResNet50	50	10

introspection works again used AlexNet [34], [35]. Since the introduction of introspection for failure prediction, more complex image classification architectures have been introduced as well. Notably, various versions of ResNet such as ResNet18 and ResNet50 have shown performance superior to AlexNet [43]. In addition, networks focusing on compactness such as MobileNet [44] have been introduced. We tested all aforementioned architectures to examine the influence of model complexity on performance. All models are pretrained on ImageNet and then finetuned with the training set of D_{img}. We replace the last layer with a binary classification layer. We only retrain the last layers to avoid overfitting. We empirically determine the number of layers to freeze using the validation data. An overview of all selected models and our training scheme is given in Table II.

We empirically chose a learning rate of 0.0005 using the adam optimizer. We trained all models until the validation error did not decrease any further, ranging from 10 epochs for MobileNetV2 to 20 epochs for ResNet50. We used data augmentation by adding random rotations of up to 10 degrees, random translation of up to 10 pixels and random cropping by up to 30% of the image size.

3) Output Filtering: For the same reasons as before, we use a moving average filter to smoothen the predictions and to detect trends in the output. The final prediction $p_{\text{img},t}$ made by the image-based failure prediction approach is then given as

$$p_{\text{img},t} = \frac{1}{\min(t, H)} \sum_{k=1}^{\min(t,H)} p_{\text{img},t+1-k}. \tag{8}$$

We again set $H = 30$, averaging over the predictions of the last three seconds. At the beginning of each 10 second sequence, we have not yet observed 30 predictions. As long as $t < 30$, we only average over the last t predictions.

C. Fusion-Based Failure Prediction

Finally, we present how we combine different modalities such as state sequences and image sequences for the best performance. Sensor data can be fused at various stages of a model. Next, we discuss the benefits of fusing either early or late in the architecture.

1) Early Fusion Vs. Late Fusion: While works such as [45] show that the performance of a model is not always affected significantly by the chosen level of fusion, early and late fusion both have distinct advantages. The main advantage of early to mid fusion compared to late fusion is the reduced computational complexity as it does not require to run multiple models in parallel. Early fusion allows to exploit the information of multiple sensors at the cost of only one inference. Additionally, early fusion allows learning cross-modal patterns, which late fusion cannot do. Recent methods such as [46] use this to improve early fusion architectures by adjusting training depending on which modality is currently most reliable.

For our use case, we argue that using a late fusion approach is more beneficial instead. Firstly, each sensor type is different and might require different architectures to be exploited best. For example, state data can be processed at a much higher frequency than images due to the significantly lower dimensionality. Secondly, it keeps the model design modular and extendable. Adding another sensor modality such as LIDAR can then be done by simply adding another trained model. If the sensors are fused at the input already, we would have to redesign and retrain the entire model. Thirdly, keeping the sensor modalities separate for as long as possible gives more insight into the source of the errors. Having separated components allows to explicitly determine which sensor reading was key for classifying the current scene. Finally, learning and relying on codependencies between weights in early layers can be detrimental to a network's performance [47]. Training separate models to learn from just a single sensor modality each forces them to learn patterns unique to the specific sensor type. While we argue that the benefits of late fusion outweigh the downsides, the increased inference time cannot be avoided and will be addressed in detail in the evaluation.

2) Proposed Approach: The modular fusion scheme we propose is straightforward. We want to fuse the results of the two classifiers C_{state} and C_{img} to assign a given set of inputs a failure probability p_{fusion}. For this, we take the failure probability predicted by each component and compute the average. The classification problem is then performed as

$$p_{\text{fusion}} = (p_{\text{state}} + p_{\text{img}})/2. \tag{9}$$

This approach requires no additional training or parameter tuning and allows for straightforward addition of other sensor modalities by addition. It lets each model influence the final prediction where it is the most confident, allowing to combine the strengths of each model. The sum in Equation 9 could be weighted using the validation set, but this did not affect the validation accuracy significantly.

Finally, we apply a moving average filter with horizon $H = 30$ as before. This way, both models need to consistently predict high failure probabilities over several seconds before the final prediction also increases. The final prediction p_{fusion} is then given as

$$p_{\text{fusion},t} = \frac{1}{\min(t, H)} \sum_{k=1}^{\min(t,H)} p_{\text{fusion},t+1-k}. \tag{10}$$

V. EXPERIMENTS AND RESULTS

In this section, we show the experiments carried out to evaluate the proposed failure prediction approaches. First, we present two state-of-the-art reference methods. Then, we evaluate the state-based and the image-based models

separately. Finally, we show the results of the fusion approach and compare the three methods in more detail.

A. Reference Approaches

Existing failure prediction methods that use both state and image data rely on human driving data and define failures as deviations from human driving [5], [32]. To the best of our knowledge, there is no existing work that explicitly predicts the disengagements of a real autonomous car. We therefore compare to two approaches that work with substitutes for real autonomous cars.

The *Arguing Machines* approach [23] predicts disengagements of a Tesla in autopilot on highways. They use an end-to-end steering angle prediction network based on PilotNet [48] trained with successful driving of the autopilot. Then, disagreement of the end-to-end network with the car's steering angle is used as a failure score. We directly follow their best-performing approach, using the data from the successful driving in the BMW data as training data for the same architecture. Following [23], we undersample driving data from angles close to zero since the cars are mostly driving straight. Our trained end-to-end network achieves a mean average error (MAE) of 1.2 on the validation set. This is comparable to the MAE of around 1.1 obtained in [23], whose training set is three times larger than ours.

We additionally implemented the approach from [33]. They predict crashes of an autonomous car in a driving simulator by monitoring the uncertainty of the same PilotNet-based network used in [23]. They obtain the uncertainty by generating multiple predictions from the same model and then computing the variance, thus referring to the approach as *Predictive Variance*. We use Deep Ensembles [4] instead of MC dropout [3] since it has repeatedly shown superior performance for uncertainty estimation [4], [36]. It also requires less inferences per input during testing. We use a Deep Ensemble of five independently trained end-to-end regression networks to obtain the Predictive Variance, with each of the five models achieving a MAE of around 1.2 on the validation set. The variance of the five models is then used as a failure score.

B. State-Based Failure Prediction

For the state-based failure prediction approach, we trained an LSTM model as described in Section IV. We first present the failure prediction performance by analyzing the Receiver Operating Characteristic (ROC) curve. Then, we perform an ablation study to investigate the importance of the individual state and variables.

1) Prediction Performance: We classified the samples from the test sequences using the LSTM classifier. Then, we applied the output filtering that was shown to significantly improve the performance in [7]. The overall result of the binary classification task is summarized in the ROC curve in Figure 5.

The model has learned to distinguish between success and failure sequences with an AUC value of 0.678, significantly outperforming both Arguing Machines with 0.544 and Predictive Variance with 0.572. We can now pick a threshold value that gives us a desired trade-off between false positives and true positives. Since safety is critical in autonomous driving, false positives in failure prediction are less harmful than false negatives. Here, we choose a threshold of 0.45 which leads to a true positive rate of 0.80 at a false negative rate of 0.76. The overall accuracy is then 78.2%. In our previous work [7], we achieved a higher accuracy with the same approach, but had a significantly smaller data set. Here, more unique failure cases are present that are more challenging to predict.

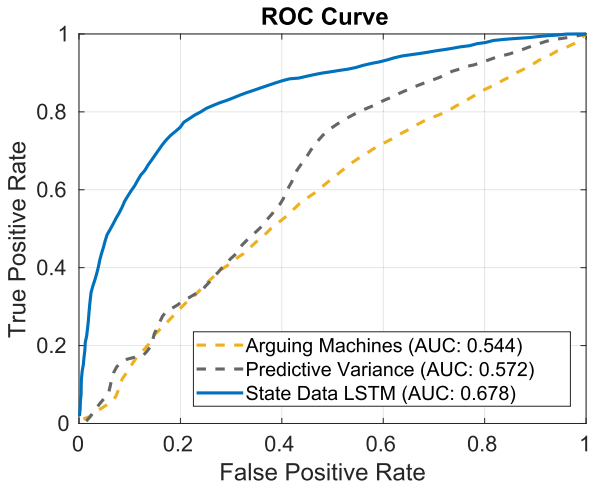

Fig. 5. ROC curve of the LSTM trained with sequences of state data compared to the reference approaches.

TABLE III
ACCURACY LOSS CAUSED BY REMOVING ONE STATE VARIABLE FROM THE TRAINING DATA

Removed State	Accuracy Loss
Speed	4.1%
Frontal acceleration	2.7%
Lateral acceleration	3.7%
Angle	4.1%
Angular speed	4.3%

2) Ablation Study: Next, we inspect the influence the different state variables had on the performance. We used speed v, angle θ, frontal acceleration a_x, lateral acceleration a_y and angular speed ω as feature components. We perform an ablation study by removing one variable from the data set at a time and retraining the LSTM. In Table III, we summarize the accuracy loss caused by removing individual state variables.

Similar to our findings with a smaller data set in [7], the accuracy is decreased by several percentage points when removing one of the states. The angular speed has the highest impact. Additionally, lateral acceleration improves the performance more than frontal acceleration, indicating that turns and fast maneuvers are important for predicting failures.

C. Image-Based Failure Prediction

Here, we show the results of failure prediction using image sequences. We retrained four architectures with the dynamical difference images introduced in Section IV to capture both temporal and spatial features in each input. First, we show

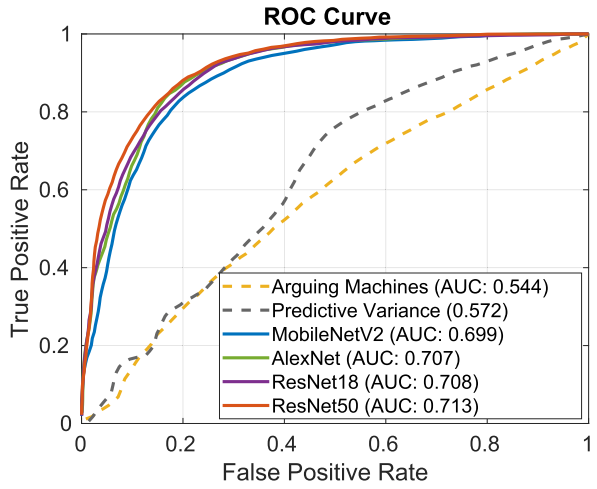

Fig. 6. ROC curves of the four image-based failure prediction models compared to the reference approaches. The performance is nearly the same regardless of the baseline architecture.

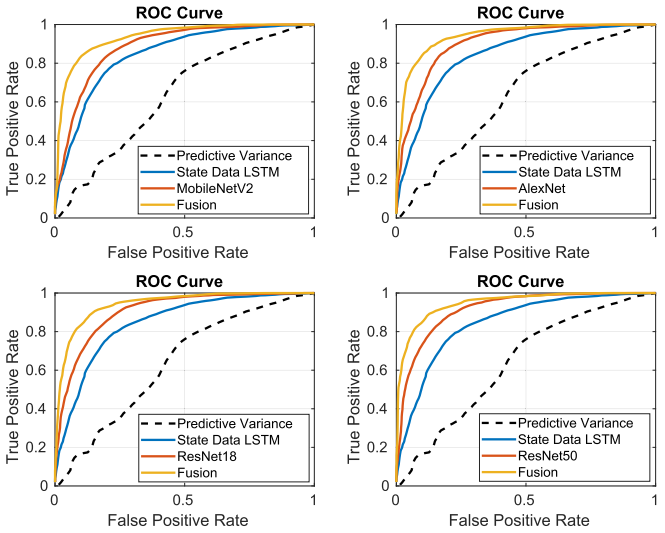

Fig. 7. ROC curves for the fusion approach with all classification architectures compared to the individual models as well as the best performing reference approach.

the prediction performance. Then, we investigate the models further by visualizing what features they learned to detect.

1) Prediction Performance: We analyze the performance by calculating the ROC curve for the four fine-tuned architectures over all test sequences. The results as well as the reference approaches are shown in Figure 6.

The AUC values are between 3.1% and 5.2% larger than for the state-based LSTM model. The reference approaches are outperformed by at least 12.7%. While the reference approaches also use images as input, they base their failure prediction on the analysis of the predicted steering angle only. The results in Figure 6 suggest that directly using the rich information of camera images for failure prediction is a promising direction. The differences between the different architectures are small. MobileNetV2 performs slightly worse than the rest at an AUC of 0.699. The finetuned AlexNet and ResNet18 models are very similar in performance at an AUC of 0.707 and 0.708. ResNet50 performs best at an AUC of 0.713, which is in line with its performance in traditional tasks such as ImageNet.

2) Feature Visualization: For a more intuitive understanding of what the image-based models learned, we visualize their behavior. We use class activation maps (CAMs) [49] to visualize which pixels have the biggest impact on the decision of each model. For AlexNet, gradient-weighted class activation maps (grad-CAMs) [50] are used instead since there are several fully connected layers after the last convolutional layer. The results give some intuition regarding what patterns the model learned to predict failures seconds in advance. We show two exemplary images that were correctly classified by each architecture from both failure and success sequences. The original input, the processed dynamic difference image and the CAMs of the four models are shown in Figure 8. It can be seen that MobileNetV2 and AlexNet use larger parts of the image to make their prediction, whereas the two ResNet architectures only focus on the most informative patch. Important areas like a car at a safe distance in the first row, the completely empty road in the second row, a car that is too close in the third row or a car cutting in from the right in the last row are detected consistently.

D. Fusion-Based Failure Prediction

Finally, we show the results of fusing both sensor modalities. The state-based approach has the advantage of being very compact, while the image-based approach achieves better performance. We show that the performance can be further improved by adding the state-based model to the image-based model. First, we compute the ROC curves for the fusion approach using the four classification architectures as before. The result for all four fused models compared to the individual models is shown in Figure 7. To keep the following plots readable, we only show the best performing reference approach of Predictive Variance from here on.

Averaging the state-based and the image-based probabilities significantly improves the classification performance regardless of the architecture. Table IV summarizes the average test accuracy for each individual model compared to fusing them. The accuracy of the fusion approach is between 3.8% and 5.8% larger than for the image-based approach alone while the confidence intervals consistently became smaller. This indicates that the two different sensor modalities learned to detect different error sources, which when combined allow us to correctly predict more errors in total. The performance gain is largest for the MobileNetV2 architecture, allowing it to achieve almost the same accuracy as the more complex ResNet50 model.

Next, we evaluate how the models perform over time when approaching a disengagement. For this, we plot the accuracy over time for both failure and success sequences. Since we have shown that the four classification architectures perform similarly, we use only the best performing architecture in the following. The results of the fusion of the LSTM and the finetuned ResNet50 compared to the individual models for success sequences are shown in Figure 9. The fusion approach outperforms the image-based approach consistently by several

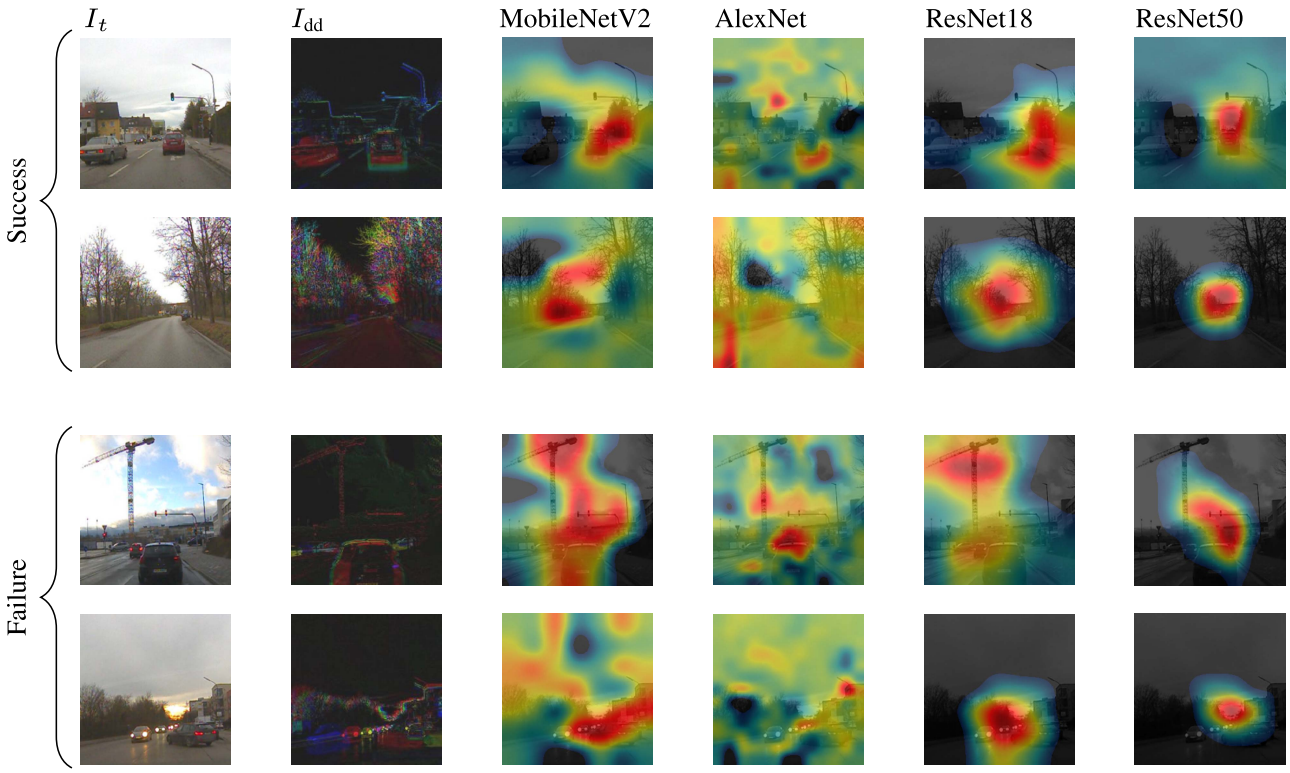

Fig. 8. CAM visualization for images I_t and the preprocessed dynamic difference images I_{dd} from two sample failure and two sample success sequences. Important areas such as the car cutting in from the right in the last row or the car at a safe distance in the first row generally lead to the strongest activations.

TABLE IV
AVERAGE ACCURACY AND CONFIDENCE INTERVAL FOR THE INDIVIDUAL MODELS AND THE CORRESPONDING FUSED MODELS

Model	Individual	Fusion
Arguing Machines	56.6 % (±0.15)	-
Predictive Variance	63.1 % (±0.09)	-
State Data LSTM	78.2 % (±0.08)	-
MobileNetV2	81.0 % (±0.07)	86.8 % (±0.05)
AlexNet	83.5 % (±0.06)	87.3 % (±0.04)
ResNet18	82.3 % (±0.06)	87.0 % (±0.04)
ResNet50	82.6 % (±0.05)	87.6 % (±0.04)

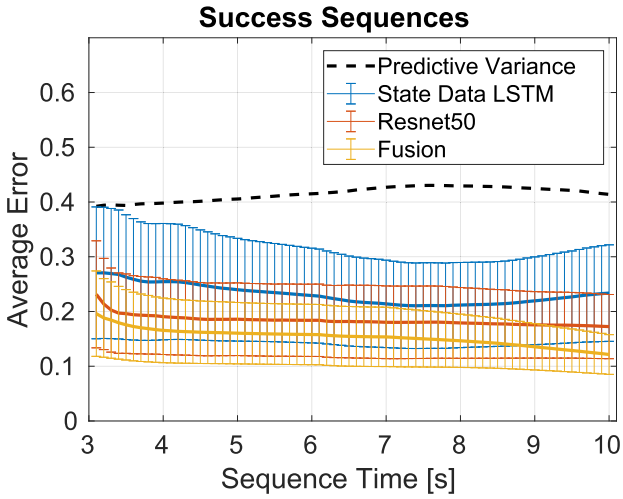

Fig. 9. Average error over time for all success sequences of Predictive Variance, the state data LSTM, the best image-based model and the fused model.

percentage points. For those models, the accuracy does not vary significantly during successful driving since there is no decisive event at the end of the sequence. Predictive Variance shows the same temporal behavior, but exhibits a significantly higher error on average. The error of the state-based model slightly fluctuates over time, showing that the state-based approach is more sensitive to noise and small changes.

Next, we show the accuracy over time for failure sequences in Figure 10. Some interesting observations can be made. The reference approach of Predictive Variance, based on the uncertainty of the steering angle, consistently improves when approaching the disengagement. While the overall performance is below the proposed explicit failure prediction methods, this result indicates that a car's uncertainty is directly correlated to its failures.

Whereas the error of the image-based approach is largely constant, both the state-based and the fusion model show a visible improvement in the last three seconds before each disengagement. Fast changes such as stronger breaking or faster turning maneuvers can be detected by the state-based model. The image-based model does not show the ability to infer these fast changes from the images of the last seconds of each failure sequence. This results in a key finding of combining state-data sequences with image data. Image data allows to generally assess the scene correctly, even up to seven seconds in advance. As seen in the CAM visualization in Figure 8, objects such as cars are well recognized and useful

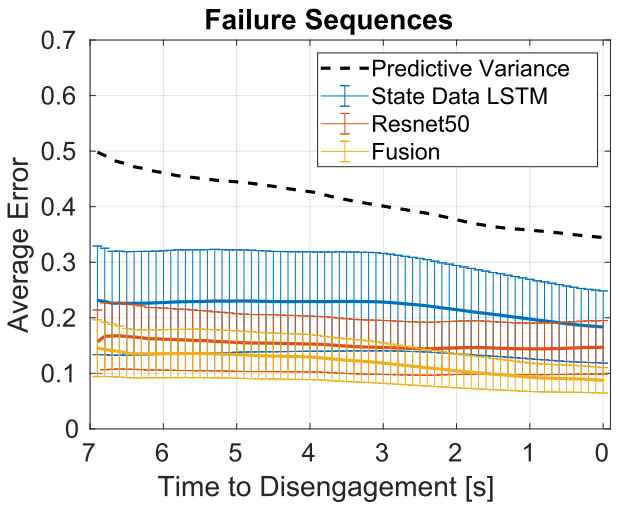

Fig. 10. Average error over time for all failure sequences of Predictive Variance, the state data LSTM, the best image-based model and the fused model.

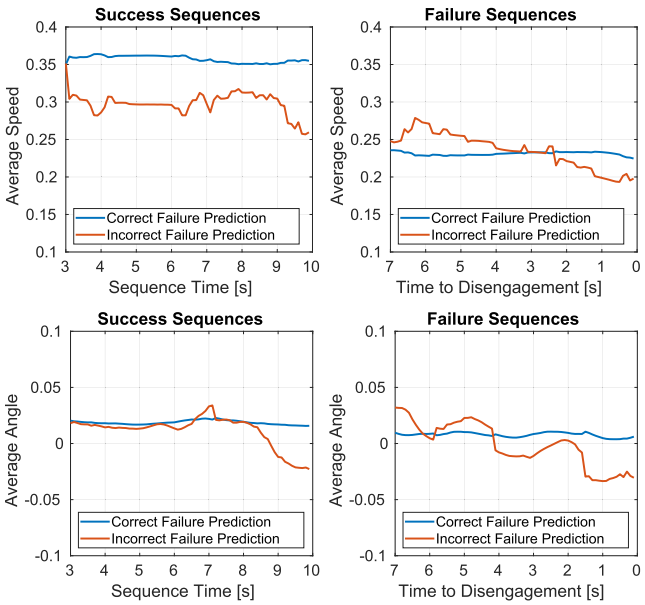

Fig. 11. Average normalized speed (top) and angle (bottom) of both correctly and incorrectly predicted samples, separated by the type of sequence they are from.

correlations between traffic and disengagements are learned. However, the specific timing of the failure is harder to infer from images. For this, the LSTM is more suitable. It is capable of noticing slight changes in the car's state that show a failure is imminent. The fusion approach allows to efficiently combine the strengths of the two sensor modalities. Failures can be predicted seven seconds in advance at an accuracy of over 85%. Two seconds in advance, the accuracy increases to over 90%.

E. Error Analysis

While the proposed methods significantly outperform existing approaches for disengagement or crash prediction of autonomous vehicles, they still incorrectly assess some situations. In this section, we investigate under what circumstances the proposed approach made errors. We restrict this analysis to our best performing model, the ResNet50 fusion, whose output was incorrect 12.4% of the time.

The errors of the proposed approach are all made in 115 of the 510 test sequences. This indicates that the errors are not evenly spread, but concentrated to the most challenging sequences. Of those errors, around 43% are made in failure sequences. This suggests that the model did not significantly overfit to either failure or success sequences. To inspect what those 115 sequences have in common, we analyze the corresponding speed and steering angle compared to the state data of the 395 correctly classified sequences. All state values are normalized with the highest absolute value in the test set. We look at data from success and failure sequences separately. The temporal development of the car's state for successful and unsuccessful failure predictions is shown in Figure 11.

The mean average speed of correctly classified samples from success sequences is over 54% larger than the speed of correctly classified failure samples. The speed difference between incorrectly classified success and failure samples is notably smaller at around 24%. The difference for the angle is less extreme, but still visible. Additionally, the temporal development of incorrectly classified samples is more varied for both success and failure sequences. This indicates that dynamic scenes with similar speed and angle developments are hardest to classify correctly as success or failure for the proposed approaches. It should be noted that some disengagements in such dynamic scenes cannot be predicted early, for example when a pedestrian suddenly enters the road. Nevertheless, these results show that the sensor modalities used in our approach are not sufficient to capture and distinguish every pattern leading to a disengagement.

TABLE V
NUMBER OF PARAMETERS OF EACH INVESTIGATED INDIVIDUAL METHOD

Model	Parameters
Arguing Machines	0.37×10^6
Predictive Variance	1.84×10^6
State Data LSTM	0.02×10^6
MobileNetV2	3.5×10^6
AlexNet	61.0×10^6
ResNet18	11.7×10^6
ResNet50	25.6×10^6

F. Computational Complexity and Time Delay

Finally, we investigate the computational complexity and the time delay of the reference approaches as well as the proposed introspective failure prediction methods. First, we show the number of parameters per model in Table V. We only show the individual models since the fusion approach merely adds the number of parameters of the state data LSTM to each of the image-based models. The state data LSTM is the most compact model at around 20 000 parameters. This is two magnitudes smaller than ResNet50, which is the largest model at over 25 million parameters.

TABLE VI
AVERAGE PROCESSING TIME PER PREDICTION FOR ALL INVESTIGATED METHODS

Model	Individual	Fusion
Arguing Machines	18.2 ms	-
Predictive Variance	81.4 ms	-
State Data LSTM	3.8 ms	-
MobileNetV2	56.1 ms	59.9 ms
AlexNet	52.4 ms	56.2 ms
ResNet18	53.9 ms	57.7 ms
ResNet50	59.5 ms	63.4 ms

However, the inference speed does not only depend on the number of parameters, but also on the type of operations that are performed. We therefore next measure the inference time for all models under identical circumstances using a Quadro P2000 GPU. Each model outputs ten predictions per second. After having observed the initial three seconds of input data, there is no time delay besides the inference time. Each model only requires the three previous seconds of input that are already available.

In Table VI, we show the average processing time for one inference. All investigated methods require less than 100ms per prediction and are therefore suitable for the desired frequency of 10Hz. The LSTM approach is by far the fastest at around 4ms per output. The speed of the different image-based models varies by up to 7ms. Due to the late fusion design, the cost of each fusion approach increases by the entire cost of the added LSTM model. However, the speed of the LSTM means the slowest ResNet50 fusion model is only 6% slower than the individual ResNet50. The main downside of the increased complexity and time delay introduced by using late instead of early fusion is therefore not a significant issue.

Arguing Machines, based on PilotNet [48], is around three times faster than our image-based approaches, but showed a significantly worse accuracy. While Predictive Variance outperforms Arguing Machines regarding classification accuracy, it is also the slowest method at over 80ms per prediction.

VI. CONCLUSION

In this paper, we presented an introspective failure prediction approach for autonomous driving using late fusion of car state and camera image data. We discussed how failure prediction can be approached in this context and established an introspective framework that can be applied to other domains as well. The proposed approach requires previous failures from the inspected system as training data. This requirement limits this approach to systems where extensive test trials are available. We argue that this makes it well suited for autonomous driving, where month-long tests are common practice. Introspection allows to exploit the evaluation of those test drives that would occur anyway. The recording of disengagements in test drives automatically generates labeled training data for our approach. The data set we constructed from data supplied by the BMW Group consists of over 14 hours of driving in autonomous mode on public roads. It contains over 2500 disengagements that we used to train the failure prediction models.

All of our approaches outperformed the accuracy of state-of-the-art failure prediction methods by at least 15%. This can be explained by the fact that existing methods do not explicitly learn from failures. Rather, they are focused on detecting deviations from the expected behavior, predicting failures only implicitly. The lack of explicit failure prediction methods in the literature can also be explained by the lack of suitable data, which is only now becoming available with more companies performing large-scale test drives on public roads. A large number of previous failures specific to the inspected system is critical for our work. While existing methods have less restrictive requirements in that regard, our results suggest that failure prediction methods highly specific to one given system are a promising approach.

We designed both a state-based and an image-based failure prediction approach. For the image-based method, we investigated four state-of-the-art architectures. We propose combining the classification results for the different sensor modalities using a late fusion approach to combine the strengths of the individual models. This straightforward fusion approach consistently outperforms the accuracy of the individual models by up to 5.8%. Our evaluation shows that the image-based approach performs better at accurately assessing a scene as challenging far in advance. However, the state-based approach is more sensitive to changes in the seconds right before the disengagement. The fusion approach combines the accurate early assessment achieved by the image-based model with the detection of fast changes enabled by the state-based model. The results with an average accuracy of 87.6% outperform our previous work [7] by almost 10% and the best performing reference approach [33] by over 24%. Even seven seconds in advance, disengagements can be predicted at an accuracy of over 85%. At an inference time of 63ms on a single GPU, the proposed prediction frequency of 10 Hz can be reasonably achieved.

Our error analysis has shown that sequences with similar state data, but different outcomes are hardest to predict by our approach. A potential solution for this issue is to add more sensor modalities such as LIDAR since the current sensor setup does not seem to capture enough information to distinguish such scenes. For each new sensor, a new introspective model can be trained and easily integrated using the proposed late fusion approach. Furthermore, our black box methods could be combined with model-intrinsic confidence measures. A failure prediction could then use both the knowledge of previous failures as well as the assessment the model is making right now. A final interesting option would be to combine introspection with Region of Interest prediction [51] to only learn from the most relevant areas. This way, the model could learn to ignore irrelevant objects when performing failure prediction for autonomous driving.

REFERENCES

[1] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3135–3151, Aug. 2020.

[2] L. Kang, W. Zhao, B. Qi, and S. Banerjee, "Augmenting self-driving with remote control: Challenges and directions," in *Proc. 19th Int. Workshop Mobile Comput. Syst. Appl.*, 2018, pp. 19–24.

[3] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[4] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6402–6413.

[5] S. Hecker, D. Dai, and L. Van Gool, "Failure prediction for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1792–1799.

[6] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1743–1750.

[7] C. Kuhn, M. Hofbauer, G. Petrovic, and E. Steinbach, "Introspective black box failure prediction for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2020, pp. 1–7.

[8] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Structural Saf.*, vol. 31, no. 2, pp. 105–112, Mar. 2009.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[10] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015. [Online]. Available: http://www.nature.com/articles/nature14541

[11] H. Wang and D.-Y. Yeung, "Towards Bayesian deep learning: A framework and some existing methods," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3395–3408, Dec. 2016.

[12] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017, pp. 57.1–57.12.

[13] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3D vehicle detection," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3266–3273.

[14] C. Gurau, A. Bewley, and I. Posner, "Dropout distillation for efficiently estimating model confidence," 2018, *arXiv:1809.10562*. [Online]. Available: http://arxiv.org/abs/1809.10562

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[16] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 1613–1622.

[17] M. Fortunato, C. Blundell, and O. Vinyals, "Bayesian recurrent neural networks," 2017, *arXiv:1704.02798*. [Online]. Available: http://arxiv.org/abs/1704.02798

[18] A. Shafaei, M. Schmidt, and J. J. Little, "A less biased evaluation of Out-of-distribution sample detectors," 2018, *arXiv:1809.04729*. [Online]. Available: http://arxiv.org/abs/1809.04729

[19] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[21] D. Saxena and J. Cao, "D-GAN: Deep generative adversarial nets for spatio-temporal prediction," 2019, *arXiv:1907.08556*. [Online]. Available: http://arxiv.org/abs/1907.08556

[22] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Proc. Int. Conf. Mach. Learn.*, vol. 34, 2017, pp. 1–5.

[23] L. Fridman, L. Ding, B. Jenik, and B. Reimer, "Arguing machines: Human supervision of black box AI systems that make life-critical decisions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019.

[24] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 204–211.

[25] A. Jain, H. S Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4Cars: Car that knows before you do via sensory-fusion deep learning architecture," 2016, *arXiv:1601.00740*. [Online]. Available: http://arxiv.org/abs/1601.00740

[26] D. Hallac, S. Bhooshan, M. Chen, K. Abida, R. Sosic, and J. Leskovec, "Drive2 Vec: Multiscale state-space embedding of vehicular sensor data," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3233–3238.

[27] D. Tian, C. Zhang, X. Duan, and X. Wang, "An automatic car accident detection method based on cooperative vehicle infrastructure systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019.

[28] X. Huang, P. He, A. Rangarajan, and S. Ranka, "Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video," *ACM Trans. Spatial Algorithms Syst. (TSAS)*, vol. 6, no. 2, pp. 1–28, 2020.

[29] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Proc. Asian Conf. Comput. Vis.* Taipei, Taiwan: Springer, 2016, pp. 136–153.

[30] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident DB," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3521–3529.

[31] A. Naidenov and A. Sysoev, "Developing car accident detecting system based on machine learning algorithms applied to video recordings data," *Large-Scale Syst. Control*, pp. 373–378, 2019.

[32] X. Huang, S. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," 2019, *arXiv:1901.05105*. [Online]. Available: http://arxiv.org/abs/1901.05105

[33] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in End-to-End autonomous driving control," 2018, *arXiv:1811.06817*. [Online]. Available: http://arxiv.org/abs/1811.06817

[34] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5824–5829.

[35] S. Rabiee and J. Biswas, "IVOA: Introspective vision for obstacle avoidance," 2019, *arXiv:1903.01028*. [Online]. Available: http://arxiv.org/abs/1903.01028

[36] C. Kuhn, M. Hofbauer, S. Lee, G. Petrovic, and E. Steinbach, "Introspective failure prediction for semantic image segmentation," in *IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–6.

[37] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Virtual, 2020, pp. 2446–2454.

[38] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: Disengagements, accidents and reaction times," *PLoS ONE*, vol. 11, no. 12, Dec. 2016, Art. no. e0168054.

[39] S. Gibbs, "Google sibling waymo launches fully autonomous ride-hailing service," *Guardian*, vol. 7, Nov. 2017. [Online]. Available: https://www.theguardian.com/technology/2017/nov/07/google-waymo-announces-fully-autonomous-ride-hailing-service-uber-alphabet

[40] S. Smith, *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Amsterdam, The Netherlands: Elsevier, 2013.

[41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[44] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[45] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2005, pp. 399–402.

[46] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," 2018, *arXiv:1805.11730*. [Online]. Available: http://arxiv.org/abs/1805.11730

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: http://arxiv.org/abs/1604.07316

[49] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[50] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[51] M. Hofbauer, C. Kuhn, J. Meng, G. Petrovic, and E. Steinbach, "Multi-view region of interest prediction for autonomous driving using semi-supervised labeling," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–6.

Christopher B. Kuhn (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree from the Technical University of Munich (TUM), in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Automated and Autonomous Driving Division, BMW Group. He is also the Chair of Media Technology with the Technical University of Munich. His research interests include the areas of video processing, machine learning, and computer vision. His work is also focused on automatic and predictive failure detection for autonomous vehicles.

Markus Hofbauer (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree from the Technical University of Munich (TUM), in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Chair of Media Technology. From 2016 to 2018, he worked as a Software Engineer with Objective Software GmbH and Luxoft Inc., in cooperation with the BMW Group in the area of Automotive and Autonomous Driving. His research interests include video processing, compression, and transmission of multicamera systems for autonomous and teleoperated driving.

Goran Petrovic received the Dipl.Ing. degree in electrical engineering and telecommunications from the University of Nis, Serbia, the M.Sc. degree in communications technology from the University of Ulm, Germany, and the Ph.D. degree in electrical engineering from the Eindhoven University of Technology, The Netherlands. From 2011 to 2015, he was a Principal Investigator with the Intel Visual Computing Institute (IVCI), Saarbrücken, Germany. He is currently with the Automated and Autonomous Driving Division, BMW Group, Munich, Germany. His research interests include the areas of multicamera systems, video processing, compression, and transmission.

Eckehard Steinbach (Fellow, IEEE) received the degree in electrical engineering from the University of Karlsruhe, Germany, from the University of Essex, Great-Britain, and from ESIEE, Paris, and the Ph.D. degree in engineering from the University of Erlangen-Nuremberg, GermanyÃ‚Â¸ in 1999. He is currently a Professor of Media Technology with the Technical University of Munich (TUM). His research interests include the area of visual-haptic information processing and communication, telepresence and teleoperation, surface haptics, tactile Internet, and networked and interactive multimedia systems. From 1994 to 2000, he was a member of the Research Staff of the Image Communication Group, University of Erlangen-Nuremberg. From February 2000 to December 2001, he was a Post-Doctoral Fellow with the Information Systems Laboratory of Stanford University. In February 2002, he joined the Department of Electrical Engineering and Information Technology, TUM. He is an Associate Editor of the IEEE TRANSACTIONS ON HAPTICS. He was a recipient of the 2011 Research Award of the Alcatel-Lucent Foundation. He was an elected Fellow of the IEEE in 2015 for his contributions to visual and haptic communications. He serves as the Chair for the IEEE standardization activity P1918.1.1 a.k.a. Haptic Codecs for the Tactile Internet.