UNCERTAINTY-AWARE MODELS FOR DEEP LEARNING-BASED HUMAN ACTIVITY RECOGNITION AND APPLICATIONS IN INTELLIGENT VEHICLES

Zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation von

ALINA ROITBERG

geboren in Chernigiv, Ukraine

Tag der mündlichen Prüfung: 23. April 2021
Hauptreferent: Prof. Dr.-Ing. Rainer Stiefelhagen
Korreferent: Prof. Dr. Mohan M. Trivedi

Alina Roitberg: *Uncertainty-aware Models for Deep Learning-based Human Activity Recognition and Applications in Intelligent Vehicles*

True wisdom is knowing what you don't know.

— Confucius

ABSTRACT

With the accuracy of action recognition CNNs gradually reaching the ceiling, the existing gap between the published methods and their applications in practice makes us wonder about important performance aspects being overlooked. When examining the previous research, we make two observations: (1) the existing algorithms are highly driven by the top-1 accuracy, skipping other relevant metrics, such as the *reliability of their confidence values*, and (2) the existing benchmarks assume that the model will be deployed under closed-set conditions (*i.e.* unknown activities cannot occur at test-time).

Uncertainty-aware activity recognition is vital in almost all real settings, for example, if the model operates in an open world or if *no certainly assigned label is better than an incorrect label* (*e.g.* in safety-critical cases). In this thesis, we specifically target one such scenario – applications inside the vehicle cabin, which would make driving more convenient and safe, but require models that are reliable in uncertain situation. Besides, presumably due CNNs being data-hungry and the manufacturers' scepticism linked to their black-box reputation, the rise of deep learning had rather a slow effect on this particular domain. We therefore first bring the conventional end-to-end computer vision approaches to the field of driver activity recognition, and, in order to train such models, we collect and release the large-scale *Drive&Act* dataset. We study the existing models in the context of driver activity recognition, enhancing their interpretability and introduce a new model for driver intention prediction, advancing state-of-the-art in this task.

We then consider action recognition from a different perspective – the perspective of uncertainty, as we develop models which do not only (1) assign the correct action category, but also (2) reliably identify incorrect predictions and (3) distinguish between the previously known and unknown behaviours, and, ideally, (4) have tools for dealing with novel concepts-of-interest without costly labelling. At the heart of this thesis are two new problems and two new models for addressing them: the *CARING* model for learning to obtain reliable confidence estimates, given that the category is known, and the voting-based *Bayesian I3D* model for detecting categories not previously seen by the classifier (open set case). As both tasks are new in the activity recognition context, we compare our models to the existing methods adopted from image classification, in both cases achieving state-of-the-art. Finally, we address the weak spot of deep CNNs – their reliance on training data and develop strategies for dealing with novel examples without additional annotations, for example, by leveraging multimodal data posted on the web as our knowledge source or leveraging language-based models via zero-shot learning.

This thesis has made an important step towards activity recognition models operating in an open world while *realistically* estimating their own prediction confidence and advanced the fields of action recognition and driver observations though new models, tasks, and datasets. Our experiment results hold great promise for uncertainty-aware end-to-end models – a crucial step towards real-life applications of such algorithms.

ZUSAMMENFASSUNG

Während die Genauigkeit der Aktivitätenerkennung mittels der Faltenden Neuronalen Netze (engl. Convolutional Neural Networks, abgekürzt CNNs) sich der Obergrenze nähert, bleibt die Kluft zwischen den veröffentlichten Methoden und deren Anwendung in der Praxis weiterhin bestehen. Dies lässt die Frage aufkommen, ob wichtige Leistungsaspekte möglicherweise übersehen wurden. Bei der bisherigen Literatur fallen uns zwei Dinge besonders auf: (1) Die existierenden Algorithmen haben als Zielsetzung eine höchstmögliche Erkennungsgenauigkeit, wobei weitere relevanten Metriken, wie z.B. die Zuverlässigkeit ihrer Konfidenz, zu kurz kommen und (2) die bestehenden Testumgebungen nehmen an, dass die Modelle zur Testzeit mit den gleichen Konzepten, wie in der Trainingsphase, konfrontiert werden, das heißt, unbekannte Aktivitäten dürfen nach dem Modelltraining nicht mehr auftauchen.

In der praktischen Anwendung ist die Berücksichtigung der *Klassifikationsunsicherheit* jedoch entscheidend, bspw., wenn ein Model in einer dynamischen Umgebung zum Einsatz kommt oder falls *keine zuverlässige Zuordnung einer bestimmten Klasse besser ist, als eine inkorrekt gewählte Klasse*, z.B. bei sicherheitskritischen Anwendungen. In dieser Arbeit befassen wir uns genauer mit einem solchen Szenario – der automatischen Fahreraktivitätenerkennung, welche die Fahrt bequemer und sicherer machen würde, aber *zuverlässige* Modelle voraussetzt. Zudem sollte erwähnt werden, dass, vermutlich aufgrund der erforderlichen großen Mengen an annotierten Daten und der Skepsis der Hersteller bedingt durch den "Black-Box"-Ruf der Deep-Learning-Modelle, der Einfluss der CNNs zur Aktivitätenerkennung in dieser Branche vergleichsweise gering war. Um das Training solcher "datenhungrigen" Modelle überhaupt zu ermöglichen, sammeln wir als erstes einen umfangreichen multimodalen *Drive&Act* Datensatz zur feingranularen Erkennung der Fahrerhandlungen, den wir mit einem hierarchischen Annotationsschema versehen. Wir implementieren und erproben mehrere konventionellen End-to-End-Ansätze im Kontext der Fahrerhandlungserkennung, verbessern deren Interpretierbarkeit und führen ein neues Modell für die Vorhersage des nächsten beabsichtigten Manövers durch die Fahrerbeobachtung ein.

Danach betrachten wir Aktivitätenerkennung aus einer anderen Perspektive – der Perspektive der Unsicherheit, und entwickeln Modelle, die nicht nur (1) die korrekten Aktivitätenkategorien zuordnen, sondern auch (2) verlässlich fehlerhafte Vorhersagen identifizieren, (3) zwischen bereits bekannten und zuvor unbekannten Aktivitäten unterscheiden können sowie, idealerweise, (4) auch Instrumente für den Umgang mit neuen, für den Anwendungszweck relevanten Konzepten bereithalten, wenn keine manuell annotierten Daten dafür verfügbar sind. Im Mittelpunkt dieser Arbeit stehen zwei neue Probleme sowie zwei neue Modelle zu deren Lösung: das CARING-Modell zum Erlangen von zuverlässigen Konfidenzwerten, sofern die Kategorie aus dem Training bekannt ist, und das Bayes'sche I3D-Modell zur Erkennung von neuen Kategorien (Open-Set-Szenario). Da beide Aufgaben im Kontext der Aktivitätenerkennung neu sind, vergle-

ichen wir unsere Modelle mit existierenden Methoden aus der Bildklassifikation und erreichen dabei Stand-der-Forschung Ergebnisse. Schließlich gehen wir auf die Schwachstelle der tiefen CNNs ein – ihre Abhängigkeit von Trainingsdaten – und entwickeln Strategien für den Umgang mit neuen Konzepten ohne zusätzlichen manuellen Annotationen, zum Beispiel durch die Nutzung der im Internet zugänglichen multimodalen Daten oder mittels Wissenstransfer durch die sprachbasierten Modelle im Kontext von Zero-Shot-Erkennung.

Diese Dissertation macht einen wichtigen Schritt hin zur Anwendung von Handlungserkennungsmodellen in realen dynamischen Umgebungen, indem der Fokus speziell auf die realistische einschätzung der Vorhersagegenauigkeit gesetzt wird und neue Modelle, Forschungsrichtungen und Datensätze im Bereich der Handlungserkennung und Fahrerbeobachtung eingeführt werden. Die Ergebnisse unserer Experimente sind vielversprechend für *unsicherheitsbetrachtende* End-to-End-Modelle – ein entscheidender Schritt zur Anwendung solcher Algorithmen in der Praxis.

ACKNOWLEDGEMENTS

The research conducted over the past three and a half years at CV:HCI, which has lead to this dissertation, would not have been possible without continuous guidance, encouragement and inspiration of my advisors, mentors, colleagues, friends and family.

First I would like to express my deepest gratitude to Prof. Rainer Stiefelhagen for being an excellent advisor during my time at KIT. I am very grateful not only for his contributions to this research but also the ability to strike the perfect balance between giving me the necessary freedom to pursue new ideas and offering valuable advice and a fresh perspective at critical moments, as both is vital for conducting original research.

I also would like to thank my second advisor, Prof. Mohan Trivedi, for agreeing to revise and give me feedback on this thesis, which is a great honour since he and his group are frontrunners in the area of driver observation. His contributions have advanced the field immensely and have been an inspiration for me for many year, serving as an important starting point in the first phase of my thesis.

This work would also not have been possible without my dearest friend and colleague Monica Haurilet, who has always been there for me in good and bad times. Lots of valuable ideas were created based on interesting discussions with Monica and I am very sad about losing her as a colleague after she has finished her dissertation.

I truly enjoyed my time as a PhD student, mostly because of the amazing group of people working at CV:HCI. I am grateful to Angela, Constantin, Corinna, Daniel, Jiaming, Kailun, Manel, Manuel, Monica, Saquib, Sebastian, Simon, Tobias, Vanessa, Vivek and Ziad. Constantin, Simon, Tobias and Monica, thank you for all our lunch conversations! A big thanks goes to Ziad, who has been a mentor to me at the beginning of my thesis and to Manel, who was the first one at the lab to propose a collaboration leading to me going to my first conference, which was a big deal! An extra thank you to Corinna!

I want to acknowledge the BMBF project PAKoS for the provided funding and especially Manuel Martin, for the collaborations during the project. Furthermore, I am thankful to the Facebook Zurich team for giving me an opportunity to work on exciting computer vision topics applied to real-world problems during my visit.

I would also like to thank my family and friends for the great moments spent together, I would not have made it without their moral support. I am deeply grateful to my parents, Inna and Sascha and my little sister Yana, I am truly blessed to have them in my life. Also big thanks to my grandparents, Rita and Yakov, who are unfortunately not with us anymore. Big thanks to my uncle Boris for inspiring me to learn programming at an early age. Ingrid, Polly, Svenja, Sandra, Marta, Ann-Kristin, Luisa, Verena, Markus, Börge, Thomas, Leo, Wolfgang – I am so lucky to have you as my friends, thank you for your friendship and patience with me.

Lastly, I want to thank David for making every day of my life better. THANK YOU!

Dedicated to my grandparents Rita and Yakov.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

BACKGROUND

1 INTRODUCTION AND MOTIVATION

The main goal of this thesis are algorithms for video-based activity recognition with deep CNNs for driver monitoring applications with special consideration of the classification uncertainty. Two common culprits hindering the integration of activity recognition models in practice are dynamic open-set environment and scarce training data, as large annotated video datasets for such specific applications are costly. In this thesis, we aim for uncertainty-aware models which are able to (1) recognize previously known activities (i.e. conventional activity recognition problem), quantify their uncertainty in order to (2) identify failure cases or (3) detect novel behaviors, and (4) find a way to handle such uncertain examples, for which not enough training data is available (e.g. via knowledge transfer from web-based sources). Figure 1 provides a high-level summary of the addressed research questions, while in the following, we will motivate our research and give an overview of the specific contributions in each of these four areas.

1.1 UNCERTAINTY-AWARE ACTIVITY RECOGNITION

Humans have a natural grasp of probabilities [40]: If we hear that a certain event is detected in a video by a neural network with 99% confidence, we automatically believe that this is the case. Such an assumption, however, would be rather naive, as the inference merely gives us values of the last fully-connected layer which are usually optimized for a high top-1 accuracy on a fixed set of previously defined action categories. As these values are usually normalized through the *Softmax*[1] function to sum up to one, they *appear* to be class probabilities but they do not depict the true confidence of the model [44]. Besides, when engineers apply such deep learning models in practice, they will quickly discover the phenomenon of *model miscalibration*, as the resulting *Softmax* estimates tend to be

1 *Softmax* function is often applied on *logits* (the output vector of the last layer of a classification network, where each value represents a category score) and normalizes them to sum up to one by computing the exponents of each output and then normalizing each of them by the sum of those exponents.

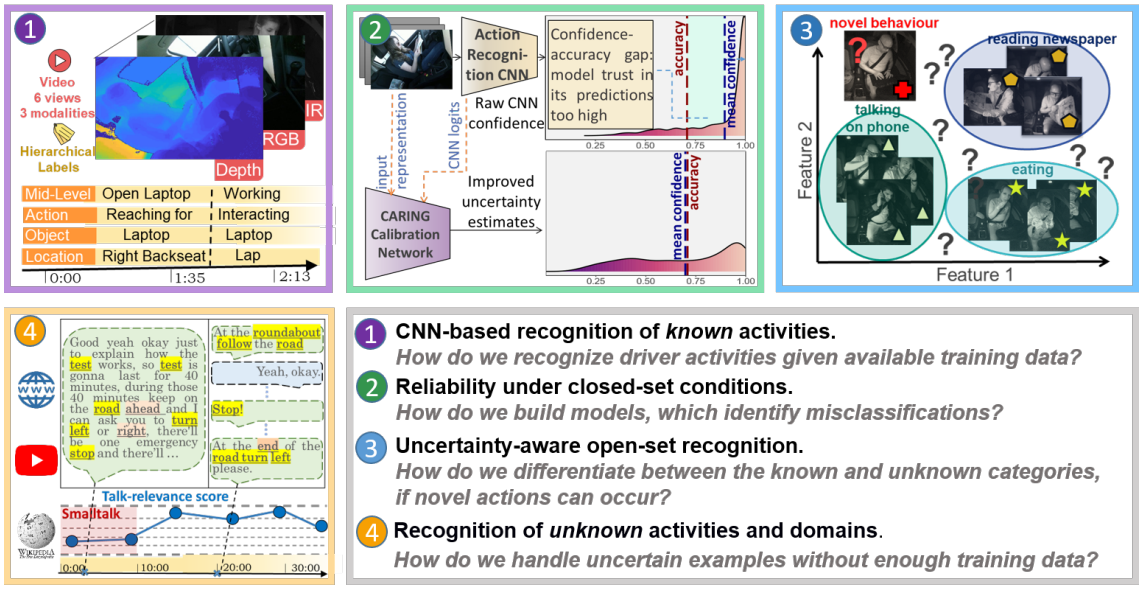


Figure 1: Overview of the contribution areas and the underlying research questions of this PhD thesis. We aim for uncertainty-aware models which are able to (1) recognize previously *known* activities, quantify their uncertainty in order to (2) identify failure cases or (3) detect novel behaviors, and (4) find a way to handle such uncertain examples despite the lack of annotated training data (e.g. via knowledge transfer from web sources).

biased towards very high values [44, 55]. Unfortunately, such high confidence outcomes are not only present for correctly predicted samples but also in case of misclassification or uncertainty.

Uncertainty-aware models are vital for safety-critical applications of *activity recognition* approaches, which range from manufacturing [176] to assistive robotics [227] and autonomous driving [143] . The impressive progress reported on the conventional action recognition benchmarks linked to the rise of deep learning [17, 58, 153] may therefore draw an artificially idealistic picture, as their validation is restricted to a static set of actions and overlooks, how well the model confidence estimates indeed correlate with the probability of a correct prediction [17, 89, 114]. Especially in the realistic open-world scenario, overly self-confident models become a burden in applications, and might lead to tragic outcomes if assessing model uncertainty in its prediction plays an important role[2]. Despite the alarm has been raised that neural networks are notably bad at detecting ambiguities [55, 62, 137, 192], examining how well the confidence values of activity recognition models indeed reflect the probability of a correct prediction has been overlooked in the past and is the main motivation of our work. Apart from the direct benefits of proper confidence values for decision-making systems, good assessment of uncertainty enhances model interpretability. For example, in the realistic scenario of open-world recognition, low-confidence input might be passed to human experts, which would pro-

2 A Tragic Loss | Tesla Inc. June 30, 2016, https://www.tesla.com/blog/tragic-loss.

vide the correct annotations (*i. e.* active learning) and therefore improve the decision boundary.

This thesis aims to elevate the role of uncertainty in the field of activity recognition and develop *uncertainty-aware models*, which do not only select the correct behavior class but are also able to identify *misclassifications* and previously *unknown behaviors* and develop strategies for dealing with such cases.

1.2 APPLICATIONS TO DRIVER MONITORING

An important application of activity recognition models is inside the vehicle cabin. Rising levels of automation increase human freedom, leading to drivers being engaged in distractive behaviors more often. The majority of traffic accidents involve secondary activities behind the steering wheel and an estimated 36% of such crashes could be avoided if the distraction had not occurred[3]. Besides identifying driver distraction for safety reasons, activity recognition may increase comfort e.g. by adjusting the driving style if the person is drinking coffee or turning on the light, when reading a book. Therefore, understanding driver behavior has strong potential to improve human-vehicle communication, dynamic driving adaptation and safety (such use-cases are discussed in Section 1.3 in detail). As in many other real-world scenarios, classification uncertainty plays a significant role in driver observation, as the set of possible behaviors is dynamic and unforeseen situations may occur at any time. Besides being potentially dangerous, false-positives caused by overly confident models are often highly disturbing for the user (*e. g.* if in the use-case of driver-centered adaptation the model repeatedly falsely recognizes that the human is reading and turns on the light).

Driver behavior understanding is closely linked to the broader field of video classification, where the performance numbers have rapidly increased due to the rise of deep Convolutional Neural Networks (CNNs) [87]. In contrast to conventional feature-based methods [185, 201], intermediate representations of such end-to-end architectures, are not defined by hand but *learned* together with the classifier [17, 58, 76, 153, 186, 197]. Presumably due to the data-hungry nature of such models and the insufficient in-vehicle datasets, the vast majority of driver activity recognition research is still grounded in the classical feature-based pipeline based *e. g.* on the body pose [4, 73, 113, 116, 142]. Furthermore, the *lack of transparency* and the inability to visualize internal decision processes resulted in CNNs being labeled as black boxes, considerably slowing down their integration in industrial systems. While studying model uncertainty is a growing area in image classification, this performance aspect has not yet received any attention in the field of activity recognition, constituting a further bottleneck for applications.

This thesis aims to bridge the gap between the CNN-based approaches for video classification, previously validated on a controlled static set of actions, and real-life applications for driver behavior recognition under presence of uncertainty.

3 Estimate taken from Dingus et al.: "Driver crash risk factors and prevalence evaluation using naturalistic driving data." Proceedings of the National Academy of Sciences 113.10 (2016): 2636-2641.

1.3 WHY AUTOMATIC DRIVER BEHAVIOR RECOGNITION?

In the light of rising automation, drivers become increasingly involved with tasks other than managing the vehicle. Understanding the situation behind the steering wheel makes human-vehicle cooperation more intuitive and safe, since automation is a gradual process and, for a long time, the human would need to remain attentive and intervene in uncertain cases. Applications of driver activity recognition models depend on the degree of vehicle automation [68] and range from improving driving comfort (*e. g.* automatically adjusting the light when the person is reading) to safety-critical functions, such as identifying distraction and, for example, sending a warning signal. We recognize four major use-cases for applications of driver activity recognition models in practice.

- **Improved safety through identified distraction.** Recent studies highlight that current activity directly affects human cognitive workload in both, general- and driving context [30, 129, 210]. For example studies by Deo and Trivedi (2019) [129] suggest that certain secondary activities such as *interacting with the infotainment unit* negatively impact the perceived readiness-to-take-over. Therefore, the key application of such algorithms at SAE levels 0 to 3 [68] is the assessment of human distraction and reacting accordingly, for example, with a warning signal.

- **Increased comfort through automatic driver-centered adaptation.** With the automation rising to SAE levels 4 and 5, increasing driver comfort by automatic adaptation of the vehicle controls becomes the more important use-case. For example, movement dynamics might automatically adjust depending on the detected activity (*e. g.* softer driving if the person is drinking tea or sleeping).

- **Novel intuitive communication interfaces.** The activity recognition task is highly related to the problem of gesture recognition. Visually recognizing gestures might lead to novel communication interfaces inside the vehicle, serving as a more intuitive alternative for the central console, as previous research identifies hand signals as a highly convenient way for human-machine interaction [49, 142, 176].

- **Prediction and prevention of dangerous manoeuvres.** A further safety-related application of driver activity recognition during manual driving is intention prediction. The majority of traffic fatalities is caused by inappropriate driving maneuvers due to human errors[105, 188]. Timely anticipation of driver intention offers a possible solution to prevent potential accidents at an early stage, allowing ADAS to notice that the person *e. g.* is going to induce a dangerous turn and prevent the accident by taking over the control or notifying the driver.

In the above overview we specifically target use-cases for recognition of either naturally happening or explicitly defined (*e. g.* gestures) activities. Of course, the broader field of visual recognition opens doors for numerous other possibilities inside the vehicle cabin. For example, facial recognition enables driver identification, while biometric measurments would allow the vehicle to automatically adjust the seat depending on the

height. Such topics, however, are outside of the scope of this thesis, as we focus on algorithms for capturing human *behavior*.

1.4 THESIS ROADMAP AND CONTRIBUTIONS

The goal of this dissertation is to develop algorithms for the visual recognition of driver activities with special regard to the resulting uncertainties in classification and finding strategies for dealing with such unknown situations without manual annotations. Specifically, we identify *four research problems*, crucial for long-term integration of activity recognition models in safety-critical or open world systems, with their underlying questions summarized in Figure 1. The main part of this thesis therefore comprises four chapters dedicated to each of these tasks, as we build models which are capable of recognizing the *known* activities in the context of standard closed-set recognition (Chapter 3), evaluate their uncertainty by identifying misclassifications (Chapter 4) or discover unknown behaviors (Chapter 5) and find a way to deal with such uncertain examples despite the scarcity of data (Chapter 6). A complete list of publications which resulted from this PhD research is provided in Appendix C.

CNN-based recognition of known activities

How can we recognize driver activities given the available training data? Chapter 3 is devoted to conventional supervised recognition of *known* driver activities, which we address with deep CNNs for the first time in the context of our application. We start by addressing lack of large-scale application-specific action recognition benchmarks, we collect and publicly release the *Drive&Act dataset*, featuring twelve hours of drivers engaged in secondary tasks while driving in manual and automated mode, for which we develop a fine-grained hierarchical annotation scheme. We adopt multiple CNN-based architectures for closed-set video classification to our task and examine them for different sensors and views, with their multi-stream fusion leading to the best recognition results (our *ICCV 2019* publication [121]).We continue with the related problem of maneuver prediction by visually observing the driver and present a new model which combines a 3D ResNet and an LSTM to forsee driver intent, achieving state-of-the-art results on the Brain4Cars benchmark (published in *IV 2019* [47]). As a side-exploration, we address multimodal gesture recognition and introduce different paradigms for connecting different modalities, including our C3D-Stitch model, which allows simultaneous information exchange at multiple network layers (published in *AMFG 2019* [175]).

Identifying failure cases under closed-set conditions

How do we build models, which identify misclassification? In Chapter 4 we go beyond the traditional goal of high top-1 accuracy and make the first step towards activity recognition CNNs capable of *identifying their failure cases* and *tracing back their root causes*.

To this intent, we measure the reliability of model confidence values and evaluate it for two prominent action recognition architectures, revealing that the raw *Softmax* values of such networks do not reflect the probability of correct prediction well. We then introduce a new model which learns to produce individual input-guided temperature values, which are used to scale the CNN logits dependent on the input representation through an additional calibration network (to appear in *ICPR 2021* [169]). Our **C**alibrated **A**ction **R**ecognition with **In**put **G**uidance (*CARING*) model consistently outperforms the native activity recognition networks and the original temperature scaling method (widely used for calibrating the confidence of image recognition models) in producing realistic confidence estimates. Furthermore, we tackle the black-box nature of 3D CNNs and introduce a diagnostic framework for analyzing the internal decision processes leading to the failure cases, *e. g.* by implementing spatiotemporal gradient-weighted class activation mapping for 3D CNNs (accepted at *ITSC 2020* [170]).

Uncertainty-aware open-set recognition

How do we differentiate between the known and the unknown categories, if novel actions can occur? A central part of this thesis are uncertainty-aware models for detecting novel behaviors introduced in Chapter 5, as in real-life we may always encounter new actions and being able to know *what we know* and *what we don't know* is decisive for the model to avoid what can be catastrophic consequences. In our next area of contributions, we therefore move to a setting, where new actions may occur at any time and introduce the concept of *open sets* to the area of driver observation and general activity recognition, where methods have been evaluated only on a static set of classes in the past. First, we formalize the problem and its evaluation testbed, presenting the *Open-Drive&Act*, *Open-HMDB51* and *Open-UCF101* benchmarks, where the model is additionally intended to identify behaviors not previously seen by the classifier. To provide strong baselines for these benchmarks, we implement a generic framework for open-set action recognition by combining closed-set models with multiple strategies for novelty detection (e.g. One Class SVM, neural network confidence). Then, we introduce a new novelty detection approach, which leverages the *uncertainty of the output neurons* using a Bayesian neural network approximation via Monte-Carlo dropout. In our *Bayesian I3D* model, output neurons decide on the novelty value of the example based on their uncertainty in a voting-like fashion. Our experiments feature different variants of the voting scheme and demonstrate clear benefits of uncertainty-based models, while *selective* uncertainty-based voting of the output neurons leads to the best recognition results (publications in *BMVC 2018* and *IV 2020* [168, 172]).

Recognition of unknown activities and domains

How can we handle unknown categories or domains, without additional annotations? Chapter 6 is dedicated to our final research question about dealing with uncertain examples, *i. e.* identified novel activity classes or changes in data distribution. We consider two

strategies for recognizing actions for which we do not have *annotated* data: transferring knowledge via language-based models (publications in *BMVC 2018*, *SiVL 2018* and *VL-LL 2020* [160, 168, 173]) and unsupervised learning from videos posted online, where we present the first framework for anticipating driver intents by learning from driving exam dialogs (under review at *IV 2021* [171]). Lastly, we address the recognition in unknown domains, as the distribution of the test and training data are rarely sampled from the same distribution in practice (*i. e.* through changes of sensor type or illumination). We formulate the problem of unsupervised domain adaptation for driver activity recognition, and present a new model for handling such distribution shifts by combining a variational auto-encoder for image translation with a classification-driven optimization strategy, leading to the best recognition results (published in *IV 2020* [161]).

While the main contributions of this thesis lie in the area of computer vision, the user and manufacturer perspective was continuously considered during the system design. Valuable input in this regard was provided by a multidisciplinary team of experts from research and automotive industry within the scope of the PAKoS project[4]. Such broader impact of the developed algorithms, *e. g.* , their adaptation in the vehicle ecosystem and the implications for human-centered control transition in highly automated vehicles is discussed in our recent book chapter [39].

A NOTE ON IMPLEMENTATION Alina Roitberg is responsible for implementation of the recognition frameworks described in Section 3.1, *all sections* of Chapter 4, *all sections* of Chapter 5, Section 6.1 and Section 6.2. Three sections are based on joint works resulting from very close collaboration with her Master Thesis students: Patrick Gebert (Section 3.2), Tim Pollert (Section 3.3) and Simon Reiß (Section 6.3). Both, Alina Roitberg and the corresponding student, have contributed substantially to this research. While it is difficult to set a precise boundary, Alina Roitberg was rather in charge of the idea while the student focused on the implementation.

4 http://www.projekt-pakos.de/

2 RELATED WORK

This thesis was influenced by a range of previously published literature, reaching from the theoretical research of uncertainty in deep learning to the more applied field of driver observation. While our results have pushed different research areas forward, it presumably had the highest impact on general- and driver activity recognition. The ways in which we advanced other fields can be roughly categorized in two groups: more *applied* contributions (*e. g.* introducing new datasets, adapting existing methods from other research fields to our task) and *algorithmic* contributions (*e. g.* our *CARING*, *Bayesian-I3D*, *CLS-UNIT* models). This chapter presents an overview of the most relevant literature.

2.1 GENERAL ACTIVITY RECOGNITION

Human activity recognition is a very active research area, strongly influenced by progress in image recognition methods, where the core classification is applied on video frames and extended to deal with the video dimension on top of it. While a high diversity of algorithms have been proposed to recognize human behaviour, the approaches can be roughly divided into two groups (see Figure 2) : (1) methods based on manually designed features and (2) end-to-end approaches based on Convolutional Neural Networks (CNNs) that act directly on the video data, so that the intermediate representations are not defined by hand but *learned* together with the classifier.

Feature-based methods, which have dominated the field for decades, follow the classical machine learning pipeline comprising two phases (illustrated in Figure 2 on the left). First, a feature vector representing the input data is estimated. The way the data is processed in this step is manually defined by human experts and is often based on the body skeleton [101, 174, 185, 220, 225], hand pose [46, 176], detected objects [132, 178] or local space-time feature descriptors[1], such as Space-Time Interest Points [95], HOG/HOF descriptor [88, 89, 96] or Dense Trajectories [167, 200]. The resulting feature

1 Note, that since this kind of feature computation often results in a varying amount of the resulting features, this group of approaches often require an intermediate third step of building a codebook, for example, with Bag-of-Words [96] of Fisher vector [201] approaches.

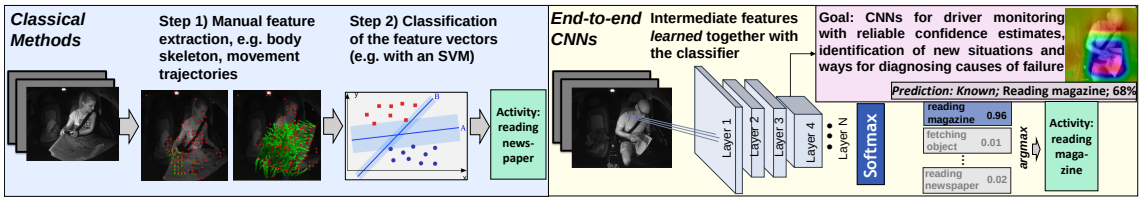


Figure 2: Classical machine learning pipeline (on the left) has dominated the driver monitoring field for years are based on features manually designed by humans (*e. g.* movement trajectories or body pose). Recent emergence of deep CNN-based architectures, which *learn* the intermediate representations automatically, has lead to a performance boost in recognition, but also to difficulties in understanding the outcome, handling domain shifts and learning from few examples.

is then passed to a machine learning framework based on, for example, Support Vector Machines (SVM) [89, 140], Hidden Markov Models (HMM) [88, 176] or Long Short Term Memory Networks (LSTM) [101, 185]. Before the deep learning revolution [86], Dense Trajectories and the derived Improved Dense Trajectories (IDT) [201] have dominated the activity recognition benchmarks[167, 179].

Similar to other computer vision fields, the methods have then shifted to deep CNNs (illustrated in Figure 2 on the right) which learn intermediate representations end-to-end. The first end-to-end architecture to outperform IDTs was the two-stream network [186], which comprises 2D CNNs operating on individual frames of color- and optical flow videos. The frame output is joined via late fusion [186, 204] or an additional recurrent neural network [32, 136]. The field further progressed with the emergence of 3D CNNs, which leverage spatiotemporal kernels to deal with the time dimension [17, 58, 76, 196, 197]. Today, approaches based on 3D CNNs deliver significantly better performance than those using hand-crafted [6, 179], with Inflated 3D ConvNet (I3D) [17], 3D Residual Network [58] and Pseudo 3D ResNet (P3D) [153] being the most prominent backbone architectures[2], (although there are highly promising *hybrid* methods [24, 217]).

Despite a remarkable number of newly published activity classification approaches, the objectives of such works are rather one-sided: the research is mostly focused on improving the accuracy on a fixed set of carefully defined actions [17, 58, 101, 185, 186, 217], with few methods targeting the computational efficiency [56, 85]. At the same time, recent studies from other areas have given alarming evidence, that the *Softmax* probability estimates of deep neural networks tend to be disproportionally high [137, 192, 193], expressing the need for new models, which are not only accurate, but also *reliable* in terms of their confidence.

2 With a high number of increasingly complex action recognition frameworks being published every month, the *raw* Inflated 3D ConvNet does not report the best results anymore, but the benchmark frontrunners are usually its modifications or extensions, so that the I3D is the most effective *backbone* architecture at the time. For example, the I3D accuracy on UCF-101 is 97.7%, while the best recent published method [24] achieves 98.2% by combining I3D with a pose based method (although the pose-based method alone achieves only 65.2%). An excellent accuracy-driven overview grouped by the individual benchmarks is provided at `http://actionrecognition.net`.

Our contribution: The above works develop algorithms with the incentive to improve the top-1 recognition accuracy on a *static* set of actions. This thesis elevates the role of uncertainty in the field of activity recognition, aiming at models, which do not only assign the correct class, but are also able to identify misclassification through *realistic confidence estimates* and function reliably under open-set conditions, being able to distinguish, between the *known* and the *unknown* concepts. We incorporate the reliability of model confidence in the evaluation of activity recognition models and demonstrate that out-of-the-box *Softmax* estimates do not reflect model uncertainty well. We then develop methods, such as our *CARING* model, which transform the biased confidence outputs into reliable probability estimates, drastically improving the results. We further introduce the new task of *open set activity recognition*, where the model is exposed to both: known concepts and action categories not present during training and propose new effective tools for dealing with it (with our *Bayesian I3D* model being the most important contribution).

2.2 DRIVER ACTIVITY RECOGNITION

Conventional activity recognition has undergone a prompt shift from machine learning approaches operating on hand-crafted features [174, 201] to end-to-end CNNs [6, 17] but this transition was comparatively slow in applications for driver monitoring. Existing algorithms perform a *coarse* classification of driver's state while focusing on a rather small set of secondary activities [98, 113, 141, 142], level of alertness [30, 110, 194], driving styles [118] or intended vehicle maneuvers [73, 74]. For example, Ohn-Bar *et al.* (2014) [141], evaluate their framework, comprising of head- hand- and eye gaze-based features classified with a hierarchical SVM, on three behaviours linked to the interior region: activities of the instrument cluster region, gear region and the steering wheel region. The proposed method achieves 94% accuracy but might draw an overly optimistic picture as the three classes are quite dissimilar. The very recent methods, such as the framework of Li *et al.* (2019) [98], applying a graph convolutional networks on skeleton data, distinguish ten driver states which are still highly distinguishable in terms of the body posture (*i. e. safe driving*, *drinking*, *reaching behind*).

Presumably due to the comparably small size of the datasets [73, 141, 214] and the data-hungry nature of CNNs, most of the approaches are based on manually defined feature descriptors, with a thorough overview published by Ohn-Bar *et al.* (2016) [143]. Such representations are often computed from the body pose [73, 98, 113, 142, 208], eye gaze [4, 141], hand location [116, 142], drivers' head pose [73, 116, 141], detected objects [208] or vehicle dynamics [73, 97], which is then passed to a classifier. Used classification approaches are fairly similar to the ones for standard activity recognition. For example, Martin *et al.* (2018) [113] and Jain *et al.* (2016) [73] classify feature vectors mostly derived from the body pose with an LSTM (although both use additional features, such as GPS position [73] or 3D interior model [113]).Other popular choices include

SVMs [141, 142], random forests [214] or HMMs [4, 73], but also more advanced methods, such as spatiotemporal graph networks applied on the body pose [98].

In end-to-end networks, which operate directly on the input video, feature extraction and classification merge *into one global model* (illustrated in Figure 2 on the right). Although CNN-based methods excelled in *general* human activity recognition benchmarks since ~ 2014 [186], feature-engineering approaches are still predominantly used for observing driver's state [98, 113, 143, 208]. Still, few works have explored application of CNNs inside the vehicle cabin concurrently with this thesis. For example, Abouelnaga *et al.* (2018) train a 2D CNNs to distinguish between ten driver postures [1], although their architecture also employs a face and hand detector. Xing *et al.* (2019)[213] recognize seven driving-related task (*e. g. normal driving* vs. *mirror checking*) by first separating drivers' body from the background with a Gaussian Mixture Model (previously trained with the purpose of segmentation), which is then passed to a 2D CNN. Besides being evaluated in a very limiting setting of few categories exclusively linked to manual driving, these approaches leverage image-based 2D CNNs, which are outdated in the field of activity recognition [17, 58]. Our goal is to adopt and systematically examine *spatiotemporal* 3D CNNs for video analysis to the field of driver monitoring at a large-scale.

Bedsides the high demand for annotated training data, one common concern when integrating CNNs in real decision making systems is their lack of transparency. The decision pathways of classical frameworks tend to be easier to interpret due to the controlled nature of the first feature-engineering phase (*i. e.* , feature calculation and -selection were designed by humans). In contrast, the obtained intermediate representations of CNNs are an enigma to the naked eye, leading to scepticism of many practitioners – a topic, which we will therefore also explore in this thesis.

Similar to the general video classification research, previous driver activity recognition works overlook the issues of uncertainty, mostly focusing on achieving high classification accuracy on a *fixed* set of driver states [1, 21, 47, 116, 121, 141, 142, 214, 218, 226]. To the best of our knowledge, no previous work has considered driver activity recognition under open set conditions.

Our contribution: This thesis enables large-scale integration of spatiotemporal CNNs in driver activity recognition frameworks, while, for the first time, also considering the classification uncertainty. As such models require high amount of labelled training data, we present the *Drive&Act* dataset for fine-grained driver behaviour analysis and conduct a systematic study of three off-the-shelf 3D CNNs, clearly demonstrating their advantages. Then, for the first time, we look at the *reliability of model confidence* for identifying cases of failure and overcome the closed-set constraint by introducing the notion of *open sets* to the field of driver observation. As already mentioned in our general activity recognition contributions, for both task, we introduce novel effective methods for addressing them (*CARING* and *Bayesian I3D*). Besides, since *interpretable* models are vital for building trust, we implement a diagnostic framework for understanding the internal decisions processes of driver monitoring CNNs, showcasing their potential to overcome

the "black-box" reputation and become more interpretable. Lastly, as part of our strategies for handling uncertain situations, we tackle the issue of domain shifts and introduce the task of unsupervised domain adaptation for cross modal driver activity recognition, for which we first leverage the existing image-to-image translation approaches and then improve them with our *CLS-UNIT* model.

2.3 MANEUVER PREDICTION THROUGH DRIVER OBSERVATION

Driver-based vehicle maneuver prediction aims to foresee the next event by learning characteristic behavioural cues preceding such the maneuvers (*e. g.* head turns). This task can be viewed as a branch of driver activity recognition, since we also need to assign a label to temporal data captured inside the cabin. Prediction of lane changes was addressed by Doshi and Trivedi (2008) [33] using a Relevance Vector Machine applied on features based on the body pose, and the vehicle dynamics, followed by Kumar *et al.* [93] using SVMs and Bayesian filtering. Jain *et al.* (2015) [72] presented the *Brain4Cars* dataset for driver maneuver prediction, featuring five distinct events (left-, and right- lane changes and turns as well as going straight). Evaluated frameworks also leveraged hand-crafted features derived from the head pose, facial landmarks and the driving context (*e. g.* GPS, car speed) and classified with a HMM in the initial work [72], which was then surpassed by using an LSTM on top of the multimodal features [74].

Our contribution: We have two contributions in the area of driver maneuver prediction. First, we present a new architecture for standard driver maneuver prediction combining a 3D CNN with an LSTM, outperforming previous *Brain4Cars* prediction methods. Second, we introduce the first framework for anticipating driver intent *without a single manually labelled* example by learning from conversations behind the steering wheel. We collect a dataset of mock road tests posted online comprising student-teacher dialogs and introduce a pre-processing technique for identifying and skipping smalltalk conversations. After the smalltalk refinement, we use the remaining relevant regions for foreseeing the next maneuver without any additional supervision. Our experiments indicate, that such multimodal data posted online can be successfully used as guides for learning novel concepts if no manual annotations are at hand.

2.4 DATASETS

The scale of video classification datasets has undergone an impressive development, growing in size by a factor of ~ 100 [3] over the past two decades. There is a variety of annotated color-based datasets for general activity recognition (usually derived from Youtube or movies) [2, 17, 89, 190] or more domain-specific purposes, such as cooking-

3 Estimated from comparison of the KTH Action dataset (2004) [183] with the Kinetics dataset (2017) [17].

	SoA conven. AR	**Multi-mod. AR**	**Driver Activity Recognition Datsets**							
	Kinetics [17]	NTU [185]	HEH [142]	Ohn *et al.* [141]	Brain4Cars [71]	D.P.-Night [218]	D.P.-Real [218]	AUC-D.D. [1]	Deo *et al.* [30]	**Drive&Act**
Year	2017	2016	2014	2014	2015	2016	2016	2017/18	2019	2019
Publicly available	✓	✓	✓	–	✓	–	–	✓	–	✓
Manual driving	–	–	✓	✓	✓	✓	✓	✓	–	✓
Autonom. driving	–	–	–	–	–	–	–	–	✓	✓
RGB/Grayscale	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Depth	–	✓	✓	N/A[b]	–	–	–	–	✓	✓
NIR	–	✓	–	–	–	✓	–	–	✓	✓
Skeleton	–	✓	–	–	–	–	–	–	✓	✓
Video	✓	✓	✓	N/A[b]	✓	✓	✓	N/A[b]	✓	✓
N° images	>76M	4M	N/A[b]	11K	2M	29K	18K	17K	> 5.6M	> 9.6M
N° synch. views	1	3	1	2	2	1	1	1	4	6
Resolution	N/A[c]	1920×1080[a]	680×480	N/A[b]	1920×1088	640×480	640×480	1920×1080	N/A[b]	1280×1024[d]
N° subjects	N/A[b]	40	8	4	10	20	5	31	11	15
Female / male	N/A[b]	N/A[b]	1 / 7	1 / 3	N/A[b]	10 / 10	N/A[b]	9 / 22	4 / 7	4 / 11
N° Classes	400	60	19	3	5	4	4	10	14[f]	83
Multi-level annot.	–	–	–	–	–	–	–	–	–[e]	✓
N° Levels	1	1	1	1	1	1	1	1	1[e]	3
Object annot.	✓	–	–	–	–	–	–	–	–	✓

[a] RGB resolution, IR/Depth resolution is 512×424
[b] information not provided by the authors
[c] variable resolution
[d] NIR-camera resolution
[e] main objective is readiness to take-over but 14 secondary activities are listed

Table 1: Comparison of driving and non-driving related datasets for action recognition with our *Drive&Act* dataset. We compare different properties of the two SoA *general* datasets for conventional- and multimodal action recognition and six datasets for *driver* activity recognition. This table is partially published in our ICCV 2019 paper [121] and is extended in this thesis.

related tasks [25, 88, 167], sports [78, 166] or robotics [77, 177]. While UCF-101 [190] and HMDB-51 [89] have been presumably the most active benchmarks, the Kinetics dataset [17] is slowly but steadily gaining popularity. The number of *multimodal* datasets is much smaller, with the NTU RGB+D dataset recorded with three synchronized Microsoft Kinect sensors being the largest one [185].

Available benchmarks become far more limited when we move to the driver observation context. In Table 1, we compare specifications of seven driving related datasets [1, 30, 141, 142, 218] to the two most prominent datasets for general action recognition: the color-based Kinetics [17] and the multi-modal NTU dataset [185]. The number of possible behaviour categories is much smaller then in general activity recognition, while most of these datasets contain only few images (under 30K, with the exception of Brain4Cars (2015) [71] and the dataset by Deo *et al.* (2019) [30] that include 2 Million and > 5 Million frames but address different tasks of maneuver prediction and readiness-to-take over estimation respectively).

With the emergence of 3D video classification CNNs with high data-demand, and many of the evaluations being conducted on *private* benchmarks [97, 113, 141, 142] there is an urgent need for large-scale datasets for driver activity recognition. Besides, with the rising vehicle automation, the *type* of behaviours used in the previous works becomes less relevant, as the drivers engage in more diverse non-driving activities. Furthermore, the above driver activity classification datasets do not cover a test setting with realistic open-set conditions.

Our dataset contribution: An important contribution of this thesis is *Drive&Act* – the first large-scale multimodal dataset for driver activity recognition specifically aimed at diverse behaviours of highly automated diving (specifications and comparison in Table 1). *Drive&Act* includes over 9.6 *million frames*, clearly more than any other previously published driver action recognition dataset. Our fine-grained hierarchical annotation scheme covers 83 *labels* in total, which is 62 more than previous driver-related benchmarks. With the unique qualities of *Drive&Act* (*e. g.* multimodal data streams, hierarchical, fine-grained annotations), we believe that the public release of our dataset will advance both, general and driver activity recognition research. Next, we introduce *Open-Drive&Act* and open set versions of *HMDB-51* and *UCF-101* – first benchmarks for activity recognition under open set conditions. Lastly, we collect the *Driver Talk* dataset by querying driving test videos from YouTube, which we use to address the new task of weakly supervised driver maneuver prediction by learning from driving exam dialogs.

2.5 CLASSIFICATION UNCERTAINTY AND NOVELTY DETECTION

2.5.1 *A Note On Uncertainty Categorization*

While various meanings have been given to the term *uncertainty* in the past literature, this thesis specifically refers to the "classification uncertainty of a discriminative model", which can be viewed as the inverse of model's confidence (as *e. g.* in Gal and Ghahramani (2016) [44], Malinin (2019) [109], Kendall and Gal (2017) [81]). Arising uncertainties of classification and regression models are often grouped into *epistemic*, or model uncertainty, and *aleatoric*, or data uncertainty, depending on the underlying source [81]. *Epistemic* uncertainty is caused by the "imperfect" model itself. It reflects our the lack of knowledge about the data and can therefore be an important cue for identifying whether we see an already known concept, or not. In other words, epistemic uncertainty can be reduced as we acquire more training examples. In contrast, *aleatoric* uncertainty is caused by the noise naturally present in the data and therefore cannot be reduced with more annotated samples. It is often further distinguished between *homoscedastic* (constant, caused , for example, by an imprecise sensor) and *heteroscedastic* (dependent on the input, *e. g.* , occlusions) aleatoric uncertainties. Note that the data-related uncertainties are often far more complex than the sensor noise. For example, imagine that we want to recognize the long-term activity of human in an autonomous vehicle but we only see him or her reaching for something on back seat. The driver might reach for a book in order to read or, maybe the driver is already watching a movie on the console screen and simply wants to additionally take something to drink. A perfect view of the person does not help as we simply do not have enough information to infer the long-term context. Note, that the line between these uncertainty categories is not always strictly defined [31].

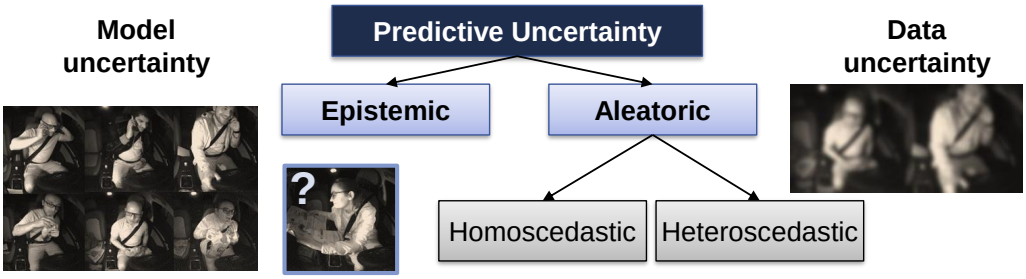


Figure 3: Sources of classification uncertainty are often categorized in two groups: *epistemic* (model or knowledge uncertainty) and *aleatoric* (data uncertainty). Aleatoric uncertainty is further divided into homoscedastic (constant for all inputs, *e. g.* , sensor noise), and heteroscedastic (different for different inputs, *e. g.* , occlusions.). Note, that the boundaries are not always strict.

2.5.2 *Quantifying Classification Uncertainty*

> *"The network learns better classification accuracy at the expense of well-modelled probabilities."*
>
> – Guo *et al.* , 2017

While multiple authors expressed the need for better uncertainty estimates in order to safely integrate deep CNNs in real-life systems [62, 137, 192], the feasibility of predicted confidence scores has been missed out in the field of activity recognition. However, this problem has been addressed before in image classification [44, 55], person identification [9] and classical machine learning [27, 138, 151].

In conventional CNNs for action recognition, output of the last fully-connected layer is normalized using the *Softmax* function, resulting in point estimates for a fixed set of classes [6, 17, 76, 136, 186]. The resulting *Softmax* scores are often directly interpreted as class probabilities, while these are only probability *estimates* used for maximizing the top-1 classification accuracy with the Cross Entropy (CE) loss [44, 54]. An off-the-shelf way to represent certainty of neural networks is to use the maximum probability estimates after the *Softmax* normalization, as done by Hendrycks and Gimpel (2017) [62]. Although the *Softmax* outcome correlates with the likelihood of a correct prediction, and, also CE is linked to the validity of confidence estimates, an alarm has been raised, that the raw *Softmax* values do not reflect the uncertainty well [44, 55, 137, 193]. Recent studies highlight that the *Softmax* values of modern CNNs tend to be strongly biased towards very high values[44, 55, 137, 193]. For example, Nguyen *et al.* (2015) [137] report state-of-the-art object recognition networks being 99.9% confident in their predictions for images unrecognizable to human eye. While giving excellent results in closed-set classification, such overly self-confident models become a burden under open-set conditions and in safety-critical applications. Nevertheless, *Softmax* estimates alone are often used as the basis for a rejection threshold in other computer vision tasks, such as obstacle detection [62, 156, 163].

Improving model confidence values has been addressed from different perspectives, often depending on the specific goal (*e. g.* obtain realistic probability estimates to reliably identify failures or detect out-of-distribution examples). The proposed algorithms can be roughly divided into (1) calibration-based methods, which learn to transform network output into realistic confidence estimates through re-calibration on held-out validation data, and (2) methods leveraging Bayesian Neural Networks (BNN) for obtaining probability *distributions*, although methods outside of these categories exist[94, 100].

CALIBRATION-BASED APPROACHES Calibration-based approaches can be viewed as post-processing models, since they usually freeze the classifier weights and learn to readjust the outcome on a held-out validation set to obtain realistic probability estimates. Recently, multiple calibration-based algorithms, such as isotonic regression [224], histogram binning [223], and Bayesian quantile binning [133] were brought in the context of CNN-based image classification by Guo *et al.* (2017) [55]. The authors introduced *temperature scaling*, a simple variant of Platt Scaling [151], where a single parameter is learned on a validation set and to rescale the neural network logits. Despite its simplicity, the temperature scaling method has outperformed other approaches in the study by Guo *et al.* [55] and has since then been successfully applied in natural language processing [91, 145] and medical applications [66]. Calibration-based approaches are highly effective as it comes to identifying success- and failure cases, given that the data of the validation set used for re-calibration and the test data are drawn from a similar distribution. Such methods therefore suit well under closed set conditions, but a different kind of approaches is needed to distinguish, between the known and the unknown categories. Calibration-based methods do not explicitly disentangle epistemic and aleatoric uncertainties, but they are better at capturing the latter one, since epistemic components can be only learnt through the differences of the training- and validation data [5, 146]. However, in an open set scenario, epistemic uncertainty is present through the *unknown* test classes, which are absent during re-calibration by definition [146].

BNN-BASED APPROACHES A powerful tool for addressing model uncertainty are Bayesian Neural Networks (BNN) [108, 135], which aim for the posterior *distributions* instead of single point estimates for each class. While exact BNN inference is computationally intractable, they are usually approximated with variational methods [11, 43, 53]. One very practical approximation is Monte Carlo dropout, introduced by Gal and Ghahramani (2016) [44], where the authors have provided a theoretical proof, that iteratively applying dropout at test-time and computing the output statistics is a variational approximation of a BNN. Furthermore, this technique is especially useful since it is able to specifically target *epistemic* uncertainty, which is very useful for detection of novel concepts. This approach has been successfully applied *e. g.* in semantic segmentation with the *Bayesian SegNet* model [80] and active learning [45]. We will later return to the Monte Carlo dropout technique, explaining it in depth in Section 5.3.

OTHER APPROACHES Several methods outside of the above groups have been proposed to quantify uncertainty of a neural network. Lakshminarayanan *et al.* (2017) [94] asses model confidence by using ensembles of several networks while also incorporating an adversarial loss function. Liang *et al.* (2017) [100] have recently shown, that the *Softmax* confidence estimates can be improved by corrupting the input.

2.5.3 *Classical Novelty Detection Methods*

Many novelty detection frameworks do not consider neural network classification uncertainty (which is a rather new research topic), but leverage a wide array of classical machine learning methods for quantifying the *normality* of a data sample, with an overview of such approaches provided by Pimentel *et al.* (2014) [150]. A lot of today's novelty detection research is handled from the probabilistic point of view [102, 128, 150, 189], modeling the probability density function of the training data, with Gaussian Mixture Models (GMM) being a popular choice [150]. The One-class SVM introduced by Schölkopf *et al.* [181] is another widely used unsupervised method for novelty detection, mapping the training data into the feature space and maximizing the margin of separation from the origin. Anomaly detection with NNs has been addressed several times using encoder-decoder-like architectures and the reconstruction error [209]. A common way for anomaly detection is to threshold the output of the neuron with the highest value [62, 112, 163]. Novelty detection through neural network confidence estimates gains popularity since Hendrycks *et al.* (2018) [62] presented a baseline for deep-learning based visual recognition using the top-1 Softmax scores and pointed out, that this area is underresearched in computer vision.

Our contribution: For the first time, we study the topic of *uncertainty* in the field of activity recognition, introducing two tasks to this field: obtaining reliable confidence estimates and identifying previously unseen activities under open-set conditions. Our *CARING* model introduced for the first task builds on the approach of Guo *et al.* (2017) [55], extending it with input-guided scaling. In contrast to [55], which uses a static temperature parameter for all data points, we introduce an additional calibration network to estimate a proper scaling parameter depending on the input. Furthermore, our benchmark examines the reliability of model confidence values in context of action recognition for the first time. To address the second task, we introduce the *Bayesian I3D* model which casts a 3D video classification CNN as a BNN using MC-Dropout [44] which we then use for novelty detection in action recognition. Furthermore, we leverage the uncertainty of multiple designated output neurons through a selective voting scheme. We also implement multiple classical novelty detection approaches, such as One-Class SVMs [181], as alternative modules for novelty detection used for comparison.

2.6 OTHER FIELDS INFLUENCING OUR WORK

2.6.1 *Image-to-Image Translation and Domain Adaptation*

Our work in Section 6.3, which is devoted to the issue of domain divergence, is influenced by the progress in unpaired image-to-image translation (*i. e.* mapping an image from a source domain to a different target space [228]), which experienced steep progress since the emergence of Generative Adversarial Networks (GANs) [50]. To this end, Zhu *et al.* introduced the concept of cycle-consistency [228], that entails the transfer back to the original representation employing a second GAN. At the same time, Liu *et al.* explored the idea of a shared-latent space, that aims to learn a joint representation of both distinct domains [103]. Image-to-image translation methods have been already successfully applied for unsupervised domain adaptation in fields such as digit recognition, semantic segmentation and person re-identification [12, 29, 65, 104, 131]. Concurrently with our work (Section 6.3.2.5) enhancing a Variational Auto-Encoder [103] with classification-based loss for cross-modal driver activity recognition, Rangesh *et al.* (2020) [157] propose a similar idea for eyeglasses removal inside the vehicle cabin, where they enhance a CycleGAN [228] with an additional loss for gaze classification.

Our contribution: We adopt and extend these image-to-image translation paradigms to handle domain changes inside the vehicle cabin, which, to our best knowledge, is explored for the first time in context of driver activity classification. We further present the *CLS-UNIT* model based on a Variational Auto-Encoder for learning domain-invariant latent representations [103], which we enhance with an additional classification-driven loss similar to the strategy employed by [65] in the context semantic segmentation. Our *CLS-UNIT* model consistently outperforms the baselines and other image-to-image translation approaches in the cross-modal setting.

2.6.2 *Multimodal Gesture Recognition*

As a side-exploration in the area of closed-set activity understanding (Section 3.3), we consider the topic of gesture recognition, specifically focusing on *multimodality*. In contrast to activity recognition, where most of the research is conducted on color-based data, gesture recognition is very often studied in multimodal context using the ChaLearn Isolated Gesture Dataset (IsoGD) dataset comprising color- and depth videos [198, 199]. In the recent gesture recognition challenge of Wan*et al.* (2017) [198], the majority of proposed methods on gesture recognition adopt the C3D [196] architecture as their backbone model. Fusing multiple modalities is done with late fusion by the vast majority of previous approaches. They train individual networks for each modality, which are then joined via score averaging [198], using Support Vector Machines (SVMs) [99], using Canonic Correlation Analysis [123] or by a employing a voting strategy [36]. Despite

the high correlation of information in the early stages of the multi-modal streams, the research of deep fusion at intermediate network layers has been scarce so far.

Our contribution: We conduct a systematic study of CNN-based methods for multi-modal fusion for gesture recognition, with the specific goal to develop strategies for *earlier*, *i. e.* convolution level-fusion. We enhance the C3D network which is the most prominent backbone in gesture recognition [198], with multiple fusion building blocks such as $1 \times 1 \times 1$ convolutions or cross-stitching units [127]. Our systematic comparison of different fusion strategies highlights the potential of fusion at earlier layers, with information exchange at multiple layers simultaneously being the most effective approach.

2.6.3 *Zero-Shot Action Recognition*

As already mentioned, the research of human activity recognition under open set conditions has been sparse so far. However, the related field of zero-shot activity recognition attempts to classify new actions (and *only new* actions) without any training data by linking visual features and the high-level semantic descriptions of a class, *e. g.* through action labels. The description is often represented with word vectors by a skip-gram model (*e. g. word2vec* [126]) previously trained on a large-scale text corpus. Zero-shot action recognition gained popularity over the past few years and has also been improving slowly but steadily [152, 205, 215, 216, 229]. The evaluation setting of such frameworks can be viewed as an opposite to conventional activity recognition: while standard action classification considers only known activities at test-time, zero shot activity recognition has been evaluated *exclusively on unknown* categories. In both cases, the distinction between *the known* and *the unknown* is assumed as given, which is not a realistic scenario. *Generalized* zero-shot learning, covering both, known and unknown concepts, has been recently studied for image recognition (Xian *et al.* (2017) [212]), reporting a drastic performance drop of classical ZSL approaches such as ConSE [139] and Devise [42].

Our contribution: We introduce the task of *generalized* zero-shot activity recognition, where the model needs to determine the correct action category, which can be both, known (*i. e.* standard supervised classification) and unknown (*i. e.* zero-shot knowledge transfer using language-based models). As an application of our *Bayesian I3D* novelty detection approach, we implement the first framework for *generalized* zero-shot activity recognition[4], where our model serves as the gate between known and unknown actions, consistently outperforming conventional zero-shot learning methods on *HMDB-51* and *UCF-101*. Additionally, we study the possibility of cross-dataset knowledge transfer for zero-shot action recognition, as utilizing large-scale external datasets for training and then generalizing to a smaller dataset of target actions might be useful for application.

4 Recently Mandal *et al.* , 2019 [111], has introduced a similar framework based on novelty detection one year *after* our *BMVC 2018* [168] publication presenting this work

Part II

UNCERTAINTY-AWARE ACTIVITY RECOGNITION

3 CNN-BASED RECOGNITION OF *KNOWN* ACTIVITIES

We begin with the classic problem of supervised activity recognition, assuming, that all test-time behaviours are *known* a priori through manually labelled training examples. The main goal of this chapter is to bridge the gab between the novel end-to-end methods for video understanding and their applications for driver activity recognition, while, at first, skipping the issues of uncertainty and an open world. This chapter is organized in four sections. Section 3.1 is based on our *ICCV 2019* publication [121] and adresses the main objective of this chapter – CNN-based driver activity recognition. We first tackle the lack of large-scale driver activity recognition datasets and collect the *Drive&Act benchmark* for which we create a fine-grained hierarchical annotation scheme. We adopt multiple CNN-based video classification architectures to our task and examine them for different sensors, views, and their combinations. In Section 3.2, we turn to the related task of maneuver prediction through driver observation and present a new model combining an optical flow network with a 3D ResNet and an LSTM (based on our *IV 2019* publication [47]). While our main goal is behaviour recognition, as a side-exploration, in Section 3.3 we tackle gesture recognition, where specifically focus on multimodality and analyse multiple existing and novel ways of fusion at convolution level (published in *AMFG 2019* [175]). Section 3.4 summarizes the scientific impact of this chapter, concluding our research of conventional supervised closed-set recognition.

3.1 DRIVER ACTIVITY RECOGNITION WITH CNNS

This section is based on our publication in *ICCV* 2019 [121], © IEEE .

3.1.1 *Motivation for a new dataset*

Like other applied research fields, driver observation is closely linked to progress in the more general area of computer vision, where recognition numbers rapidly increased due

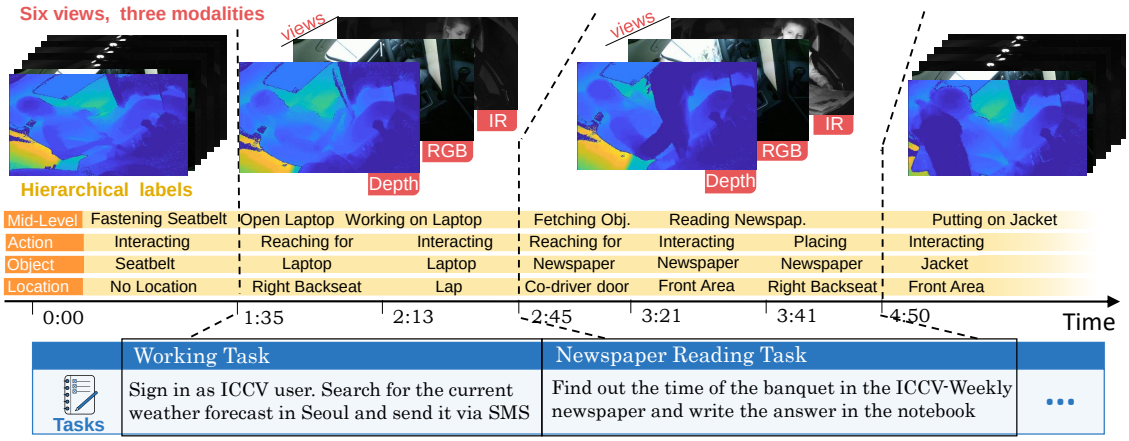


Figure 4: Overview of the *Drive&Act* dataset for fine-grained driver behaviour recognition. *Drive&Act* covers frame-wise hierarchical labels of 9.6 Million frames and fifteen drivers.

to the rise of deep learning [17, 60, 186, 196]. While 3D CNNs conquered almost every general activity recognition benchmark, they were rather left behind by driver behaviour understanding research. One cause of such delayed integration is that a large amount of accurately labelled data is a key to CNNs' success. Existing driver activity recognition works are often evaluated on private benchmarks [141, 218] and are limited to the classification of very few lower-level states (*e. g.* whether the person is holding the steering wheel, or switching gear [141]), while being considerably smaller in size compared to the general action classification datasets (details are given in the related work, Section 2.4). Besides, standard activity recognition research mostly considered color-based sensors, which rely heavily on sunlight and are therefore not applicable inside a vehicle.

In the face of rising automation leading to an increased driver freedom, benchmarks with small restrictive sets of possible behaviours become obsolete. We do not only require *large-scale* datasets in terms of the amount of labelled examples needed to train the data-hungry CNNs, but should also look at the *type* of driver activities from a different perspective. Rising levels of automation increase human freedom, leading to drivers being engaged in distractive behaviors more often while the type of activities become increasingly diverse. For example, *working on laptop* or *reading magazine* behind the steering were almost unthinkable until now, but these behaviours become more common as the driver is gradually relieved from actively steering the car. Although distractions become safer as the vehicle becomes more intelligent, this change does not happen from one day to another and is a rather long-lasting transformation [195]. Over-reliance on artificial intelligence might lead to catastrophic consequences, and, for a long time, the driver will need to intervene in case of uncertainty [106, 154, 195]. However, there are important long-term application scenarios of driver monitoring even in fully-autonomous cars. For example, understanding the situation inside the vehicle cabin may increase comfort *e. g.* by adjusting the driving style if the person is drinking tea or being an intuitive communication interface via gestures. Our goal is therefore to cover diverse behaviours not bounded by the need of *active* steering (*e. g. changing clothes* or *reading newspaper*),

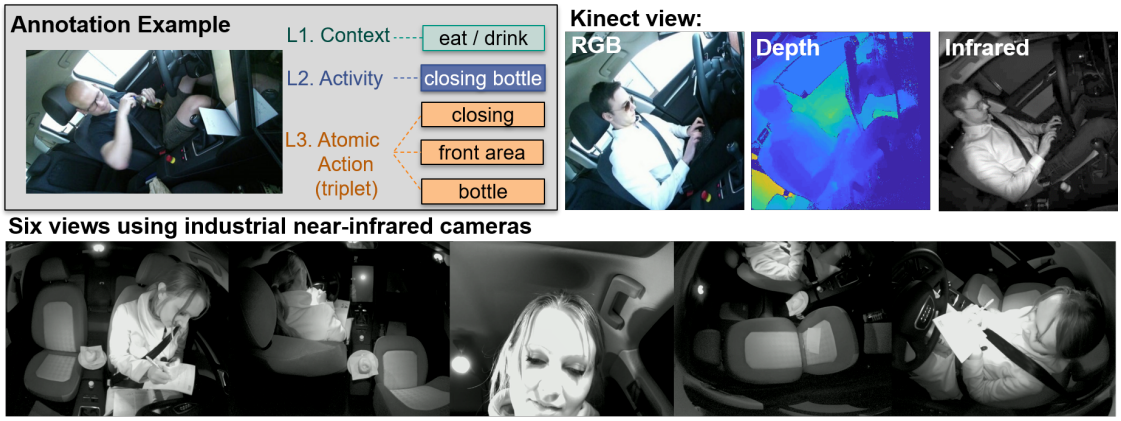


Figure 5: Captured data streams of the *Drive&Act* dataset. Dix distinct views cover the vehicle cabin, recording three different modalities: near-infrared (×6), color (×1) and depth (×1) data. Video frames are densely annotated with a hierarchical annotation scheme.

while specifically considering highly automated driving, which, to the best of our knowledge is not covered by any of the existing datasets.

To tackle the lack of large-scale driver activity recognition datasets, we create and publicly release the *Drive&Act* benchmark[1], featuring twelve hours of people engaged in secondary tasks behind the steering wheel (overview in Figure 4). *Drive&Act* covers challenging recognition tasks linked to practical applications of the video classification models and is the first publicly available dataset, with the following qualities:

- **High-level driver activities** in context of automated driving (83 labels).
- **Multi-modality**: color-, depth- and infrared-data, as the models trained on standard color-based action recognition datasets would strongly rely on sunlight.
- **Multi-view**: six calibrated cameras cover the vehicle cabin from different perspectives to deal with limited body visibility.
- **Hierarchical activity labels** consider three levels of abstraction and complexity, from the long-term tasks to primitive interactions with the environment.
- **Fine-grained** distinction between the individual categories (*e. g.* *opening bottle* and *closing bottle*) and **high diversity** of action duration and complexity, which is typical for application but makes the recognition especially challenging.

1 In the *Drive&Act* bechmark, Alina Roitberg is responsible for implementation and experiments regarding the end-to-end models (which are described in depth), while Manuel Martin implemented the body pose-based approaches, which results are given for comparison only. As it comes to the *Drive&Act* creation, both have contributed significantly in all phases of data collection as part of their PhD research. While setting a strict line is hard, Alina Roitberg focused more on creation of the activity vocabulary used for the annotation, while Manuel Martin focused on the sensor setup. The accents of this thesis is set accordingly.

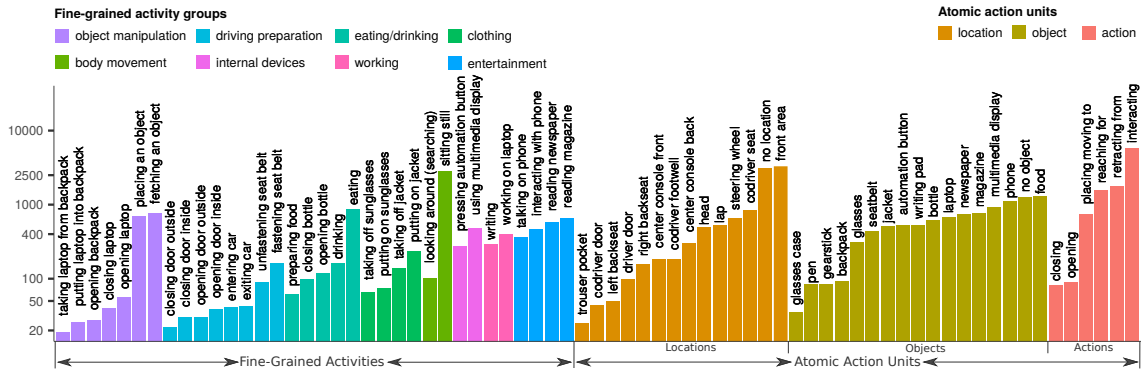


Figure 6: Sample frequency of fine-grained activities (left) and atomic actions (right) by class (we use logarithmic scale). One sample corresponds to a 3 second video snippet with the assigned label. Colors of the fine-grained activities group them roughly by their meaning (*e.g.* food-related).

Apart from the autonomous driving applications, our dataset fills the gap of large multi-modal benchmarks for fine-grained action recognition with multiple levels of abstraction. Our benchmark is therefore of interest for general computer vision research, while our evaluation of modern approaches for video classification highlights the difficulty of *Drive&Act*. Our dataset is publicly available at www.driveandact.com.

3.1.2 *Hierarchical Vocabulary of Driver Actions*

The first question to ask when building a visual recognition system is *what do we want to recognize?* Ideally, we would identify behaviours which (1) are indeed typical for driving and (2) would be useful for the manufacturers, for example, since they have a strong effect on accident odds or are linked to an interesting multimedia application. To adequately represent real driving situations, we conducted a thorough literature review on secondary tasks during manual driving using three types of sources: (1) driver interviews, (2) police reviews of accidents, as well as, (3) naturalistic car studies [10, 51, 67, 84]. Key factors for the choice of the in-cabin scenarios have been the *frequency* of activity engagement while driving and action *impact* on drivers' attention (*e. g.* via increased accident odds). Furthermore, we asked five experts from car manufacturing industry and research to rate individual activities in terms of their usefulness for future applications on a numerical scale and provide additional feedback [2]. The results indicate high interest in classes such as *talking on a mobile phone*, *working on a laptop*, *searching for something* and recognition of basic body movements (*e. g. reaching for something on the floor*), while actions such as *smoking cigarette* were rated as less useful. Certain categories, such as *sleeping*, were omitted due to technical feasibility. Following the literature review and the expert survey, we define a vocabulary of relevant driver activities from eight areas: *eating and drinking*, *clothing and accessories*, *working*, *entertainment*, *entering/exiting and car adjustment*, *body movement*, *object manipulation* and *using vehicle-internal devices*.

Guided by this analysis, we derived a hierarchical three-level vocabulary of driver actions, covering 83 labels in total. The three levels represent different degrees of granularity, building a complexity- and duration-based hierarchy. The *scenarios/tasks* (level 1) are linked to the goals

2 We have questioned the expert partners from the PAKoS project: www.projekt-pakos.de

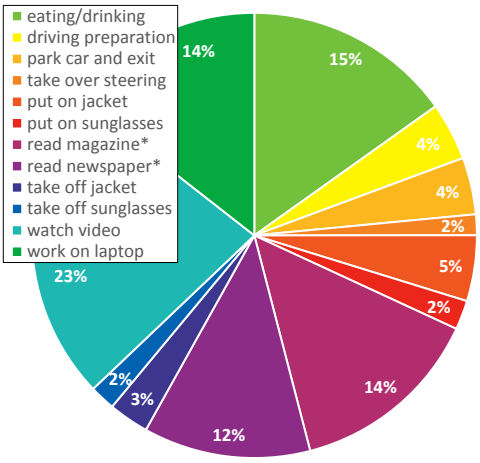


Figure 7: Distribution of the scenarios/tasks (1st hierarchy level). * these tasks consist of both finding information about a previously asked question by reading a newspaper/magazine and of writing the answer into a notebook.

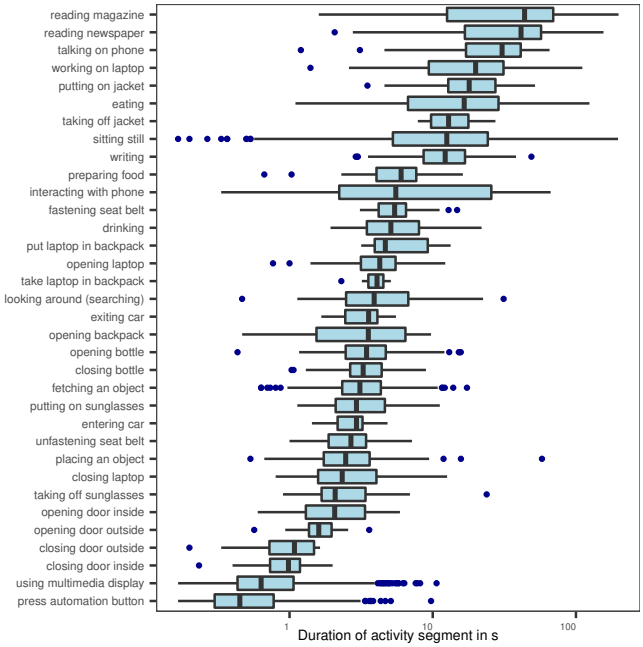


Figure 8: Duration statistics of the *fine-grained activities* (2nd hierarchy level) as boxplot (log. scale).

the driver was asked to achieve and are therefore long-term behaviours, which comprise more concise *fine-grained activities* (level 2), serving as our main evaluation level. The fine-grained activities are further broken down into atomic action units (level 3) – primitive building blocks of human behaviour, such as simple object interactions (examples in Figure 5 and Figure 4). We will now explain the three abstraction levels in depth.

3.1.2.1 *Level 1: Scenarios / Tasks*

The twelve tasks our participants were asked to complete in each session[3] form the *first level* of our hierarchy and are either scenarios typical during manual driving (*e. g. eating and drinking*) or highly distracting situations which are expected to become more frequent with the increasing level of automation (*e. g. using a laptop*). Figure 7 illustrates the frame-wise frequency statistics of the *scenarios* as a pie plot. This analysis reveals that our drivers spent most of the time (23%) in the entertainment task (*i. e.* watching a video), while the most rare segment was driving manually after a take over request. The *take over* scenario is special, because the subject was unexpectedly asked to interrupt what he or she was doing to take over and switch to manual driving.

3.1.2.2 *Level 2: Fine-grained Activities*

The *second level* portrays the *fine-grained activities*, breaking down the coarse *scenarios / tasks* into 34 more precise categories. Unlike the upcoming third level of *atomic action units*, behaviours of the second level preserve a clear semantic meaning. Certainly, different degrees of abstraction are causally connected, as complex composite behaviors are often built from multiple more simple

3 the recording process will be explained in Section 3.1.3

activities. In contrast to the predefined scenarios/tasks, which the subject needed to accomplish, the fine-grained activities alternate freely in one session, as the participants are *not* instructed *how* to solve the given task.

A key challenge for recognition at this level arises from the fine-grained nature of the classes, as we differentiate between *closing bottle* and *opening bottle* or between *eating* and *preparing food*. We believe, that many applications require such concise discrimination, as the basic elements of the scene (*i. e.* the vehicle cabin or the loose body position) often stay similar, while the relevant category-differences occur at a smaller scale than in standard video recognition datasets. As a consequence of such concise discrimination, the frequency of the categories in the dataset is not even, as seen in Figure 6, which illustrates the class distribution. On average, our dataset features 303 samples per class, with *taking laptop from backpack* being the rarest (only 19 samples) and *sitting still* being the most frequent category (2797 samples). Note, that we refer to the three second chunks as our samples (as explained in the forthcoming sectionSection 3.1.3.3 regarding the dataset splits). The fine-grained activities are also diverse in terms of their duration: (Figure 8) reveals the statistics of how long the individual activity segments last. For example, *reading a magazine* often takes almost a minute, while other activities, such as *pressing the automation button*, on average, last less than one second.

3.1.2.3 *Level 3: Atomic Action Units*

The *atomic action units* portray the basic building blocks of complex human behaviour of the previous levels and are primitive interactions with the environment. Such action units represent the lowest degree of abstraction and are disconnected from a higher semantic meaning. We define an atomic action unit as a triplet of *action*, *object* and *location*. We cover 5 types of actions (*e. g. reaching for*), 17 object classes (*e. g. writing pad*) and 14 location annotations (*e. g. co-driver footwell*), with their distribution statistics provided in Figure 6. In total, 372 distinct combinations of action, object and location were recorded in *Drive&Act*.

ADDITIONAL ANNOTATIONS We provide multiple additional annotations which we did not directly use *yet*, but which might be helpful in the future: the *driving-state*, indicating whether the driver is in the automated driving mode or steering with the left, right or both hands, time stamps of the take over requests and simulator-internal signals *e. g.* the steering wheel angle.

3.1.3 *Data Collection*

3.1.3.1 *Recording Procedure*

To motivate diverse distractive behaviours without putting the pedestrians or the driver at risk, *Drive&Act* was recorded in a *driving simulator*, where the simulated scene was projected on the screen. At the same time, the driving session took place in a *real* vehicle (Audi A3) placed inside the simulation environment and modified so that the vehicle controls (*i. e.* steering, brake) are directly translated into the simulation.

With the created annotation hierarchy in mind, we designed a data collection protocol which would allow us to capture the desired classes, while also giving the driver enough freedom and keep the environment realistic. Each driver completed two sessions, where he or she was asked to complete twelve different tasks, which shape our first annotation level (two instruction exam-

ples are illustrated in Figure 35) The first task comprises entering the car, making adjustments, beginning to drive manually and switching to the autonomous mode after several minutes. All following instructions (*e. g.* look up the current weather forecast with the laptop and report it via SMS), were displayed on a tablet placed inside the central console, with the task order being randomized in every session. Most of the tasks are completed while driving autonomously. However, in every session, four *unexpected take over requests* were triggered through an audio signal. As a result, the journey was continued manually for at least one minute. We want to remind, that while the sequence of the coarse tasks was directly given through instructions, the person could freely decide *how* to approach them (*i. e.* exact way of their execution, which is often reflected in the fine-grained activities, was left to the subject).

Overall, fifteen drivers (four females and eleven males) were recorded in *Drive&Act*. With an exception of one subject, all participants completed the procedure twice, resulting in 29 driving sessions with an average duration of 24 minutes (the first session oftentimes lasted longer than the second one, since the subject was not very familiar with the environment).

3.1.3.2 *Sensor Setup and Recorded Data Streams*

Multi-modality is an essential concept in our framework, since each sensor has its individual advantages and limitations. For example, a large number of recognition models available for *color* images are well-suited for adaptation to other application domains via transfer learning, while such RGB sensors require active illumination and fail at night. Depth images, on the other hand, are less influenced by the texture (*e. g.* clothing), and Near-Infrared (NIR) cameras are less dependent on the illumination.

The multimodal *Drive&Act* setting features an Audi A3 equipped with six cameras of two types: (1) a Microsoft Kinect v2 sensor is mounted on the passenger side of the A-pillar, facing the driver and (2) five NIR cameras are placed to complement each other so that the complete cabin is covered (Figure 5). The NIR cameras[4] record 30 Hz videos at a 1280×1024 pixel resolution. In the long run, we aim to disentangle activity recognition from conventional color input in the favor of lightweight near-infrared cameras, which are also effective at low lightning conditions. These industrial cameras would also be our favourite modality of choice. Still, we acquire and release data acquired by Kinect, which is less practical but popular in the computer vision community as it delivers three modalities simultaneously: color (950×540 pixel at 15 Hz), infrared (512×424 at 30 Hz) and depth (512×424 at 30 Hz).

Additionally to the video streams, we provide dense frame-level annotations – the videos were labeled manually by humans using the hierarchical annotation scheme described in Section 3.1.2.

3.1.3.3 *Dataset Splits*

We randomly split our dataset *person-wise* into *train* (20 subjects), *val* (4 subjects) and *test* (20 subjects) subsets, so that our benchmark validates generalization to new drivers. *train* is used for model optimization, *val* for hyperparameter tuning and potentially selecting the checkpoints (in case of early stopping) and *test* is used for the final evaluation. Since the activities vary greatly in their duration (as shown in Figure 8), we split each annotated action segment into chunks of 3 seconds (or less, if the chunk is shorter) and use them as samples in our benchmark.

4 Camera specs: en.ids-imaging.com/store/ui-3241le.html

3.1.4 *Neural Architectures*

In this section, we implement and adapt off-the-shelf CNNs for general activity recognition to the driver observation task. In contrast to feature-based approaches, CNNs operate directly on the video data *i. e.* intermediary representations are not explicitly defined, but *learned* through the convolution filters. Our goal is to systematically evaluate whether such end-to-end architectures suit the task of driver activity recognition, where the feature-based are still predominantly used [73, 75, 113, 115, 141, 142].

C3D The C3D model [196] is the first widely-used CNN leveraging 3D convolutions for action recognition. C3D consists of 8 convolutional layers ($3 \times 3 \times 3$ kernels) and 5 pooling layers ($2 \times 2 \times 2$) followed by two fully-connected layers. Besides being the first framework for generic spatiotemporal feature extraction from videos, it is compact and efficient through the small kernel sizes. It takes a 112×112 video snippet of 16 frames as input and produces a 4096-dimensional video feature, which is then classified with a fully connected layer.

INFLATED 3D CONVNET Inflated 3D ConvNet (I3D) [17] is presumably the most widespread backbone activity recognition architecture at the time. The model builds upon the Inception-v1 network [69] by extending the 2D filters with an additional temporal dimension. I3D stacks 9 characteristic Inception modules: small sub-networks which execute $5 \times 5 \times 5$, $3 \times 3 \times 3$ and $1 \times 1 \times 1$ convolution operations in parallel and concatenate the output, while keeping the number of operations low by reducing the input dimensions via $1 \times 1 \times 1$ convolutions. The complete I3D network consists of 27 layers: three convolution layers at the beginning and one fully-connected layer at the end, four max-pooling layers at the beginning and one average pooling layer preceding the last fully-connected and nine inception modules which themselves are two layers deep. A useful quality of I3D is its ability for knowledge transfer from 2D datasets. First, a two dimensional version of I3D is trained on an image recognition dataset, such as ImageNet[28]. Then, the convolution kernels are inflated to become 3D-kernels, with the learned weights being copied along the time dimension. According to this procedure, one could reuse the knowledge learned from still images, where very large datasets are available, and fine-tune the model for the application-specific video data. The input to I3D are 64 frames at 224×224 resolution.

PSEUDO 3D RESNET Apart from such 2D-to-3D inflation, another way for reusing the available 2D CNNs on spatiotemporal video data is to first convolve them spatially (where pre-trained 2D models can be used), and then convolve the time dimension only. Following this paradigm, Pseudo 3D ResNet (P3D) [153] "mimics" 3D convolutions by combining a filter on the spatial domain (*i. e.* $3 \times 3 \times 1$) with one in the temporal dimension (*i. e.* $1 \times 1 \times 3$). Furthermore, P3D ResNet leverages residual connections due to improve the gradient flow, allowing a remarkable depth of 152 layers. P3D operates on 64 frame snippets with 160×160 pixel resolution.

3.1.5 *Experiments*

Our task is to assign the correct activity label given a video segment of 3 seconds or less (in case of shorter events). Following the standard practice, we use the *balanced accuracy* as our main performance measure: first, we obtain the accuracy for every category individually (*i. e.* the recall) and further use the mean over all classes as our final metric.

Type	Model	Validation	Test
Baseline	Random	2.94	2.94
Pose	Interior	45.23	40.30
	Pose	53.17	44.36
	Two-Stream [202]	53.76	45.39
	Three-Stream [113]	55.67	46.95
End-to-end	C3D [196]	49.54	43.41
	P3D ResNet [153]	55.04	45.32
	I3D Net [17]	**69.57**	**63.64**

Table 2: Fine-grained Activities recognition on our Drive&Act dataset. We group the examined models into: (1) baselines, (2) feature-based approaches and (3) CNN-based end-to-end methods that operate directly on the input videos.

Camera	View	Validation	Test
NIR Cameras	front top	69.57	63.64
	right top	65.16	60.80
	back	54.70	54.34
	face view	49.73	42.98
	left top	68.72	62.83
	combined	72.70	67.17
Kinect Color	right top	69.50	62.95
Kinect Depth		69.43	60.52
Kinect IR		72.90	64.98
Combined		73.80	68.51
All combined (score averaging)		**74.85**	**69.03**

Table 3: Fine-grained activity level recognition results for different modalities and views and their combination (I3D model).

We compare the end-to-end models described in Section 3.1.4 to the random classifier (varying between 0.31% and 16.67% depending on the hierarchy level) and to a feature-based approach using the drivers skeleton and the vehicle interior models [113, 202]. The framework leverages a Long Short-Term Memory (LSTM) Unit [63] on top of the extracted features and was introduced by Wang *et al.* [202] and further adopted by Martin *et al.* [113] for driver observation[5]. In addition to the body skeleton (obtained with the *OpenPose*[6] [16] and *OpenFace*[7] [8] libraries), this method includes the 3D interior model of the car, *i. e.* the distances of the hands to certain vehicle cabin sections, such as the gear stick or the seats, as in [113].

We evaluate our models separately for every hierarchy level: 12 scenarios/tasks (first level), 34 fine-grained activities (second level) and atomic action units with 372 possible combinations of the *{Action, Object, Location}* triplets (third level). Because the amount of triplet combinations is very high, we also report the performance for correctly classified Action, Object and Location separately (6, 17 and 14 classes, respectively). We view the fine-grained activities as our main evaluation level and therefore do not only compare the models among each other, but also conduct a thorough evaluation of different modalities, views, and their combinations.

3.1.5.1 *Fine-grained Activities*

We begin with our main evaluation level of the *fine-grained activities*, first comparing different model among each other using the *front top* NIR view for the end-to-end models in Table 2. Overall, we achieve a mean per-class accuracy between 40.3% and 63.64%, compared to 2.94% of the random baseline. The Inflated 3D Model yields the best recognition rate (63.64% on *test*, 69.57% on on *val*), surpassing the feature-based approaches and other end-to-end models by a large margin, while not using any additional information (such as the 3D vehicle model).

5 All implementations and experiments of the *feature-based* methods were conducted by Manuel Martin and are reported in this thesis to provide a comparison to the classical feature-based framework.

6 https://github.com/CMU-Perceptual-Computing-Lab/openpose

7 https://github.com/TadasBaltrusaitis/OpenFace

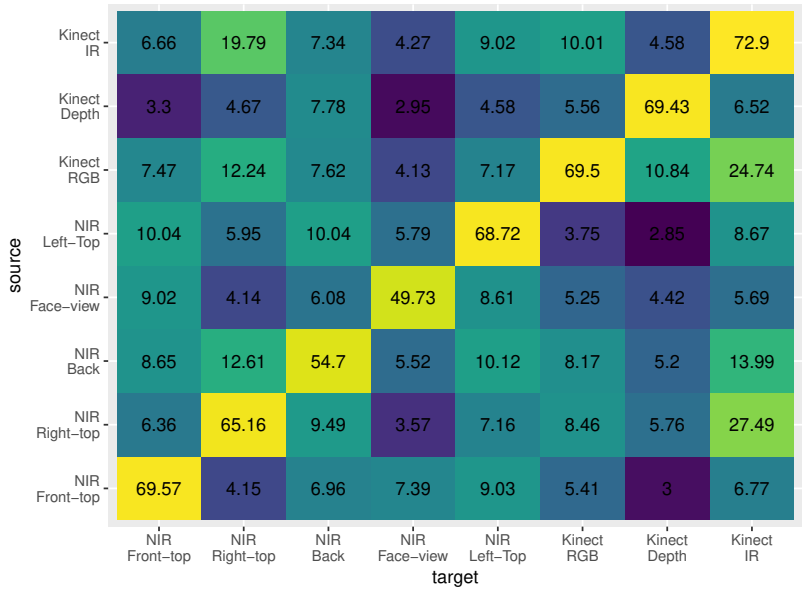


Figure 9: Validation accuracy of cross-view recognition: the I3D model trained on data from *source* is evaluated on the *target* view. Note, that random baseline is at 2.49%. Although all models surpass the random classifier, it is evident that domain shifts are a significant weak spot of such approaches. Domain-invariant end-to-end is therefore an important future research direction.

We now focus on the best-performing I3D model and examine it for the individual views and modalities and their combinations through averaging of the *Softmax* output scores (multimodal results in Table 3). There is a clear correlation between the general scene visibility and the classification success. For example, the *face view* setting yields the lowest performance (42.98%) as only the driver's face is recognizable. In contrast, the *front top* camera has an excellent view of driver's body and close objects. While the best single-view results were achieved using the *Kinect IR* data (64.98%), for real world applications, we would recommend the *front top* view of the *NIR* camera, because it is far smaller, easier to integrate and less dependent on the illumination, while losing only 1.3% in accuracy. While the best single-view results are achieved using the *Kinect IR* data (64.98%), late fusion of multiple inputs consistently improves the recognition (69.03% using all sources). Automatic activity recognition clearly benefits from multimodality, as the using all sources simultaneously leads to the best recognition rate (69.03% on *test*, 74.85% on *val*), outperforming the best feature-based model by 22.08% on *test* and 19.18 on *val*.

Our next area of investigation is a rather novel setting of cross-modal and cross-view recognition. In this testbed, we evaluate our best performing end-to-end method (I3D) on a view not previously seen during training (results in Figure 9). As data-driven models are susceptible to changes in data distribution, cross-view recognition is an exceptionally hard task and the performance drops significantly (although all models achieve better results than the random classifier). 27.49% of the fine-grained activities were correctly identified in the *Kinect IR to right top NIR* view setting and 24.74% in the cross-modal *Kinect color to Kinect IR* setting. Our results demonstrate the sensitivity of modern CNN-based action recognition models to domain shifts and highlight the need for further research of methods for handling such changes. We will later address this issue when targeting the *unknown* activities and domains (Section 6.3) and propose measures to improve recognition under such challenging conditions.

Model	Scenario/Task		Action		Object		Location		All AAUs	
	val	test	val	test	val	test	val	test	val	test
Random	8.33	8.33	16.67	16.67	5.88	5.88	7.14	7.14	0.39	0.31
Pose	35.76	29.75	57.62	47.74	51.45	41.72	53.31	52.64	9.18	7.07
Interior	37.18	32.96	54.23	49.03	49.90	40.73	53.76	53.33	8.76	6.85
Two-Stream	39.37	34.81	57.86	48.83	52.72	42.79	53.99	54.73	10.31	7.11
Three-Stream	41.70	**35.45**	59.29	50.65	55.59	45.25	**59.54**	**56.5**	11.57	8.09
I3D Net	**44.66**	31.80	**62.81**	**56.07**	**61.81**	**56.15**	47.70	51.12	**15.56**	**12.12**

Table 4: Recognition results for the Atomic Action Units (AAU) defined as *{Action, Object, Location}* triplets (the four left columns) and coarse Scenarios/Tasks (the right column) .

3.1.5.2 *Atomic Action Units Classification*

Next, we provide the results of the *atomic action units* classification in Table 4. We show the performance of each value in the {Action, Object, Location} triplet individually, as well as, the overall accuracy of the triplet values combined. The CNN-based methods consistently show better results for *object* (56.15%) and *action* classification (56.07%), while feature-based approaches are better in inferring the *location* (56.5%). This is a rather expected result, as *e. g.* the three-stream method takes the 3D interior features as input, which is highly useful for inferring the location, while the end-to-end methods often employ pooling, causing a loss of exact location information. However, in the combined evaluation (*i. e.* a sample counts as correctly classified if *all* three components are accurately predicted), the I3D model clearly achieves the best results.

3.1.5.3 *Scenarios/Task Recognition*

The results of the task classification (level 1, highest level of abstraction) are reported in Table 4 (left column). Surprisingly, the *test* and *val* results are inconsistent in this case (best *test* results of 35.45% are achieved using the three-stream approach, best *val* results of 44.66% are obtained with the I3D model), while the overall recognition rate is lower than in other levels. We assume, that the 3 s chunks are too short for high abstraction of the task level and the results are strongly influenced by noise, so that the networks rather learn biases instead of capturing the underlying nature of such long-lasting tasks. We therefore presume that this hierarchy level would strongly benefit from a time window longer than the current 3s segments.

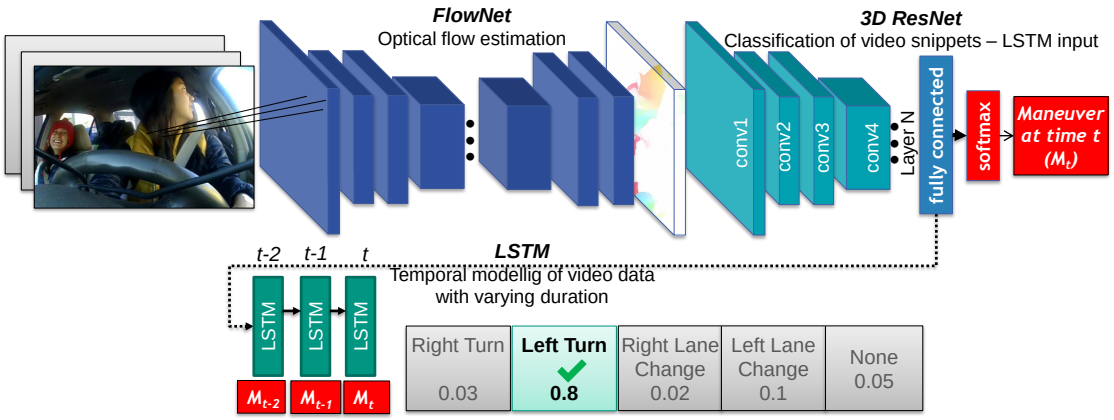


Figure 10: Overview of the proposed neural network-based framework for maneuver prediction.

3.2 DRIVER MANEUVER PREDICTION

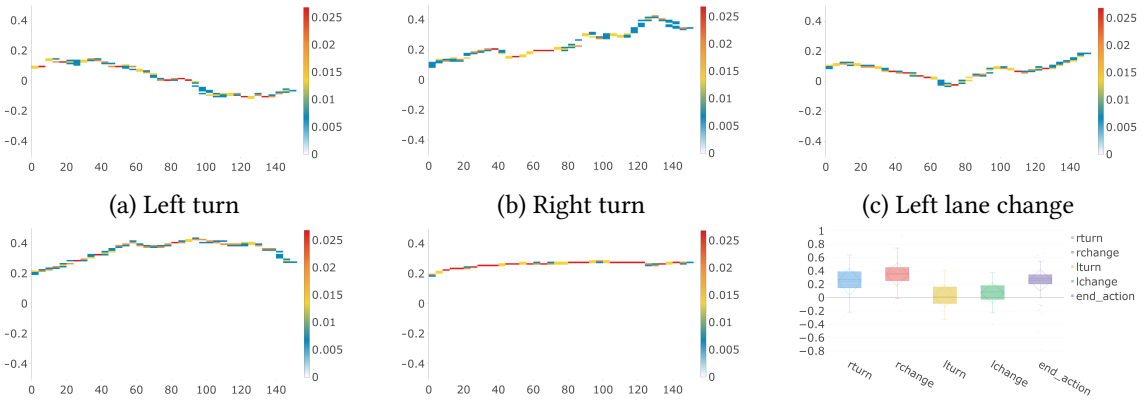
This section is based on our publication in *IV* 2019 [47], © IEEE .

Can we foresee dangerous events before they were induced? While the previous section focused on driver activity recognition, we now tackle the problem of vehicle maneuver *prediction* by observing the driver, which aims to find the future driving event based on a given video sample. Despite extraordinary progress of Advanced Driver Assistance Systems (ADAS), an alarming number of over 1.3 million people are still fatally injured in traffic accidents every year[8]. Human error resulting in inappropriate manoeuvrers is the main cause of such misfortune [162, 187], as by the time the ADAS system has alarmed the driver, it is often too late.

Vehicle maneuver prediction has been widely researched in the last decades, mostly handled by classifying hand-crafted features,*e. g.* , eye gaze, head and body pose or context features, such as GPS and car speed [33, 72, 74, 93, 143]. We leverage the recently emerged computer vision approaches for end-to-end video analysis and present a deep-learning based framework for predicting driver's intent. Our model comprises three components: a neural network for estimating optical flow, a very deep video classification network based on 3D convolutions and residual connections and an LSTM for handling data of varying input length (overview in Figure 10). We apply our method to both, driver observation data from inside the vehicle cabin and the street view data from an outside camera by fusing both sources via a multi-stream network. We demonstrate the effectiveness of our CNN-based approach on the publicly available Brain4Cars [72] benchmark, being able to predict the maneuver with an accuracy of $83,12\%$ and $4,07s$ in advance, outperforming previous approaches while using the inside view only.

Next, we provide a formal definition of the driver maneuver anticipation task (Section 6.3.1) and describe the network components of the proposed model (Section 3.2.2).

8 The World Health Organization: https://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en/



(a) Left turn (b) Right turn (c) Left lane change

(d) Right lane change (e) No induced maneuver (f) Box plot of mean yaw angle

Figure 11: Drivers' motion prior to different maneuvers. Figures 11a-11e are a 2D-histograms of the head trajectory distribution 6s before the event was induced (statistics calculated over the Brain4Cars dataset using multiple drivers). The X and Y axes show the time frame and the yaw angle of the drivers' head respectively, the color stands for the frequency in the corresponding 2D-bin. Figure 11f summarizes the mean head pose preceding the events as a boxplot.

3.2.1 *Maneuver Anticipation Task*

Consider that a person behind the steering wheel is inducing a certain driving event (*e. g.* a *left turn*) at time point T while being monitored with a video camera. Our goal is to correctly predict the executed event *before* it took place. Given a training set with m known maneuver categories $\mathcal{A}\{1, ..., m\}$, our task is to assign the correct event label $a_t \in \mathcal{A}$ to the video frames $(x_0, ..., x_t)$ preceding the driver action by $T - t$, where $t < T$.

The basic task consists of classifying the complete observation sequence from the start time of the video up to the last point before the action took place $t = T - 1$. Since the core application idea of our model is to intervene, when the human is about to induce a dangerous maneuver, it is useful to predict such event *earlier* than $T - 1$ and therefore restrain the video snippet to earlier time points. The difficulty of our maneuver prediction task is dependent on the duration to the next maneuver (*i. e.* it is more challenging for smaller t). We therefore consider both tasks: 1) maneuver prediction directly before a maneuver event ($t = T - 1$) and 2) assessments up to 6 seconds prior to the maneuver ($t < T - 1$).

3.2.2 *Neural Architecture*

Conceptually, our maneuver prediction model (see Figure 10) consists of three components: 1) an optical flow extraction network, 2) a convolution neural network based on 3D convolutions with residual connections for classification of the maneuver label and 3) an optional LSTM for handling video data of variable lengths, which we describe in the following.

3.2.2.1 *Motion Representation*

MOTIVATION – BODY MOTION AND MANEUVERS The driver is oftentimes *active* while planning to induce next steering maneuver (*e. g.* looking over the shoulder to check for bypassing

Figure 12: Optical flow visualization of motion inside the cabin prior to a *left turn*.

vehicles). We therefore aim for movement-based video representation instead of still images. To validate this, we statistically analyse head movement patterns preceding different maneuvers. Figures 11a-11e depict the distribution of the drivers' *yaw head angle* five seconds before the event computed using the Brain4Cars dataset [72]. The histogram statistics illustrate that the drivers' head motion preceding these maneuver types is often different.

Figure 11f summarizes the statistics of the head yaw angle prior to the driving event, showing that the head is often turned in the direction of the maneuver. The lane change maneuver is also clearly linked to the head motion but the characteristic pattern is softer than for the turns. Furthermore, constructing a simple threshold classifier based on the yaw angle of the last 6 seconds only, allows us to correctly assign future maneuver 37, 6% of the times, which is clearly above random chance (20%). This statistical analysis confirms the role of human motion as a discriminative feature for foreseeing driving intention.

OPTICAL FLOW NETWORK Inspired by this analysis, we leverage *motion* videos by employing an *optical flow* extraction network as the first component of our architecture. The optical flow obtained from consecutive frames describes a displacement vector in the X- and Y-direction for each pixel in the image. We use the FlowNet 2.0 [35] architecture and transform the obtained 2D displacements into the RGB space[9]. In Figure 12 we provide the example output of the optical flow network for consecutive video snippets prior to a *left turn* translated into the color space. We observe the person turning the head to the left and back before the event is induces.

3.2.2.2 *Maneuver Classification via 3D CNN - LSTM model*

3D RESNET FOR FEATURE LEARNING Our framework is based on the ResNeXt-101 architecture – a 3D convolutional residual neural network for action recognition proposed by Hara *et al.* , which has demonstrated highly promising results in standard activity recognition, especially when using optical flow videos [57]. This neural network learns to assign the forthcoming maneuver on 16 frame optical flow snippets. The 101-layered 3D convolutional architecture consists of an ensemble of shallow ResNeXt blocks – a series of convolution layers with ReLu, residual connections and batch normalization. Each ResNeXt block consists of three 3D convolutional layers and utilizes a *group convolution* in the middle layer, which divides the feature maps into smaller groups with a cardinality of 32. The complete ResNeXt-101 architecture consist of an initial convolutional layer with $7 \times 7 \times 7$ kernels and 64 feature channels, followed by 3 ResNeXt blocks with 128 channels, 4 blocks with 256 channels, 23 blocks with 512 channels and 3 blocks with 1024 channels. The resulting 2048 channels are combined in a global average pooling layer, followed by a fully connected and a *Softmax* layer with m neurons, where m corresponds to the number of classes (in our case $m = 5$). Since the datasets for driver intention prediction are too small for training end-to-end models from scratch, we use a model pre-trained on the large-scale

9 We employ the publicly available PyTorch implementation of [159] pre-trained on the Sintel dataset [122].

Kinetics dataset for human activity recognition [17]. Then, we fine-tune the model on the optical flow visualization of the Brain4Cars video samples, which are always 150 frames long. We train the network with cross-entropy loss and SGD with a learning rate of 0.1 which is divided by 10 after the validation loss saturates, a weight decay of 0.001 and a momentum of 0.9.

LSTM FOR HANDLING VARIABLE VIDEO LENGTHS. While the described CNN architecture is suitable for prediction from fixed-sized videos, it is unable to handle time series of varying length. In practice, a good model needs to predict the maneuver as soon as possible. Sometimes the cues for the future action are visible six seconds before, while in other cases, the drivers' intention might not be visible until very close to the event execution. Ideally, we would continuously handle the incoming data and predict the maneuver at different time steps.

To deal with varying input sizes, we combine the 3D ResNet architecture with a Long Short-Term Memory (LSTM) network [64]. LSTM is a recurrent neural network (RNN) based on a gated network cell unit. Gate units control the internal network state by regulating the data flow through the cell unit. A basic cell unit comprises an *input*, *forget* and *output gate* which control the amount of input information to be stored and thus the amount of past knowledge that can be "forgotten". In contrast to RNNs, the cell gates enforce a constant error flow back through the network graph, therefore mitigating the vanishing (or exploding) gradients problem. LSTMs can connect long time intervals without losing the ability to learn short time dependencies.

The last fully connected layer of the ResNet is used as an input to the LSTM network with two layers and 30 hidden units each. Then, a fully connected layer is used with the number of neurons set to the number of maneuver classes (five for the Brain4Cars dataset), followed by Softmax normalization. The input is split up into blocks of 25 frames which get passed to the LSTM at each time step. The LSTM network is trained jointly with the 3D ResNet using stochastic gradient descent for 150 epochs and an initial learning rate of 0.001 which gets divided by 10 after the validation loss saturates. The loss is calculated after every time step in order to ensure both late and early predictions. We use a momentum of 0.9 and a dropout rate of 0.5.

3.2.3 *Experiments*

EVALUATION SETTING We evaluate our model on the publicly available Brain4Cars benchmark [72] – a naturalistic dataset for driver-focused maneuver prediction. Brain4Cars covers both – inside and outside videos 6 seconds prior to the event[10]. The inside video data captures the frontal view of the driver, while the outside camera faces the street scene ahead. Originally, [72] reports that the dataset compromises 700 vehicle maneuvers sampled from 10 test subjects. However, we found that a portion of training data is missing and only 594 maneuver videos are publicly available: 234 videos of driving straight, 124 videos of left lane change, 58 videos of left turn, 123 videos of right lane change and 55 videos of right lane change. Following the experimental setting of Jain *et al.* [72], we evaluate our model using 5-fold cross validation. In the final results, we report the mean and standard deviation over the five test sets, while one of the four remaining splits were used as a validation set for the hyper-parameter tuning.

10 The Brain4Cars dataset is available at http://brain4cars.com. Please note, that a part of training data is missing (only 594 of the reported 700 videos are made available).

Method	Inside	Outside	*Acc* [%]	±*SE*	F_1 [%]	±*SE*
Baseline Methods						
Chance	–	–	20,0	–	20,0	–
Prior	–	–	39,0	–	–	–
Feature-based Methods from [72] and [74]						
IOHMM	✓	✓	–	–	72,7	–
AIO-HMM	✓	✓	–	–	74,2	–
S-RNN	✓	✓	–	–	74,4	–
F-RNN-UL	✓	✓	–	–	78,9	–
F-RNN-EL	✓	✓	–	–	80,6	–
Our 3D-ResNet-based Architecture						
Outside-only		✓	53,2	± 0,5	43,4	± 0,9
Inside-only	✓		**83,1**	± 2,5	**81,7**	± 2,6
Two Stream	✓	✓	75,5	± 2,4	73,2	± 2,2

Table 5: Zero time-to-maneuver results: accuracy and F_1 score computed of different models on the Brain4Cars dataset using 5-fold cross-validation (mean and standard deviation over the folds). The approaches in the second group are the best performing models of Jain *et al.* [72, 74].

METRICS We evaluate our approach with two metrics: the multi-class classification accuracy, as in driver activity recognition, and the *adjusted* F_1 score for detecting the maneuvers (*i. e.* the harmonic mean of the precision and recall), since this was the metric used by related work [72]. The latter metric is *adjusted* to the task of maneuver detection, so that the *driving straight* maneuver is handled differently. Specifically, Jain *et al.* [72] define the precision and recall as follows:

- true prediction (tp): correct prediction of the maneuver
- false prediction (fp): prediction is different than the actual performed maneuver
- false positive prediction (fpp): a maneuver-action predicted, but the driver is driving straight
- missed prediction (mp): a driving-straight predicted, but a maneuver is performed

$$Pr = \underbrace{\frac{tp}{tp + fp + fpp}}_{\text{Total \# of predictions}}, \quad Re = \underbrace{\frac{tp}{tp + fp + mp}}_{\text{Total \# of maneuvers}} \tag{1}$$

Using precision and recall, we calculate the F_1 score as:

$$F_1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{2}$$

ZERO TIME-TO-MANEUVER PREDICTION First, we evaluate our model in the *zero time-to-maneuver* setting, where the complete videos up to the last frame before the starting point of the maneuver are used as input (see Table 5). Additionally to the comparison with the previous work, which focused on hand-crafted features, we investigate our model's performance on the inside- and outside view as well as the combination of both via the multi-stream networks.

The model trained on the inside view achieves the best performance, surpassing the multi-stream architecture by 7,6% and the outside view by 29,9%. We link this surprisingly strong performance of the inside-only model to the knowledge transfer by pre-training our model using the

Maneuver	Pred. Frame	Pred. Time [s]	FTM	TTM [s]
Left turn	52,0 ± 6,7	2,1 ± 0,3	98,4 ± 6,7	3,9 ± 0,3
Left change	49,2 ± 4,3	2,0 ± 0,2	100,8 ± 4,3	4,0 ± 0,2
Right turn	56,0 ± 3,7	2,2 ± 0,2	94,1 ± 3,7	3,8 ± 0,2
Right change	40,8 ± 2,3	1,6 ± 0,1	109,2 ± 2,3	4,4 ± 0,1
End action	43,4 ± 3,3	1,7 ± 0,1	106,6 ± 3,3	4,3 ± 0,1
Avg. ± SE	48,3 ± 2,8	1,9 ± 0,11	101,7 ± 2,8	4,1 ± 0,1

Table 6: Varying time-to-maneuver evaluation: prediction time, Time-To-Maneuver (TTM) and Frame-To-Maneuver (FTM) of the final LSTM network using a 3D ResNet for feature extraction.

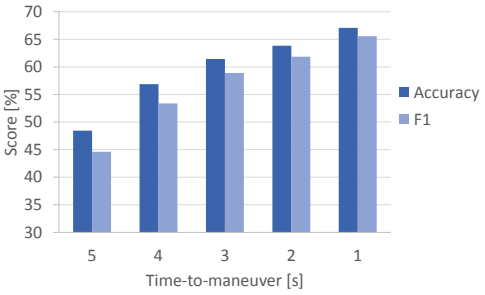


Figure 13: The accuracy and the F_1 score depending on the time-to-maneuver. The recognition rate is higher, the closer we are to the event starting point.

Kinetics dataset [17] for human activity recognition. The Kinetics dataset [17] is highly human-centered and the transferred structures might be less applicable to the street images. Of course, another important reason for this high recognition accuracy remains the usefulness of observing the driver, as human motion and behavior patterns differ significantly in the preparation stages of different maneuvers. We want to mention, that pre-training the 3D ResNet component on a large-scale dataset focused on classification of *outside* scenes might lead to better results for the street view variant of our model.

Our end-to-end model is able to predict driver intention with an accuracy of 83,12% and an F1 score of 81,74%, advancing the state-of-the-art. Note, that, at the same time, our model was optimized on 15% less training data than the reference approaches [72, 74] and did not use any additional context features, such as GPS coordinates or vehicle speed.

3.2.3.1 *Varying time-to-maneuver prediction*

We now move to the *varying time-to-maneuver* setting, where our goal is to predict driver event up to 5s earlier, taking different time steps into consideration. Such prediction is only possibly by employing an LSTM on top of the ResNet model to handle data input of varying durations (see Section 3.2.2). Following the *time-to-maneuver* evaluation procedure from [72] we find the earliest time step when a test sample is predicted correctly.

On average, the maneuvers were predicted correctly 4.1 seconds before the event has been induced by the driver (as reported in Table 6). Additionally, to the *time-to-maneuver* analysis, we provide the accuracy and F1 score depending on the period of time before the beginning of the maneuver in Figure 13. Of course, prediction at an earlier stage is a more difficult task and the accuracy drops compared to the zero-time to maneuver case. Still, over 60% of events are correctly predicted 3 seconds before their occurrence, which is significantly higher than the random classifier (20% for five maneuver types).

Figure 14: Example of a gesture in the multimodal IsoGD dataset. The data captured by a color-based camera is strongly influenced by the illumination conditions *e. g.* the shadows produced by the light source to the left. However, the depth data can have problems recognizing even the hand if the hand has the same depth as other objects close to it *e. g.* the hand touching the wall.

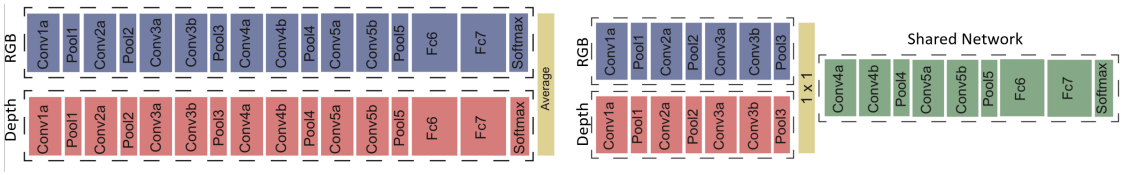
3.3 MULTIMODAL GESTURE RECOGNITION

This section is based on our publication in *CVPR AMFG Workshop* 2019 [175], © IEEE .

The automotive sector clearly benefits from gesture recognition systems. Besides classifying the implicit drivers' behavior in order to, *e. g.* improve safety through identified distraction, explicitly communicated hand gestures provide a novel medium for human-vehicle interaction, attempting to detach the input from conventional devices, such as board computers. This assumption is empirically supported by studies highlighting gestures as an intuitive and highly efficient way of human-machine communication [49].

Gesture recognition models are often studied in multimodal context using the color and depth videos provided by the "Chalearn Isolated Gesture Recognition Dataset" (IsoGD) [99, 123, 198]. While multiple published CNN-based frameworks recognize gestures from color and depth data simultaneously, they usually comprise separately trained models for each modality, which are afterwards joined at the very last stage (*e. g.* late fusion via score averaging). The majority of architectures evaluated on IsoGD (see [198] for an overview), leverage C3D as their backbone for each of the modality streams (note, that C3D was originally developed for activity recognition and previously described in Section 3.1.4), while all multimodal frameworks leverage the late fusion. After the CNNs are trained individually for each input type, their last output layer is used for fusion, usually via score averaging [198, 203], but also with Support Vector Machines [99], using Canonic Correlation Analysis [123] or by a employing a voting strategy [36]. Despite a high correlation between data streams, the options of fusing the information at earlier stages has not been explored in the area of gesture recognition yet. The main objective of our work is to implement and systematically examine different CNN-based fusion strategies for multimodal gesture recognition with deep neural networks, covering both, the conventional *late* combination of the results and multiple models, where the information exchange happens *earlier*.

Given the high correlation of the input data from the beginning, we argue that gesture recognition models might benefit from fusion at intermediate layers and take a closer look at different fusion strategies for gesture recognition especially combining information at *intermediate* layers. To examine our premise, we adopt the C3D architecture [196] based on 3D convolutions as our backbone model, which is widely used in gesture recognition [99, 198]. We systematically compare three fusion strategies on the widely used C3D architecture to the single-model counterpart: 1) late fusion which combines the streams in the final layer; 2) information exchange at a single



(a) Standard late fusion: separate depth- and RGB-networks do not exchange information up until the final prediction (*i. e.* the Softmax layer) where the confidences for each gesture class are averaged.

(b) Intermediate single-layer fusion : we combine the two streams at a chosen layer via $1 \times 1 \times 1$ convolutions. After this the two streams are merged to a single shared network using concatenation.

Figure 15: Overview of the single layer fusion architectures for gesture recognition.

intermediate layer using $1 \times 1 \times 1$ convolution, which is then passed to a shared late network; and 3) linking information at multiple layers simultaneously using the *cross-stitch units*, which were originally proposed for multi-task learning. Our proposed *C3D-Stitch* model achieves the best recognition rate, demonstrating the effectiveness of sharing information at earlier stages.

We begin by describing the backbone architecture (Section 3.3.1.1) and analyzing the conventional late fusion approach (Section 3.3.1.2). Next, we adress the question, "how can we fuse the information at an earlier stage" and explore two strategies. First, we consider linking the streams earlier at a single layer via $1 \times 1 \times 1$ convolutions, so that the two networks become a single shared network after the fusion (Section 3.3.1.3). As our second strategy, we propose to learn to exchange the information at multiple layers simultaneously via cross stitch units and introduce a new *C3D-stitch* architecture (Section 3.3.1.4).

3.3.1 *Fusion Strategies for Multimodal Gesture Recognition*

3.3.1.1 *Backbone Architecture and Preprocessing*

We use C3D [196] as our backbone architecture, since it has been very popular specifically for gesture recognition (architecture details already described in Section 3.1.4). More precisely, our pipeline leverages *two* C3D networks – one each modality, while our goal is to find good ways for linking their information at different stages. We train the model with a learning rate of 0.0001, momentum of 0.9 and a batch size of 10. We initialize the weights for both, color and depth streams, using a model pre-trained on the Sports-1M [79] dataset for large-scale action recognition.

As we aim to fuse the output at earlier stages, and therefore at the *convolution* layers, correct spatial alignment between the feature maps of different modalities is important. However, the color- and depth frames of the IsoGD dataset are not perfectly aligned. In order to register the different views, we calculate the homography between the RGB and depth frames via multiple corresponding points. This operation aligns the views, therefore increasing their correlation. Following the original C3D implementation [196], we first rescale the videos to a resolution of 128×171 pixel. The input to the C3D network are then 16 cropped frames of 116×116 pixel.

3.3.1.2 *Late Fusion Approach*

We begin by implementing the standard *late fusion* paradigm, where we combine the outputs of the two networks through their last fully-connected layer by score averaging – a common method in gesture and action recognition [186, 198, 203]. We investigate three different policies to train

the model: 1) individual training of the two networks with two separate losses, 2) joint training of both networks in an end-to-end fashion, with a single loss estimated after averaging, and 3) a multi-step technique, where we first pre-train the networks on each modality individually and thereafter fine-tune them jointly. An overview of the C3D network with the late fusion paradigm is illustrated in Figure 15a.

3.3.1.3 *Mid-level Fusion with Shared Late Network*

Our main incentive was to develop approaches which exchange the stream information *earlier*, so that the useful early correlations are preserved. Our first intuition is to use separate streams at early layers and, then, fuse them into a joint model in a later stage (as depicted in Figure 15b). This can be achieved by using $1 \times 1 \times 1$ convolutions followed by concatenation of the two output feature maps. The input shape for a single shared network of the next layer (after the fusion) should have the same dimensionality as each of the two inputs to the fusion modules. Thus, we reduce the number of output filters by half in each $1 \times 1 \times 1$ convolution layer (*i. e.* we divide the number of filters by the number of streams). In other words, we employ the $1 \times 1 \times 1$ convolutions to decrease the dimensionality within the filter space. The final architecture therefore consists of three components: two early-stage networks corresponding to each individual modality and a shared network for the final stage, which leverages the shared input representation.

An important question is *when* to fuse the information, *i. e.* we need to select a layer after which the two networks become one. To examine this, we implement and compare different variants of the model, with fusion placed different stages. We follow the same learning procedure as for the late fusion (Section 3.3.1.2). Furthermore, similar to Section 3.3.1.2, we evaluate both variants, with and without pre-training on the individual modalities.

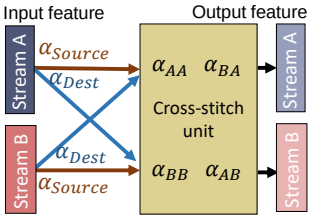
3.3.1.4 *Fusion on multiple Levels via Cross-stitch Units*

Until now, we manually selected the *point-of-fusion* (*i. e.* the layer, after which the two separate networks fuse the data and the shared network starts). Our next idea is to build a model, where both, the *individual* or *joint* data flow is enabled as the information exchange is learned at *multiple layers simultaneously*. We present a novel multi-stream paradigm, which consists of individual C3D networks for each modality passing the information to each other after each convolution and fully connected layer. In this architecture, the output of each of these layers is combined via a learned weighted average called cross-stitch units[11] [127] (see overview of the *C3D-Stitch* model in Figure 16b). In other words, at every stage, all networks contribute to each other's input pairwise, while the extend of this contribution is learned end-to-end.

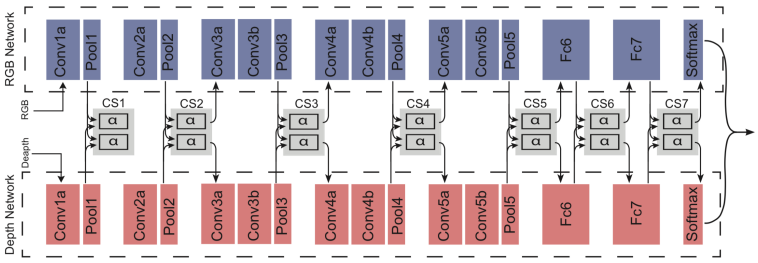
The cross-stitch units take two activation maps from both streams and pieces them together through a linear combination with learned weights, which is passed to the next layer of each stream (*i. e.* we learn *two* linear combinations for two of the output streams). More formally, let x_A, x_B be the feature maps of the two networks after layer ℓ (*e. g.* output of one of pooling following a convolution layer). The objective is to learn the linear combination $\hat{x}_A$, $\hat{x}_B$ of the two feature maps x_A, x_B:

$$\begin{bmatrix} \hat{x}_A^{i,j} \\ \hat{x}_B^{i,j} \end{bmatrix} = \begin{bmatrix} \alpha_{AA}^{\ell} & \alpha_{AB}^{\ell} \\ \alpha_{BA}^{\ell} & \alpha_{BB}^{\ell} \end{bmatrix} \begin{bmatrix} x_A^{i,j} \\ x_B^{i,j} \end{bmatrix}, \tag{3}$$

11 The cross-stitch units were first introduced used for *multi-task* learning [127], and are utilized in our framework for *multimodal* fusion of *single-task* C3D networks



(a) A cross-stitching unit exchanges information by learning to linearly combine the input activation maps.



(b) C3D-Stitch links the information of two C3D streams via multiple cross-stitch units installed after the pooling and fully connected layer.

Figure 16: Overview of the proposed multi-layer fusion *C3D-Stitch* architecture.

where i, j are location coordinates in the feature maps, while the α learned weights show the amount of information flow of each filter between the streams. The parameters α_{AA}, α_{BB} weight the information flow in the same modality, while α_{AB}, α_{BA} control the impact of the external modality stream on the current one. In other words, the α-values denote the degree of contribution of each pair of streams. A close-to-zero α_{AB} or α_{BA} value indicates that the amount of information shared between the modalities is low, while, high positive or low negative α_{AB} or α_{BA} weights are linked to a high amount of information exchange between the networks.

The core structure for each C3D model remains almost unchanged, as only *extend* it with connections to another network via cross-stitch units after each convolution-pooling clock and in-between the fully-connected layers. As the *C3D-Stitch* consists of two individual networks which actively share the information along the layers, the direct forward pass outputs two predictions. We therefore average the resulting *Softmax* scores of both network and unify the prediction score. We follow the same learning procedure as for the late fusion (Section 3.3.1.2) and choose a cross-stitch layer learning rate of 0.01, similar to [127].

3.3.2 *Experiments*

We evaluate both our fusion policies and the single-stream baseline methods on the publicly available Isolated Gesture Dataset (IsoGD) [198, 199] for multimodal gesture recognition. This benchmark consists of both color- and depth videos of 249 hand signs, where each video corresponds to a single isolated gesture. IsoGD is a large-scale dataset that provides a high variety of different gesture types of multiple applications ranging from sign language to diving and more specialized ones like gestures used for communication by Italians.

In this work, we focus on the potential of multi-layer fusion and conduct a systematic evaluation of various methods at different stages in the network. To this intent, we do not aim at improving the performance of current approaches, but selected a popular neural network often used in this task without any extensions such as skeleton extraction or hand cropping, which are often employed to improve the recognition rate.

In order to systematically evaluate fusion at different levels, we conduct our experiments on ten gestures, which are most frequent in the IsoGD dataset for mainly two reasons. First, the IsoGD dataset is highly unbalanced and considers classes, which occur only a few times in the dataset. This unbalance might influence the outcome of our evaluation, as the task gradually becomes few-shot learning. Secondly, due to the high computational cost of training on the entire

Modality	Train. Proc.	Validation	Test
Baselines			
RGB	–	52.3	58.0
Depth	–	49.0	71.6
Late Fusion Methods			
RGB+Depth	separate	49.3	70.3
	combined	54.9	66.7
	sep.+comb.	**64.6**	**75.2**

Table 7: Different late fusion strategies (val accuracy): 1) train the models separately and combine the prediction only; 2) train the depth and RGB-model together by averaging the cross entropy loss of both networks; 3) first train the networks separately and, then, fine-tune them together.

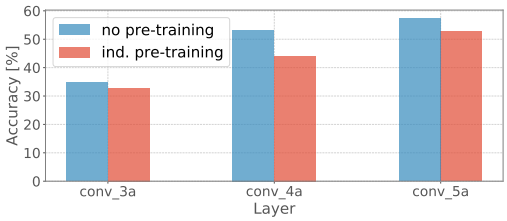


Figure 17: Validation accuracy of the intermediate single-layer fusion using $1 \times 1 \times 1$ convolutions and a shared late network. We compare different placements of the fusion. Furthermore, we differentiate between models that were first pre-trained individually and ones that were directly trained together.

dataset, we opt to include more experiments on a subset of the data instead of providing only a scarce analysis on the complete IsoGD. Thus, we evaluate our idea on the ten most frequent gestures from IsoGD, resulting in a dataset of 3711 gesture videos. We adopt the training, validation and test splits provided by the IsoGD benchmark and also use the recognition accuracy as our default metric for comparing our fusion methods [198].

3.3.2.1 *Late Fusion*

We first examine the commonly used late fusion approach depending on their training policy with and compare them with the single-stream models. The considered training schemes were previously described in Section 3.3.1.2: training two models separately, jointly, and a combination of both (first, they are trained separately and then, they are fine-tuned by averaging the losses). Table 7 summarizes the results, clearly demonstrating the benefit of multimodal fusion. Training both networks jointly after single-modality pre-training leads to the best accuracy of 75.2%, outperforming the depth-only model by over 3% and the RGB-only model by more than 17%.

3.3.2.2 *Early and Mid-Fusion via* $1 \times 1 \times 1$ *convolutions*

Next, we explore the effect of the proposed mid-level fusion via $1 \times 1 \times 1$ convolutions (as described in Section 3.3.1.3). We add the fusion layer at different depths of the networks and report results for fusion at layers *conv_3a*, *conv_4a* and *conv_5a* of the C3D model. The position of the fusion layer has a great impact on the overall performance on the test set, ranging from 53.4% at the earliest layer to 78.6% at *conv_5a* layer (Table 8). We see a trend for better classification results deeper in the network for both validation and test set. Still, information exchange via $1 \times 1 \times 1$ convolution at later stages surpasses the standard late fusion by over 3%.

Modality	Ind. pre-train.	Layer	Validation	Test
Baselines				
RGB	–	–	52.3	58.0
Depth	–	–	49.0	71.6
Late Fusion Methods				
RGB+Depth	✗	softmax	54.9	66.7
	✓		64.6	75.2
1 × 1 × 1 Convolutions				
RGB+Depth	✗	conv_3a	32.8	42.7
	✓		34.7	53.4
	✗	conv_4a	44.1	64.8
	✓		53.2	70.5
	✗	conv_5a	52.8	75.2
	✓		57.4	78.6
Cross-stitch Units				
RGB+Depth	✗	multi-layer	56.6	77.1
	✓		**66.0**	**79.8**

Table 8: Results of C3D using the different fusion methods. We group our fusion methods into three categories: 1) late fusion where we combine the prediction of the networks after the final fully connected layer by simply averaging the confidences for each class; 2) early- and mid-level fusion using 1 × 1 × 1 convolution layers to bridge the information between our two networks; 3) we apply cross-stitch units after each pooling and fully connected layer of the two C3D streams.

3.3.2.3 *Fusion via Cross-Stitching Units*

Finally, we evaluate the effectiveness of the proposed *C3D-Stitch* model, where the networks share the information on multiple layers simultaneously. In Table 8, we compare the *C3D-Stitch* network with the late- and single-layer mid-level fusion approaches and the baseline methods. Similarly to previously considered methods, *C3D-Stitch* benefits from combining both, individual modality-specific pre-training and final joint optimization. As expected, our model outperforms single-model baselines by a large margin (17% for validation, 8.2% for testing) and are also more effective than the conventional late fusion strategy (1.4% for validation, 4.6% for test). Overall, the proposed *C3D-Stitch* network yields the best recognition rate of 79.8% . This outcome shows that modern multimodal gesture recognition models would benefit from deeper research of fusion methods at the earlier network stages. It further shows that it is helpful to employ a method like cross-stitch units that allow the network to learn end-to-end where and how much the different streams should interact with each other.

3.3.2.4 *Learned Shared C3D-Stitch Representations*

Networks with cross stitch units share the information through a linear combination of activation maps, where the corresponding weights are learned during training in an end-to-end fashion. We now investigate the amount of information shared by the network as we take a look at the learned cross stitch unit's weights. The parameters α_C and α_D (Section 3.3.1.4) denote the weight each of the streams contribute to the output (C denotes color- and D depth network input). The weights are initialized in such a way that a small amount of information is shared between the

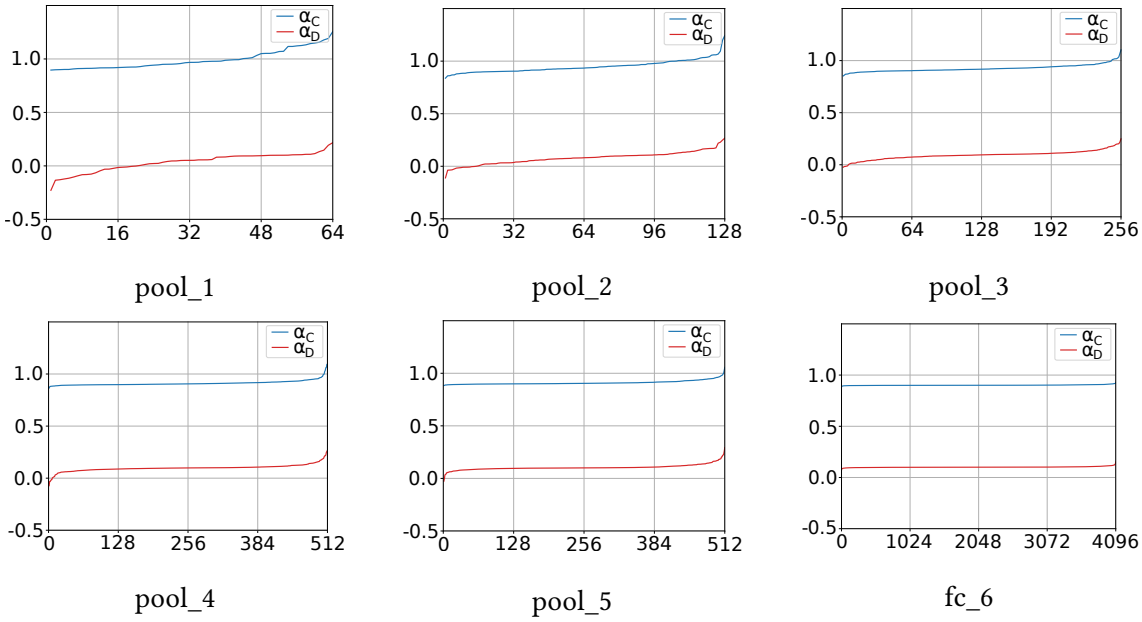


Figure 18: Learned weights for the *color network input*. The sorted cross stitch units weights for different layers, where α_C are the weights of the RGB-C3D model, while α_D the depth architecture weights. The higher the values for α_C the more the network chooses information from the *native* RGB model, while higher α_D show a stronger weight of the *foreign* depth stream.

two networks, as done in [127]. During training, the α values are *learned* to assure the optimal information sharing for the task.

We visualize the learned weights of the cross-stitch units for the input to the color stream in Figure 18. The figure illustrates the *sorted* weights of each individual layer, where the cross stitch units are applied. We see in Figure 18, that while overall, internal features (in this case, color data), have a stronger contribution to the input of the next layer, we observe a clear mixture of the two modalities. The weights of the foreign depth network contain values of up to 0.25, while some α values of the color network have a value of over 1.0. Individual features of the same modality are weighted differently, *i. e.* our model has learned to select and share the most useful information. This exchange pattern is present along all layers, except for the last convolution layer *fc_6*, where the representation is still mixed, but the features seem to be weighted uniformly (around 0.9 for color and 0.1 for the foreign depth stream). In conclusion, these results demonstrate, that both the RGB and the depth model benefit from the knowledge sharing at multiple stages.

Overall, our analysis of different fusion strategies for gesture recognition from color and depth videos, has given three main findings: 1) we confirm the assumption, that gestures recognition benefits from multimodality, as even simple multimodal approaches surpass single-modal ones; 2) we show, that involving mid-level features in the information exchange with an additional $1 \times 1 \times 1$ convolution layer further boosts the performance; 3) sharing the information at multiple layers simultaneously consistently outperforms single-layer fusion, which we demonstrate with the proposed *C3D-stitch* architecture.

3.4 CHAPTER CONCLUSION

The central goal of this chapter was to uncover the potential of end-to-end models in the area of driver activity recognition – a field still dominated by the classical feature-based pipeline. In order to train such data driven models, we collect and publicly release the first large-scale multimodal dataset for recognizing drivers' behavior at multiple levels of granularity, which we call *Drive&Act*. We started the dataset design by reviewing previous studies on common activities behind the steering wheel and their influence on the accident odds, after which we have questioned experts from automotive research and manufacturing. The results of these phases have guided us as we created a hierarchical vocabulary of driver actions, which captures human behavior on three levels of granularity and serves as our annotation scheme. After recording and labelling the dataset, we examine whether deep CNNs are suitable in our application and adopt multiple off-the-shelf video classification networks to our task, which we compare among each other and with the feature-based models. Our findings highlight that *Drive&Act* is a difficult dataset presumably due to the concise annotations (*e. g. opening bottle* versus *closing bottle*), while CNNs have been proven to be a powerful tool for our application, setting state-of-the art results in the majority of settings, including the main evaluation level of fine-grained activities.

We then moved to the task of anticipating future vehicle maneuvers by visually observing the driver. Although the *Brain4Cars* benchmark provides training data in this case, previously evaluated approaches were exclusively based on hand-crafted features. Since motion cues are especially relevant for maneuver prediction, we combine a neural network for optical flow extraction with a 3D ResNet and an LSTM model, to allow varying-time-to-maneuver inference. With an overall accuracy of 83.12% and an F_1 score of 81.74%, our model outperforms previous state-of-the-art approaches and is able to handle input sequences of variable temporal duration, on average, anticipating future maneuvers $4,07s$ in advance.

While this thesis focuses on recognition of naturally occurring human activities, as a side-contribution, we look at automatic gesture recognition, as such systems would provide a novel communication interface inside and are an intuitive alternative for the central console. Specifically, we focus on multimodal gesture recognition models and conduct the first systematic study of fusion methods at the convolution level. Among our contributions, we propose the C3D-Stitch model for multimodal gesture recognition, which "glues" the streams together at multiple network layers simultaneously by learning a linear combination of the activation maps.

Overall, our experiments show that deep CNNs are highly effective for driver monitoring, setting state-of-the-art results for maneuver prediction through driver observation on *Brain4Cars* and in the majority of *Drive&Act* tasks, including the recognition of *fine-grained* activities. However, we also want to highlight, that *Drive&Act* was collected in a driving *simulator*, which is a potential limitation since challenges due to changes in illumination and activity appearances will presumably arise when moving to a real-life application.

We outline the scientific impact of this chapter in four main contributions:

Contribution 1 : *Drive&Act* – the first large-scale publicly available dataset for fine-grained driver behavior analysis, densely labelled with a hierarchical annotation scheme.

Contribution 2: Implementation and study of multiple off-the shelf video classification CNNs in the context of driver activity recognition. The architectures are benchmarked against each other, the feature-based methods and compared for different modalities, views, and their combination.

Contribution 3: A new model for maneuver prediction through driver observation, combining a 3D ResNet with an LSTM and an optical flow computation network.

(Side-) Contribution 4: A systematic study of of different deep fusion strategies for multimodal *gesture* recognition, including our proposed C3D-Stitch architecture, where the information exchange happens on multiple network layers simultaneously through cross-stitch units.

Although this chapter makes the decisive first step towards integration of deep CNNs inside the vehicle cabin, it considers a highly controlled environment of supervised closed set classification, while being mainly guided by the recognition accuracy. A real-world model should be able to strike a good balance between delivering high recognition rates and being able to identify its own limits, *e. g.* , through realistic confidence estimates. In fact, in many safety-critical applications, false positives are more damaging than false negatives, and revealing cases of failure and uncertainty is oftentimes more important for the model than predicting the correct class. Motivated by this, our main goal for the forthcoming chapters are *uncertainty-aware models* which can reliably identifying their own classification mistakes or discover unknown activity classes.

4 RELIABILITY UNDER CLOSED-SET CONDITIONS

Automated activity understanding opens doors for new ways of human-machine interaction but requires models that can identify uncertain situations and trace back their root causes. In this chapter, we aim for activity recognition models capable of *identifying their failure cases*, *i. e.* the resulting probability estimates should indeed reflect the likelihood of the prediction being correct. Furthermore, we tackle the black-box nature of 3D CNNs for driver monitoring and examine methods for analyzing internal decision processes leading to such failure cases. The chapter is organized in three sections. In Section 4.1 we incorporate the *reliability of model confidence* in the activity recognition evaluation and develop methods which transform the oftentimes overly-confident outputs of the original networks into reliable probability estimates (results accepted for publication (oral) in *ICPR 2021* [169]). In Section 4.2 we implement a diagnostic framework, where we analyze, *e. g.* , the learned internal representations and high-activation video regions with the incentive to uncover the reasons of failing (results published in *ITSC 2020* [170]). Section 4.3 revisits the scientific contributions and gives an outlook on the next research steps.

4.1 RELIABILITY OF MODEL CONFIDENCE ESTIMATES

This section is based on our publication in *ICPR* 2021 [169], © IEEE .

Beyond assigning the correct class, an activity recognition model ought to be able to determine, how certain it is in its predictions. While the recognition accuracy of video classification networks has rapidly improved due to the rise of deep learning [6, 17, 121, 153, 186], examining how well the confidence values of such models indeed reflect the probability of a correct prediction has been overlooked in the past. Obtaining realistic uncertainty measures is vital for building trust and is a serious concern for the integration of activity recognition CNNs in real-life systems.

In this section, we present the first study of how well the confidence values of modern action recognition architectures indeed reflect the probability of the correct outcome and propose a learning-based approach for improving it. First, we extend two popular action recognition datasets with a *reliability* benchmark in form of the expected calibration error and reliability

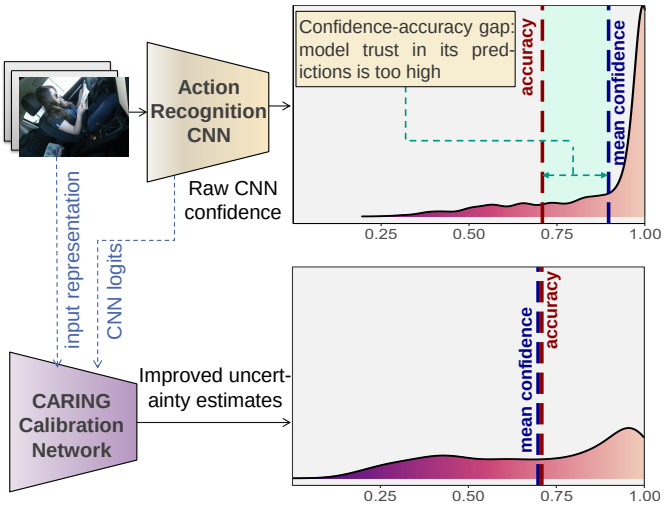


Figure 19: Softmax confidence distribution of a popular video classification network (P3D) before and after the improvement through our **C**alibrated **A**ction **R**ecognition with **In**put **G**uidance model. Native confidence values underestimate model uncertainty (the majority of samples was rated with > 90% confidence, while the accuracy is significantly lower). We propose to incorporate the *reliability* of model confidence in the activity recognition evaluation protocols and develop algorithms for improving it.

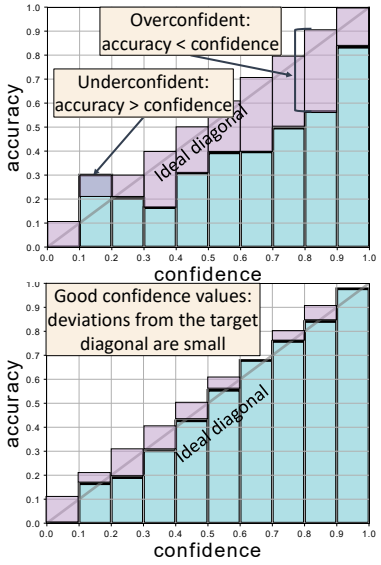


Figure 20: Reliability diagrams of a model with poor confidence estimates (top) and a well-calibrated model (bottom) . The illustrated data are the confidence values of P3D on the *Drive&Act* validation split before and after the improvement with our CARING calibration network.

diagrams. Since our evaluation highlights that confidence values of standard action recognition architectures do not represent the uncertainty well (see Figure 19), we introduce a new approach which learns to transform the model output into realistic confidence estimates through an additional calibration network. The main idea of our Calibrated Action Recognition with Input Guidance (CARING) model is to learn an optimal scaling parameter *depending on the video representation*. We compare our model with the native action recognition networks and the temperature scaling approach – a widespread calibration method utilized in image classification [55]. While temperature scaling alone drastically improves the reliability of the confidence values, our CARING method consistently leads to the best uncertainty estimates in all benchmark settings.

4.1.1 *Problem Definition: Reliable Confidence Measures*

In order to identify cases of failure, our model needs to produce proper confidence values which indeed reflect the likelihood of a correct prediction. We therefore introduce the *reliability of model confidence benchmark* to supervised multi-class activity recognition, where the models are usually validated via top-1 accuracy only [17, 89, 121, 153, 186]. Given an input video clip x with a ground-truth label a_{true} and the set of all possible target classes $a \in \mathcal{A}\{1, ..., m\}$, let f_θ be our activity recognition model predicting an activity label a_{pred} and the corresponding model confidence,*i. e.* , the probability estimate $\hat{p}(a_{pred})$: $f_\theta(\mathbf{x}) = [a_{pred}, \hat{p}(a_{pred})]$. A *reliable* model ought to not only learn to predict the correct activity (*i. e.* $a_{pred} = a_{true}$), but also give us well-calibrated

confidence estimates $\hat{p}(a_{pred})$, which indeed reflect the true probability of a successful outcome $\mathbb{P}(a_{pred} = a_{true})$. A perfectly calibrated *i. e.* a perfectly *reliable* model is often formalized as [55]:

$$\mathbb{P}(a_{pred} = a_{true} | \hat{p}(a_{pred}) = p) = p, \quad \forall p \in [0, 1] \tag{4}$$

In other words, the model's inadequacy to produce reliable probability estimates is directly linked to the gap between the average model's confidence and the achieved accuracy (see Figure 19). Reaching perfect model calibration is practically impossible and, in addition, we cannot even *perfectly* evaluate it, since in Eq. 4.1.1 the space of all possible probability values p is continuous, while we only have a finite amount of the measured estimates $\hat{p}(a_{pred})$. However, we can approximate this value by discretization of the probability space. To quantify the calibration quality of the models' confidence scores, we use the Expected Calibration Error (ECE) metric [55]. To compute ECE, we divide the space $[0, 1]$ of possible probabilities into K segments (in our case, $K = 10$). We then compute the model accuracy and average model confidence for samples belonging to each individual segment (see Figure 20). In a perfectly calibrated model, the difference between accuracy and average confidence of the individual segments would be zero. We therefore compute the distance between the mean confidence and the measured accuracy in each bin and then calculate the average over all such segments, weighted by the number of samples in each bin. Formally, the expected calibration error is defined as:

$$ECE = \sum_{i=1}^{K} \frac{N_{bin_i}}{N_{total}} |acc(bin_i) - \hat{p}(bin_i)|, \tag{5}$$

where N_{bin_i} is the number of samples with probability values inside the bounds of bin_i, $acc(bin_i)$ and $\hat{p}(bin_i)$ are the accuracy and average confidence estimates of such examples respectively and N_{total} is the total number of data points (in all bins).

The expected calibration error can be visualized intuitively using *reliability diagrams* (example provided in Figure 20). First, the space of possible probabilities (X-axis) is discretized into K equally sized bins, as previously described for the ECE calculation. Samples with predicted confidence between 0 and 0.1 fall into the first bin, between 0.1 and 0.2 into the second bin and so on. For each segment, we plot a bar with height corresponding to the accuracy in the current segment. In an ideal case, the accuracy should be equal to the average confidence score inside this bin, meaning, that the bars should have the height of the diagonal. As we see in Figure 20, these are often beyond the diagonal if the Pseudo 3D ResNet [153] model probabilities are used out-of-the-box. This means that the model tends to be overly confident, as the accuracy in the individual bins tends to be *lower* than the probability produced by the model.

4.1.2 *Backbone Neural Architectures*

We consider two widely used spatiotemporal CNNs for activity recognition: Inflated 3D ConvNet (I3D) [17] and Pseudo 3D ResNet (P3D) [153], which we have already utilized for driver activity recognition in the previous chapter. Both architectures directly operate on the video data and learn the intermediate embeddings together with the classifier layers in an end-to-end fashion (see Section 3.1.4 for more details). As in other CNNs, the neurons of the last fully-connected layer are referred to as a *logit vector* $\mathbf{y}$ with its activations y_a representing *not normalized* scores of activity a being the current class. A straight-forward way to obtain the model

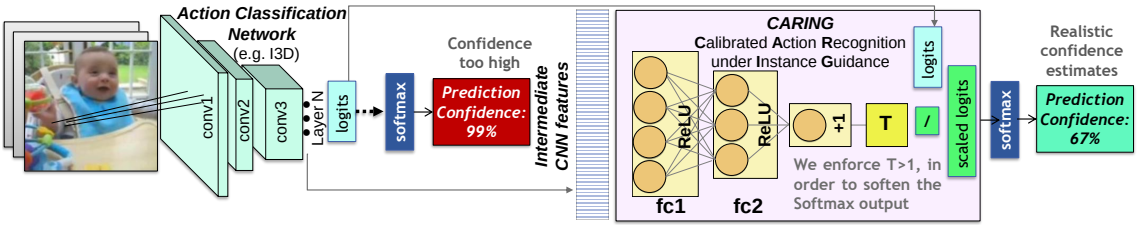


Figure 21: Overview of the Calibrated Action Recognition under Instance Guidance Model (*CARING*). *CARING* is an additional neural network which learns to infer the scaling factor $\mathcal{T}$ depending on the instance representation. The logits of the original activity recognition network are then divided by T, giving better estimates of the model uncertainty.

confidence estimates which mimic a probability function, is to normalize the scores using *Softmax*: $\hat{p}(a_{pred}) = \max_{a \in \mathcal{A}} \frac{exp(y_a)}{\sum_{\hat{a} \in \mathcal{A}} exp(y_{\hat{a}})}$. During training, the cross-entropy loss is computed using the *Softmax*-normalized output, optimizing the network for high top-1 accuracy. While both architectures have demonstrated impressive results in activity recognition [17, 114, 153] in terms of their accuracy, an evaluation of how well their Softmax-confidence values indeed reflect the model uncertainty remains an open question and is therefore the central topic of this work.

4.1.3 *Calibration via Temperature Scaling*

A popular way for obtaining better confidence estimates from CNN logits in image recognition is *temperature scaling* [55]. Temperature scaling simplifies Platt scaling [151], and is based on learning a single parameter τ which is further used to "soften" the model logits. The logits are therefore divided by τ before applying the *Softmax* function: $\mathbf{y}_{scaled} = \mathbf{y}/\tau$. With $\tau > 1$ the resulting probabilities become smoother, moving towards $\frac{1}{m}$, where m is the number of classes. Contrary, scaled probability would approach 1 as τ becomes closer to 0. After the neural network is trained for supervised classification in a normal way, we fix the model weights and optimize τ on a held-out validation set using Negative-Log-Likelihood. Despite the method simplicity, temperature scaling has been highly effective for obtaining well-calibrated image recognition CNNs, surpassing heavier methods such as Histogram binning and Isotonic Regression [55].

As temperature scaling has not been explored for spatiotemporal CNNs used for video classification yet, our first intuition is to combine it with the existing activity recognition models. We therefore augment the I3D and P3D models with a post-processing temperature scaling module. We optimize τ using SGD with a learning rate of 0.01 for 50 epochs.

We want to notice that as the networks are fully trained and their weights remain fixed while learning the scaling parameter τ, transformation of the logits does not influence their order and therefore the *model accuracy stays the same*. In other words, while temperature scaling gives us better uncertainty estimates, the predicted activity class does not change as all logits are divided by the same scalar.

4.1.4 *Calibrated Action Recognition with Input Guidance (CARING)*

In this section, we introduce a new model for obtaining proper probability estimates by learning the logit scaling *depending on the input*. While our evaluation described in the next section reveals

that the temperature scaling method clearly improves model confidence calibration, it *does not consider the representation of the current sample* when deciding how to scale the output, meaning that the logits are always divided by the same global scalar τ.

We believe that the input itself carries a useful signal for inferring model confidence and build on the temperature scaling approach [55] with one crucial difference: the scaling factor is not global but different for varying input. Our main idea is therefore to learn acquiring the scaling parameter $\mathcal{T}(\mathbf{z})$ on-the-fly at test-time depending on the input representation $\mathbf{z}$, so that the scaled logits become $\mathbf{y}_{scaled} = \mathbf{y}/\mathcal{T}(\mathbf{z})$.

To learn the input-dependent temperature value $\mathcal{T}(\mathbf{z})$, we introduce an additional *calibration neural network*, which we refer to as the *CARING* model (**C**alibrated **A**ction **R**ecognition under **In**put **G**uidance), as it guides the scaling of the logits depending on the current instance. An overview of our model is provided in Figure 21. The *CARING* network comprises two fully-connected layers, with the output of the second layer being a single neuron used to infer the input-dependent temperature scalar. Note, that we extend the last *relu* activation with an addition of 1 to enforce $\mathcal{T}(\mathbf{z}) \geq 1$, required to soften the probability scores. The input-dependent temperature $\mathcal{T}(\mathbf{z})$ is therefore obtained as:

$$\mathcal{T}(\mathbf{z}) = 1 + relu(\mathbf{W}_2\ relu(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2), \tag{6}$$

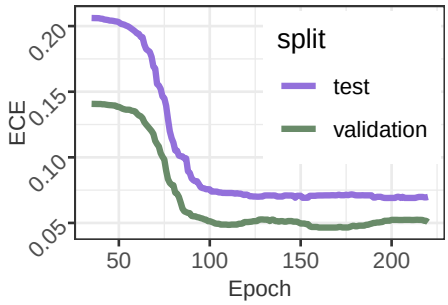
where $\mathbf{W}_1$,$\mathbf{W}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the network weight matrices and bias vectors and $\mathbf{z}$ is the input representation, for which we use the intermediate features of the original activity recognition network ($\mathbf{z}$ has a size of 1024 for Infalted 3D ConvNet and 2048 for Pseudo 3D ResNet).

We then scale the logits by the inferred instance-dependent temperature $\mathcal{T}(\mathbf{z})$ and our prediction probability estimate becomes:

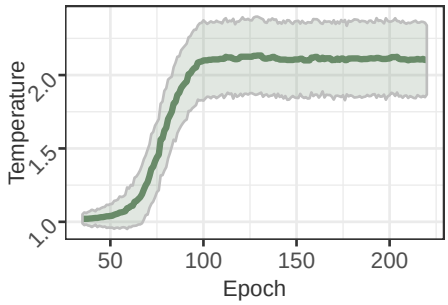
$$\hat{p}(a_{pred}) = \max_{a \in \mathcal{A}} \frac{exp(\frac{y_a}{\mathcal{T}(\mathbf{z})})}{\sum_{\tilde{a} \in \mathcal{A}} exp(\frac{y_{\tilde{a}}}{\mathcal{T}(\mathbf{z})})}. \tag{7}$$

We train the *CARING* model on a held-out validation set with Negative Log Likelihood (NLL) loss for 300 epochs (learning rate of 0.005, weight decay of $1e^{-6}$). Similarly to the approach described in Section 4.1.3, *CARING* can be viewed as a post-processing step for obtaining better uncertainty confidence and *does not affect the predicted activity class* and the model accuracy, as the order of the output neurons does not change.

Why does NLL-based calibration improve the confidence estimates? Since both the main classifier and the calibration models are trained using the NLL loss, we need to clarify why such optimization gives unrealistic *Softmax* scores after the standard classifier training but leads to reliable confidence estimates after the calibration phase. Indeed, the NLL itself indirectly reflects model miscalibration [55] and we therefore employ it as an additional metric. The main reason for the confidence-accuracy disarray rising during the classifier training is *overfitting on the training set combined with the target labels being exclusively 0 or 1* [55, 92]. After the classifier training has converged in terms of *accuracy* (which is usually very high on the training data) it further optimizes the NLL-criteria to match the 0/1 labels, which will usually be 1 for the predicted class on the training data, leading to overly confident models. Calibrating on a held-out validation set while restricting the model to further improve the accuracy (as the classifier weights are frozen) enforces us to learn *Softmax*-scaling while having a realistic distribution of correct and incorrect predictions. Note, that while such training-data-overfitting to the 0/1 targets is the main known reason for high confidence values, experimental results [55] highlight other factors, such as batch normalization, although the exact reason why is not well understood.



(a) Expected Calibration Error improvement during the training procedure for validation and test data.



(b) Average temperature and its standard deviation estimated by our model during training.

Figure 22: *CARING* model evolution during training for one *Drive&Act* split. Both average value and standard deviation of the learned input-dependent scaling parameter $\mathcal{T}(z)$ rise as the training proceeds (right figure). Jointly with the decrease of the calibration error (left figure), this indicates the usefulness of learning different scaling parameters for different inputs.

Does our CARING model indeed scale different inputs differently? To validate that our model indeed leverages the input as the signal for confidence scaling, we examine the evolution of different model metrics during training. Figure 22 illustrates changes of the expected calibration error (defined in Section 4.1.1) as well as the mean and standard deviation of the inferred scaling parameter $\mathcal{T}(z)$ measured over all validation examples in the corresponding epoch. Figure 22b reveals that both, the mean and standard deviation of the learned temperature, rise during training, leading to a lower calibration error (Figure 22a). The observed increase in standard deviation of the learned scaling factor indicates that handling the logits differently dependent on the input is beneficial in our task, which we will confirm empirically in the following experiment section.

4.1.5 *Experiments*

4.1.5.1 *Benchmark settings*

We propose to incorporate the reliability of model confidence in the activity recognition evaluation and adapt our *Drive&Act* dataset and the *HMDB-51* [89] benchmark for standard action recognition to our task. We select the *Drive&Act* testbed for driver activity recognition as our main benchmark, as it is application-driven and encompasses multiple challenges typical for real-life systems (*e. g.* fine-grained categories and unbalanced data distribution). *Drive&Act* comprises 34 fine-grained activity classes (see Section 3.1.2.2), which, however are highly unbalanced as the number of examples ranged from only 19 examples of *taking laptop from backpack* to 2797 instances of *sitting still*. As CNNs have a lower performance when learning from few examples, we sort the behaviors by their frequency in the dataset and divide them into two groups: *common* (top half of the classes) and *rare* (the bottom half). We subsequently evaluate the models in three modes: considering *all activities*, as it is usually done, using only the *overrepresented*- or only the *rare* classes.

We further validate the models on *HMDB-51* [89], a standard activity recognition dataset comprising of more "everyday" behaviours. The benchmark covers 51 activity classes, which are

Model	ECE		NLL	
	validation	test	validation	test
Drive&Act - Common Classes				
P3D [153]	16.9	19.39	1.63	1.85
I3D [17]	10.22	13.38	0.90	1.27
P3D + Temperature Scaling [55]	5.65	5.7	1.28	1.48
I3D + Temperature Scaling [55]	5.31	6.99	0.57	0.83
CARING - P3D (ours)	4.81	**4.27**	1.19	1.42
CARING - I3D (ours)	**2.57**	5.26	**0.50**	**0.78**
Drive&Act - Rare Classes				
P3D [153]	31.49	37.25	3.43	4.68
I3D [17]	31.48	43.32	3.41	4.54
P3D + Temperature Scaling [55]	17.83	21.09	2.26	2.99
I3D + Temperature Scaling [55]	24.97	32.38	1.96	2.62
CARING - P3D (ours)	**13.73**	**19.92**	2.12	2.93
CARING - I3D (ours)	18.34	23.6	**1.55**	**2.17**
Drive&Act - All Classes				
P3D [153]	17.89	21.09	1.77	2.12
I3D [17]	11.72	15.97	1.10	1.56
P3D + Temperature Scaling [55]	5.89	6.41	1.35	1.63
I3D + Temperature Scaling [55]	6.59	8.55	0.68	0.99
CARING - P3D (ours)	4.58	**5.26**	1.26	1.57
CARING - I3D (ours)	**3.03**	6.02	**0.58**	**0.9**
HMDB-51				
I3D [17]	10.29	20.11	0.98	1.97
I3D + Temperature Scaling [55]	4.00	7.75	**0.81**	1.57
CARING - I3D (ours)	**3.38**	**5.98**	**0.81**	**1.54**

Table 9: Reliability of confidence values on *Drive&Act* and *HMDB-51* datasets for original activity recognition models and their extensions with uncertainty-aware calibration algorithms.

more discriminative in their nature (*e. g.* laughing and playing football) and are perfectly balanced (three splits with 70 training and 30 test examples for every category).

Following the problem definition described in Section 4.1.1, we extend the standard accuracy-driven evaluation protocols [89, 114] with the expected calibration error (ECE), depicting the deviation of model confidence score from the true misclassification probability. In addition to ECE, we report the Negative Log Likelihood (NLL), as in previous works where higher NLL values have been linked to model miscalibration [55]. Since HMDB-51 does not contain a validation split, we randomly separate 10% of the training data for this purposes. As done in the original protocol definitions [89, 114], we report the average results over the three splits for both testbeds.

4.1.5.2 *Confidence Estimates for Action Recognition*

In Table 9 we compare CNN-based activity recognition approaches and their uncertainty-aware versions in terms of the expected calibration error and NLL for *rare*, *overrepresented* and *all Drive&Act* classes as well as in the *HMDB-51* setup. First, we verify our suspicion that native activity recognition architectures provide unreliable confidence estimates: confidence scores produced by I3D score have a misalignment of 15.97% for *Drive&Act* and 20.11% for HMDB-51. Sim-

Activity	Number of Samples	Recall	I3D			CARING-I3D		
			Mean Conf.	ΔAcc	ECE	Mean Conf.	ΔAcc	ECE
Five most common activities								
sitting_still	2797	95.1	97.96	2.86	2.86	93.84	-1.26	**1.84**
eating	877	86.42	93.26	6.84	9.33	80.99	-5.43	**5.75**
fetching_an_object	756	76.03	93.77	17.74	18.28	79.42	3.4	**5.32**
placing_an_object	688	66.77	93.03	26.25	26.25	75.9	9.13	**9.25**
reading_magazine	661	92.93	98.58	5.65	6.09	93.35	0.42	**2.87**
Five most underrepresented activities								
closing_door_inside	30	92.31	98.51	6.21	**8.22**	86.00	-6.31	8.30
closing_door_outside	22	81.82	93.55	11.73	20.97	86.86	5.04	**19.81**
opening_backpack	27	0	98.82	98.82	98.82	82.69	82.69	**82.69**
putting_laptop_into_backpack	26	16.67	92.67	76.00	76.00	76.46	59.8	**59.80**
taking_laptop_from_backpack	19	0.00	85.25	85.25	85.25	70.08	70.08	**70.08**

Table 10: Analysis of the resulting confidence estimates of the initial I3D model and its CARING version for individual common and rare *Drive&Act* activities. *Recall* denotes the recognition accuracy of the current class, while *Mean Conf.* denotes the average confidence estimate produced by the model. Supplemental to the Expected Calibration Error (*ECE*), we report the difference between the mean confidence value and model accuracy (denoted Δ*Acc*). While in a perfectly calibrated model ΔAcc is 0, ECE is a better evaluation metric, as *e. g.* if a lot of samples have too high and too low confidence values, their average might lead to a misconception of good calibration. While there is room for improvement for underrepresented and poorly recognized activity classes, the CARING model consistently leads to better uncertainty estimates.

ilar issues are present in P3D: 21.2% ECE on *Drive&Act*, an error far too high for safety-critical applications.

Model reliability is clearly improved by learning to obtain proper probability estimates, as all uncertainty-aware variants surpass the raw *Softmax* values. Interestingly, although I3D has better initial uncertainty estimates than P3D (ECE of 21.09% for P3D, 15.97% for I3D), P3D seems to have a stronger response to both, temperature scaling and *CARING* approaches than I3D (ECE of 5.26% for *CARING-P3D*, 6.02% for *CARING-I3D*). However, as this difference is very small (< 1%), we would rather recommend using I3D, as it mostly gives higher accuracy [17, 114, 153]. While we consider the expected calibration error to be of vital importance for applications, we realize that this metric is complementary to model accuracy and encourage taking both measures into account when selecting the right model. We want to remind that both temperature scaling and the *CARING* method *do not influence the model accuracy* (see Sections 4.1.3 and 4.1.4). For Pseudo 3D ResNet we achieve an overall accuracy of 54.86% (validation) and 46.62% (test) on *Drive&Act*, which does not change through our uncertainty-based modifications. Consistent with the previous chapter we observe a clearly higher recognition rate of I3D and its variants with 68.71% (validation) and 63.09% (test) accuracy[1].

As expected, the reliability of model confidence estimates correlates with the amount of training data (see distinguished areas for *common*, *underrepresented* and *all* classes of *Drive&Act* in Table 9). For example, the *common classes* setting encounters the lowest expected calibration error for both original and uncertainty-aware architectures (13.38% for I3D, 5.26% for *CARING-I3D*). Leveraging intermediate input representation via our *CARING* calibration network leads

1 The slight deviation from the accuracy reported in Section 3.1.5 (between 0.18% and 1.3%) is due to random factors in the training process.

to the best probability estimates on both datasets and in all evaluation settings. Thereby, the *CARING* strategy surpasses the raw neural network confidence by 9.95% and the temperature scaling method by 2.53% on *Drive&Act*, highlighting the usefulness of learning to obtain probability scores *depending on the input*.

We further examine model performance for the individual classes, considering the five most frequent and the five most uncommon *Drive&Act* activities separately in Table 10. In addition to ECE, we report the accuracy for samples belonging to the individual class, the average confidence value they obtained with the corresponding model and the difference between them (denoted ΔAcc). Although such global confidence-accuracy disagreement is interesting to consider (and is 0 for a perfectly calibrated model) it should be viewed with caution, as it might lead to an incorrect illusion of good confidence calibration, as *e. g.* a lot of samples with too high and too low confidence values might cancel each other out through averaging.

Reliability of the confidence scores is significantly improved through the *CARING* method and is connected to the amount of training data and the accuracy. Models have significant issues with learning from few examples (*e. g.* 76% I3D and 59.80% *CARING-I3D* ECE for *putting laptop into backpack*). For both, over- and underrepresented classes, the ECE of easy-to-recognize activities (*i. e.* the ones with high accuracy) is lower. Before calibration, the average confidence value is always higher than the accuracy (positive ΔAcc) disclosing that the models are too optimistic in their predictions. Interestingly, after the *CARING* transformation is applied, the average model confidence is lower than the accuracy for some classes, such as *eating*. *CARING* models therefore tend to be more conservative in their assessment of certainty.

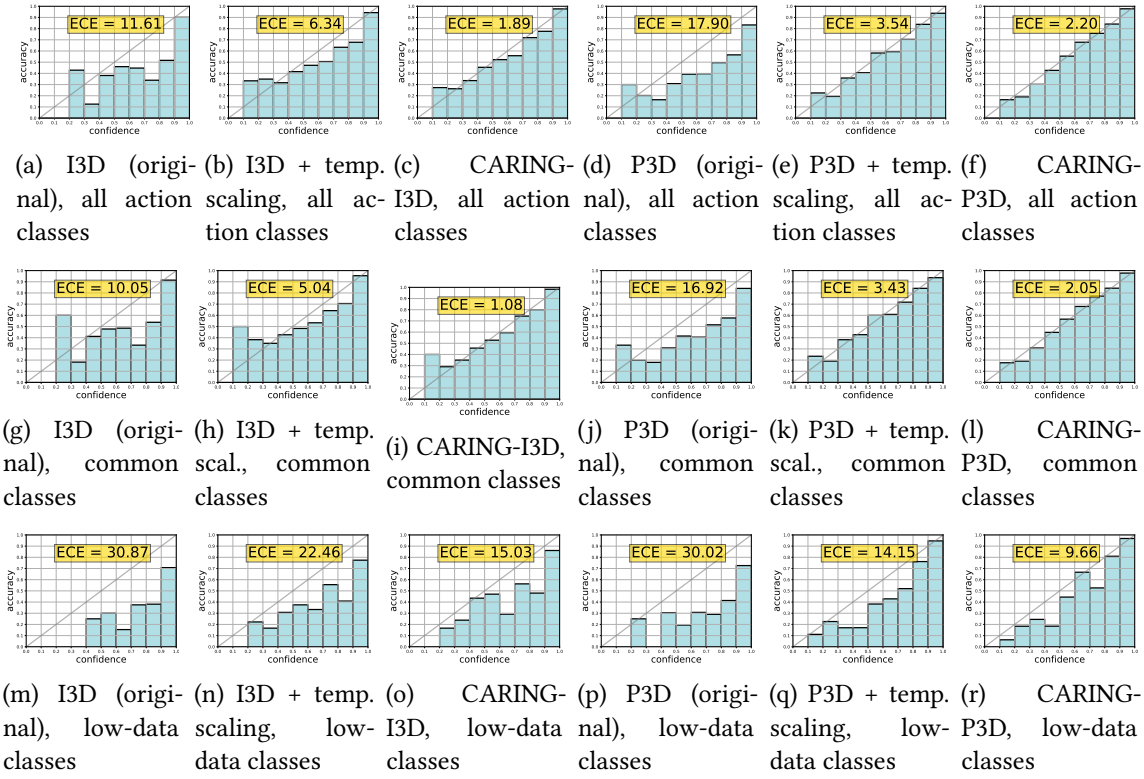
4.1.5.3 *Calibration Diagrams*

In Figure 23 we visualize the agreement between the predicted model confidence and the empirically measured probability of the correct outcome via reliability diagrams (explained in Section 4.1.1) . In case of good estimates, the result will be close to the diagonal line. Values above the diagonal are linked to models being overly confident in their prediction, while values below indicate that the model doubts the outcome too much and the accurate prediction probability is higher than assumed.

First, we discuss the reliability diagrams of the original action recognition networks. Both P3D and I3D confidence values deviate from the target, with a clear bias towards too optimistic scores (*i. e.* values are oftentimes below the diagonal in Figures 23a, 23d, 23g, 23j, 23m, 23p). One exception is an above-diagonal peak in the low probability segment for *all* and *common* classes, meaning that in "easier" settings, low confidence examples often turn out to be correct (23a, 23d, 23g, 23j). In the "harder" setting of *rare* activities (Figure 23m, 23p), the bias towards too high probabilities is present for all values.

We see a clear positive impact of temperature scaling (Figures 23b, 23e, 23h, 23k, 23n, 23q) and our *CARING* model (Figures 23c, 23f, 23i, 23l, 23o, 23r). *CARING* models outperform other approaches in all settings and lead to almost perfect reliability diagrams for *all* and *common* classes. Still, both temperature scaling and *CARING* methods have issues with rare classes, with model confidence still being too high, marking an important direction for future research.

Note, that ECE might be in a slight disarray with the visual reliability diagram representation, as the metric weighs the misalignment in each bin by the amount of data-points in it, while the reliability diagrams do not reflect such frequency distribution. For example, while the *CARING-I3D* model in Figure 23i slightly exceeds the target diagonal, it has lower expected calibration



(a) I3D (original), all action classes (b) I3D + temp. scaling, all action classes (c) CARING-I3D, all action classes (d) P3D (original), all action classes (e) P3D + temp. scaling, all action classes (f) CARING-P3D, all action classes

(g) I3D (original), common classes (h) I3D + temp. scal., common classes (i) CARING-I3D, common classes (j) P3D (original), common classes (k) P3D + temp. scal., common classes (l) CARING-P3D, common classes

(m) I3D (original), low-data classes (n) I3D + temp. scaling, low-data classes (o) CARING-I3D, low-data classes (p) P3D (original), low-data classes (q) P3D + temp. scaling, low-data classes (r) CARING-P3D, low-data classes

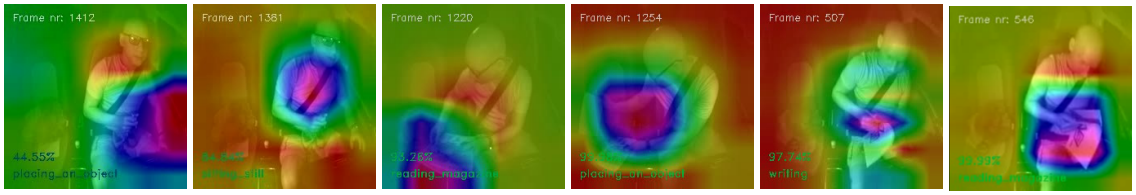
Figure 23: Reliability diagrams of different models reflect the agreement between the confidence values and the empirically measured probability of correct prediction (results of one *Drive&Act* validation split). A model with perfectly reliable probability estimates would match the diagonal (see Section 4.1.5.3). The ECE values slightly deviate from Table 9, as they visualize a single split, while the final results are averaged over all splits. While temperature scaling clearly improves the outcome, our CARING model leads to the lowest calibration error in all settings.

error than *CARING-P3D* which seems to produce nearly perfect results in Figure 23l. As there are only very few examples in the low-confidence bin, they are overshadowed by smaller differences in the high-confidence bins, which contribute much more as they have more samples.

4.2 A DIAGNOSTIC FRAMEWORK FOR IDENTIFYING CAUSES OF FAILURE

This section is based on our publication in *ITSC* 2020 [170], © IEEE .

Methods developed in the previous section allow us to identify incorrect predictions through realistic confidence estimates. In this section, we aim to trace back the *reasons* leading to network failures, as this is the first key step for preventing them. Despite the well-deserved reputation as visual recognition front-runners, the lack of transparency and the inability to efficiently visualize internal decision processes resulted in CNNs being labelled as black boxes, considerably slowing down their integration in industrial systems. In contrast to conventional feature-based methods [73, 113, 142], intermediate representations of such end-to-end architectures are not defined by hand but *learned* together with the classifier (see Figure 2), considerably hindering the interpretation of the decision pathways. Understanding the limitations of such networks is



(a) Prediction: **placing object (incorrect ✗)** (b) Prediction: **sitting still (correct ✓)** (c) Prediction: **read. magazine (incor. ✗)** (d) Prediction: **placing object (correct ✓)** (e) Prediction: **writing (incorrect ✗)** (f) Predict.: **reading magazine (corr. ✓)**

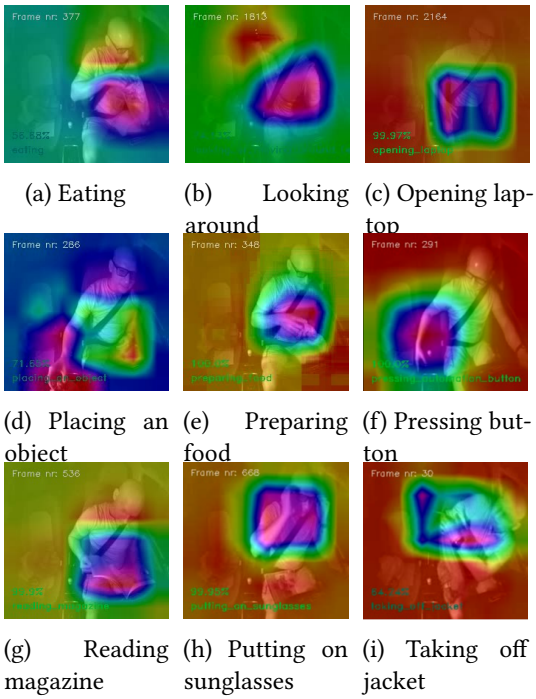
Figure 24: Correct vs. Misclassified Predictions: Analysis of video segments using our spatiotemporal version of gradient weighted class activation maps technique [184], where samples were close to each other and comprised the same behavior, but resulted in different predictions.

vital for applications and studying how such architectures function internally becomes increasingly important for overcoming data biases [23, 61], identifying most relevant data [184] and explaining failure cases [38].

In this section, we make a first step towards transparency behind spatiotemporal CNNs for driver monitoring, and implement a diagnostic framework for understanding the factors leading to network's mistakes. We gain insight into (1) *where did the network look*, *i. e.* which video regions have guided the current decision in cases of both, success and failure (2) *what did the network learn*, *i. e.* , exploring the intermediate layer representations with unsupervised methods and detecting relationships between different behaviors, and (3) a detailed performance analysis focused on common misclassifications of the individual classes and the relation to data scarcity.

First, we aim for *visual explanations* of the internal decisions and analyze where the network attended when it predicted the specific behavior. To this end, we set our target as the predicted class and backpropagate the gradient to the last convolution layer, building on the method of [184] and extending it to the temporal dimension. We then weigh the individual feature activation maps at that particular layer based on the gradient. We examine the resulting heatmaps which indicate the image regions directing the specific decision and compare the focus of the network in cases of success and incorrect predictions. We then consider the *representation point of view* and examine what the network has learned internally for three different models previously used for driver monitoring. To interpret hundreds of neurons of the last network layer, we reduce the dimensionality using t-SNE [107] and examine the resulting clusters, which are far more discriminative for the Inflated 3D Net. We further identify relationships between the learned representations of individual classes by using Ward's hierarchical agglomerative clustering.

Finally, we conduct a comprehensive study of the model performance, going beyond the top-1 accuracy by analyzing the top-5 generalization and the most common confusion of the individual classes. We distinguish between classes that are rare and ones that occur frequent during training. Our findings indicate that the main failure cases can be traced back to either semantic similarity combined with underrepresentation in the training set (*e. g. closing* versus *opening bottle*) or a learned movement-, object- or position bias (*e. g.* misclassification as *reading magazine* if a magazine is somewhere in the scene), highlighting the need of more diverse object placement in the datasets. We aim to make the first step to detach deep CNNs for driver observation from



(a) Eating (b) Looking around (c) Opening laptop

(d) Placing an object (e) Preparing food (f) Pressing button

(g) Reading magazine (h) Putting on sunglasses (i) Taking off jacket

Figure 25: Activation maps of the last Inflated 3D ConvNet convolution layer weighted by the gradient. Heatmaps overlayed over the original frame illustrate, which region has contributed to the network's decision.

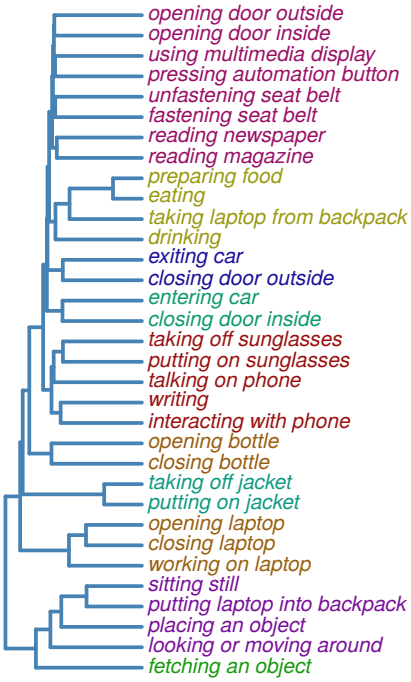


Figure 26: Results of Ward's Hierarchical Agglomerative Clustering reveal learned relationships between the individual classes. We cluster the mean vector of the intermediate Inflated 3D Net embedding for each activity.

their black box reputation and provide experimental evidence, that such models have the power to become highly interpretable through certain diagnostic tools.

4.2.1 *Evaluated CNNs and Testbed*

We integrate our diagnostic framework into the *Drive&Act* testbed introduced in Section 3.1, focusing on the 34 fine-grained activities and the frontal near-infrared camera view as our evaluation setup. We consider the three spatiotemporal CNNs we have previously adopted for driver activity recognition (Section 3.1.4): C3D [196], Inflated 3D ConvNet [17] and Pseudo3D ResNet [153]. Although our previous experiments have shown excellent driver monitoring performance of such spatiotemporal CNNs in terms of accuracy, understanding, what has driven the neural network decision in case of misclassifications remains an open question. Thereby, we analyze where the network has attended when different decisions were made (Section 4.2.2) and what it learned at the embedding level (Section 4.2.3). Lastly, we extend our previous performance analysis with additional metrics and settings, analyzing individual class confusions and their dependence on the amount of training data (Section 4.2.4). We analyze all three models in Section 4.2.3 and Section 4.2.4, and we choose the Inflated 3D ConvNet for the visual explanations in Section 4.2.2, as it has shown the best recognition results in previous work.

4.2.2 *Where did the network look?*

We build on the method of [184] and introduce a three-dimensional version of the gradient-weighted class activation map technique, providing visual explanations of *spatiotemporal* CNNs for the first time. Given an input video, we first conduct a conventional forward pass and obtain the predicted class c *i. e.* the class with the highest activation. Then, we estimate the gradient over y_c (the output before the *Softmax* layer) with respect to each individual value in the kth feature map A_k of a layer in the CNN. This is used to obtain the *feature importance* w_c^k for each individual feature map k by averaging the gradients over all its n values:

$$w_c^k = \frac{1}{n} \sum_{i,j,t} \left(\frac{\partial y_c}{\partial A_k^{i,j,t}} \right), \tag{8}$$

where $A_k^{i,j,t}$ is the activation at position in space i, j and time t. In each location (i, j, t) we linearly combine the values in the feature map by the importance estimate w_c^k. The final weights $V_c^{i,j,t}$ are obtained by passing the computed values to a *relu* function to remove negative values, as we are only interested in pixels that increase y_c. More formally, we calculate the final weights as follows:

$$V_c^{i,j,t} = relu \left(\sum_k w_c^k A_k^{i,j,t} \right). \tag{9}$$

To be able to visualize the resulting explanations as images, we average the resulting heatmaps over the time dimension. We provide the resulting visual explanations of the Inflated 3D ConvNet decisions for different classes in Figure 25, while Figure 24 illustrates key differences between correct and failed predictions. For example, the network features characteristic for *eating* are focused around both, hands and head (probably due to chewing), while *preparing food* is linked to the hands only. The network attention is different depending on the activity, but in general, we observe increased focus on human hands and head. There is also a visible object bias, which is useful in many cases (*e. g.* a laptop or a newspaper in the scene increases the chances of an activity involving these objects). However, such object bias might lead to mistakes, if *e. g.* the human is only *placing* a magazine but *reading magazine* is predicted (Figure 24c). Figure 24e reveals that a specific hand movement leads to the network predicting *writing*, while the person is actually *reading*. While in most cases the network seems to make the predictions for the right reasons, specifically looking at uncertain cases helps us to draw useful conclusions for improvement. For example, our analysis highlights the need for diversification of training data in terms of object placement, so that the network predicts object-related activities if the human interacts with them, and not if they are simply present in the scene.

4.2.3 *What did the network learn?*

We now gain insight into the intermediate features of the CNNs, to verify whether they provide good generic representations of driver behavior. We use the first validation split of Drive&Act and extract the features of the fully connected layer of the C3D, Pseudo 3D ResNet and Inflated 3D ConvNet. To make sense of hundreds of neurons, we first reduce the dimensionality using t-SNE [107]. We then visualize each video clip in two-dimensional space in Figure 27, marking behavior classes with different colors. We qualitatively observe that Inflated 3D Net captures the nature of activities better, as its features form far more discriminative clusters. Still, samples of

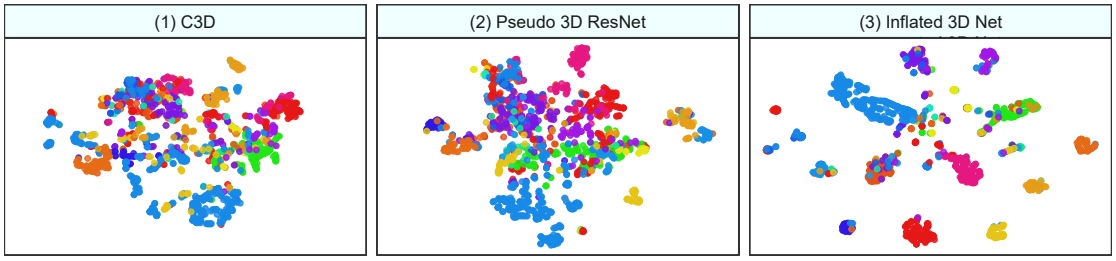


Figure 27: Visualizations using t-SNE of the intermediate representations learned by different CNN models. Different behavior classes are marked with different colors. While all models have clear correlations of the embedding values and the activity, such "class-specific cluster" are much more discriminative for the Inflated 3D Net.

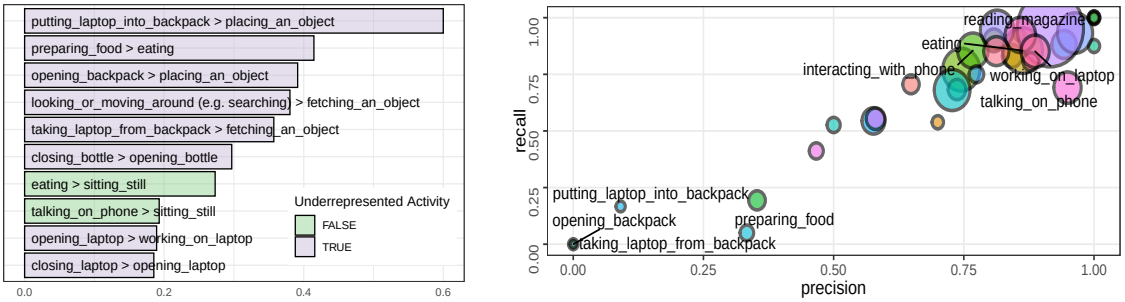
the same activities also shape visible groups for C3D and Pseudo 3D ResNet, but the boundaries are far less concise.

We now examine how different behaviors are connected from the CNN point of view. First, we compute the class centroid vector by averaging the fully connected I3D features of each activity. We apply the Ward's Hierarchical Agglomerative Clustering method [207] on the class centroids. The resulting class hierarchy, illustrated in Figure 26, reveals how the classes are connected in the model internally. While most of the semantically related activities are also placed together in the cluster hierarchy (*e. g. opening* and *closing bottle*), such similar cases often lead to high confusion, as we will show quantitatively in the next section. We can also understand how the network operates by looking at these relations, *e. g.* the activities *writing*, *talking on phone* and *putting on sunglasses* all fall into the same red cluster (Figure 26), while they do not match semantically at first glance. As the network groups these behaviors, we infer that it has learned them as fine-grained hand-centric actions and makes its decisions based on the concise hand movements. This is confirmed by the visual explanation in Figure 24e, where the model inaccurately predicts *writing* by focusing on a very small area around the hand instead of the object. The model view of some activities is surprising, for example, *taking laptop from backpack* is connected to *eating*, *preparing food* and *drinking*. The quantitative analysis in our next section will uncover that this action is indeed very poorly recognized. The way the network interprets this behavior is therefore simply incorrect. We assume that the model has learned a certain place bias, as a lot of coarse movements in front of the torso is typical for these actions. Extending the dataset with more diverse examples of this action (*e. g.* taking out the laptop in other locations) might therefore be beneficial.

4.2.4 *A Detailed Misclassification Analysis*

To examine the strengths and weaknesses of CNN-based algorithms, we extend our initial *Drive&Act* evaluation procedure with multiple settings and metrics. *Drive&Act* comprises 34 fine-grained activity classes, which, however are highly unbalanced. As CNNs have well-known issues when learning from few examples, we sort the behaviors by their frequency in the dataset and divide them into *common* (top half of the classes) and rare (the bottom half)[2] and benchmark

2 Note that we have already used such grouping in the reliability of model confidence evaluation (Section 4.1.5.1)



(a) Most common confusions of the I3D model. Purple color indicates that the class was underrepresented during training.

(b) Precision and recall of the individual classes. Circle size corresponds to the number of samples in the dataset (for readability only a random subset of activities is labelled).

Figure 28: Misclassification statistics of the Inflated 3D ConvNet on the Drive&Act dataset

Model	**Common**		**Rare**		**All classes**	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Validation						
C3D	54.44	87.53	45.70	75.82	50.07	81.67
Pseudo 3D ResNet	58.00	86.61	52.08	74.77	55.04	80.69
Inflated 3D Net	**80.62**	**95.83**	**58.50**	**87.88**	**69.67**	**91.85**
Test						
C3D	47.97	83.75	38.86	74.02	43.41	78.89
Pseudo 3D ResNet	52.43	84.05	38.20	65.09	45.32	74.57
Inflated 3D Net	**77.88**	**95.06**	**49.41**	**81.93**	**63.64**	**88.49**

Table 11: Top-1 and top-5 accuracy for fine-grained activity recognition on the *Drive&Act* dataset, evaluated separately for classes over- and underrepresented during training.

these groups separately. In addition to the conventional top-1 accuracy, we evaluate the *top-5 accuracy*, *i. e.* , we consider the sample as correctly classified if any of the five classes with the highest probabilities match the ground truth. The top-5 accuracy might be useful if we want to overlook confusions of highly similar classes (*e. g. fastening* and *unfastening seatbelt*) and are only interested in coarse recognition. We further extend the original evaluation protocol with the Precision P, Recall R and $F1$ score of the individual classes. Formally, our metrics (including the balanced multi-class accuracy Acc) are defined as:

$$Acc = \frac{\sum_{i=1}^{m} \frac{A_i^{corr}}{A_i^{total}}}{m} \qquad P = \frac{A_i^{corr}}{A_i^{pred}}, \qquad R = \frac{A_i^{corr}}{A_i^{total}} \qquad F1 = 2 \times \frac{P \times R}{P + R} \tag{10}$$

where m is the total number of classes, A_i^{pred} the total number of examples which were assigned the label i, A_i^{corr} is the number of correctly predicted instances of class i, and A_i^{total} depicts the total frequency of class i in the test set.

In Table 11 we compare different architectures in terms of their top-5 and top-1 accuracy for rare, overrepresented and all activity classes. While the Inflated 3D ConvNet outperforms other approaches in all metrics (63.64% top-1 test accuracy for all classes), C3D seems to be stronger than Pseudo 3D ResNet in terms of the top-5 accuracy, while the latter model is better in top-

True Activity Class	Validation					Test				
	Precision %	Recall %	F1 %	Most Common Confusion Class	%	Precision %	Recall %	F1 %	Most Common Confusion Class	%
closing_bottle	0.65	0.71	0.68	opening_bottle	0.12	0.57	0.47	0.51	opening_bottle	0.30
closing_door_inside	0.86	0.92	0.89	taking_off_jacket	0.08	0.70	0.82	0.76	entering_car	0.06
closing_door_outside	1.00	1.00	1.00	closing_bottle	0.00	0.73	0.73	0.73	exiting_car	0.18
closing_laptop	0.70	0.54	0.61	working_on_laptop	0.23	0.67	0.37	0.48	opening_laptop	0.19
drinking	0.84	0.83	0.84	placing_an_object	0.07	0.93	0.88	0.90	closing_bottle	0.05
eating	0.86	0.85	0.86	sitting_still	0.07	0.76	0.59	0.67	sitting_still	0.27
entering_car	1.00	1.00	1.00	closing_bottle	0.00	0.77	0.74	0.75	closing_door_inside	0.11
exiting_car	1.00	1.00	1.00	closing_bottle	0.00	0.83	0.80	0.82	closing_door_outside	0.08
fastening_seat_belt	0.81	0.90	0.85	taking_off_jacket	0.04	0.77	0.82	0.79	placing_an_object	0.04
fetching_an_object	0.75	0.77	0.76	placing_an_object	0.07	0.64	0.67	0.65	placing_an_object	0.13
interacting_with_phone	0.77	0.85	0.81	sitting_still	0.05	0.92	0.86	0.88	eating	0.04
looking_moving_around	0.35	0.19	0.25	fetching_an_object	0.29	0.14	0.04	0.06	fetching_an_object	0.38
opening_backpack	0.00	0.00	–	placing_an_object	1.00	0.14	0.09	0.11	placing_an_object	0.39
opening_bottle	0.74	0.68	0.71	closing_bottle	0.24	0.72	0.68	0.70	closing_bottle	0.13
opening_door_inside	1.00	0.88	0.93	closing_door_inside	0.06	0.65	0.57	0.60	closing_door_inside	0.09
opening_door_outside	1.00	1.00	1.00	closing_bottle	0.00	0.89	0.89	0.89	exiting_car	0.11
opening_laptop	0.50	0.53	0.51	working_on_laptop	0.21	0.54	0.51	0.53	working_on_laptop	0.19
placing_an_object	0.73	0.68	0.70	fetching_an_object	0.09	0.59	0.72	0.64	fetching_an_object	0.11
preparing_food	0.33	0.05	0.09	eating	0.65	0.19	0.07	0.11	eating	0.41
pressing_button	0.94	0.88	0.91	placing_an_object	0.05	0.89	0.98	0.93	using_mm_display	0.02
putting_laptop_backpack	0.09	0.17	0.12	placing_an_object	0.83	0.27	0.20	0.23	placing_an_object	0.60
putting_on_jacket	0.58	0.54	0.56	taking_off_jacket	0.14	0.43	0.62	0.51	taking_off_jacket	0.15
putting_on_sunglasses	0.77	0.75	0.76	interacting_with_phone	0.09	0.88	0.71	0.79	closing_bottle	0.05
reading_magazine	0.96	0.93	0.95	reading_newspaper	0.04	0.89	0.88	0.88	reading_newspaper	0.08
reading_newspaper	0.81	0.95	0.88	reading_magazine	0.03	0.79	0.90	0.84	placing_an_object	0.05
sitting_still	0.92	0.95	0.93	using_mm_display	0.02	0.87	0.93	0.90	using_mm_display	0.02
taking_laptop_backpack	0.00	0.00	–	fetching_an_object	0.60	0.40	0.14	0.21	fetching_an_object	0.36
taking_off_jacket	0.58	0.55	0.57	putting_on_jacket	0.32	0.45	0.70	0.55	putting_on_jacket	0.15
taking_off_sunglasses	0.47	0.41	0.44	interacting_with_phone	0.15	0.75	0.56	0.64	fetching_an_object	0.16
talking_on_phone	0.95	0.69	0.80	sitting_still	0.15	0.85	0.71	0.77	sitting_still	0.19
unfastening_seat_belt	0.88	0.81	0.84	fastening_seat_belt	0.06	0.84	0.68	0.75	putting_on_jacket	0.08
using_mm_display	0.86	0.92	0.89	sitting_still	0.04	0.87	0.98	0.92	sitting_still	0.01
working_on_laptop	0.89	0.85	0.87	interacting_with_phone	0.08	0.90	0.76	0.82	fetching_an_object	0.06
writing	0.81	0.85	0.83	eating	0.05	0.86	0.58	0.70	reading_newspaper	0.13

Table 12: Extended performance analysis of the I3D model: Precision, Recall, F1 score as well as the most common confusion are calculated for each individual class. Most of the mistakes occur in semantically close classes or in cases, where one activity is a specialization of another one (*e. g. taking laptop from backpack* as a special type of *fetching an object*). We link this issues to too large receptive fields of the current architecture and too fast reduction of the image size

1 classification. C3D therefore is well suited for coarse classification but has issues discovering fine-grained structures. As expected, the top-1 recognition rate is significantly lower than the top-5 results, but this gap grows by a large margin for rare classes (*e. g.* this difference is 32.52% for uncommon- and 17.18% for common actions when considering the Inflated 3D ConvNet test setting). In general, activity recognition models seem to perform well for coarse behavior recognition (over 80% top-5 recognition rate in all settings for Inflated 3D ConvNet), while there is room for improvement in detecting fine-grained structures, especially for underrepresented classes (top-1 Inflated 3D ConvNet accuracy for rare categories under 50%). Still, identifying half of the actions which only had few training samples correctly is a good result, as CNNs are known for being data-hungry and the random baseline is only 100/34 = 2.94%, as we have 34 actions in total.

We now examine model performance *for the individual classes*, with exact precision, recall, F1-score and most common confusion provided in Table 12. We see in Figure 28b that while *all* of the very poorly recognized actions are underrepresented (frequency in the training set is illustrated through the circle size), well-recognized behaviors can be both: common and rare classes. The models are therefore *surprisingly tolerant to learning from few examples in case of highly discriminative actions*. For example, *closing door from outside* only has around 20 examples in the complete dataset (see the *Drive&Act* sample frequency statistics in Figure 6). However it is recognized correctly in 73% of the cases in test set and in all cases in the validation set (Table 12),

probably since the human is acting outside of the vehicle, which is easy to distinguish from the other activities. The combination of low discriminativeness and underrepresentation are fatal for a class: *e. g. taking laptop from backpack* and *preparing food*) are recognized correctly in only 14% and 7% of the test set cases. In Figure 28a we summarize the most common Inflated 3D ConvNet confusions, disclosing that eight out of ten most frequent mistakes entail an underrepresented ground-truth class. Oftentimes, the confusion happens when the two behaviors are semantically very close and one of them is rare. In this case, the model tends to predict the more frequent class (*e. g. preparing food* classified as *eating* in 41% of cases). Another cause of confusion is if one action being a special case of another: *putting laptop into backpack* is a specialization of *placing an object* and is classified as such in 60% and 83% of times in the test- and the validation set respectively. Similarly, *taking laptop from backpack* is marked as *fetching an object* in 36% of the test set samples. This might be connected to the fact that modern architectures downsample the image relatively fast to obtain large receptive fields and therefore focus on classification of coarse structures. Developing models which fit well for *fine-grained* recognition would therefore be beneficial. Some of the common confusions in Table 12 are surprising and uncover potential biases. For example, the most common confusion of *putting on sunglasses* is not *taking off sunglasses*, but *interacting with phone* on the validation set and *closing bottle* on in the test set. The model has presumably learned a bias of concise hand-centric movements, which are the common pattern of all these actions. Expanding the training set with more diverse examples might be important for learning to predict these activities *for the right reasons*, such as a *combination* of typical hand location, -movement and the correct object being held.

4.3 CHAPTER CONCLUSION

Applications in industrial systems require activity recognition models to not only be accurate, but also to determine, how likely they are to be correct in their prediction through realistic confidence estimates. In this chapter, we have opened a new research direction by elevating role of classification uncertainty in the field of activity recognition. Our final goal are models, which do not only select the correct behavior class but are also able to *identify misclassifications*. We measure the *reliability of model confidence* and evaluate it for two prominent action recognition architectures, revealing that the raw *Softmax* values of such networks do not reflect the probability of correct prediction well. We further implement two strategies for learning to convert poorly calibrated confidence values into realistic uncertainty estimates. First, we combine the native action recognition models with the off-the-shelf temperature scaling [55] approach which divides the network logits by a single learned scalar. We then introduce a new approach which learns to produce individual input-guided temperature values dependent on the input representation through an additional calibration network. We show in a thorough evaluation, that our model consistently outperforms the temperature scaling method and native activity recognition networks in producing realistic confidence estimates.

Besides identifying system failures, understanding its root cause it crucial to prevent them from happening in the future. With this notion in mind, we ease the secrecy of spatiotemporal CNNs through a diagnostic framework for shedding light on their internal reasoning processes when recognizing driver behavior. With a thorough inspection of the automatically learned inner representations, we are able to reason about the learned connections between the categories through the lens of deep models. Here, we evaluate current CNN-based approaches in their capabilities of capturing rare occurring activities. With our extension of the gradient-weighted class

activation maps into the temporal space, the visual inspection of spatiotemporal cues leading to failed predictions becomes much more tangible. With this diagnostic framework in place, narrowing down causes of failures enables testing pipelines to preemptively identify shortcomings in the data-distribution or detect present object- or location biases.

The experiments hold great promise that deep CNNs can be enhanced so that they can effectively reason about their uncertainty, with our CARING approach achieving state-of-the-art results in obtaining realistic confidence estimates for activity recognition. Besides, *transparency* and *explainability*, especially in case of incorrect outcomes, is vital for building trust in the problem solving abilities of deep CNNs. Although the original deep activity recognition CNNs have deficiencies in both, interpretability and adequately quantifying their confidence, our findings indicate that they have strong potential to become both, highly *reliable* and *interpretable* when improved with the proposed methods.

The scientific impact of this chapter can be summarized in four main contributions:

Contribution 1 : Integration of the ECE metric in the action recognition evaluation and first study of how well the confidence of the modern activity recognition architectures indeed reflects the likelihood of a prediction being correct.

Contribution 2: Combining two action recognition CNNs with the temperature scaling method [55] for network calibration, clearly improving the confidence values.

Contribution 3: A new method referred to as **C**alibrated **A**ction **R**ecognition with ***In**put **G**uidance* (CARING), which entails an additional calibration network learning to produce *temperature values dependent on the input representation*, leading to the most reliable confidence estimates.

Contribution 4: A diagnostic framework for analyzing and narrowing down causes of failures by *e. g.* through visual explanations with gradient-weighted class activation maps which we extended into the temporal space.

However, *identifying misclassifications* among the training classes is not the same as *identifying novelty* [146]. The essence of calibration-based methods, such as temperature scaling or our *CARING* model, is learning realistic confidence values on a held-out validation set. As shown in a recent study from the area of image classification [146], the reliance on this held-out validation set also becomes the greatest weakness of such models when facing domain shifts. In other words, confidence calibration is effective, as long as the test data roughly reflects the distribution of the validation set. Identifying behaviors not previously seen by the classifier is therefore a different challenge that we will meet in the next chapter.

5

UNCERTAINTY-AWARE OPEN-SET RECOGNITION

While the calibration-based methods of the previous chapter yield highly realistic confidence estimates under closed-set conditions, their weak spot are distributional shifts, as the outcome relies on the scaling learned on a validation set [146]. In this chapter, we move to a setting, where new actions may occur at any time and introduce the concept of *open sets* to the areas of driver observation and general activity recognition, where the methods have been evaluated on a static set of classes in the past. This chapter is based on our *BMVC 2018* publication [168] considering general activity recognition and the *IV 2020* publication [172] focusing on driver observation and is structured as described in the following. Section 5.1 motivates and formally defines the task, introducing the Open-Drive&Act benchmark and open set extensions of general activity recognition datasets. Section 5.2 describes a generic framework for open set action recognition, which enhanced current closed set models with novelty detection algorithms. In Section 5.3, we introduce *Bayesian-I3D* – a new approach for detecting previously unknown behaviors based on Bayesian uncertainty of the output neurons approximated through Monte Carlo-dropout sampling. Finally, Section 5.4 evaluates our models and Section 5.5 draws conclusions of this chapter.

All following sections of this chapter consolidate our publication in *IV* 2020 [172] (best student paper runner-up award), © IEEE and our BMVC 2018 publication [168].

5.1 OPEN SET ACTIVITY RECOGNITION: MOTIVATION, DEFINITION AND OVERVIEW

How can we deal with activities that were not learned by our model? As we will never be able to capture and annotate all possible driver behaviors in our training data, we need to find a way to handle such *unknown* examples. While this task is vital for practical applications of driver activity recognition models, previous approaches merely focused on optimization on a fixed set of carefully designed actions [1, 21, 47, 116, 141, 142, 214, 218, 226]. Exploring what happens if a video containing a new behavior is passed to the classification model, has been overlooked in the past. Activity recognition has a variety of applications inside the vehicle cabin, ranging from perceiving distraction and sending a warning to increasing comfort during autonomous driving. However, if a model developed for closed set recognition is utilized directly in an open world,

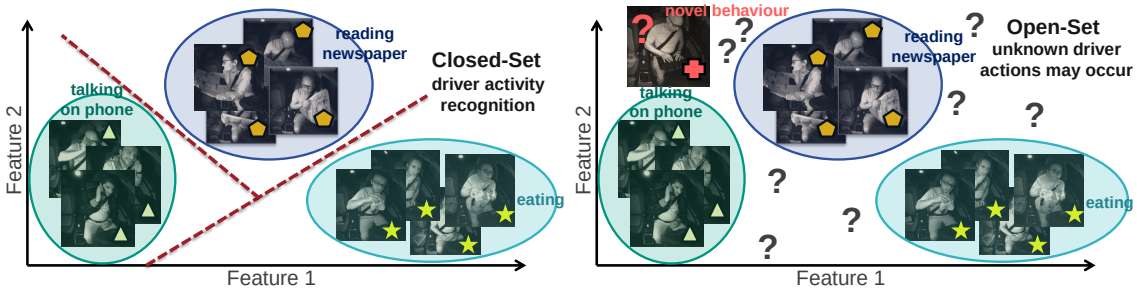


Figure 29: **Closed- vs. open set action recognition.** Standard *closed set* benchmarks only consider classes previously seen during training (left). We propose the task of *open set activity recognition*, where behaviours not previously seen by the classifier are also present at test-time (right).

it will be quickly exposed to uncertain situations. This might result in a high number of false positive detections which are both highly disturbing for the user and potentially dangerous. Beyond driver monitoring, similar open-world circumstances are faced in numerous applications of activity recognition models, such as human-robot interaction or assisted living [20, 176, 227]. The field of general activity recognition experiences a similar research gap – the existing approaches [17, 58, 153] assume, that the model will be deployed under closed-set conditions and no previously unseen behaviours will occur.

Several recent works raise concerns about this research gap, underlying the importance of studying the behavior of such models when exposed to previously unseen classes and highlighting their limits in cases of uncertainty [44, 137, 147, 192]. Although multiple datasets for both, general human activity recognition [17, 90, 185, 190] and driver monitoring applications [1, 121], including *Drive&Act*, have been published in recent years, they all represent a setting where the action categories in the training and test set are exactly the same (Figure 29, left). For example, in our *Drive&Act* benchmark [121], which is the largest publicly available driver activity recognition dataset at the time, all of the 34 fine-grained activity classes are used for both: evaluation and training. The impressive results we have achieved by using deep 3D CNNs on the conventional datasets in the first chapter, may therefore draw an artificially rosy picture, as the closed set constraint represents a significant bottleneck in the *dynamic* real-life environment. We therefore propose to incorporate previously unseen behaviors in the evaluation of both, driver observation and general activity models and expose them to *open set* conditions (Figure 29, right).

The distinguishing component of an open set model lies in its ability to identify previously unseen classes, which is directly linked to model's *epistemic uncertainty* (see Section 2.5). In conventional CNNs for action recognition, output of the last fully-connected layer is normalized using the *Softmax* function, resulting in point estimates for a fixed set of classes from which Cross-Entropy loss is computed. As previously mentioned (Section 4.1.4), the resulting Softmax scores are often inaccurately denoted as class probabilities and tend to be biased towards very high values [44, 55, 169]. Still, these estimates alone are often used as the basis for a rejection threshold in other computer vision tasks, such as obstacle detection [62, 156, 163]. Calibration-based methods, such as our *CARING* model, produce far better uncertainty estimates *given that the behavior is known*, but an eminent drawback arises from their dependence on data distribution of the held-out examples used to calibrate the model. An unknown situation is, by definition, absent during training, and the achieved calibration improvement, therefore, looses its power under distributional shifts [146]. Since such calibration-based algorithms are not directly applicable

under open set conditions, we aim for novelty detection approaches specifically targeting models epistemic uncertainty.

In this section, we aim to introduce the concept of *open sets* to the area of driver observation, where the methods have been validated only on a static set of classes in the past. First, we formulate the problem of open set activity recognition, where a model is intended to identify behaviors previously unseen by the classifier and present the novel *Open-Drive&Act* for open set driver observation as well as two benchmarks for the general activity recognition case, extending the HMDB-51 [89] and UCF-101 [190] datasets. We integrate the existing 3D CNNs for closed-set activity recognition in a generic open set activity recognition framework, where we enhance them with multiple strategies for novelty detection. Our framework is capable of both, standard supervised classification of familiar activities and novelty detection serving as a filter to distinguish between *the known* and *the unknown*. We further introduce a new model for novelty detection based on approximation of the epistemic uncertainty via Monte-Carlo dropout. Besides using the uncertainty of the top-1 neuron alone, we incorporate the approximated posterior distributions of *all* the output neurons, so that they selectively contribute to the resulting novelty score in a voting-like manner. Our experiments demonstrate clear benefits of uncertainty-based models, while leveraging the uncertainty of the designate output neurons in a voting-like fashion leads to the best recognition results.

5.1.1 *Problem Formulation and Testbed*

The vast majority of published methods [17, 153, 196] and datasets [17, 89, 121, 190] are developed under the assumption, that all categories are known a priori. We believe, that distinguishing between the *known* and the *unknown* is decisive for the model to be deployed in real applications and explore the field of both driver- and general activity recognition under open set conditions, a setting which has been little-noticed before.

5.1.1.1 *Problem Formulation*

Given a model trained on the available *known* behaviors, our framework is exposed to both, *known* and *novel* activities at test-time. Let $\mathbf{x}$ be a test input video representation and f_θ a classification model optimized on a training set of *known* action categories $\mathcal{A}\{1, ..., m\}$. With a_{true} being the true label of x, the assumption $a_{true} \in \mathcal{A}$ *does not hold* anymore under open set conditions. An open set model ought to not only learn to predict the correct activity (*i. e.* $a_{pred} = a_{true}$) in case a_{true} is a *known* class, but also quantify, how likely it is, that the depicted behaviour is *unknown*. We therefore need to additionaly solve the task of novelty detection,*i. e.* produce a *newness score* $\nu(\mathbf{x}, f_\theta)$, where high values indicate that $a_{true} \notin \mathcal{A}$.

We consider two different task formulations to validate the open set model performance: (1) novelty detection and (2) open set multi-class recognition. (1) The novelty detection benchmark validates, how well the newness score $\nu(\mathbf{x}, f_\theta)$ can be used to distinguish the *known* from the *unknown* and is therefore treated like binary classification during evaluation, using the area under curve (AUC) values of the receiver operating characteristic (ROC) curve. (2) In the latter setting of open set multi-class recognition we treat *unknown* as an additional category. The task therefore extends the standard classification with an unknown class,*i. e.* our goal is $\tilde{a}_{pred} = \tilde{a}_{true}$, where $\tilde{a}_{pred}, \tilde{a}_{true} \in \tilde{\mathcal{A}}\{1, ..., m, m+1\}$ and $m+1$ portrays the *unknown* class. Multi-class accuracy is used as the recognition metric in this case. Since the category distribution in *Drive&Act* is not balanced

(see category statistics of the dataset in Figure 6), we evaluate both, standard- (*i. e.* unbalanced-), and balanced accuracy (meaning a separate measurement for each ground-truth category averaged over all classes).

5.1.1.2 *Open Set Activity Recognition Testbed*

OPEN-DRIVE&ACT BENCHMARK. To address the lack of open set benchmarks for driver observation, we introduce *Open-Drive&Act* – the first driver activity recognition testbed in which the evaluation procedure comprises both *known* and *unknown* behaviors. Thereby, we extend our *Drive&Act* dataset, previously comprising only a closed set recognition setting, to the open set scenario and formalize the evaluation process to handle *unseen* classes as follows. We employ the available *Drive&Act* videos captured by a NIR-camera facing the person and all 34 annotations on the fine-grained activity level (as defined in Section 3.1.2.2). Note, that this level also serves as the primary evaluation mode in our previous chapters. We then split the dataset into 24 *seen* and 10 *unseen* categories, of which 5 *unknown* classes are used for validation and 5 for testing. Videos of *unseen* activities are not available during training, while samples of the remaining *seen* activities are further split into training (60%), validation (20%) and testing (20%). We randomly generate 10 splits and report the average and the standard deviation of the recognition metrics. To create an avenue for future work, we will make *Open-Drive&Act* public at www.github.com/aroitberg/open-set-driver-activity-recognition.

BENCHMARKS FOR GENERAL OPEN SET ACTIVITY RECOGNITION. We then broaden our task to the *general* activity recognition setting. Same as in driver observation, there is no established evaluation procedure available for action recognition under open-set conditions. We therefore modify the existing evaluation protocols of two well established datasets, HMDB-51 [89] and UCF-101 [190] to suit our task. We evenly split each dataset into *seen* and *unseen* categories (26/25 for HMDB-51 and 51/50 for UCF-101), using 70% of the *seen* examples for training and 30% for testing and creating one additional split for validation. In the same manner as *Open-Drive&Act*, we evaluate the models using 10 randomly generated splits. We will make the splits publicly available at https://cvhci.anthropomatik.kit.edu/~aroitberg/novelty_detection_action_recognition.

5.2 FRAMEWORK

5.2.1 *Architecture*

In this section, we present a generic framework for open set driver activity recognition, incorporating both facets necessary for such systems: conventional supervised classification of previously *seen* activity classes and the ability to identify *novel* activities, which were not present during the classifier training (*i.e.* novelty detection). To achieve this, we augment a closed-set architecture, in our case a 3D CNN, with a *novelty detection module*, which objective is to quantify the *newness* of an input sample.

First, we compute an encoding of the input, by using an intermediate layer of a neural network f_θ trained for supervised classification of the *known* classes. The embedding is then passed to the novelty detection module, which decides whether the represented class is familiar. This module produces a *newness score* $\nu(\mathbf{x}, f_\theta)$ and then determines whether the instance is seen or unseen

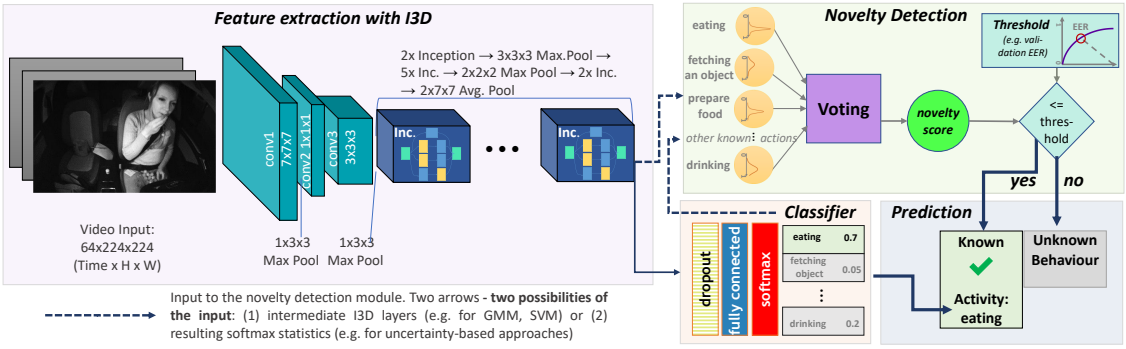


Figure 30: Overview of the proposed framework for open set driver activity recognition.

by thresholding it (see Section 5.2.2 for more detail). The threshold can be chosen, *e.g.*, using the Equal Error Rate (EER) of the ROC curve on the validation set. EER is the point on the ROC curve that corresponds to have an equal false positive and true positive rates. This point is computed by intersecting the ROC curve with the descending diagonal.

Depending on the outcome, the video is either conveyed to a supervised classification module or marked as *unknown*. Figure 30 provides an overview of our architecture with novelty detection using uncertainty-based selective voting of the output neurons (Section 5.3). While Figure 30 illustrates the best performing variant of the novelty detection module, we also adopted other popular novelty detection approaches, such as One-Class SVM [182], as described in Section 5.2.2 and Section 5.3.

For feature learning and classification, we adopt the Inflated 3D architecture (I3D) proposed in [17], as it had repeatedly proven to give the best closed-set recognition outcome in previous chapters. The network takes as input a video snippet of 64 frames with a resolution of 224×224 and learns to assign one of the *known* activity labels. For more details on the I3D architecture, please refer to Section 3.1.4. Figure 31 visualizes the obtained I3D embedding representations for both *known* and *unknown* classes for one *Open-Drive&Act* validation split using t-SNE [107], where we see a clear correlation between the computed features and action semantics. The known behaviors are colored, while the five unknown categories are gray. This representation gives us first qualitative clues, that such open set recognition is difficult, as the intermediate representations of the *unknown* categories are often placed in the imminent neighbourhood of *known* behaviors (*e.g.* the embedding of a novel activity *working on laptop* is very close to *opening* and *closing laptop*). Still, the learned relationships are logically coherent: the model places the *unkown* activity *drinking* between the familiar actions *eating* and *preparing food*.

5.2.2 *Novelty Detection Variants*

A novelty detection module takes as input an intermediate CNN representation of a video, quantifies its *newness* and decides, whether the instance is *seen* or *unseen* by thresholding the *newness score*, as previously described. We implement three popular methods for novelty and our outlier detection for quantifying the sample *newness*: (1) a One Class Support Vector Machine (SVM) [181]; (2) a Gaussian Mixture Model (GMM) [230]; and (3) neural network *Softmax* probability estimates [62, 163] as the value for thresholding. We further introduce (4) a new method for

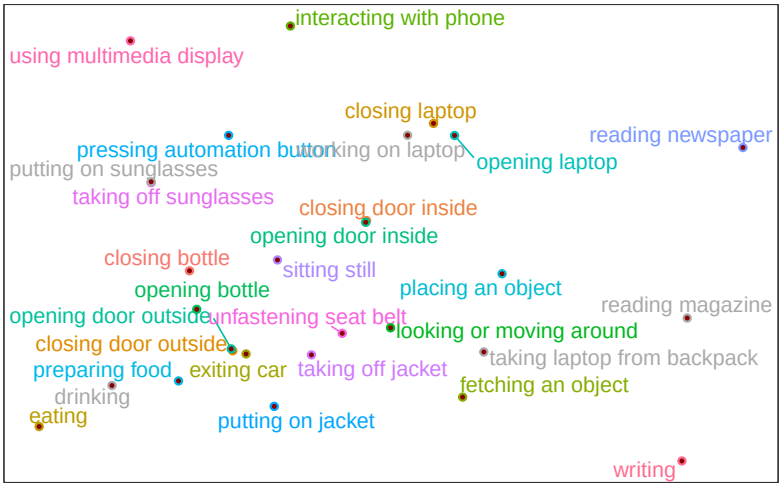


Figure 31: T-SNE [107] representation of the I3D video embeddings of one Open-Drive&Act validation split. Labeled dots depict the mean computed from of all samples of the corresponding category. Activities *known* from training are in *color*, the *unknown* behaviours are in *gray*).

novelty detection based on classifier uncertainty obtained through a Bayesian Neural Network approximation, which we first briefly explain and then thoroughly describe in Section 5.3.

ONE-CLASS SVM The One-class SVM introduced by Schölkopf *et al.* [181] is a widely used non-probabilistic method for novelty detection. The model learns to transform video embeddings into a feature space defined by a boundary hyperplane aiming to increase the separation margin from the origin. The novelty estimate is then quantified as the signed distance to the separating hyperplane, which is positive, if the data point is inside the boundary (*i. e.* a *known* class). We use a Radial-Basis-Function kernel and train the SVM using the intermediate I3D-embeddings as our video representation.

GAUSSIAN MIXTURE MODELS We consider a classical generative approach for novelty detection using Gaussian Mixture Models (GMMs) [150]. We use a mixture of 24 Gaussian distributions (*i.e.* number of the known categories) and estimate model parameters using the Expectation-Maximization algorithm to fit our *known* activities, represented as the intermediate embeddings of the I3D model. We then use the negated estimated probability density function to quantify the novelty.

NEURAL NETWORK SOFTMAX CONFIDENCE Multiple works detect novel concepts by thresholding the *Softmax* value of the neuron with the highest activation, *i.e.*, the *Softmax*-normalized output of the last fully-connected layer [62, 112, 163]. The resulting *Softmax* scores are often denoted as class probabilities [44], since they satisfy the properties of a probability function: they range between 0 and 1 and sum up to one. The input is assigned the class with the maximum *Softmax* score and can be directly used to quantify the data normality [156, 163]. We therefore use the I3D model to distinguish between the known activities directly through the negated top class *Softmax* score as our novelty measure.

NOVELTY DETECTION BASED ON BAYESIAN UNCERTAINTY Apart from the existing approaches, we introduce *Bayesian-I3D* — a new method for novelty detection in action recognition. Our main idea is to leverage the approximated Bayesian uncertainty of the output neurons, which is estimated using Monte-Carlo Dropout Sampling [44] and let the output neurons (which are linked to the *known* categories) vote about the sample novelty. Furthermore, we differentiate three strategies: voting of *all* output neurons, selective voting of a designated group of neurons or letting the maximum-confidence neuron decide on the uncertainty alone. The next section explains our approach in depth.

5.3 DEEP PROBABILISTIC NOVELTY DETECTION

5.3.1 *Background: Bayesian Neural Networks*

In conventional neural networks, the resulting *Softmax* scores represent single point estimates for each of the m categories and are often denoted as the class probabilities. As already pointed out in the light of identifying misclassification (Section 4.1), such *Softmax* estimates tend to be unjustifiably high [44, 62, 108, 137, 169, 193]. While giving excellent top-1 classification results, overly self-confident models become a burden in applications, where high number of false positives is highly inconvenient and potentially dangerous. The calibration-based methods of the previous chapter give us highly realistic confidence estimates, but are not applicable in an open-set case, as they are grounded in the confidence calibration step on a held out validation set [146] and data allowing such calibration in an open world case is absent by definition. Despite this drawback, the magnitude of the output neuron with highest Softmax score is often used directly to quantify sample novelty [7, 62, 112, 163] (and is therefore also adopted as our baseline).

An alternative way to model network confidence is to aim for the posterior *distributions* instead of single point estimates for each class, which can be achieved using Bayesian Neural Networks (BNN) [108, 135], first introduced by Mackay in 1992. Instead of having a fixed set of parameters, BNN applies a prior distribution over the weights and biases and aims for the predictive posterior given the data. Therefore, both weights and outputs of these networks are probabilistic distributions. The predictive probability of a BNN is obtained by integrating over the parameter space θ (Eq. 11). Since the posterior $p(\theta|\mathcal{D}_{train})$, where $\mathcal{D}_{train}$ denotes the training data and annotations, is computationally intractable, it is often replaced with a variational distribution $q_\omega(\theta)$ and approximated using Monte-Carlo sampling (Eq. 12 and Eq. 13):

$$p(a_i|\mathbf{x}, \mathcal{D}_{train}) = \int_\omega p(a_i|\mathbf{x}, \theta) p(\theta|\mathcal{D}_{train})\, d\theta \tag{11}$$

$$\approx \int_\theta p(a_i|\mathbf{x}, \theta) q_\omega(\theta) d\theta \tag{12}$$

$$\approx \frac{1}{T} \sum_{t=1}^{T} p(a_i|\mathbf{x}, \theta_t), \text{ with } \theta_t \sim q_\omega(\theta) \tag{13}$$

where $a_i \in \mathcal{A}$ denotes an activity category known from $\mathcal{D}_{train}$.

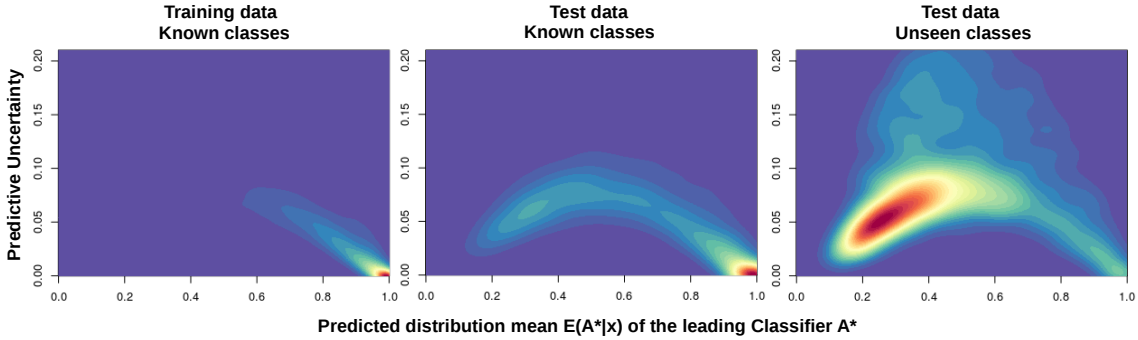


Figure 32: Predicted distributions as a 2D histogram. Distribution of the predictive mean and uncertainty of the assigned action category (*i. e.* the class with highest *Softmax* confidence mean) after 100 stochastic forward passes for the known and unseen actions (computed on HMDB-51 dataset). Red denotes common cases (high frequency), while blue illustrates unlikely cases.

5.3.2 *Bayesian I3D - Approximation via Probabilistic Dropout Sampling*

We approximate the BNN posterior with networks parameters modelled as a Gaussian Process (GP) [158] using the method proposed by Gal and Ghahramani [44]. This method is based on Dropout [191], a widely used regularization technique for training neural networks, which has proven to be very effective against overfitting. Typically, dropout is only active during training and is disabled at test-time, yielding deterministic network weights and involvement of all layer inputs. Gal and Ghahramani (2016) have provided a theoretical proof, that iteratively applying dropout *at test-time* and then computing the output statistics of the model, is a variational approximation of a BNN posterior distribution with network parameters modeled as a GP [44]. Dropout sets the nodes to zero with a probability ρ making the network non-deterministic (therefore $q_\omega(\theta)$ follows a Bernoulli distribution for model weights to approximate the BNN posterior). Computing the mean and variance of multiple probabilistic forward passes leads to the outcome being a Gaussian distribution instead of a single point estimate, where the variance can be viewed as a measure of *epistemic* uncertainty [44]. We leverage such approximation of Bayesian Neural Networks in order to use the resulting epistemic uncertainty estimates for identifying activities not present during training. In the following, we refer to such approximation as Monte-Carlo Dropout (MC-Dropout).

We equip the I3D action recognition model with MC-Dropout at test-time after the last average pooling layer and use its Bayesian version to quantify the normality of driver behavior. Let $\mathbf{x_{emb}}$ be the embedding generated by I3D after the last average pooling layer and $\mathbf{W}$, $\mathbf{b}$ be the weight matrix and bias vector of the last fully connected layer. Typically, network weights $\mathbf{W}$ are deterministic at test time and dropout is only active during training. Instead, we apply dropout interactively at test time, which below is formalized by multiplying $\mathbf{W}$ with a diagonal matrix $\mathbf{D}$, with diagonal values set to 0 with probability ρ and otherwise to 1. The probabilistic I3D no longer gives single point estimates, but now predicts *Gaussian distributions* for each known

driver behaviour. The mean of the computed distribution is now used to assign the known class to the data and is computed over T stochastic iterations as follows[1]:

$$\mathbb{E}(a_i|\mathbf{x}) \approx \frac{1}{T}\sum_{t=1}^{T} \text{softmax}(\mathbf{W}\ \mathbf{D}\ \mathbf{x}^{\mathbf{t}}_{\mathbf{emb}} + \mathbf{b}) \tag{14}$$

Similarly, we compute the predicted distribution variance and use it as an estimate of the model's epistemic uncertainty:

$$U(a_i|\mathbf{x}) \approx \frac{1}{T-1}\sum_{t=1}^{T} [\text{softmax}(\mathbf{W}\ \mathbf{D}\ \mathbf{x}^{\mathbf{t}}_{\mathbf{emb}} + \mathbf{b}) - \mathbb{E}(a_i|\mathbf{x})]^2 \tag{15}$$

Figure 32 illustrates the distribution of the predictive mean $\mathbb{E}(a^\star|\mathbf{x})$ on the X-axis and epistemic uncertainty estimates $U(a^\star|\mathbf{x})$ on the Y-axis computed with our probabilistic version of I3D for examples of *known* and *novel* activities. Note, that we visualize the values of *assigned* known class activity $a^\star = \underset{a_i \in \mathcal{A}}{\text{argmax}}\ \mathbb{E}(a_i|\mathbf{x})$, *i.e* the neuron with the highest *Softmax* score (based on the average of T stochastic forward passes). We observe contrasting patterns of the resulting probability distributions for these two cases, giving us first qualitative evidence that approximation of Bayesian uncertainty is an effective signal for detecting previously unseen actions.

5.3.3 *Uncertainty-based Selective Voting of Output Neurons*

Until now, we have considered the approximated uncertainty of *one* output neuron linked to the category with the highest confidence, which therefore would become the predicted class if our framework marks the sample as *known*. In this section, we explore the idea of consolidating the uncertainty of *all* output neurons in contrast to utilizing the top-1 uncertainty alone. Intuitively, we do not only consider the certainty of the model that the input indeed belongs to the predicted class, but also, how sure it is, that it *does not* belong to one of the other classes. To implement this idea, we let the output neurons vote about the novelty of a sample. Furthermore, the voting is privileged only to the subset of output neurons, which usually have stable uncertainty values given the predicted class.

Given an input video representation $\mathbf{x}$, and a model f_θ trained to predict one of the *known* classes $\mathcal{A}\{1, ..., m\}$ (in our case I3D), our goal is to estimate the novelty score $v(\mathbf{x}, f_\theta)$. After a single forward pass and *Softmax* normalization, the values of the *m output neurons* depict the probability estimates $\hat{p}(a_i|\mathbf{x})$ for each known action class $a_i \in \mathcal{A}$. Similarly, in our *Bayesian I3D*, T stochastic forward passes estimate the probability distributions as the predictive mean $\mathbb{E}(a_i|\mathbf{x})$ and uncertainty $U(a_i|\mathbf{x})$ for each of the *m output neurons*.

We now introduce two concepts: the *leader* and the *council*. The *leader* $a^\star$ denotes the "winning" neuron, *i. e.* the output neuron with the highest value. The *leader* therefore marks the predicted *known* action and assuming that the class of $\mathbf{x}$ is one of the *known* categories, *i. e.* $a_{pred} = a^\star \in \mathcal{A}$. In the next step, other output neurons are going to question the correctness of the *leader* by voting about whether they are confident about its prediction, or not. One way of achieving this is to let all neurons contribute equally to the decision. However, should all neurons indeed contribute equally, when checking the *leader*? We observe varying levels of stability of different output neurons depending on the *leader* and therefore introduce the *council*

1 For readability, we abbreviate the predicted distribution mean for an a_i action class $\mathbb{E}(\hat{p}(a_i|\mathbf{x}))$ as $\mathbb{E}(a_i|\mathbf{x})$.

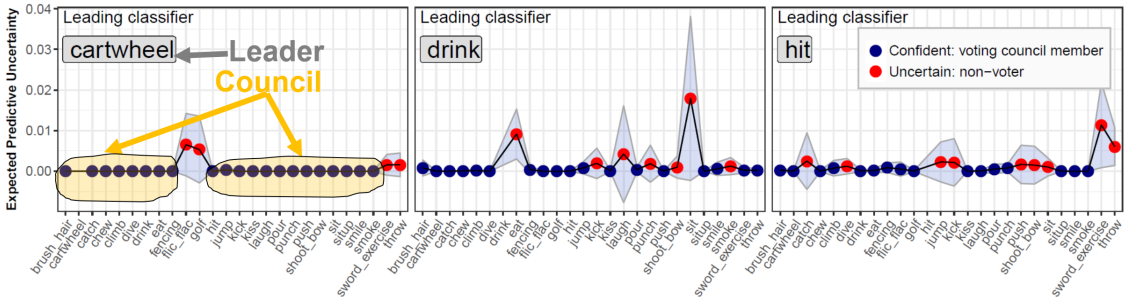


Figure 33: Council members and uncertainty statistics for three different leaders (HMDB-51). The classifier's *average* uncertainty and its *variance* (area surrounding the point) illustrate how it changes its belief in the leader for different data inputs. Blue points are in the council of the current leader, while red points are classifiers that did not pass the credibility threshold.

– a subset of *informed neurons* $\mathcal{I}_{a^\star}$, which are classes with the output neuron uncertainty usually not changing much for different inputs with the same leader $a^\star$. Such informed *council* neurons will help us validating the decision of a specific *leader* and are chosen for each *leader* individually. Formally, $a_i \in \mathcal{I}_{a^\star}$ if the uncertainty variance of the a_i neuron observed during training is below a credibility threshold ξ .

Main steps of our approach can be summarized in following way: we first decide on the predicted category (choosing the *leader*), form a group of "informed" neurons depending on the *leader* (choosing the *council*) and let the *council* vote, whether to trust the *leader* (known activity) or not (novel activity) based on their uncertainty.

CHOOSING THE LEADER. We select the leader $a^\star$ as the neuron with the highest mean of the resulting distribution. In other words, the leader corresponds to the category with highest expected *Softmax* prediction measured over T MC-dropout iterations:

$$a^\star = \underset{a_i \in \mathcal{A}}{\operatorname{argmax}}\, \mathbb{E}(a_i|\mathbf{x}), \tag{16}$$

where $\mathbb{E}(a_i|\mathbf{x})$ is estimated according to Eq. 14.

CHOOSING THE COUNCIL. Guided by the idea, that we should not limit ourselves to the leader confidence alone, we let the rest of the category neurons contribute to the decision. We notice that the distributions of different category neurons exhibit specific patterns depending on the current leader (Figure 33). For example, for the leader *hit*, the variance of neuron uncertainty for categories such as *eat* or *drink* is very low. If a sample is classified as *hit*, but unexpectedly high uncertainty is measured for a neuron of *eat*, the video is likely to depict a behaviour we have never seen before. The neuron of *sword exercise*, on the other hand, has highly fluctuating uncertainty values for the leader *hit* and is therefore not very reliable. Motivated by this, we select the *council* of the current *leader* – a subset of categories which are *informed* $\mathcal{I}_{a^\star}$, meaning that their matched output neurons produce highly consistent uncertainty values (*i. e.* the variance of the uncertainty for the specific leader measured during training is low).

To select the *council* of each *leader*, we randomly divide the initial training set into a main training set $\mathcal{D}_{train}$ used for optimization of our CNN f_θ, and a holdout set $\mathcal{D}_{holdout}$ used for choosing the informed neurons of the councils (ratio 9 : 1). After we have estimateed the parameters of our deep model on $\mathcal{D}_{train}$, we run an evaluation on all samples from $\mathcal{D}_{holdout}$. Independently for each

true category $a_{true} \in \mathcal{A}$ in our model, we select the correctly classified examples $\mathcal{D}_{corr}^{a_{true}} \subseteq \mathcal{D}_{holdout}$ (which means, that the *leader* is correct and $a_{true} = a^\star$). For each data sample $\mathbf{x} \in \mathcal{D}_{corr}^{a_{true}}$, we estimate the uncertainty $U(a_i|\mathbf{x})$ of the rest of the category neurons $a_i \in \mathcal{A} \setminus \{a^\star\}$ using the MC-Dropout approach. To determine how stable the uncertainty values of these neurons' usually are for the current leader we measure their uncertainty variance:

$$Var(a_i|a_{true}) = \frac{1}{N}\sum_{n=1}^{N}(U(a_i|\mathbf{x}) - \mathbb{E}[U(a_i|\mathbf{x})])^2 \tag{17}$$

where $N = |\mathcal{D}_{corr}^{a_{true}}|$ and $\mathbb{E}[U(a_i|\mathbf{x})]$ is the expected value of the uncertainty of the category neuron a_i computed from $\mathcal{D}_{corr}^{a_{true}}$. The output neurons with uncertainty variance under a certain credibility threshold $Var(a_i|a_{true}) < \xi$ are then elected to the council of the leader $a_{true} = a^\star$. We choose $\xi = 0.004$ empirically using the validation set.

Figure 33 illustrates the uncertainty statistics of three ground-truth categories (*i. e.* which are also the *leaders*, as we only use correctly classified examples for choosing the informed neurons) and their elected councils. For instance, eight classifiers did not pass the credibility threshold for the leader *drink* and were excluded from its council. The variance of the uncertainty is especially high for *sit* and *eat* in this case, which is unsurprising, since those actions are often observed in the same context.

Algorithm 1 Novelty Detection by Voting of the Council Neurons

Input: Input sample $\mathbf{x}$, Classification Model f_θ, m sets of *Council* members for each *Leader* : $\mathcal{I}_{a^\star} \forall a^\star \in \mathcal{A}$

Output: Novelty score $\nu(\mathbf{x})$

Inference using MC-Dropout
    Perform T stochastic forward passes: $\hat{p}_{a_i}^t = \hat{p}(a_i|\mathbf{x}, f_\theta^t)$;
for all $a_i \in \mathcal{A}$ **do**
    Calculate the prediction mean and uncertainty: $\mathbb{E}(a_i|\mathbf{x})$ and $U(a_i|\mathbf{x})$
end for
Find the *Leader*: $a^* = \underset{a_i \in A}{\operatorname{argmax}}\ p(a_i|\mathbf{x})$
Select the *Council*: $\mathcal{I}_{a^\star}$
Compute the *novelty score* : $\nu(\mathbf{x}) = \frac{\sum_{a_i \in \mathcal{I}_{a^\star}} U(a_i|\mathbf{x})}{|\mathcal{I}_{a^\star}|}$

VOTING FOR NOVELTY Given the trained deep model f_θ and the sets of all council members $\mathcal{I}_{a^\star} \forall a^\star \in \mathcal{A}$ from the previous step, we can now estimate the newness $\nu(\mathbf{x})$ of a new input $\mathbf{x}$. First, through MC-Dropout with T stochastic forward passes we obtain the resulting Gaussian distribution by computing the mean $\mathbb{E}(a_i|\mathbf{x})$ and variance (*i. e.* uncertainty) $U(a_i|\mathbf{x})$ for the output neurons of each category $a_i \in \mathcal{A}$. Then, the *leader*, which is the classifier with the maximum predicted mean is selected. Finally, the council members of the chosen leader vote for the novelty of sample $\mathbf{x}$ based on their estimated uncertainty (see Algorithm 1).

Examples of such voting outcome for three different leaders are illustrated in Fig. 34. In case of category *cartwheel*, we can see that when the leader is voting indeed for the correct category, all council members show low uncertainty values therefore resulting in a low novelty score, as uninformed neurons (marked in red) are excluded. However, we see a very different outcome for

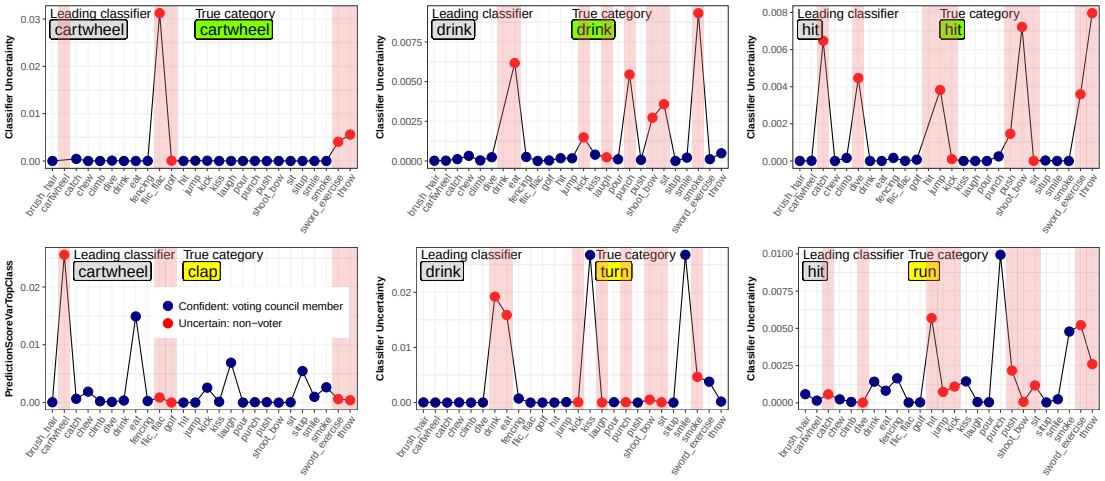


Figure 34: Examples of selective voting for the novelty score of different activities. The first row depicts the case where the samples are of *known* classes and second row for those of *novel* classes. Red points highlight classifiers, which were *excluded* from the council of the current leader. Their uncertainty is, therefore, ignored when inferring the novelty score.

an example from an *unknown* category *clap* which is also predicted as *cartwheel* by our closed-set model. Here, multiple neurons which are in the council (marked in blue) show unexpected high uncertainty values (*e.g. eat, laugh*), therefore overruling the *leader's* decision and voting for a high newness score.

MODEL VARIANTS. While our main idea is the Bayesian version of action recognition 3D CNNs via MC-Dropout approximation, we distinguish three model variants based on how the uncertainty of the individual output neurons contributes to the final novelty assessment:

1. The *Informed Democracy* model: this is our main approach with the previously described *informed* voting, which is restricted to the *council* of the current *leader*.

2. The *Uninformed Democracy* model: here, we leverage the uncertainty and *all* output neurons equally. The step 7 in Algorithm 1 is therefore replaced with $\nu(\mathbf{x}) = \frac{\sum_{a_i \in \mathcal{A}} U(a_i|\mathbf{x})}{m}$.

3. The *Dictator* model: in this model, the decision is made based on the leader's uncertainty alone, the newness score is therefore $\upsilon = U(a^\star|\mathbf{x})$.

5.4 EXPERIMENTS

We evaluate the open set activity recognition pipeline extensively using our testbed (Section 5.1), equipping the I3D model with different variants of the novelty detection module. Examined methods range from standard approaches (One-Class SVM, GMM, using CNN confidence directly) to the novel uncertainty-based methods we have introduced in Section 5.3 and are compared with each other and a random classifier baseline.

Novelty Detection Model	Drive&Act		HMDB-51		UCF-101	
	AUC [%]	± SE	AUC [%]	± SE	AUC [%]	± SE
Standard Methods and Baselines						
Random Chance	50.00	-	50.00	-	50.00	-
One-class SVM	59.35	± 11.19	54.09	± 3.0	53.55	± 2.0
Gaussian Mixture Model	65.73	± 7.09	56.83	± 4.2	59.21	± 4.2
Conventional NN Confidence	81.05	± 4.64	67.58	± 3.3	84.28	± 1.9
Our Proposed Model based on Bayesian Uncertainty						
Bayesian I3D - Dictator	82.69	± 4.36	71.78	± 1.8	91.43	± 2.3
Bayesian I3D - Uninformed Democracy	83.52	± 3.84	73.81	± 1.7	92.13	± 1.8
Bayesian I3D - Informed Democracy	**84.33**	± 3.85	**75.33**	± 2.7	**92.94**	± 1.7

Table 13: Results for the detection of unknown behaviors as a binary decision task on *Open-Drive&Act* and the open-set versions of *HMDB-51* and *UCF-101* (average and standard deviation of the area under the ROC curve computed over the ten splits). Our Bayesian I3D models with different voting schemes consistently outperform the standard approaches.

5.4.1 *Novelty Detection*

We first examine our approaches in terms of novelty detection *i. e.* binary decision whether the observed behavior was present during training, or not. We use the area under the ROC curve computed from the produced newness scores as our evaluation metric and report the mean and standard deviation over ten splits in Table 13.

We begin by discussing the driver observation results (*Drive&Act* benchmark). While all models surpass the random classifier, neural network-based approaches show clear advantages, as even using the neural network *Softmax* score alone outperforms a GMM by 15.32% . The recognition rate is further improved by using probabilistic approaches, as all model variants based on Bayesian uncertainty surpass using conventional *Softmax* confidence. While we report the results for $T = 100$ stochastic forward passes, our further evaluation has shown, that while reducing T to 10 indeed adversely affects the performance, the difference is small (below 1%) and might be omitted in favor of better computation speed. Leveraging uncertainty of all the output neurons via informed voting leads to the best recognition rates, surpassing the raw neural network confidence by 3.28% with a total area under ROC of 84.33% on the test set.

An even higher advantage of *Bayesian-I3D* is observed in the case of general activity recognition (*HMDB-51* and *UCF-101* benchmarks). All versions of our model clearly outperform the conventional methods with a ROC-AUC gain of over 7% on both datasets. Along our model variants, uncertainty-based selective voting (*Informed Democracy*) has proven to be the most effective strategy, outperforming the *Dictator* by 5.5% and 1.4%, while *Uninformed Democracy* achieved second-best results. We believe that smaller differences in performance gain on the *UCF-101* data are due to the much higher supervised classification accuracy on this dataset. Since the categories of *UCF-101* are easier to distinguish visually and the confusion is low, there is more agreement between the neurons in terms of their confidence.

Method	Balanced Accuracy		Normal Accuracy	
	Acc [%]	±*SE*	Acc [%]	±*SE*
Conventional Methods and Baselines				
Random Chance	4.00	–	4.00	–
One-class SVM	24.49	±8.68	54.78	±7.90
Gaussian Mixture Model	23.61	±6.82	62.39	±5.98
Conventional NN Confidence	44.02	±5.16	74,83	±6.86
Deep Models based on Bayesian Uncertainty				
Bayes. I3D – Dictator	49.31	±9.08	75.55	±6.40
Bayes. I3D – Uninformed Democracy	48.94	± 7.24	77.78	± 4.36
Bayes. I3D – Informed Democracy	**57.55**	±9.54	**77.62**	±4.55

Table 14: Accuracy for the multi-class recognition with an *unknown* categories on the *Open-Drive&Act* dataset (24 known classes + unknown). Normal accuracy is the recognition rate across all test samples, while balanced accuracy is the mean of the accuracy for each individual category.

5.4.2 *Open Set Multi-class Recognition*

Our next area of investigation is the open set multi-class recognition, where we use accuracy as our evaluation metrics and treat *unseen* as an additional label (Table 14). Uncertainty-based selective voting outperforms other approaches in both metrics with a remarkably strong lead in the balanced accuracy. This reflects, that the important question is not only *"how do we distinguish between known and unknown?"*, but also "*if we reject a known class by mistake, are we missing out an otherwise correctly predicted sample or a misclassification?*". In case of selective voting, such false positives are usually samples, which would have been incorrectly classified by I3D anyway. Their incorrect categorization as *unknown* is therefore not very damaging, and oftentimes even practical. Of course, open set recognition is a harder task and the balanced accuracy is lower than in the closed set case (see our these results in Section 3.1.5). Still, 77.62% of the test examples are correctly classified (57.55% after balancing), which is significantly higher than the random baseline of 4% for 25 categories (24 plus the *unknown* class).

5.5 CHAPTER CONCLUSION

Digitalizing human actions has strong potential to make driving more convenient and safe but requires models which can handle a constantly changing world, as unforeseen situations may occur at any time. In this chapter, we introduced the new task of open set activity recognition, which extends driver observation [121, 141, 218] and standard video classification [17, 89, 190] with presence of previously unseen behaviors. We enriched our *Drive&Act* dataset with open set splits and formalized evaluation protocols in our *Open-Drive&Act* benchmark. To broaden the impact of our work, we have also introduced the open set versions of the popular *HMDB-51* and *UCF-101* datasets for general activity classification. To tackle the proposed task, we implemented a generic pipeline for open set driver activity recognition, which combines modern closed set 3D CNNs with an additional component for quantifying the input newness. Besides the versions of this novelty detection module based on already existing novelty and outlier recognition techniques, we propose *Bayesian-I3D* – a new voting-based model for novelty detection leveraging Bayesian uncertainty approximation via Monte-Carlo dropout. Our extensive evaluation reveals

clear benefits of the uncertainty-based models for open set recognition – a vital step for applications of such algorithms in real-life driver monitoring systems.

This chapter exposes the *driver-* and *general* activity recognition to open set conditions for the first time and has three main scientific contributions:

Contribution 1 : A new *open set activity recognition* task and creation of the corresponding benchmarks (*Open-Drive&Act* and open set versions of *HMDB-51* and *UCF-101*).

Contribution 2: A generic framework for open set activity recognition, combining action recognition CNNs with different versions of a novelty detection module.

Contribution 3: *Bayesian I3D* – a new voting-based model for novelty detection in action recognition based on epistemic uncertainty of the output neurons approximated via MC-dropout.

While our evaluation considers all novel activities as a single *unseen* category, methods for transfer from external sources (*e. g.* via zero-shot learning [173]) would potentially allow us to distinguish different *unknown* behaviours among each other. Such prediction of *unknown* activities by distilling knowledge from sources other than task-specific annotated datasets marks an important direction for future research and therefore lays the foundation for our next chapter.

6 RECOGNITION OF *UNKNOWN* ACTIVITIES

Despite extraordinary success in almost every area of computer vision, the amount of annotated training data remains the everlasting Achilles' heel of deep neural networks. Previous chapters allowed us to draw the line between *the known* and *the unknown*. But what can we do, if we are continuously detecting a certain novel behaviour and want to recognize it or the data distribution has changed? In this chapter, we examine ways of recognizing such new activities without any manual annotations – recognizing *the unknown*. We follow two strategies: 1) "webly"-supervised learning by querying Youtube for videos containing the desired activities and learning from natural dialogs (Section 6.1, under review at *IV 2021* [171]) and 2) knowledge transfer from external models using word vectors, where we consolidate our generalized zero-shot learning framework published in *BMVC 2018* [168] with our publication at *SiVL 2018* [173] targeting zero-shot transfer from external datasets (Section 6.2). We further address the problem of domain divergence, where we *do have* annotated training data for the desired actions in a *source* domain, but high uncertainty is present as the input data belongs to an *unknown target* domain, *i. e.* due to changes in camera type (Section 6.3, based on our *IV* 2020 paper [161]). In Section 6.4 we review the main ideas and contributions of this chapter.

6.1 MANEUVER PREDICTION BY LEARNING FROM DRIVING EXAM DIALOGS

This section is based on our paper submitted for review to *IV* 2021 [171].

In this section, we explore the potential of "webly"[1]-supervised learning of new concepts. We specifically focus on the *driver observation* setting, where manually annotated and clean datasets are regarded as the default starting point [1, 21, 34, 71, 116, 121, 141, 142, 214, 218, 226]. For example, existing research on future maneuver prediction through driver observation is often restricted to a static sensor setup or detecting a single maneuver type [34, 71]. Applications of such methods at a large-scale are mostly hindered by expensive data collection requiring accurate temporal localization of the events. *Can we skip expensive domain specific annotations?*

1 "Webly"-supervised learning is a type of weakly supervised learning, where the data and *loose* annotations are automatically collected by crawling the Web, as defined in [19].

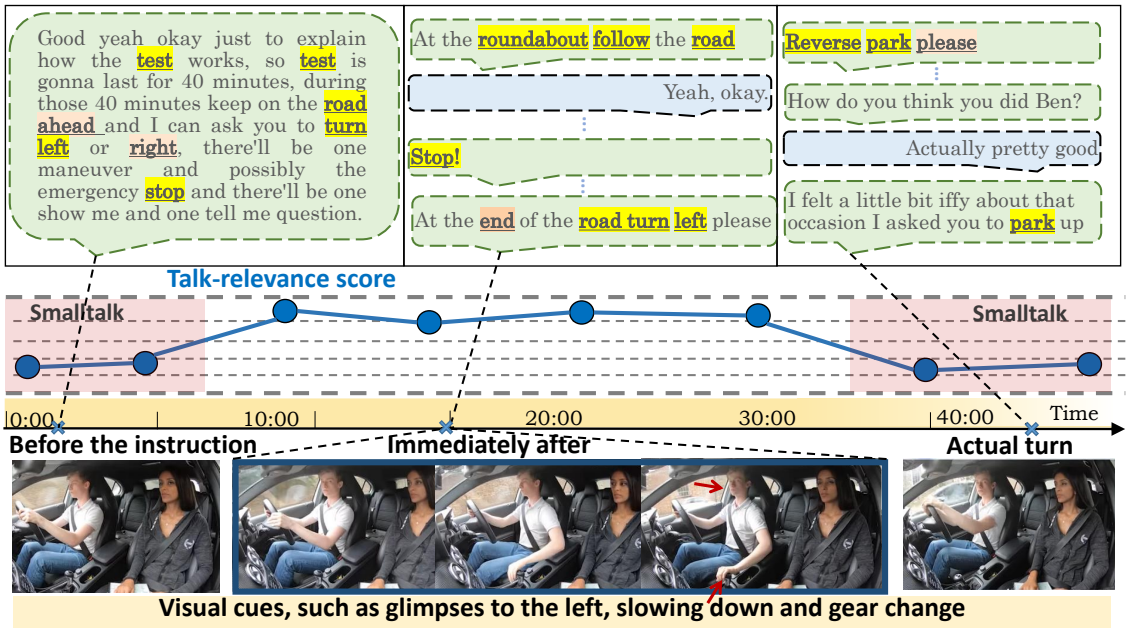


Figure 35: Example of a driving exam dialog. The teacher verbally directs the student driver, triggering an imminent behavioral reaction. We leverage such conversations to learn common visual cues preceding the maneuver. To mitigate adverse effects of unrelated casual talk, we propose a technique for rating segment relevance based on the likelihood of spoken words being present in everyday conversations compared to our setting.

The intersection of vision and language allows us to leverage weak labels inherently present in social media, where multimodal posts hold an enormous amount of information about objects, activities and concepts of our daily life. One example of such content are mock driving exams often publicized on YouTube by driving schools in order to give advice to a wider audience. Such tests build on active student-teacher interaction, where the instructor gives verbal directions about the next action to take (Figure 35).

We view conversations happening during driving sessions as an unprecedented opportunity for connecting speech to changes in human action and present the first framework for anticipating driver intent without a single manually labeled example. We collect a multimodal dataset of mock driving exams by querying YouTube and use conversation transcripts to learn characteristic visual cues preceding certain maneuvers. A driving exam usually starts with casual talk and the teacher explaining the procedure, followed by the driving session, where the student is verbally directed about the next action (for example, *turn right at the traffic lights*) and is concluded by further everyday conversation and feedback. While a large portion of such road tests contain the teacher giving concise directions, a challenge arises from the unrestricted casual talk which is also common. To mitigate its adverse effects, we propose a technique to detect such smalltalk dialogs by relating the odds of words spoken during driving conversation to their likelihood in everyday speech. We empirically analyze the dialogs and use frequency and domain-distinctiveness of used terms to derive seven maneuvers which we aim to predict. As our visual model, we adopt multiple video classification architectures, which we train using ten second videos immediately following the request. While our experiments reveal that visually recognizing human intent through dialog supervision is a challenging task, all evaluated models

Figure 36: Snapshot examples of driving exams video recordings we have collected.

surpass the random baseline by a large margin, while *learning from less but better data* with our smalltalk refinement consistently leads to better recognition.

While recent works use social media content as weak labels [13, 22, 124], our dataset offers a distinct setting where a *dialog* between two people has an imminent effect on *future actions*. Besides, we are the first to use *weak dialog supervision* for driver maneuver prediction, resulting in our dataset being considerably larger and less restrictive than previous work bound by the cost of manual annotation[2]. For example, [71, 117] predict lane changes and turns, but consider a fixed camera view, while others focus on a single maneuver or use a simulated environment [34, 219]. Due to the free nature of web content, our dataset covers diverse views, people (almost all recordings have different drivers) and situations. While we focus on seven common events in our evaluation, new maneuvers can be added by issuing suitable dialog queries.

Besides the applications inside the vehicle, our dataset of driving exam dialogs represents an interesting new avenue for research of speech and multimodality. Our environment is unique as the student-instructor dialogs trigger an immediate visual response, therefore, opening an excellent opportunity for connecting vision and language to actions.

6.1.1 *Web Mining Mock Driving Exams*

We aim to unveil the task of visually foreseeing future human actions by learning from naturally occurring dialogs and introduce a new dataset of driving exam conversations. We have issued YouTube queries with terms such as “mock driving exam” or “road test”, and used a publicly available API to collect the data. The only restriction we made is bypassing videos where the YouTube preview shot indicates that they obviously do not focus on humans, as we aim to study human behavior-only.

Our dataset covers both: 1) transcripts of conversations between the driver student and the examiner and 2) visual recordings inside the vehicle cabin (diverse views, see Figure 36). In 98 cases the transcripts were available through the YouTube API, while for the remaining recordings we used the *autosub*[3] library for automatic speech recognition. Although 120 sessions were initially collected, 14 were omitted as our smalltalk detection technique (described in Section 6.1.2) indicated that they did not contain any relevant conversations (*e.g.* the teacher giving the student tips

2 The largest public dataset for maneuver prediction through driver observation has 700 events [71], we cover over 4K examples, although our labels are noisier.

3 github.com/agermanidis/autosub

Figure 37: *What are people talking about during driving exams*? Domain-salient dialogs visualized, word size highlights term occurence frequency.

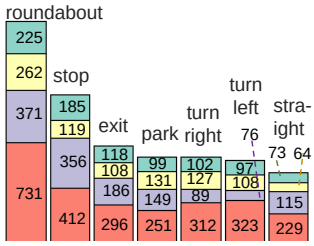


Figure 38: Statistic of dialog lines containing maneuver commands by split. Colors: **train_refined**, **train_smalltalk**, **val**, **test**

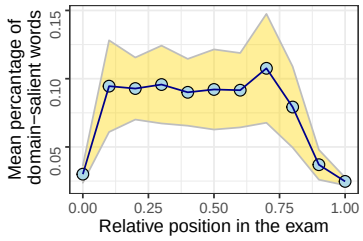


Figure 39: Proportion of domain-salient words for different relative positions in the exam; mean (blue points) and variance (yellow area).

without any actual driving). In Figure 36 we provide multiple examples of snapshots taken from our dataset recordings. The examples cover the driver and, in certain cases the driving instructor and have a variety of views due the unrestricted nature of videos posted online .

6.1.2 *Detecting Smalltalk*

Everyday conversations, such as exchanging pleasantries and giving feedback are common at the beginning and end of a driving exam. Such casual dialog, which may also occur in the middle, introduces additional noise and is presumably unfavorable for both, constructing a fair evaluation set and training the model [82]. While the structure of smalltalk is different from the driving conversations, they still may contain keywords of maneuvers we may want to recognize therefore adversely affecting the model. For example, if we want to recognize drivers searching for a parking spot, the feedback line "*I felt a bit iffy about that occasion I asked you to park*", which contains the word *park* is not helpful, as visuals do not match the mentioned action.

To meet this challenge without additional annotations, we propose a simple yet effective preprocessing technique for detecting smalltalk. Conceptually, our method comprises two steps: 1) computing *domain-salient words i. e.* terms which are distinctive for driving dialog, and 2) determining *domain-salient dialog segments* by estimating a score $s^\star(m, d)$ for every minute m of every dialog d based on the pace of domain-salient words inside the segment and its neighbor segments. The method is described in detail and formalized in Algorithm 2.

PREPROCESSING AND WORD OCCURRENCE PROBABILITIES First, we set all words to lower case, remove common English stop words and compute the vocabulary $\mathcal{V}_{drive}$ of all words that appear in the exam conversations ($|\mathcal{V}_{drive}| = 9492$). In a similar fashion, we set up $\mathcal{V}_{norm}$ with words spoken during movie dialogs, obtained from [26] as our reference for everyday speech. We therefore explicitly distinguish between the *normal* and the *driving dialog corpora*. For each word $w_i \in \mathcal{V}_{drive}$ we compute $p(w_i|norm)$ and $p(w_i|drive)$, which are probability estimates of each word being present in a *normal* dialog sentence or *driving* corpora. To achieve this, we treat *dialog lines as documents* and compute the *document-term matrix* to model such probabilities.

COMPUTING THE DOMAIN-SALIENCY SCORE OF WORDS Next, we aim to quantify the *domain-saliency* of a word, *i. e.* , how specific a certain term is to our *driving domain*. When designing such metric, we aim for words which (1) are *common* in driving conversations and (2)

Algorithm 2 Detecting Smalltalk

Input: $\mathcal{W}_{d,m}$ – words spoken at minute m of a dialog d; $\mathcal{V}_{drive}$ – driving vocabulary; $p(w_i|drive)$ and $p(w_i|norm)$ – line occurrence probabilities for driving and normal settings (computed beforehand for all $w_i \in \mathcal{V}_{drive}$); parameters α (salient word threshold); r (window range); ϵ (for num. stability) and $\mathcal{V}^{\star}_{drive} := \emptyset$.
Output: region-saliency scores $s^{\star}$ for each exam minute
// Compute domain-salient word set $\mathcal{V}^{\star}_{drive}$
for all $w_i \in \mathcal{V}_{drive}$ **do**
  $f(w_i) := p(w_i|drive) \cdot \log\left(\frac{p(w_i|drive)}{p(w_i|norm)+\epsilon}\right)$
  if $f(w_i) < \alpha$ **then** $\mathcal{V}^{\star}_{drive} := \mathcal{V}^{\star}_{drive} \cup \{w_i\}$
  end if
end for
// Compute domain-salient dialog regions
for all minutes m in all exam dialogs d **do**
  $s(m,d) := \frac{|W_{d,m} \cap \mathcal{V}^{\star}_{drive}|}{|W_{d,m}|}$
end for
// Connect the neighboring regions via sliding window
for all minutes m in all exam dialogs d **do**
  $s^{\star}(m,d) := \sum_{i=m-r}^{m+r} s(m,d) \cdot \frac{1}{2r+1}$
end for

are *specific* to this context. In information retrieval systems, these requirements are often met by the *Term Frequency – Inverse Document Frequency (tf–idf)* technique [164, 165] which quantifies how important a word is to a document in a collection of documents by weighting its frequency in a document by inverse document frequency of the word across a set of documents. We cannot use *tf-idf* as-is due to a sligtly different setting (we want do detect domain-relevant lines and therefore compute word occurrences in a dialog line, treating lines as documents, while having *two* distinct corpora). Inspired by *tf-idf* we derive a very similar metric to obtain the domain-saliency of a word in our setting. We compute the domain-saliency score $f(w_i)$ as the likelihood of w_i occurring in a driving conversation sentence weighted by the logarithm of ratio by which the probability estimates increases if the conversation has driving context:

$$f(w_i) = \underbrace{p(w_i|drive)}_{\text{How likely in a driving situation?}} \underbrace{\cdot \log\left(\frac{p(w_i|drive)}{p(w_i|norm)+\epsilon}\right)}_{\substack{\text{How specific to a driving situation?} \\ \text{Log used to soften the effect esp. if } p(w_i|norm)=0}}, \tag{18}$$

where ϵ is an arbitrary small constant added for numerical stability. Note, that $p(w_i|drive)$[4] is the precomputed probability estimate of the word being present in a *driving* dialog sentence

4 Our $p(w_i|drive)$ would approximately correspond to the term-frequency in the *tf-idf* perspective, although *tf-idf* would deal with counts in the document while we view lines as documents and estimate probability of a word being present in a line over the complete driving corpus.

Word	Probability of occurence	Probability increase	Domain-saliency score
Top 5 with highest occurence probability in a driving dialog line			
right	6.9	2.58	0.07
just	6.75	1.52	0.03
yeah	6.53	2.13	0.05
okay	5.13	2.41	0.05
left	4.92	12.43	0.12
Top 5 with highest *increase* of line occurence probability estimates in case of driving context compared to everyday dialog			
road	3.05	60.93	0.13
roundabout	2.53	50.54	0.1
exit	2.14	42.85	0.08
lane	1.12	22.41	0.03
yards	1.03	20.63	0.03
Top 5 words with highest domain-saliency score			
road	3.05	60.93	0.13
left	4.92	12.43	0.12
roundabout	2.53	50.54	0.1
turn	3.93	13.23	0.1
exit	2.14	42.85	0.08

Table 15: Word statistics in our driving exam conversations dataset by different metrics. The proportion of used domain-salient words (*i. e.* expressions with high domain saliency-score) serve as the basis for determining whether a dialog region is relevant or not.

and $\frac{p(w_i|drive)}{p(w_i|norm)}$ [5] signals the significance of the word in the driving domain, as it depicts by how much the occurrence probability increases if the conversation happend during driving. An additional logarithm transformation is used to "soften" the effect of the term $\frac{p(w_i|drive)}{p(w_i|norm)}$ as it otherwise would become very high for words which have never been observed in the normal corpus (this technique is also often used in *tf-idf*) [165].

We then refer to a word as *domain-salient* if its relevance score surpasses a threshold α (set to 0.002 producing ~ 150 domain-salient words). Examples of such terms are *roundabout* and *exit*, which occur in *driving* conversations 51 and 43 times more often.

IDENTIFYING SMALLTALK SEGMENTS To identify smalltalk segments of a dialog (we use on minute temporal resolution), we rate these one minute segments as the *proportion of the used domain-salient* words and connect the neighboring regions via sliding window smoothing (window range $r = 5$). We view a segment as smalltalk, if $s^\star(m, d) < 0.05$, *i. e.* the portion of domain salient words is less than 5% (after the region smoothing).

In Table 15, we shed light on our smalltalk refinement step by listing the top five most common words in comparison to the expressions with highest computed saliency score and the highest increase of the probability estimates in case of driving contex. The proportion of such

5 Our $\frac{p(w_i|drive)}{p(w_i|norm)}$ resembles the *idf* term in the *tf-idf* perspective. In contrast to *tf-idf*, we use probability estimates in a dialog line and compare the driving and the normal corpora, while in *tf-idf* the frequency over all documents would be used in the denominator.

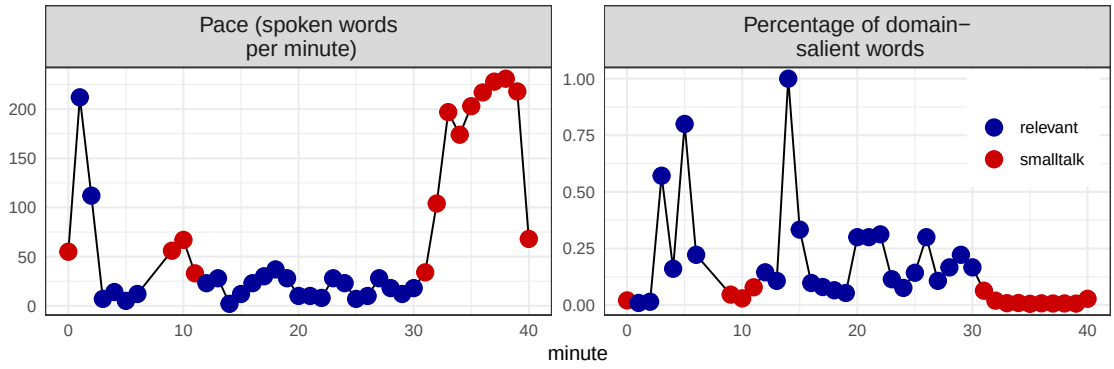


Figure 40: Detected smalltalk regions (red) for one driving session example. The x-axis depicts time (in minutes), while the y-axis is the speech pace (words per minute) for the left graph and the percentage of used domain-salinet words for the right graph. High pace (right) and rare usage of domain-salient words (left) are characteristic for smalltalk conversations.

domain-salient words is then used to determine, whether a dialog region contains a relevant- or a smalltalk conversation. In Figure 40, we further illustrate an example of the detected smalltalk regions for one session, in relation to the speech pace and the percentage of domain-salient words.

6.1.3 *Dialog Analysis and Split Statistics*

Using driver exam conversations as weak annotations allows us to query the specific maneuver type. For example, if we want to recognize searching for a parking spot or exiting the highway, we might look for terms such as *park* or *exit* in the dialogs. While the set of possible maneuvers is dynamic *i. e.* events can be added on-demand by issuing corresponding requests, we need to fix a category set for the evaluation. To achieve this, we took into consideration the terms with the highest domain-saliency score (see Table 15), as well as, studies of maneuver impact on accident odds [188]. Finally, we inferred seven maneuvers: *stop*, *exit*, *park*, *turn right*, *turn left*, *straight* and *roundabout*.

Our dataset is split into *train*, *val*, and *test* sets with a 6:2:2 ratio of exams. As we always desire clean test data, *val* and *test* contain non-smalltalk dialog segments only. The *train* dialogs cover both, *train_smalltalk* and *train_refined*, which are regions that our approach marked as smalltalk or non-smalltalk. As our training data, we compare two options: all dialogs *train_refined* ∪ *train_smalltalk* and domain-salient *train_refined* only. Sample statistics by category and split, *i. e.* the number dialog lines containing the seven target commands, is provided in Figure 38.

The collected 106 recordings last 62.6 min on average (ranging from 7 to 120 min.). The mean speech pace of 71.9 words per minute (wpm) is significantly higher for the detected smalltalk (96.7 wpm) and lower for the instructional dialogs (29.6 wpm). Frequently used terms are illustrated in Figure 37: driving-related expressions (*i.e. road*, *turn*) overshadow the dialogs, while certain casual terms accompanying friendly request-response conversations are also common (*e.g. please*, *thanks*). The average proportion of domain-salient words spoken in a dialog line (7.8% overall) is, unsurprisingly, higher for regions we estimate to be relevant (18%) and lower for smalltalk (only 1.8%). There is also a connection between the point in time and the speech relevance (Figure 39).

Model Type	Smalltalk refined	Top1 Acc Bal.	Top1 Acc Unbal.
Random	–	33.33	33.33
CNNs without training set refinement			
C3D	no	47.62	44.02
Pseudo 3D Resnet	no	45.40	43.73
Inflated 3D Net	no	50.17	**55.43**
CNNs with training set smalltalk deletion			
C3D	yes	51.92	54.89
Pseudo 3D Resnet	yes	50.63	54.35
Inflated 3D Net	yes	**53.42**	**55.43**

Table 16: Recognition results for the three-maneuver-setting (classes *straight*, *exit* and *stop*).

Model Type	Smalltalk refined	Top1 Acc Bal.	Top1 Acc Unbal.	Top3 Acc Bal.	Top3 Acc Unbal.
Random	–	14.29	14.29	42.86	42.86
CNNs without training set relevance refinement					
C3D	no	28.15	23.11	61.38	61.19
Pseudo 3D Resnet	no	22.43	20.32	56.87	56.93
Inflated 3D Net	no	32.76	34.79	64.18	69.34
CNNs with smalltalk deletion in the training set					
C3D	yes	31.09	30.78	64.61	67.15
Pseudo 3D Resnet	yes	31.68	33.21	61.8	67.64
Inflated 3D Net	yes	**36.05**	**39.66**	**65.64**	**70.56**

Table 17: Results for all seven maneuvers. Smalltalk refinement improves while models and Inflated 3D Net performs the best.

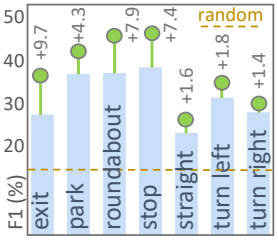


Figure 41: Per-category I3D F1 score with (green) and without (blue) smalltalk refinement.

6.1.4 *Visual Model*

We use 10 second videos right after the teacher's request, to learn visual cues preceding the specific maneuvers. For our visual models, similar to Section 3.1.4, we implement three approaches based on spatiotemporal CNNs, initially developed for activity recognition: C3D [196], Inflated 3D ConvNet [17] and Pseudo3D ResNet [153]. Section 3.1.4 already covers the details of these video classification architectures. All models were trained using stochastic gradient descent with momentum and cross entropy loss.

6.1.5 *Experiments*

We use balanced accuracy (mean accuracy over all classes) as our main metric and, additionally, report the unbalanced accuracy and the $F1$-score (harmonic mean of precision and recall) for the individual classes. To determine, whether it is better to learn from more but noisier data or from a smaller amount of highly relevant data, we consider two training settings: using complete training set dialogs or limiting them to the non-smalltalk regions (*i.e.* skipping around 35% of the data where a maneuver term was present, but, presumably in a non-driving context). We learn model parameters on *train*, select checkpoints and hyperparameters on *val*, and present final evaluation on *test*.

We report the results for a simpler setting with three maneuvers in Table 16, then move to a harder task with seven distinct events in Table 17 and, finally, examine the performance for individual classes in Figure 41. While the Inflated 3D Net is a clear frontrunner (53.42% for three and 36.05% for seven categories), *learning from less but better* data through our smalltalk refinement improves the recognition for all architectures. While it is evident that learning visual cues of the intended maneuvers guided only by exam dialogs is a hard task, all models outperform the random baseline by a large margin.

6.2 KNOWLEDGE TRANSFER WITH LANGUAGE-BASED MODELS

This sections consolidates and extends the generalized zero-shot learning part of our BMVC 2018 publication [168] and our ECCV SiVL Workshop 2018 publication on cross-dataset knowledge transfer, © Springer.

"Never memorize something that you can look up."
– Albert Einstein

Humans learn different from modern recognition algorithms. While deep CNNs have surpassed human performance in many computer vision tasks [14, 60, 134], they struggle when it comes to learning from very few examples. Humans, on the other hand, are excellent in transferring learned concepts to new categories. Furthermore, we are able to recognize new examples without any training data, if a suitable *description* allows us to link this class to the already existing knowledge. For example, if a human has never seen someone *riding an elephant*, but is familiar with *riding a bike* and the animal *elephant*, connecting the dots and categorizing the event without any explicit training data is not a problem at all.

In visual recognition, such generalization to new examples without any training data is referred to as *Zero-Shot Learning (ZSL)*. Zero-shot action recognition aims to classify actions not previously seen during training by building a visual model for the *seen* classes and establishing an association to the *unseen* classes through a high-level semantic description, *e. g.* the action labels. The description is often represented with word vectors and a skip-gram model (e.g. *word2vec*[125, 126]), previously trained on web text data. ZSL has many flavours, but, in general, most approaches follow a similar paradigm: the model would first compute the word vector by mapping a visual representation of a new example to the common semantic space and then assign it to one of the previously unseen categories by finding a category with the closest semantic representation (see overview in Figure 42).

ZSL for action recognition gained popularity over the past few years and has also been improving slowly but steadily, usually dividing the dataset into *seen* categories for training and *unseen* categories for evaluation[152, 168, 205, 215, 216]. In all of these works, however, all classes used for testing are *unseen*, so that none of the *seen* classes used for training the visual model are present in evaluation. This restriction is a bottleneck in many applications, where both, known and novel behaviours might occur. A more challenging and at the same time more realistic task of *Generalized* Zero-Shot Learning (GZSL) has not been considered in the field of activity recognition prior to this work but is increasingly studied in other areas of computer vision, such as object detection [18, 212]. In GZSL, the evaluation is conducted on both, unseen and seen classes, so that the goal is to maximize the performance across both sets of classes (*i.e.* it can bee seen as an open set extension of the standard ZSL). Prominents ZSL algorithms, such as ConSE [139] and Devise [42] are capable of GZSL by design, but a significant performance decline on the unseen classes has been reported in the generalized setting [212]. Specifically, if both seen and unseen categories are treated equally, off-the-shelf ZSL approaches have a very strong bias towards the known classes.

Figure 42: Zero-shot action recognition paradigm: instances of the new *unseen* classes are recognized without any training data by linking visual features learned from the *seen* categories with a language-based representation of the action labels.

In our work, we extend the task of zero-shot action recognition to the generalized case, which, to our best knowledge, has not been studied yet[6]. We implement a framework for generalized zero shot action recognition by combining two standard ZSL approaches with our novelty detection method based on Bayesian uncertainty (Section 5.3) to distinguish between *known* and *unknown* actions, leading to a significant improvement in the recognition results.

Furthermore, as a side-contribution, we loosen the assumption, that the seen and test classes originate from the same dataset. Motivated by the emergence of large-scale datasets, the idea of cross-dataset ZSL, where the model built from a high amount of external data is classifying examples from a smaller, potentially application-specific dataset, becomes more realistic [229]. We therefore examine such transfer in the ZSL context and formalize the multiple evaluation regimes for incorporating such external knowledge. We demonstrate, that zero-shot action recognition benefits immensely from cross-dataset transfer, but certain precautions must be considered in evaluation to satisfy the ZSL premise of disjoint seen and unseen classes.

6.2.1 *Problem Definition*

We start by formalizing the zero-shot and the generalized zero-shot learning tasks. Let $\mathcal{A}^{seen} = \{a_1^{seen}...a_m^{seen}\}$ be a set of m previously seen action categories with the available labelled training data (*i.e.* we can train a visual classifier). Given the set of previously unseen categories $\mathcal{A}^{unseen} = \{a_1^{unseen}...a_n^{unseen}\}$ and a new data sample $\mathbf{x}$, our goal is to predict the correct unseen action category $a_{true} \in \mathcal{A}^{target} = \mathcal{A}^{unseen}$ without having any training data (*i. e.* labeled visual ex-

6 One work introducing a similar framework [111] has been published one year *after* our *BMVC 2018* [168] publication presenting this work.

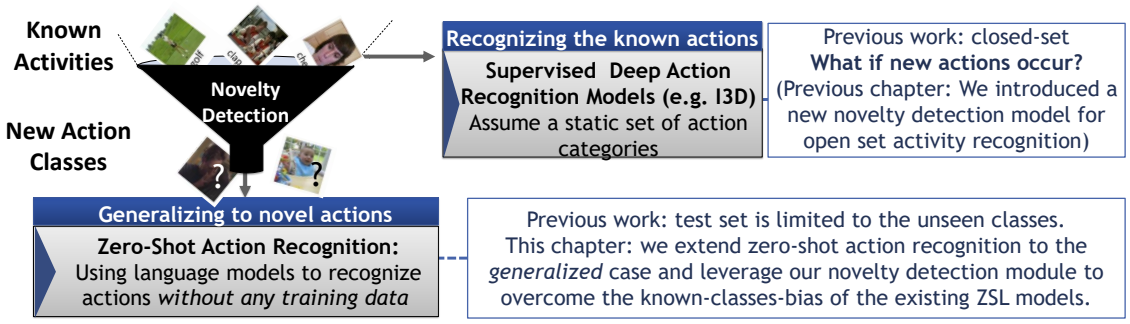


Figure 43: Generalized zero-shot action recognition framework. We address the task of generalized zero-shot action recognition for the first time, where test samples can belong to either *unseen* or *seen* categories. We leverage our novelty detection algorithm to draw the line between the *seen* and the *unseen*, therefore mitigating the effect of the known-classes-bias.

amples) for this class (*i. e.* we can only use the action labels as their semantic description). The set of the target test categories $\mathcal{A}^{target}$ is therefore restricted to the unseen classes only. Since the core idea of ZSL is to recognize unseen visual categories, source labels and target labels are set to be strictly disjoint. This is known as the *zero-shot premise* and is formalized as: $\mathcal{A}^{unseen} \cap \mathcal{A}^{seen} = \emptyset$.

The harder but more realistic task of *generalized* zero-shot action recognition, encompasses the same training setup, but the evaluation resembles an open set scenario, where the new examples may be both, known or unknown: $a_{true} \in \mathcal{A}^{target} = \mathcal{A}^{unseen} \cup \mathcal{A}^{seen}$. Note, that this does not affect the zero-shot premise, *i. e.* the sets of seen and unseen categories are disjoint, while our model is not restricted to any group of target labels and is evaluated on seen and unseen categories. The standard evaluation for the GZSL is the *harmonic mean* of accuracies for seen and unseen classes [212].

6.2.2 *Generalized Zero-Shot Action Recognition*

We adress the task of generalized zero-shot action recognition for the first time and present a framework based on our novelty detection module introduced in the previous chapter (Section 5.3). The main purpose of our novelty detection algorithm is to reduce the effect of the inherent bias towards the seen action classes, which is often observed in GZSL research from other fields [212]. It therefore serves as a filter which distinguishes whether the observed example should be classified with the visual model in the standard classification setup, or mapped to one of the unknown classes via a ZSL model (Figure 43).

ZSL connects a visual model trained on a dataset of known (*source*) classes to the unknown (*target*) classes through the high-level semantic information about an action. Such additional information might be given in form of attributes, textual descriptions, or action labels. We consider the latter case of category labels, as it does not require any additional supervision except for the name of the behaviour we want to recognize. Conceptually, our framework comprises three neural networks: (1) a visual model, trained to distinguish between the *seen* classes, (2) a language model, for embedding the language labels used to associate the unseen class to the previously learned seen concepts and (3) our novelty detection module, which differentiates between the seen and the unseen classes.

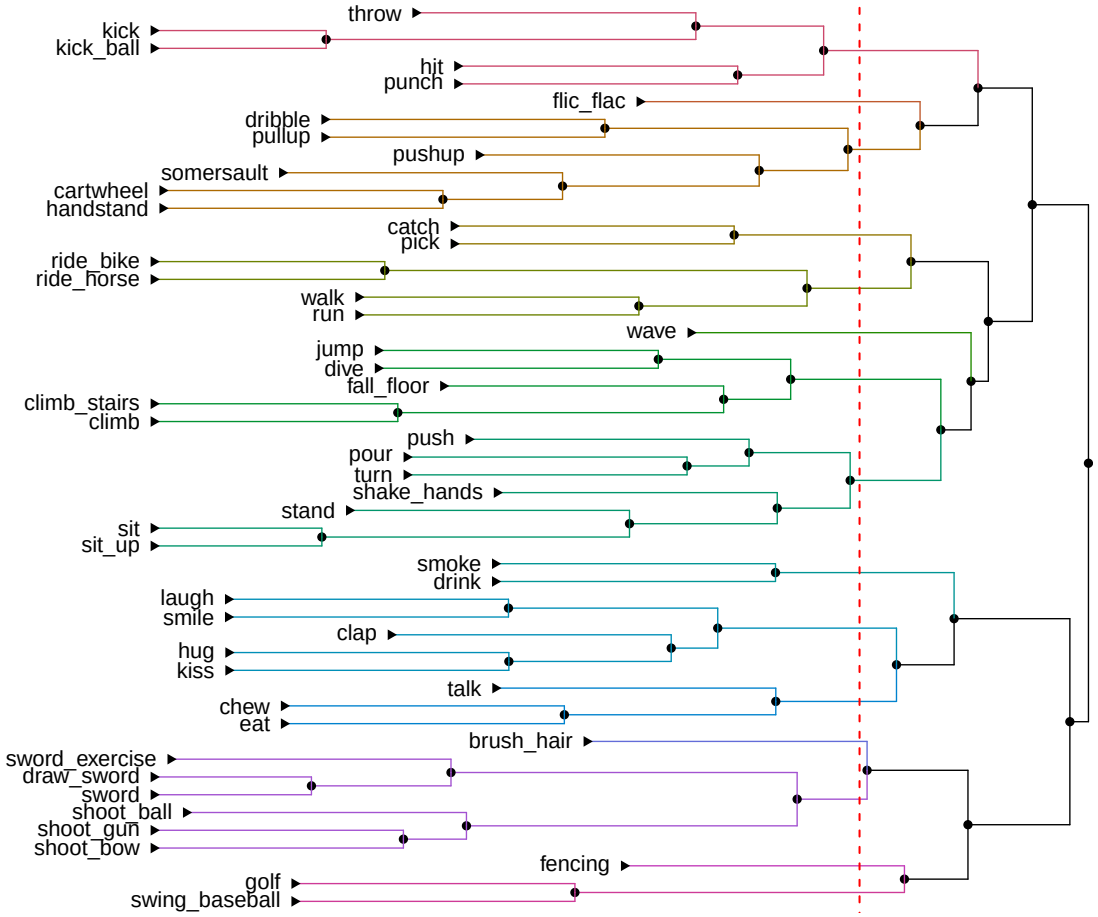


Figure 44: Clustering representation of the label embeddings. Ward's Hierarchical Agglomerative Clustering method [207] of the activity labels using *word2vec* (HMDB-51 dataset).

6.2.2.1 *Language Model: Representing Activities with Word Vectors*

We represent the label description with word vectors (in our case *word2vec*[126]) – a skip-gram model, previously trained on web data. *Word2vec* is a shallow two-layer network trained to predict the neighbouring words of a given context word. Words are represented as a one-hot-encoding, so that the goal of the first layer is to embed them into a more efficient representation. The second layer predicts the target words from this intermediate embedding using hierarchical Softmax. This intermediate embedding, referred to as *word2vec* representation $\omega(\cdot)$, is widely used to quantify how two words relate, with cosine similarity (*i. e.* the normalized dot product) being the most wide-spread metric. High cosine similarity of two *word2vec* representations means that the underlying words often appear in a similar context. To compute the word vectors embeddings of the action categories, we use the publicly available *word2vec* model trained on 100 billion words from Google News articles, which maps the input into 300 dimensional semantic space [126][3]. In case the label comprises multiple words, we average their embeddings.

The main idea of zero-shot learning, is to leverage the correlation between the similarity in the language model space and the visual model space, *i. e.* the concepts which are similar in terms of language tend to also be visually close. We now examine how different word2vec representa-

3 *word2vec* model obtained from: https://code.google.com/p/word2vec/.

tions of the HMDB-51 [89] activity labels are connected in the language space. First, we compute the *word2vec* representation of each label and its pairwise cosine similarities to the other action names. We then apply the Ward's Hierarchical Agglomerative Clustering method [207] on the resulting vectors containing the cosine similarities. The resulting class hierarchy, illustrated in Figure 44, reveals how the classes are connected in the language space. Many of the action relationships uncovered by the language model would also make sense for visual recognition. For example, *hit* and *punch* or *laugh* and *smile* are closely linked.

6.2.2.2 *Visual Model and Zero-Shot Inference Algorithm*

As the backbone of our visual recognition model, we use I3D [17], which consistently remained the frontrunner in our previous chapters. We consider two prominent ZSL methods: ConSE [139] and DeViSE [42], which are explained in the following.

CONVEX COMBINATION OF SEMANTIC EMBEDDINGS (CONSE) The first ZSL approach we integrate into our framework is Convex Combination of Semantic Embeddings (ConSE) [139]. ConSE starts by predicting probabilities of the seen classes using the previously trained visual classifier, and then takes the convex combination of word embeddings and selects its nearest neighbor from the novel classes in the *word2vec* space. While ConSE has been used for zero-shot action recognition before [216], where the underlying visual model was based on dense trajectory features encoded as Fisher Vector, we employ a visual model based on CNNs (in our case I3D), which we first train to distinguish between the known activities.

For each test sample $\mathbf{x}$, we use a visual recognition model to obtain the probability estimates $\hat{p}(a_i^{seen}|\mathbf{x})$ for each of the known classes $a_i^{seen} \in \mathcal{A}^{seen}$. In the next step, we synthesize a word vector embedding $w^*(\mathbf{x})$ by taking a linear combination of the predicted probabilities and the semantic representation of source classes: $w^*(\mathbf{x}) = \sum_{i=1}^{m} \hat{p}(a_i^{seen}|\mathbf{x})\, \omega(a_i^{seen})$, where $\omega(\cdot)$ is the *word2vec* representation of a category. The input $\mathbf{x}$ will be classified as the category whose semantic representation is most similar to the synthesized word embedding:

$$a_{pred} = \underset{a_i \in \mathcal{A}^{target}}{\operatorname{argmax}}\, sim\left(\omega\left(a_i\right), w^*(\mathbf{x})\right) \tag{19}$$

where $sim(\cdot)$ is the cosine similarity of two vectors. Recall, that $\mathcal{A}^{target}$ differs depending on the task ($\mathcal{A}^{target} = \mathcal{A}^{unseen}$ in the standard case and $\mathcal{A}^{target} = \mathcal{A}^{unseen} \cup \mathcal{A}^{seen}$ the generalized case).

DEEP VISUAL-SEMANTIC EMBEDDING MODEL (DEVISE) The main idea of DeViSE [42] is to train a single-layer projection model to directly regress *word2vec* representations from the visual features. After we have trained the visual classifier using the seen categories, we remove its softmax prediction layer and train a linear transformation g_θ (*i. e.* a single projection layer), to map the I3D representation into our 300 dimensional *word2vec* space. For a new example $\mathbf{x}$, we first do a forward pass with our visual model (without the softmax layer) to obtain the embedding $\mathbf{x}_{emb}$ and then estimate the semantic *word2vec* representation by using the learned projection $g_\theta(\mathbf{x}_{emb})$. The final inference is almost identical to Eq. 19, except for $w^*(\mathbf{x})$, which is replaced with our learned projection into the language space $g_\theta(x_{emb})$.

6.2.2.3 *Experiments*

While there are no existing protocols for GZSL for action recognition, standard ZSL is often evaluated using the *HMDB-51* and *UCF-101* action recognition datasets [205]. We therefore adapt the evaluation framework of [205], to the *generalized* case. Similar to the previous chapter, we evenly split each dataset into seen/unseen categories (26/25 for HMDB-51 and 51/50 for UCF-101). Samples of unseen classes will not be available during training, while samples of the remaining set of seen classes is further split into training (70%) and testing (30%) sets. As in [205], we randomly generate 10 splits and report the average and standard deviation of the recognition accuracy.

Note, that both ConSE and DeVISE are capable of generalized zero-shot action recognition by design but previous research indicates, that raw ZSL methods struggle with strong bias towards the seen classes [212]. We therefore compare both the native ZSL methods and our framework enhancing them with a novelty detection module. Similarly to the novelty detection task of the previous chapter, we compare different novelty detection modules: 1) a One Class SVM [181, 182]; 2) a GMM [150, 230]; 3) *Softmax* probability estimates [62, 163] and 4) our uncertainty-based *Bayesian I3D* method with the informed neurons voting (described in Section 5.3).

For consistency, we first report the results for the standard ZS case (marked as U⟶U) and further extend to the generalized case as shown in Table 18. In the more realistic GZSL setup (marked as U+S⟶U+S), our model is not restricted to any group of target labels and is evaluated on actions of seen and unseen category using the *harmonic mean* of accuracies for seen and unseen classes as proposed by [212]. Additionally, we report the results in semi-generalized setting (marked as U+S⟶U), where our model is allowed to predict both, known and unknown categories, but the evaluation set actually contains only the unknown activities. Table 18 shows a clear advantage of employing novelty detection as part of the GZSL framework. While failure of the original ConSE and DeViSE models might be surprising at first glance, such performance drops have been discussed in previous work on ZSL for image recognition [212] and is due to the fact that both models are biased towards labels that were used during training. Our *Informed Democracy* model yields the best recognition rates in every setting and can therefore be indeed successfully applied for multi-label action classification in case of new activities.

6.2.3 *Knowledge Transfer from External Datasets*

In the previous section, we have extended the conventional zero-shot the activity recognition to the generalized setting. In this section, we aim to study a different application-relevant aspect of zero-shot transfer – leveraging knowledge from *external* datasets. In activity recognition, the zero-shot setting has been achieved by splitting an existing dataset category-wise into seen- and unseen actions [152, 168, 205, 215, 216]. Recent emergence of large-scale action recognition datasets has lead to an increasing interest in the field of domain adaptation and cross-dataset recognition, where the model built from a high amount of external data is classifying examples from a smaller, potentially application-specific dataset [229].

At the first glance, one would assume, that classifying data from a foreign source would be a harder problem because of the potential domain shift. However, recent works using data from foreign datasets for training of the visual recognition model, report extraordinary results in zero-shot action recognition, doubling the performance of the previous models focused on the inner-dataset split [229]. In order to draw a clear line between the *zero-shot* and the conventional *supervised* classification, the *source* and *target* categories must be disjoint (Section 6.2.1). Ensuring

Zero-Shot Approach	HMDB-51			UCF-101		
	U→U	U→U+S	U+S→U+S	U→U	U→U+S	U+S→U+S
Standard ConSe Model	21.03 (±2.07)	0 (±0)	0 (±0)	17.85 (±1.95)	0.07 (±0.10)	0.13 (±0.20)
Standard Devise Model	17.27 (±2.01)	0.26 (±0.37)	0.52 (±0.73)	14.48 (±1.13)	0.81 (±0.36)	1.61 (±0.71)
ConSe + Novelty Detection						
One-class SVM	21.03 (±2.07)	10.99 (±1.83)	17.40 (±2.41)	17.85 (±1.95)	10.37 (±1.59)	16.55 (±1.91)
Gaussian Mixture Model	21.03 (±2.07)	13.30 (±2.58)	19.91 (±3.32)	17.85 (±1.95)	9.31 (±1.30)	15.98 (±1.99)
Conventional NN Confidence	21.03 (±2.07)	10.96 (±0.87)	18.56 (±1.22)	17.85 (±1.95)	12.19 (±1.72)	20.91 (±2.59)
Informed Democracy (ours)	21.03 (±2.07)	**13.67 (±1.31)**	**22.27 (±1.79)**	17.85 (±1.95)	**13.62 (±1.94)**	**23.42 (±2.97)**
Devise + Novelty Detection						
One-class SVM	17.27 (±2.01)	8.92 (±1.89)	14.67 (±2.74)	14.48 (±1.13)	8.65 (±1.59)	14.25 (±2.00)
Gaussian Mixture Model	17.27 (±2.01)	10.61 (±2.22)	16.72 (±3.1)	14.48 (±1.13)	7.26 (±0.84)	12.88 (±1.40)
Conventional NN Confidence	17.27 (±2.01)	8.68 (±1)	15.17 (±1.56)	14.48 (±1.13)	10.08 (±1.59)	17.69 (±2.33)
Bayesian I3D – Informed Voting (ours)	17.27 (±2.01)	**10.73 (±1.47)**	**18.18 (±2.21)**	14.48 (±1.13)	**11.03 (±1.42)**	**19.48 (±2.21)**

Table 18: Generalized Zero-Shot Action Recognition Results. U⟶U: test set consists of unseen actions, the prediction labels are restricted to the unseen labels (standard). U⟶U+S: test set consists of unseen actions, both unseen and seen labels are possible for prediction. U+S⟶U+S: generalized ZSL case, both unseen and seen categories are among the test examples and in the set of possible prediction labels (harmonic mean of the seen and unseen accuracies reported.)

this is not trivial, especially when the source data origin is an external dataset. A single dataset would not contain the same activity twice. Action labels of an external dataset, on the other hand, possibly intersect with the test categories, violating the ZSL premise of assigning action classes not seen during training and turning the problem into supervised classification. We believe that leveraging external knowledge for zero-shot recognition is a key step towards creating global models, but it is also important to draw the line between *zero-shot* recognition and standard *supervised* recognition and take a closer look at the similarity of action categories of source and target data in order to honor the ZSL premise.

In the following, we study zero-shot action recognition in the cross-dataset setting and introduce an evaluation procedure that enables fair use of external data. First, we highlight the evaluation difficulties of such knowledge transfer. We quantitatively analyze the similarities of the *seen* and *unseen* labels in the inner-dataset and cross-dataset setup and demonstrate, that *external* labels tend to have categories very similar to the unseen *target* classes, therefore violating the ZSL assumption of disjoint source and target categories. We then propose a corrective protocol allowing integration of external data for zero-shot action in a *fair* way, using the maximum semantic similarity within the target dataset labels as a restrictive threshold for classes of external origin. Besides, we propose a novel *hybrid* ZSL regime, where the model is allowed to use all the internal labels and additional large-scale external data, consistently increasing the accuracy. We evaluate our method on the HMDB-51 dataset, and show how using external data improves the ZSL performance, even in our more fair evaluation setting.

6.2.3.1 *Evaluation protocols for ZSL*

We extend the task of zero-shot action recognition defined in Section 6.2.1 with the possibility of training data (*i. e.* examples of the seen classes) originating from a dataset different from the unseen categories. Formally, when referring to the set of seen categories $\mathcal{A}^{seen}$, we distinguish

between $\mathcal{A}^{seen}_{intra}$ (categories and their training examples come from the same dataset as the unseen categories) and $\mathcal{A}^{seen}_{cross}$ (categories and their training examples obtained from an external dataset). We now define different zero-shot learning regimes linked to their use of external sources.

INTRA-DATASET PROTOCOL. A common way to evaluate ZSL approaches is to divide a dataset into seen and unseen categories. That is, while a subset of unseen categories is held out during training, both the source and target labels belong to the same dataset: $\mathcal{A}^{seen} = \mathcal{A}^{seen}_{intra}$. In this setting, source and target categories do not overlap and the ZSL premise is satisfied automatically, since well designed datasets contain no duplicated categories. This is confirmed in Figure 45, as the maximum cosine similarity of source and target labels is at most 0.8 in case the categories originate in the same dataset.

CROSS-DATASET PROTOCOL. The long-term goal of ZSL, however, is to apply knowledge from available data to tasks from a different domain where labelled data is difficult to obtain. This setting is evaluated by training on one dataset and evaluating on a different dataset: $\mathcal{A}^{seen} = \mathcal{A}^{seen}_{cross}$. In that case, however, the zero-shot premise is not given by default. In the most extreme case, it might occur that $\mathcal{A}^{unseen} \subset \mathcal{A}^{seen}_{cross}$, where no semantic transfer is needed as the problem becomes standard supervised classification.

HYBRID PROTOCOL: INTRA- AND CROSS- DATASET. Recently, several approaches in other computer vision areas have been presented that investigate ways of increasing the performance by mixing the available domain-specific datasets with large amounts of training data from external sources [155]. We transfer this paradigm to zero-shot action recognition and formalize this *hybrid* evaluation regime as: $\mathcal{A}^{seen} = \mathcal{A}^{seen}_{intra} \cup \mathcal{A}^{seen}_{cross}$. Similarly to the previous setting, the zero-shot premise is not automatically ensured.

6.2.3.2 *Proposed protocol to incorporate external datasets*

In the intra-class protocol, compliance with the zero-shot premise is generally well accepted [152, 205, 215] since *a single dataset does not contain the same category twice*. However, when external datasets are involved, ensuring that we are still within the terms of zero-shot learning has to be taken care of. For example, Zhu *et al.* [229] excludes classes from the training dataset whose category label overlaps with a tested label. This procedure would remove the action *brushing hair*, present in both ActivityNet [15] and Kinetics [17], since the label *brush hair* is present in the target classes of HMDB-51 [89].

It is not trivial to determine if a source class should be excluded and eliminating *direct* category matches may not be enough. External datasets often contain slightly diverging variants or specializations of the target actions (*e.g.*, *drinking beer* and *drink*), leading to a much closer relation of source and target actions compared to the inner dataset protocol, even when excluding direct matches. We argue, that taking into account the similarity of source and target labels is a key element for evaluation of zero-shot action recognition when external datasets are used.

We propose a standardized procedure to decide whether an external class should be used or discarded when training the visual model. Our corrective method is based on the fact that zero-shot learning is well defined for the intra-class protocol, thus all *source* categories of the intra-dataset split can always be used to train our model. We remove source categories if their labels

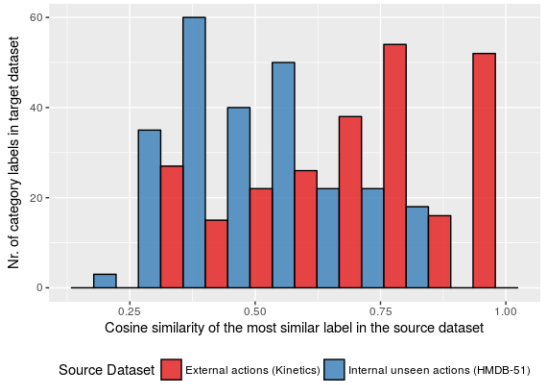


Figure 45: Histogram of pairwise semantic similarities between all *unseen* labels and their most similar *seen* label for external (red) and intra-dataset (blue) seen actions.

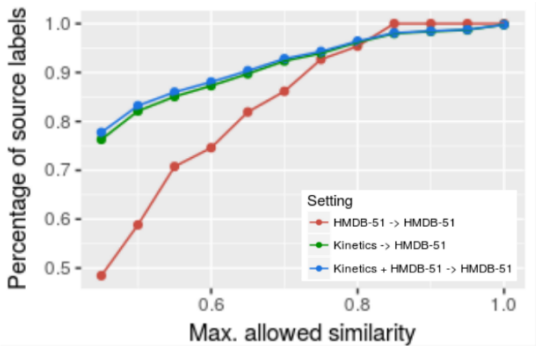


Figure 46: [%] of the allowed seen categories depending on the similarity rejection threshold τ . No labels are excluded after $s_{th} \approx 0.8$ for the inner-dataset split, while analogue actions are still present in the cross-dataset regime.

are semantically too similar to any of the target categories by leveraging the maximum similarity observed inside the same dataset as a rejection threshold for categories of foreign origin.

Formally, a *seen* category $a_i \in \mathcal{A}^{seen}$ is allowed if and only if following condition is satisfied:

$$\forall a_j \in \mathcal{A}^{unseen},\ sim(\omega(a_i), \omega(a_j)) \leqslant \tau \tag{20}$$

where $sim(\cdot)$ is the cosine similarity of two vectors, $\omega(\cdot)$ is the *word2vec* representation of a label and τ is the similarity rejection threshold. We set the similarity threshold τ as the maximum pairwise similarity between the source and target labels in the *intra-class* setting:

$$\tau = \max_{a_i \in \mathcal{A}^{seen}_{intra}, a_j \in \mathcal{A}^{unseen}} sim(\omega(a_i), \omega(a_j)) \tag{21}$$

Note, that τ is dataset-specific and amounts to around 0.8 for HMDB-51.

6.2.3.3 *Experiments*

EXPERIMENTAL SETUP. For evaluation, we adopt the popular ConSE approach, described in Section 6.2.2.2, and examine the impact of using external data and, more precisely, the influence of source-target label similarity, on the recognition performance under the premise that the model itself remains exactly the same. We use I3D [17] as our visual recognition model, and the 300 dimensional *word2vec* model trained on Google News articles as our language model [126]. We use the HMDB-51 [89] as our target dataset, following the zero-shot learning setup of Wang *et al.* [205] (25 known and 24 unknown categories) and ten random splits, for which we report mean and standard deviation statistics. As a foreign data source we use the Kinetics dataset [17], which covers 400 activity categories.

INTRA- AND CROSS-DATASET CLASS SIMILARITY. First, we reassure our assumption that labels of seen actions tend to be significantly closer to the unseen categories if they originate from a foreign dataset. Figure 45 shows the distribution of the maximum pairwise source-target similarity for each source label. As we see, foreign actions are far closer, often even identical, to

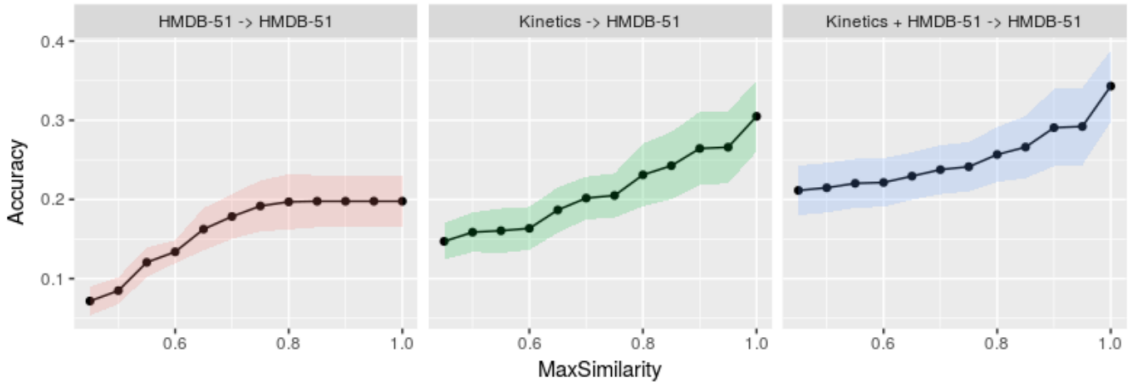


Figure 47: Effect of eliminating *familiar* concepts on the zero-shot accuracy (*i. e. upper* bound for the allowed source-target label similarity). We distinguish intra-dataset, cross-dataset and hybrid protocol and compute the average accuracy (over ten splits) for different upper thresholds for the allowed labels. X-Axis denotes the semantic similarity threshold s_{th} above which source categories are excluded. Having similar classes in the seen and unseen sets strongly affects accuracy, an effect that is more pronounced when using external datasets.

the target classes dataset in comparison to the same dataset case. We explain this distribution by the nature of datasets design, as a single dataset would not contain duplicates or activities that are too close to each other.

EFFECT OF THE SIMILAR ACTIVITIES ON THE CLASSIFICATION ACCURACY. Our next area of investigation is the influence of such analogous activities and external data on the classification results. We report the average and standard deviation of the recognition accuracy over the splits for different similarity thresholds s_{th} for restricting the target categories (Figure 47 and Table 19). Extending the model trained on the native data (intra-dataset) with external datasets (intra- and cross-dataset regimes) increases the accuracy by almost 15%, with 10% accuracy increase observed when an external source is used alone (cross-dataset regime). Excluding direct matches (s_{th} of 0.95) leads to a performance decline of 4% for cross-dataset scenario, although only around 1% of external action categories are excluded (Fig. 46).In other words, only 1% of external action labels (which are extremely similar to the target) account for almost half of cross-dataset vs. inner-dataset performance boost.

The accuracy saturates at a similarity threshold of around 0.8 in the inner-dataset regime, as no duplicate activities are present (Figure 46). Our evaluation procedure leverages this maximum inner-dataset similarity to effectively eliminate synonyms from external sources, while not influencing the inner-dataset performance. In our framework, the majority of the external dataset is kept 384.7 of 400. However, the influence of analogue activities is clearly tamed, leading to a performance drop from 34.77% to 25.67% for the inner- and cross-dataset protocol. Still, using external data is very beneficial for the recognition results and using both internal and external data consistently outperforms the single-source model. A clear standardized protocol for defining allowed external source classes without violating the ZSL rules, is a crucial step towards more adequate model evaluation.

CONTEXT OF PREVIOUS WORK. In this work, our goal is to highlight the ambiguities which arise when external datasets come into play in zero-shot action recognition and solve this

Exclusion protocol	Source	# source labels	Accuracy	ZSL premise
n. a.	HMDB-51	26	19.92 (±3.3)	✓
Use all source labels	Kinetics +	400	30.72 (±4.4)	–
	HMDB-51	426	**34.77 (±4.5)**	–
Exclude exact labels	Kinetics +	≈394.8	26.6 (±4.6)	–
	HMDB-51	≈420.8	**29.22 (±4.9)**	–
Exclude similar labels **(ours)**	Kinetics +	≈384.7	23.1 (±3.9)	✓
	HMDB-51	≈410.7	**25.67 (±3.5)**	✓

Table 19: Zero-shot recognition results with different evaluation regimes. While leveraging external sources clearly improves the results, additional measures should be taken so that the unseen categories are indeed *unseen*. Our corrective procedure automatically excludes overlapping concepts, ensuring the ZSL premise. Even after our corrective measure, transfer from external datastes is highly beneficial. The number of source labels sometimes contains decimal digits, because we report the *mean over ten splits*.

problem by formalizing a corrective protocol, therefore facilitating further research of such cross-dataset transfer. The vast majority of evaluated methods has used the inner-dataset split, *e. g.* a similar ConSE model employed by [216] which reaches 15.0%, while our model with underlying deep shows an improvement of 19.92%. The state-of-the-art approach using inner-dataset evaluation achieves 22.6% [152], while the recent work of Zhu *et al.* [229] reports highly impressive results of 51.8% employing an external data source (ActivityNet). We want to note, that our model also consistently outperforms state-of-the-art which uses inner-dataset split only, however, we find that systematic elimination of synonyms is crucial for a fair comparison, as we do not know, which actions were allowed in the setting of [229] and we show, that few analogue actions might lead to a clear performance boost.

ELIMINATING TOO *UNFAMILIAR* CONCEPTS. We have previously considered setting an *upper bound* on the allowed seen-unseen category similarity for the purposes of a fair evaluation (in order to ensure the ZSL premise of disoint seen and unseen classes). As a side-observation, we have found that using an additional *lower bound* on the similarity of the external and target categories leads to a performance increase of around 2% for every evaluation setting (Fig. 48). In other words, concepts which are too unfamiliar to our target dataset are rather a distraction and eliminating such classes leads to a better recognition. It might therefore be worth to investigate this aspect thoroughly in future research.

Effect of eliminating *familiar* concepts on the zero-shot accuracy (*i. e. upper* bound for the allowed source-target label similarity). We distinguish intra-dataset, cross-dataset and hybrid protocol and compute the average accuracy (over ten splits) for different upper thresholds for the allowed labels. X-Axis denotes the semantic similarity threshold s_{th} above which source categories are excluded. Having similar classes in the seen and unseen sets strongly affects accuracy, an effect that is more pronounced when using external datasets.

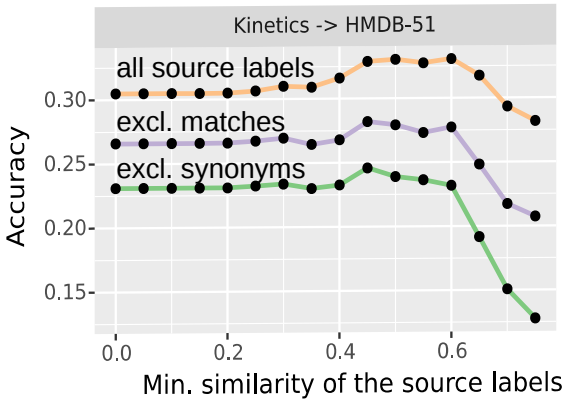


Figure 48: Side-exploration: effect of eliminating *unfamiliar* concepts on the zero-shot accuracy (*i. e. lower* bound for the allowed source-target label similarity) for the cross-dataset setting (Kinetics to HMDB-51). The x-axis indicates the *minimum* allowed unseen-seen label similarity (all seen categories *below* this threshold are excluded). For example, x-axis value of 0.0 means that there is no filtering due to labels being too *unfamiliar* with the target labels, while a threshold of 0.9 indicates that only source labels which are extremely similar to the target labels are allowed. This can be viewd as the opposite to Figure 47, where we set an *upper* threshold to meet the zero shot premise. The three colored lines correspond to different handling of the *maximum* allowed similarity (upper threshold): the orange line indicates that no labels are excluded based on the *upper* threshold, the purple one excludes exact label matches and the green line indicates our corrective procedure based on intra-dataset similarity. Interestingly, categories with similarity to the target labels *below* ~ 0.4 begin to hinder the performance, presumable being additional noise.

6.3 CROSS DOMAIN RECOGNITION

This section is based on our publication in *IV* 2020 [161], © IEEE .

Beyond handling the unknown *activities*, as addressed in the previous sections, one should also keep in mind potential challenges of unknown *domains*. Activity recognition algorithms rely heavily on the premise that examples used for training and testing are drawn from the same distribution [116, 141, 143]. In practical applications, it is rather rare for the target data to be of the same distribution as during training. For example, inside the vehicle cabin, such domain shifts may arise from alterations of illumination, sensor type and -placement, posing a significant obstacle for data-driven models. Research of learning domain-invariant representations is therefore crucial for long-term integration of such models in practice.

Capturing driver behavior by observing the person through cameras benefits from the advancements in computer vision, but also inherits its challenges, such as sensitivity to domain shifts. While combining multiple cameras consistently leads to improvement in recognition results [121, 175, 176], introducing a novel sensor into the setup often requires costly data collection, annotation and model re-training for the new modality type. *Could we skip labelling of new data and instead transfer the already existing knowledge to our domain?* Such unsupervised domain adaptation for cross-modal driver behavior recognition is the key goal of this section and consid-

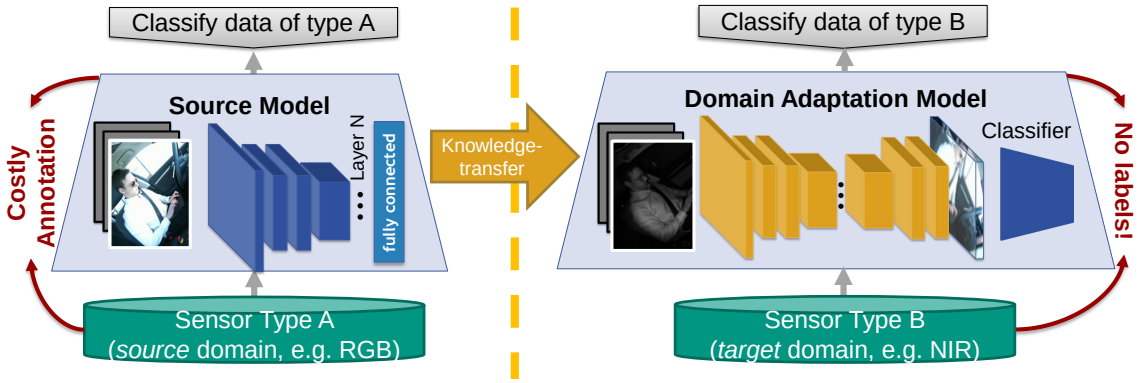


Figure 49: We introduce the task of unsupervised domain adaptation for cross-modal driver behaviour recognition, aiming to learn a mapping and enable knowledge transfer between two sensor types (delivering data in *source* and *target* domains). In our setting, the annotations are available only for the *source* domain, but the deployed model faces *target* domain data.

ers a scenario where annotated training data from the source distribution is given, but our task is to classify examples from a different target distribution with only unlabeled data (Figure 49).

In this section, we aim for driver activity recognition models which are able to classify data from domains other than the ones seen during training. Our final objective is to enable cross-modal knowledge transfer: given an existing model trained on annotated data from the *source* domain, we aim to adjust it to classify data examples from a different *target* domain where new annotations would be too costly to acquire (overview in Figure 49). First, we formulate the problem of *unsupervised domain adaptation for driver activity recognition*, where a model trained on labeled examples from the source domain (*i. e.* color images) is intended to adjust to a different target domain (*i. e.* infrared images) where only unlabeled data is available during training. To enable such a knowledge transfer, we leverage current progress in image-to-image translation (mapping an image from a source domain to a different target space), which experienced steep progress since the emergence of Generative Adversarial Networks (GANs) [50, 65, 103, 228]. We adopt and extend prominent image-to-image translation paradigms to handle domain changes inside the vehicle cabin.As our long-term goal is not high realism of the converted image but robust cross-domain *classification*, we present a novel **CL**a**S**sification-driven model for **UN**supervised **I**mage **T**ranslation *CLS-UNIT*. Our model is based on a Variational Auto-Encoder (VAE) for domain adaptation [103], which we enhance with an additional classification-driven loss influenced by a similar strategy employed successfully in previous semantic segmentation works [65][7]. To evaluate our idea, we explore two settings, in which the test data is captured by a sensor different from the one used during supervised training: classification of (1) *near-infrared* and (2) *depth* videos with annotated examples only available for *color* data. Our *CLS-UNIT* model consistently outperforms the baselines and other image-to-image translation approaches.

We now define the task of unsupervised domain adaptation for driver behavior recognition (Section 6.3.1) and describe our proposed strategy for leveraging generative image-to-image translation models for our task (Section 6.3.2). Note, that Section 6.3.2 covers both, existing image-

7 Concurrently with our work (both contributions published in IV 2020) Rangesh *et al.* (2020) [157] propose a similar idea for eyeglasses removal inside the vehicle cabin, where they enhance a CycleGAN [228] with an additional loss for gaze classification, consistently improving the accuracy.

to-image translation models from other fields which have adapted to suit our driver activity recognition task, and our proposed *CLS-UNIT* framework (Section 6.3.2.5).

6.3.1 *Cross-Modal Driver Activity Recognition*

We address the problem of *unsupervised domain adaptation for cross-modal driver activity recognition*, which aims at inferring the correct driver behavior from a modality type not seen by the classifier. Our assumption is, that for the target domain, we have no labelled data for training a classifier, but raw *unlabelled* videos are available (which is a realistic scenario, as we can always collect videos without annotation on-the-fly).

Conventional action recognition aims at assigning an activity label $a \in \mathcal{A}\{1, ..., m\}$ to input data $\mathbf{x} \in X$, where the training and evaluation samples are generated by the same underlying probability distribution, *i. e.* $\mathbf{x} \sim P_{data}$ for both training and test examples $\mathbf{x}$. In *cross-modal action recognition*, on the other hand, test and training data are sampled from distinct probability distributions. Formally, our training set comprises labeled instances from the *source* domain: $(\mathbf{x}_s, a_s)$, with $x_s \in \mathcal{X}_s$ and $a_s \in \mathcal{A}$, and *unlabeled* data from the *target* domain $\mathbf{x}_t \in \mathcal{X}_t \sim P_{source}$, where the samples $\mathbf{x}_t$ do not have any corresponding activity labels. Our goal is to classify a new instance $\mathbf{x}_t^{test}$, which data representation belongs to the *target* domain $\mathbf{x}_t^{test} \in \mathcal{X}_t \sim P_{target}$. In the following, we will abbreviate the distributions P_{source} and P_{target} as P_s and P_t respectively.

6.3.1.1 *Directions of the Mapping Functions*

A key challenge in domain adaptation is determining, how these domains are related. As we address the task of cross-modal driver activity recognition by leveraging image-to-image translation models, which learn to map an image from one domain to another, we explore two different types of mapping functions, determined by their transfer direction: learning to transfer (1) from *source* to *target* $m_{s\rightarrow t} : \mathcal{X}_s \rightarrow \mathcal{X}_t$ and (2) from *target* to *source* $m_{t\rightarrow s} : \mathcal{X}_t \rightarrow \mathcal{X}_s$. After we learn these domain-mapping-functions (which is the main topic of our work and will be discussed in Section 6.3.2 and Section 6.3.2.5) the inference for a new example $\mathbf{x}_t^{test}$ from the unlabelled target domain can be done in two ways, depending on the chosen mapping function direction:

(1) The *source-to-target* function $m_{s\rightarrow t}$ can be directly used on the labeled training data. That is, we translate the labeled source examples x_s into the target domain, which we use for training a classifier $c_t : \mathcal{X}_t \rightarrow \mathcal{A}$ on $(m_{s\rightarrow t}(\mathbf{x_s}), a_s)$.

(2) Another strategy is to leverage the opposite (*i.e. target-to-source*) translation $m_{t\rightarrow s}$ to convert an instance $\mathbf{x_t}^{test}$ from the target domain of our test set into the source domain. A classifier $c_s : \mathcal{X}_s \rightarrow \mathcal{A}$ trained on the labelled source data $(\mathbf{x_s}, a_s)$ is subsequently used to yield the class-prediction for $m_{t\rightarrow s}(\mathbf{x_t}^{test})$.

6.3.1.2 *Benchmark*

We modify the *Drive&Act* benchmark to suit our domain adaptation conditions. We remind, that the dataset comprises color, NIR- and depth- videos of 15 drivers, which are densely annotated with 34 fine-grained activity labels. As previously described, our training data consists of: (1) *labeled* data in the *source* domain and (2) *unlabeled* recordings in the *target* domain. We select color videos as our source modality and both, NIR and depth as our target domains, resulting in two

distinct experimental setups. For our training data, we therefore randomly select color data of 7 drivers with the corresponding activity annotations and unlabeled videos of 3 drivers in the target (*i. e.* NIR or depth) domain. To evaluate our model, we then use NIR and depth footage of the remaining 5 drivers for validation (2 subjects) and testing (3 subjects). As done in the standard setting (see Section 3.1.3.3), we divide the recordings into 3 second chunks, compute the prediction for each chunk and then use the *balanced accuracy* as our performance metric.

6.3.2 *Neural Video Translation*

We now examine how to learn the mapping-functions for video frame transfer from the *source* domain (*e. g.* RGB) to the *target* domain (*e. g.* NIR) and vice versa. As we deal with an unsupervised setting and there are no labels for target domain, we also lack access to pairwise registered videos between the two modalities. We therefore leverage the concept of cycle-consistency [228] , which allows us to learn the mapping without the available ground-truth pairs.

6.3.2.1 *Generative Adversarial Networks (GANs)*

We model the mapping functions $m_{s\rightarrow t}$ and $m_{t\rightarrow s}$ with generator networks, which use convolution layers to translate the frames. Using a generator alone is a viable solution but has two drawbacks: (1) it is prone to learn a transfer to a single instance point (*e. g.* mapping to sepia in case of image colorization), and (2) it requires paired ground-truth data, which is impractical in many applications. The discriminators D_s and D_t are neural networks that learn to decide if a sample stems from the probability distribution of the source or target domain respectively or if it was produced by the generators. The discriminator output is therefore the estimated likelihood of a *real* image in $(0, 1)$. The architecture for source-to-target mapping of the images is trained using the $\mathcal{L}_{GAN}$ loss:

$$\begin{aligned}\mathcal{L}_{GAN}^{s\rightarrow t} = \mathbb{E}_{\mathbf{x_t}\sim P_t}[\log \underbrace{D_t(\mathbf{x_t})}_{\text{Discriminator output for true target domain data.}}] \\ +\mathbb{E}_{\mathbf{x_s}\sim P_s}[\log(1 - \underbrace{D_t(m_{s\rightarrow t}(\mathbf{x_s}))}_{\text{Discriminator output for ``fake'' target domain data generated by } m_{s\rightarrow t}.})].\end{aligned} \tag{22}$$

The loss comprises: (1) a target-based loss, which penalizes the discriminator for not classifying data sampled from the target domain correctly; and (2) a loss that includes both the generator and the discriminator, in such a way that they oppose each other during training. While the generator produces data intending to fool the discriminator, the discriminator learns to distinguish between the synthesized and real instances. For the inverse direction (*i. e.* from target to source), the loss is computed by interchanging our two domains in Equation 22.

6.3.2.2 *Cycle-consistency paradigm*

When using the loss from Equation 22 as it is, we do not enforce the generator to use the input map for fooling the discriminator. Thus, the model can fool the discriminator by, for example, producing previously unseen noise. To enforce the generator to keep relevant information in the translation process, we employ the cycle-consistency paradigm [228] known as the *CycleGAN*, incorporating the cyc-loss in our training:

$$\mathcal{L}_{cyc}^{s \longrightarrow t} = \mathbb{E}_{\mathbf{x_s} \sim P_s}[\| m_{t \rightarrow s}(m_{s \rightarrow t}(x_s)) - \mathbf{x_s} \|_1] \tag{23}$$

where $\|\cdot\|_1$ denotes the L_1 distance. This term encourages the network to retain information from the input image by encouraging the mappings to reproduce the original sample.

6.3.2.3 *Semantic consistency loss*

The previously described objective yields a realistic mapping between the domains. In our case, such realism of the resulting videos does not serve as an end in itself, but as a means for the cross-domain *classification*. We therefore augment the cycle-consistency loss with additional semantic information extracted from our labeled *source* data, similarly to [65]. To this intent, we design a classifier $c : X_s \cup X_t \longrightarrow \mathcal{A}$ for fusing the class-information into the training procedure:

$$\begin{aligned} \mathcal{L}_{sem} = \mathbb{E}_{(\mathbf{x_s}, a_s) \sim P_s}[CE(c(m_{s \rightarrow t}(\mathbf{x_s})), a_s)] \\ + \mathbb{E}_{x_t \sim P_t}[CE(c(m_{t \rightarrow s}(\mathbf{x_t})), \hat{a}_t(\mathbf{x_t}))], \end{aligned} \tag{24}$$

where $\hat{a}_t(x_t) = \text{argmax}(c(x_t))$ infers a label of a target instance using our classifier c. The cross-entropy loss denoted with $CE(\cdot, \cdot)$ is calculated with respect to the classification result of c on the instance mapped into the other domain.

Overall, the building blocks of the final loss, as employed in [65], cover the adversarial loss for realistic image reconstruction and a cycle-consistency loss to compensate for the lack of paired data. Moreover, a semantic consistency loss takes advantage of the labeled source training data and enforces similar classification scores of images before and after the translation. The final loss therefore not only aims to realistically map between the domains, but is also *classification-driven* as it computes the loss by summing the previously defined terms:

$$\mathcal{L}_{CLS} = \mathcal{L}_{GAN}^{s \longrightarrow t} + \mathcal{L}_{GAN}^{t \longrightarrow s} + \mathcal{L}_{cyc}^{s \longrightarrow t} + \mathcal{L}_{cyc}^{t \longrightarrow s} + \mathcal{L}_{sem}. \tag{25}$$

We refer to this framework as CyCADA, as done in the original work [65] in context of semantic segmentation.

6.3.2.4 *Shared-latent space models*

Instead of estimating the mapping functions directly, shared-latent space models [103] take a detour through an intermediate representation shared by both the source- and target domain. That is, the direct mapping function $m_{s \rightarrow t}$ is divided into a convolutional encoder $m_{s \rightarrow \ell}$ and a decoder network $m_{\ell \rightarrow t}$. This encoder-decoder setup condenses the input to a compact latent representation and is often implemented using Variational Auto-Encoders (VAEs) [103]. Thus, two VAEs underlie $m_{s \rightarrow \ell}$, $m_{\ell \rightarrow s}$ and $m_{t \rightarrow \ell}$, $m_{\ell \rightarrow t}$ respectively. To encourage the encoder networks $m_{s \rightarrow \ell}$ and $m_{t \rightarrow \ell}$ to have a *common representation space*, the parameters are shared throughout the later layers. An additional regularization constraint that is frequently applied to VAEs, restricts the output of the encoder to follow a standard normal distribution, *i. e.* $z \sim P_\eta(\cdot)$, where $P_\eta(z) = \mathcal{N}(z;\ 0, I)$. The outcomes are penalized when deviating from standard normal distribution via the KL-divergence between $P_\eta(\cdot)$ and the latent parameters (averages and deviations). The final latent representation of an image therefore encompasses sampling from this estimated distribution. The encoder models the distributions by estimating mean vectors of unit Gaussians

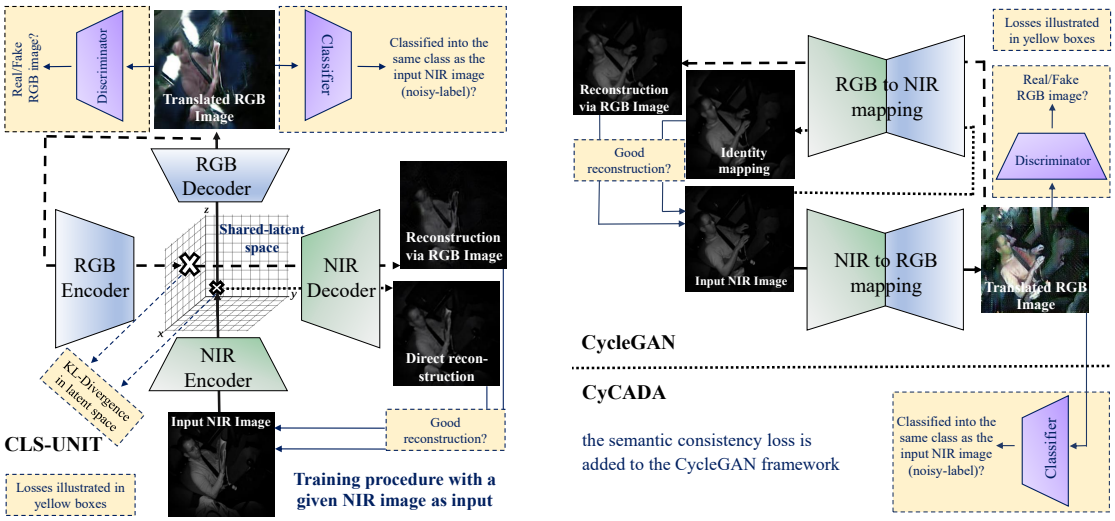


Figure 50: Overview of our *CLS-UNIT* architecture (left) and other considered image-to-image translation models (CycleGAN and CyCADA) on the right. The main difference between CyCADA and CycleGAN is the semantic consistency loss. Our *CLS-UNIT* model extends the conventional UNIT model with the classification-driven loss in a similar way. The training procedure for learning a mapping function from the *target* to *source* domain is depicted for all models.

$\mathcal{N}(z_s;\ m_{s\to\ell}, I)$ and $\mathcal{N}(z_t;\ m_{t\to\ell}, I)$, of which the outputs are used in the decoders. Finally, we encourage the latent representation to follow a standard distribution and the reconstructed image to resemble the input data as follows:

$$\begin{aligned}\mathcal{L}_{VAE}^{s,\ell} = \lambda_1 KL(\mathcal{N}(\mathbf{z_s};\ m_{s\to\ell}(\mathbf{x_s}), I) \,||\, P_\eta(\mathbf{z})) - \\ \lambda_2 \mathbb{E}_{z_s\sim\mathcal{N}(\cdot;\ m_{s\to\ell}(\mathbf{x_s}),I)}[\log p_{m_{\ell\to s}}(\mathbf{x_s};\ \mathbf{z_s})]\end{aligned} \tag{26}$$

where $KL(\cdot||\cdot)$ computes the Kullback-Leibler divergence between two probability distributions. As stated in [103], $p_{m_{\ell\to s}}(x_s;\ z_s)$ is modelled as a Laplacian distribution which when minimizing its log-likelihood is equivalent to minimizing the absolute distance between the original image and its reconstruction using $m_{\ell\to s}$. A matching loss $\mathcal{L}_{VAE}^{t,\ell}$ sets up the second VAE of the framework.

In addition to this, two GANs are employed to ensure that the decoder networks produce samples that fit into their assigned domains. This is established by augmenting our framework with additional discriminator networks for each domain, leading to a loss similar to Eq. 22.

To ensure that the mapping of an image to the other domain and back into the original domain results in the input image, a variant of the cycle-consistency paradigm is added as follows:

$$\begin{aligned}\mathcal{L}_{vaecyc}^{s\to t\to s} = \ & \lambda_3 KL(\mathcal{N}(\mathbf{z_s};\ m_{s\to\ell}(x_s), I)||p_{st}(z)) \\ & + \lambda_3 KL(\mathcal{N}(z_t;\ m_{t\to\ell}(m_{s\to t}(x_s)), I)||P_\eta(\mathbf{z})) \\ & - \lambda_4 \mathbb{E}_{\mathbf{z_t}\sim\mathcal{N}(\cdot;\ m_{t\to\ell}(m_{s\to t}(x_s)),I)}[\log p_{m_{\ell\to s}}(\mathbf{x_s};\ \mathbf{z_t})],\end{aligned} \tag{27}$$

with a cross-domain mapping of $m_{s\to t}(x_s) = \mathbb{E}_{\mathbf{z_s}\sim\mathcal{N}(\cdot;\ m_{s\to\ell}(x_s),I)}[m_{\ell\to t}(\mathbf{z_s})]$. The hyperparameters $\lambda_{\{1\text{–}4\}}$ provide measures to weight the different components of the losses $\mathcal{L}_{VAE}$, $\mathcal{L}_{GAN}$ and $\mathcal{L}_{vaecyc}$ that are all optimized in both directions composing the loss for [103]. This shared-latent space framework is referred to as UNIT.

Figure 51: CLS-UNIT training progression. The progression in training of a mapping from NIR- to color images. The training iterations increase from left to right.

6.3.2.5 *CLS-UNIT: Classification-driven domain transfer learning with VAEs*

While UNIT provides a meaningful mapping from the source to the target domain, it has issues with preserving the class information. In contrast, CyCADA [65] is able to preserve the semantic information during the mapping procedure, but encounters difficulties bridging between the source- and target domain. We aim to capture the advantages of both techniques and introduce the **CL**a**S**sification-driven framework for **UN**supervised **I**mage **T**ranslation *CLS-UNIT* (overview in Figure 50). Our model enhances the VAE-based UNIT network for learning a shared-latent space with a *classification-driven loss*, similar to [65]. Our ***CLS**-UNIT* loss is defined as:

$$\begin{aligned}\mathcal{L}_{CLS\text{-}UNIT} = \lambda_{cls}\mathcal{L}_{sem} + \lambda_{unit}(&\mathcal{L}_{GAN}^{s\to t} + \mathcal{L}_{GAN}^{t\to s} \\ &+ \mathcal{L}_{VAE}^{s,\ell} + \mathcal{L}_{VAE}^{t,\ell} \\ &+ \mathcal{L}_{vaecyc}^{s\to t\to s} + \mathcal{L}_{vaecyc}^{t\to s\to t})\end{aligned} \tag{28}$$

where λ_{unit} and λ_{cls} are parameters for weighting the losses set empirically using the validation data to 0.6 and 0.4 respectively. In the *NIR-to-color* testbed, larger λ_{unit} causes the translation to be more colorful while resulting in a blurry image. Overall, setting a higher value for λ_{cls} leads to the preservation of structure in the mapped images lowering the emphasis on faithful colorization.

6.3.2.6 *Implementation Details*

VIDEO EMBEDDING SCHEME We embed the input videos using the I3D network pretrained on Kinetics [17]. Depending on the mapping strategy (see Section 6.3.1), we either finetune the model on the labeled source data (*i. e.* color videos) or on the frames translated into the target domain (*i. e.* NIR or depth). The training hyperparameters are adopted from Section 3.1.

SEMANTIC SIGNAL For determining the semantic consistency loss of our mapping network, we use an auxiliary ResNet pretrained on ImageNet [60]. The backpropagated signal flows through the parameters of the mapping network, encouraging it to preserve information about the action semantics. As the auxiliary classifier has not learned useful semantic information early in training, we only backpropagate the signal if its loss falls below a threshold γ (in our case $\gamma = 3.4$, *i. e.* $\log(\#classes)$, the loss of uniform classification).

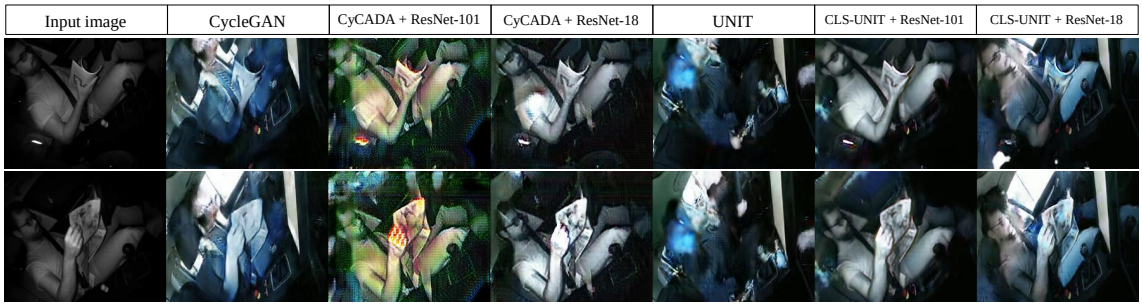


Figure 52: NIR-to-color translation results. Example translations of different models from NIR- to color images; the proposed CLS-UNIT model (last two columns) achieves meaningful colorization while preserving the structure and shapes in the input image.

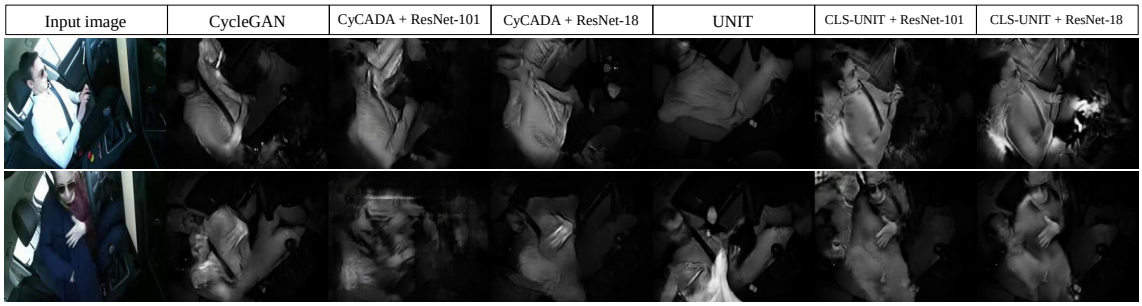


Figure 53: Color-to-NIR translation results. Example translations of different models from RGB to NIR images; our CLS-UNIT model (last two columns) with classification signal and shared-latent space is better at capturing fine structures in faces and fingers.

Frame sampling scheme As the input to our mapping network is an image, we need a strategy for sampling from the video while training. Selecting the frame from *Drive&Act* uniformly is problematic, because the class distribution is highly unbalanced. To tackle this, we perform class-wise sampling for the source domain data which draws frames of each class with the same probability. In case of the target domain data (*i. e.* NIR or depth), we draw instances uniformly over all frames as we do not have associated class labels.

GENERATOR AND DISCRIMINATOR Architectures for the mapping networks in CycleGAN- and UNIT-based methods are adapted from [228] and [103]. They were trained for 20 epochs with $10K$ sampled images of size 256×256 per epoch. We use the initial learning rate of 0.0001 for the first 10 epochs, and linearly decay it afterwards. The weights are initialized using He-initialization [59]. The mapping network and the classification stream with the semantic consistency loss are optimized using Adam [83] with a weight decay of 0.0001.

6.3.3 *Experiments*

6.3.3.1 *Image-to-Image Translation Results*

While our main focus lies in recognizing human activities we also present qualitative results regarding the learned mapping functions. Figure 51 illustrates the optimization progress of our

Translation Model	Direction	Classifier	Val	Test
Baseline Methods				
–	–	Random	3.03	2.94
–	–	Color-I3D	10.97	15.57
–	–	Grayscale-I3D	17.91	17.22
CycleGAN-based Networks				
CycleGAN	NIR⟶Color	Color-I3D	16.52	15.06
CyCADA + RN-18	NIR⟶Color	Color-I3D	16.94	22.33
CyCADA + RN-18	Color⟶NIR	NIR-I3D	29.14	24.58
Shared-Latent Space Models				
UNIT	NIR⟶Color	Color-I3D	4.11	4.03
CLS-UNIT + RN-18 (ours)	NIR⟶Color	Color-I3D	24.88	23.06
CLS-UNIT + RN-18 (ours)	Color⟶NIR	NIR-I3D	**31.52**	**29.32**

Table 20: Cross-modal activity recognition results with knowledge transfer from *color-to-NIR*. The *translation model* and the *direction* can be derived from the recognition procedure employed. RN denotes the ResNet architecture used for the classification-driven loss.

CLS-UNIT model in the *NIR-to-color* setting, showing how it incorporates color better at later stages while still learning to retain details relevant for activity recognition due to the semantic consistency loss.

In Figure 52 and Figure 53, we compare the *NIR-to-color* and *color-to-NIR* translation of all previously described models. Most of the networks ignore fine structures, such as the hands of a person, in their translations, with the exception of our *CLS-UNIT* model, where such classification-relevant cues are preserved. The *NIR-to-color* mapping schemes in CycleGAN and UNIT that do not employ a classification signal, generate colorful images, however, at the expense of blurring the driver. The balance between retaining details relevant to classification (*e. g.* person- and object-related cues) and meaningful colorization is done best by our *CLS-UNIT* approach with an auxiliary ResNet-18 classifier. An example of the *color-to-depth* translation and the corresponding ground-truth depth map are visualized in Figure 54.

6.3.3.2 *Cross-Modal Recognition Results*

We demonstrate the effectiveness of our model in Table 20. Additionally to prominent image-to-image translation approaches [65, 103, 228], we compare our model to three baselines: (1) a random classifier, (2) the I3D network trained on the source data (*i. e.* color) classifying the data in target domain directly and (3) an I3D trained on source data transformed to grayscale directly classifying the target domain data. The third baseline (grayscale) yields a fair *color-to-NIR* evaluation, as NIR data might seem similar to grayscale images with the naked eye.

The I3D model achieves an accuracy of 67.76% in the conventional (*i. e.* color-to-color) setting, which may be seen as the upper bound for our cross-modal approach. I3D performance drops to only 15.57%, when applied in our cross-modal setting without any additional transfer, as CNNs per se are highly susceptible to domain shifts. Converting the training images to grayscale clearly helps, as they appear similar to the target IR frames (17.22%), but the recognition rate still remains low. When using CycleGAN-based methods and the mapping direction *NIR-to-color*, only the CyCADA model with an auxiliary ResNet-18 classifier outperforms the grayscale I3D baseline. The

Figure 54: Color-to-depth translation. Source color image, ground-truth depth map (estimated with a Kinect) and the *CLS-UNIT* translation result.

Model Type	Val	Test
Baseline Methods		
Random	3.03	2.94
Color-I3D	8.21	8.42
Grayscale-I3D	10.21	9.50
Shared-Latent Space Model		
CLS-UNIT + ResNet-18 (ours)	**17.23**	**17.57**

Table 21: Cross-modal activity recognition results, *color-to-depth* setting.

conventional UNIT framework could not carry the relevant information for classification through the mapping functions at all and obtains an accuracy slightly better than random. However, our proposed extension of UNIT with a classification-driven loss (*CLS-UNIT*) heavily increases the performance.As described in Section 6.3.1 we are flexible in choosing the mapping direction for classifying the unfamiliar modality. Utilizing a mapping from *color-to-NIR* we translate the labeled color videos and train a NIR classifier on top of them for our best CycleGAN-based and shared-latent space models. In addition to consistently producing better results (see *color-to-NIR* models in Table 20) this scheme eliminates the necessity to compute the translation of incoming data in an online scenario as the classifier directly operates on the target domain. Overall, our model with the recognition rate of 31.52% on validation set and 29.32% on test set surpasses other translation models and baselines by a significant margin.

To examine broader applicability, we conduct a second experiment in the *color-to-depth* setting. (Table 21) shows consistent recognition confirming the benefits of our approach. Using a *color-to-depth* transfer function with our *CLS-UNIT* model and then learning to classify the translated frames boosts the native I3D performance by 9.15% (test) and 9.02% (validation).

While the CLS-UNIT model surpasses other image-to-image translation methods and the native I3D baseline (*i. e.* cross-modal recognition without additional knowledge transfer), we need to acknowledge, that the recognition rate has room for improvement. Of course, the achieved accuracy is far below the same-domain-recognition (*e. g.* we achieve ~ 30% in the *Color-to-NIR*, while the standard *NIR-to-NIR* recognition results are over ~ 65%, although the random baseline is only ~ 3%). Since deep CNNs are driven by the training set distribution by nature, cross-modal activity recognition is a hard yet vital task with a strong necessity for increased research. Still, our improvement of over ~ 13% compared to the native I3D model reveals the potential of VAE-based image-to-image translation models, presenting them as a powerful tool for learning domain-invariant latent representations.

6.4 CHAPTER CONCLUSION

The reach of supervised activity recognition algorithms is limited by costly annotations. In practical applications, an action recognition model should be able to solve three problems: 1) standard classification of previously seen categories; 2) identifying uncertain cases, such as previously unknown behaviours, and 3) dealing with such uncertain examples, especially if repeatedly labelling them with new concepts is not an option. In this chapter, we have addressed the latter task of learning novel activities or handling domain shifts without any manual annotations.

Our first idea for learning without a single explicitly labelled example is to leverage content posted online, in our case, YouTube. We introduced the problem of weakly supervised driver intention prediction from videos by learning from conversations behind the steering wheel and presented a novel dataset of driving exam dialogs. To meet the challenge of noisy smalltalk conversations, we introduced a pre-processing technique for identifying and skipping such regions. We showed through experiments featuring different deep models, that such dialogs can be successfully used as guides for learning to foresee human intent.

Our second strategy for generalization to new unseen classes without any additional training data is knowledge transfer from language models via zero-shot learning. We present a first framework for *generalized* zero-shot action recognition, where the model is evaluated on both, instances of known and novel classes. This opens new challenges for the native ZSL approaches, because they tend to be strongly biased towards the seen classes. We follow a gating strategy to address this: our framework combines a novelty detection model (serving as the gate between the *known* and the *unknown*) with off-the-shelf zero-shot approaches, leading to significant improvements in classification accuracy. In addition, we study the possibility of cross-dataset knowledge transfer for zero-shot action recognition, guided by the idea of utilising large-scale external datasets for training and then generalizing to a smaller dataset of target actions.

Finally, we address activity recognition in unknown domains and explore generative models which learn a *shared representation* space of the source and target domain. We formalize the problem of unsupervised cross-domain driver activity recognition and extend our *Drive&Act* testbed with this setting. To enable knowledge transfer between the two domains, we leverage current progress in image-to-image mapping and implement multiple off-the-shelf image translation models. We then introduce a novel approach for cross-modal activity recognition in context of driver observation. We leverage activity labels of the source domain training data and learn a shared-latent space of both modalities with a VAE-based model extended with an additional *classification-driven loss*. Enhancing the UNIT-VAE model training with the *classification-driven loss* encourages the network to learn a shared representation which reflects the semantic nature of the activity classes and leads to the best recognition results.

Scientific impact of this chapter can be summarized in following main contributions:

Contribution 1 : A new dataset and approach for *recognizing human intention by learning from driving exam dialogs* without a single manually annotated label. Our framework includes a novel pre-processing algorithm for detecting smalltalk and therefore identifying the conversation segments relevant for training.

Contribution 2: An extension of the zero-shot activity recognition task to the generalized case and a framework combining the popular ZSL methods with our model for novelty detection achieving the best recognition results.

Contribution 3: Formalization of cross-dataset and hybrid regimes for zero-shot-action recognition and a corrective filtering algorithm ensuring a fair evaluation (*i. e.* disjoint seen and unseen categories) in case of external knowledge transfer.

Contribution 4: A new task and a new VAE-based model for cross-modal driver observation, where no annotated examples are available in the domain we want to recognize, but a labelled training set is given in a different domain.

Part III

INSIGHTS

7 IMPACT ON THE FIELD

This thesis has advanced activity recognition- and driver observation research in different ways: we opened new tasks relevant for applications, introduced new datasets to the community and proposed new theoretical models for solving the underlying problems. In case the task or the benchmark has already existed, we compared our performance to the previously published work, to the best of our ability and as far as we were aware of it. In case of a novel benchmark, we have implemented challenging baseline models used in the related fields. This chapter revisits the main contributions from the perspectives of opened research directions, datasets, and models.

7.1 NEW RESEARCH DIRECTIONS

RELIABLE CONFIDENCE ESTIMATES IN ACTIVITY RECOGNITION With the incentive to facilitate research of action recognition models which identify, whether the prediction is correct, or not, we have proposed to incorporate the *reliability of model confidence* in the evaluation of action classification approaches in the form of expected calibration error and reliability diagrams (we formalized this task in Section 4.1.1).

OPEN SET ACTIVITY RECOGNITION Beyond identifying misclassifications, we aim for models which can detect new behaviors which were not present during training and introduce the concept of open sets to the field of activity recognition. We explore both, the general setting and driver observation context, formalizing the open set activity recognition task in Section 5.1.1.1.

GENERALIZED ZERO-SHOT ACTIVITY RECOGNITION For the first time, we consider the task of *generalized* zero shot activity recognition, where the model needs to not only classify the *known* and identify the *unknown* classes, but also assign an *unknown* category by establishing a semantic connection to external language models through word vectors (Section 6.2).

"WEBLY"-SUPERVISED DRIVER MANEUVER PREDICTION With rapidly growing amount of multimodal data posted on the web, our idea is to explore the intersection of language and vision for leveraging such on-line content in order to surpass costly annotation. In

Section 6.1 therefore introduce the problem of weakly supervised driver intention prediction by learning from conversations during driving exam dialogs without any manual labelling.

UNSUPERVISED DOMAIN ADAPTATION FOR DRIVER OBSERVATION As in real-life we always encounter changes of environmental conditions or sensor type, research of domain-invariant representations is vital for applications of CNNs which are known to be highly susceptible to distributional shifts. In Section 6.3 we formalize the problem of unsupervised cross-domain driver activity recognition, where a model trained on labeled examples from the source domain is intended to adjust to a different target domain, where only *unlabeled* data is available.

7.2 NEW DATASETS

This thesis has lead to two completely new datasets and multiple extensions of the existing benchmarks, most of which are already public or will be made available to the community upon publication to encourage future research. The *Drive&Act* dataset, introduced in Section 3.1 is the first large-scale dataset for driver activity recognition, featuring a hierarchical annotation scheme on multiple levels of granularity with 83 labels in total. The *Driver-Talk-To-Action* database from Section 6.1 is a multimodal dataset of mock driving exam conversations collected from YouTube, which we view as a unique opportunity for connecting speech to changes in human actions. Since no established benchmarks existed for several of our tasks, we have extended multiple existing datasets to suit our requirements. In this manner, we have introduced *Open-Drive&Act* as well as the open set versions of the prominent action recognition benchmarks *HMDB-51* and *UCF-101* (Section 5.1.1.2). We further introduced the unsupervised domain adaptation split of the Drive&Act benchmark (Section 6.3).

Note, that the above list covers new datasets, or existing datasets with substantial modifications, *e. g.* , different split construction, introduced *unknown* classes. It does not include extensions of published datasets with new metrics, as done, *e. g.* , in our confidence reliability benchmark.

7.3 NEW MODELS, FRAMEWORKS AND QUANTITATIVE COMPARISON

At the heart of this thesis are two new models: the *CARING* model for learning to obtain reliable confidence estimates, given that the category is known, and *Bayesian-I3D* with different voting variants for detecting categories not previously seen by the classifier. Both methods hold state-of-the art results in their respective field. The *CARING* model (Section 4.1.4) consistently outperforms the native action recognition networks and the widely used temperature scaling method [55] in terms of the expected calibration error. The *Bayesian-I3D* model (Section 5.3) leads to the best open set activity recognition results in terms of both, novelty detection as binary classification task and multi-class open set recognition, where *unknown* is treated as an additional category, surpassing neural network confidence and classical novelty detection approaches, such as One Class SVM [181]. Besides, when integrated in our generalized zero-shot activity recognition framework in Section 6.2, we achieve state-of-the art results, strongly mitigating the bias of standard zero-shot models towards the known classes.

An important part of our framework for annotation-free learning from YouTube in Section 6.1 is the new preprocessing method for detecting smalltalk. This method is used to identify less relevant conversation segments and automatically refine the training set. Our experiments indicate,

that *learning from less but better data* through our smalltalk refinement leads to better recognition results when learning from the web.

In the case of domain adaptation for cross-domain driver activity recognition, we have implemented multiple *existing* image-to-image translation methods and also introduced the new *CLS-UNIT* approach, which combines a Variational Autoencoder with classification-driven optimization strategy (Section 6.3), outperforming the raw network inference in cross-modal setting and other image-to-image translation methods, such as CycleGAN [228] or CYCADA [65].

To detach deep neural networks from their black-box reputation, in Section 4.2 we make a step towards *transparency* behind the spatiotemporal CNNs for driver monitoring, and implement a diagnostic framework for tracing back the reasons of failures. Our qualitative analysis indicates, that the main problems are caused by either learned object-, movement- or position- bias or high action similarity in combination with underrepresentation in the training set.

We also have algorithmic contributions in the conventional closed-set context, specifically, in vehicle maneuver prediction based on driver observation. To adress this task, in Section 3.2 we introduce a new framework by combining an optical flow network with a 3D ResNet and an LSTM, outperforming previous approaches on the *Brain4Cars* [73] benchmark. As a side-contribution, we also explore the related task of gesture recognition, as a gateway for novel communication interfaces inside the car. We notice, that while gesture recognition is often studied in multimodal context, previous methods are based on the late fusion paradigm. We focus on the potential of *earlier* knowledge exchange and conduct a systematic evaluation of various fusion methods at different stages in the network. While in Section 3.1.4 we adopt and study the *existing* end-to-end activity recognition CNNs for the *Drive&Act* task, it is worth mentioning, that such end-to-end methods outperform feature-based approaches (based on the skeleton and the interior model of the vehicle) in the majority of cases, including the main *Drive&Act* evaluation setting of the *fine-grained activities*. While the feature-based methods are better in inferring *location*, this is expected, since such information is inherently present in the 3D vehicle model, while activity recognition CNNs detach the output from location through pooling. We believe, that leveraging CNN architectures specifically designed for learning relations in space, such as Spatial Transformer Networks [70], would greatly improve the recognition in the future.

8 APPLICATIONS TO OTHER FIELDS

This work has an array of applications, reaching from autonomous driving to robotics and assistive technologies. In the proof-of-concept experiment from our *ACVR 2017* publication [120], we showcased an example of successful transfer of algorithms initially developed for autonomous vehicles to help the visually impaired people. While this thesis has focused primarily on the automotive industry (specifically, driver observation), the developed tools can be directly passed on to other fields requiring action recognition, facing the open world uncertainty, or safety-critical systems needing to detect their failure. In this chapter, we briefly describe three of such areas.

8.1 ASSISTIVE COMPUTER VISION FOR THE VISUALLY IMPAIRED

Humans rely strongly on the biological vision as their main sense for perceiving the environment. Every day, around 285 million people[1] worldwide face substantial challenges due to visual impairment. Tasks appearing straightforward to many of us, such as safe exploration of new spaces, obstacle detection or grasping non-verbal communicating cues when interacting with others, require additional assistance from peers or technology, if the biological vision is low.

Computer vision algorithms can, almost by definition, assist the visually impaired people as they transform the visual world into semantic concepts which can be communicated to the user, for example, through audio or haptics [119]. Some computer vision algorithms have already been explored in this domain, mostly targeting navigation tasks [119, 180, 221] or recognizing objects crucial for navigation,*e. g.* , staircases [130, 206] or paedestrian traffic lights [48, 222].

While activity recognition might improve social interactions by perceiving the current state of other people, to the best of our knowledge, it has not been researched in the context of visually impaired technologies yet. Even more surprising – the question of model uncertainty has been overlooked in this field, although overly-confident models would place the reliant person in imminent danger. Imagine a model trained to recognize paedestrian traffic lights in one country, being used in a different country, where the signs slightly diverge. A good model would

1 Estimate obtained from the World Health Organization: www.who.int/blindness/publications/globaldata/en/

identify, that it is facing a problem, as a false positive prediction of a green pedestrian light might put human life at stake. We argue that there is a clear gap in the literature as it comes to the visually impaired application regarding both, activity recognition and, more importantly, consideration of the model uncertainty. In our *ACVR 2017* publication [120], we have already demonstrated through a proof-of-concept experiment, that technology initially developed for autonomous driving can be successfully transferred to help navigate blind people. We therefore believe, that adapting the driver activity recognition models we have developed in this thesis to assist the visually impaired would also greatly help this cause.

8.2 ROBOTICS

The addressed challenges of neural networks, such as reliable confidence estimates, handling of unexpected situations or model interpretability, are also directly applicable in the field of cognitive robotics. In 2018, the worldwide demand of industrial robots has reached the mark of 400.000 units[2]. Although the operation area of suchs robots and humans is mostly separated, the potential of collaborative human-robot workspaces has become more and more clear over the past years, as it detaches robot re-configuration from the needed expertise in the specific programming language [148, 149]. This kind of cooperation requires high level of situation awareness through visual sensors and recognition algorithms.

While a substantial portion of research has been devoted to conventional activity recognition in context of human-robot interaction [3, 52, 174, 176], uncertainty-aware open set models would strongly benefit this field, since in case of incorrect recognition, industrial high-energy machines might be physically dangerous for humans. In addition, novelty detection might notify the human worker in case of an unusual situation. Besides, Sünderhauf *et al.* (2018) [192] have recently expressed their concerns regarding the limits of the existing deep models in robotics, explicitly naming *uncertainty estimation* and *identifying unknowns* as two primary issues. Motivated by this, we believe, that our proposed approaches would be helpful in robot vision. Furthermore, frameworks proposed in the context of recognizing new categories and domains without the use of labels might be useful for *active* robot agents, *e. g.* in the context of social robotics, where the robot moves through uncontrolled environments, constantly receiving new types of input. Leveraging knowledge available on the web, or automatic adjustment to new domains are important steps towards independent agents, which do not require constant supervision.

8.3 VIRTUAL AND MIXED REALITY

Visual recognition models become increasingly important in the rapidly growing field of mixed reality [144], which aims at merging immersive simulations of virtual reality with elements from the physical world. While such use-cases might seems less important often being grouped into the "entertainment" category, applications of computer vision go beyond improving user convenience and amusement, as free walking with such headsets might result in dangerous situations when people, *e. g.* , collide with the surrounding obstacles and fall [211]. As mixed reality headsets equipped with cameras become more affordable and simulations are often experienced at home, visual recognition might prevent less careful users from injuring themselves. As there are almost no restrictions in terms possible environments, the common machine learning assumption that

2 Estimates from www.statista.com/topics/1476/industrial-robots/

the model will be deployed under closed-set and same-domain conditions does not hold anymore. Such applications almost certainly encounter new concepts, situations, textures, or the surrounding conditions, therefore being an excellent use-case for our uncertainty-aware approaches.

Building *reliable* models is needed in almost every *Computer-Vision-for-VR* scenario, while, specifically, digitalizing human behaviour would enable novel interaction techniques. For example, in simulation-based learning [37, 41], automatic activity understanding would allow the system to track the progress of the current task and identify when the human is struggling or making a mistake in order to correct the user. Last but not least, while lower-level human understanding, such as body tracking, has already found resemblance in video games[3], higher-level assessment of the players' behaviour might lead to completely new gaming experiences.

3 One example is the VR dancing game *Beat Saber*: beatsaber.com.

Part IV

APPENDIX

A

EXAMPLE INSTRUCTIONS FROM THE DRIVING EXAMS DATASET

In the following, we showcase examples of recorded requests said by the driving teacher which contained our terms of interest (*i. e.* the maneuver we want to predict, which are *park, roundabout, right turn, left turn, stop, straight* and *exit*). Note, that while we select these seven categories for evaluation, further events can be added by querying the dialogs with the corresponding expressions. We present five examples for each evaluated category.

PARK

- *reverse parallel park within two columns*
- *I would like you to pull over and park in a safe place*
- *now park on the left please*
- *park in a safe place for me please*
- *There is a few parking spaces, take one please*

ROUNDABOUT

- *at the roundabout follow the road ahead please*
- *we've got three roundabouts coming off*
- *cross the roundabout second exit then*
- *go right on the roundabout and take the third exit*
- *and were gonna go to the roundabout*

RIGHT TURN

- *at the end of the road turn right please*
- *and turn right at the traffic lights at*
- *turn right then take the second left*
- *after one hundred yards turn right*
- *at the t-junction turn right*

LEFT TURN

- *turn left*
- *at the end of the road turn left please*
- *turn left then take the second right*
- *from now just turn left in here*
- *and let's ask you to turn left all right*

STOP

- *and just stop here again*
- *okay stop now at the end*
- *stop*
- *stop again*
- *stop on the left in a safe and convenient place*

STRAIGHT

- *continue straight at all*
- *go straight through*
- *straight ahead*
- *go straight away*
- *keeping reasonably straight as you do it*

EXIT

- *exit*
- *the road ahead second exit please*
- *second exit then take the second left*
- *take the first exit*
- *right third exit please so right third*

B

SHORT CV

Alina Roitberg

Education and career highlights

Apr 2017 - today	**PhD Student in Computer Vision**, *Karlsruhe Institute of Technology* , Research topic: **"Uncertainty-aware Models for Deep Learning-based Human Activity Recognition in Autonomous Vehicles"**.
Sep 2020 - Nov 2020	**PhD Internship - Computer Vision for AR/VR**, *Facebook Zurich (remote)*.
Feb 2016 - Mar 2017	**Data Science Consultant**, *MHP - A Porsche Company, Big Data and Analytics*.
Oct 2009 - Jul 2015	**B.Sc. and M.Sc. in Computer Science**, *Technical University Munich*, Specialisation: "Artificial Intelligence and Robotics", **with distinction** .
Sep 2012 - Feb 2013	**Erasmus exchange**, *Chalmers University of Technology, Gothenburg, Sweden*, .
2009	**School Graduation (Austrian Matura)**, Billrothgymnasium, Vienna, Austria.

Awards and other achievements

Nov 2020	**Best Student Paper First Runner Up Award**, *Intelligent Vehicles Symposium*, Awarded Paper: "Open Set Driver Activity Recognition".
Apr 2017 - today	**Supervision of student theses**, successfully supervised computer science students (four Master and one Bachelor) during their final theses projects .
Sep 2020 - today	**Official TUM Mentor for master's students interested in pursuing a PhD**, *Mentoring program of the Technical University of Munich*.
May 2014	**Google Summer of Code**, a programming scholarship awarded by Google to support students working on open-source software (project: Point Cloud Library).
Apr 2014	**Deutschlandstipendium** , *German national scholarship for talented students*.
2004-2005 (school-time)	**Ukrainian Mathematical Olympiada**, three awards for the second and one award for the third place in the Ukrainian Mathematical Olympiadas (7th and 8th grade).

C

AUTHORED PUBLICATIONS

This doctoral research resulted in the following thesis-related publications (coarsely sorted by relevance, although such order is not easy to define.

⋆ indicates that A. Roitberg is an *equal first co-author*, first two authors are listed alphabetically.

1. Alina Roitberg, Chaoxiang Ma, Monica Haurilet and Rainer Stiefelhagen. **Open Set Driver Activity Recognition.** *Intelligent Vehicles Symposium (IV)*, IEEE, October 2020, 🏆**Best Student Paper Runner-Up Award**.

2. Manuel Martin⋆, Alina Roitberg⋆, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit and Rainer Stiefelhagen. **Drive&Act: A Multi-modal Dataset for Fine-grained Driver Behavior Recognition in Autonomous Vehicles.** *International Conference on Computer Vision (ICCV)*, IEEE, October 2019.

3. Alina Roitberg, Monica Haurilet, Manuel Martinez and Rainer Stiefelhagen. **Uncertainty-sensitive Activity Recognition: a Reliability Benchmark and CARING models.** *International Conference on Pattern Recognition (ICPR)*, IEEE, January 2021, **oral**.

4. Alina Roitberg, Monica Haurilet, Simon Reiß and Rainer Stiefelhagen. **From Driver *Talk* To Future *Action*: Vehicle Maneuver Prediction by Learning from Driving Exam Dialogs** *Intelligent Vehicles Symposium (IV)*, 2021, IEEE.

5. Alina Roitberg⋆, Ziad Al-Halah⋆ and Rainer Stiefelhagen. **Informed Democracy: Voting-based Novelty Detection for Action Recognition**. *British Machine Vision Conference (BMVC)*, Newcastle upon Tyne, UK, September 2018.

6. Alina Roitberg, Monica Haurilet, Simon Reiß and Rainer Stiefelhagen. **CNN-based Driver Activity Understanding - Shedding Light on Deep Spatiotemporal Representations .** *Intelligent Transportation Systems Confrence (ITSC), IEEE September, 2020.*

7. Alina Roitberg, Manuel Martinez, Monica Haurilet, Rainer Stiefelhagen. **Towards a Fair Evaluation of Zero-Shot Action Recognition using External Data**. *ECCV Workshop on Shortcomings in Vision and Language (SiVL)*, Springer, September 2018.

8. Alina Roitberg⋆, Tim Pollert⋆, Monica Haurilet and Rainer Stiefelhagen. **Analysis of Deep Fusion Strategies for Multi-modal Gesture Recognition.** *CVPR Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, IEEE, June 2019.

9. Simon Reiß*, Alina Roitberg*, Monica Haurilet and Rainer Stiefelhagen. **Deep Classification-driven Domain Adaptation for Cross-Modal Driver Behavior Recognition.** *Intelligent Vehicles Symposium (IV)*, IEEE, October 2020, **to appear**.

10. Patrick Gebert*, Alina Roitberg*, Monica Haurilet and Rainer Stiefelhagen. **End-to-end Prediction of Driver Intention using 3D Convolutional Neural Networks.** *Intelligent Vehicles Symposium (IV)*, IEEE, June 2019.

11. Simon Reiß*, Alina Roitberg*, Monica Haurilet and Rainer Stiefelhagen. **Activity-aware Attributes for Zero-Shot Driver Behavior Recognition.** *CVPR Workshop on Visual Learning with Limited Labels (VL-LL)*, IEEE, June 2020.

12. Manuel Martinez, Alina Roitberg, Daniel Koester, Boris Schauerte and Rainer Stiefelhagen. **Using Technology Developed for Autonomous Cars to Help Navigate Blind People**. *ICCV Workshop on Assistive Computer Vision and Robotics (ACVR)*, IEEE, 2017.

13. Manuel Martin, Michael Voit, Julian Ludwig, Alina Roitberg, Michael Flad, Sören Hohman and Rainer Stiefelhagen. **Innenraumbeobachtung für die kooperative Übergabe zwischen hochautomatisierten Fahrzeugen und Fahrer.** *VDI-Berichte 2360*, 2019.

14. Michael Flad, Philipp Karg, Alina Roitberg, Manuel Martin, Marcus Mazewitsch, Erdi Kenar, Lenne Ahrens, Boris Flecken, Luis Kalb, Burak Karakaya, Julian Ludwig, Achim Pruksch, Rainer Stiefelhagen, and Sören Hohmann. **Personalisation and Control Transition between Automation and Driver in Highly Automated Cars.** *Smart Automotive Mobility*, Springer, 2021. ISBN: 978-3-030-45130-1. **Book chapter.**

The following scientific publications were co-authored by Alina Roitberg during her PhD research but are out of the scope of this thesis (chronological order).

15. Robin Ruede, Verena Heusser, Lukas Frank, Alina Roitberg, Monica Haurilet, and Rainer Stiefelhagen. **Should I have another piece of cake? Multi-task Learning for Calorie Estimation**. 2020, *International Conference on Pattern Recognition (ICPR)*, IEEE, January 2021.

16. Monica Haurilet, Alina Roitberg, Simon Reiß, Manuel Martinez, Constantin Seibold, Rainer Stiefelhagen. **Node Matching for Graph Generation from Single Feature Vectors.** 2020, **under review**.

17. Monica Haurilet, Alina Roitberg, Rainer Stiefelhagen. **Towards End-to-end Document Analysis: A New Model and the SlideQA Dataset.** 2020, **under review**.

18. Monica Haurilet, Alina Roitberg, Rainer Stiefelhagen. **It's not about the Journey; It's about the Destination: Following Soft Paths under Question-Guidance for Visual Reasoning.** *Conference on Computer Vision and Pattern Recognition (CVPR)* , IEEE, 2019.

19. Monica Haurilet, Alina Roitberg, Manuel Martinez, Rainer Stiefelhagen. **WiSe - Slide Segmentation in the Wild.** *International Conference for Document Analysis and Recognition (ICDAR)*, IEEE, 2019.

20. Sidney Bender, Monica Haurilet, Alina Roitberg, Rainer Stiefelhagen. **Learning Fine-Grained Image Representations for Mathematical Expression Recognition** *ICDAR Workshop on Graphics Recognition* , IEEE, 2019.

21. Elena Wolf, Manuel Martinez, Alina Roitberg, Rainer Stiefelhagen and Barbara Deml. **Estimating Mental Load in Passive and Active Tasks from Pupil and Gaze Changes using Bayesian Surprise.** . *ICMI Modeling Cognitive Processes Workshop (ICMI-MCPMD)*, ACM, 2018.

BIBLIOGRAPHY

[1] Yehya Abouelnaga, Hesham M Eraqi, and Mohamed N Moustafa. "Real-time distracted driver posture classification." In: *NeurIPS Workshop on Machine Learning for Intelligent Transportation Systems* (2018).

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark." In: *arXiv preprint arXiv:1609.08675* (2016).

[3] David Ada Adama, Ahmad Lotfi, Caroline Langensiepen, Kevin Lee, and Pedro Trindade. "Learning human activities for assisted living robotics." In: *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. 2017, pp. 286–292.

[4] Naoki Akai, Takatsugu Hirayama, Luis Yoichi Morales, Yasuhiro Akagi, Hailong Liu, and Hiroshi Murase. "Driving Behavior Modeling Based on Hidden Markov Models with Driver's Eye-Gaze Measurement and Ego-Vehicle Localization." In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 949–956.

[5] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. "Uncertainty Estimation Using a Single Deep Deterministic Neural Network." In: (2020).

[6] Maryam Asadi-Aghbolaghi, Albert Clapes, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei, and Sergio Escalera. "A survey on deep learning based approaches for action and gesture recognition in image sequences." In: *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. IEEE. 2017, pp. 476–483.

[7] MF Augusteijn and BA Folkert. "Neural network classification and novelty detection." In: *International Journal of Remote Sensing* 23.14 (2002), pp. 2891–2902.

[8] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. "Openface 2.0: Facial behavior analysis toolkit." In: *International Conference on Automatic Face & Gesture Recognition*. 2018, pp. 59–66.

[9] Aayush Bansal, Ali Farhadi, and Devi Parikh. "Towards transparent systems: Semantic characterization of failure modes." In: *European Conference on Computer Vision*. Springer. 2014, pp. 366–381.

[10] Chad Barker. "Key findings from focus group research on inside-the-vehicle distractions in New Zealand." In: *Distracted Driving, S* (2007), pp. 213–254.

[11] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. "Weight uncertainty in neural networks." In: *arXiv preprint arXiv:1505.05424* (2015).

[12] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. "Unsupervised pixel-level domain adaptation with generative adversarial networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3722–3731.

[13] Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. "Finding Microaggressions in the Wild: A Case for Locating Elusive Phenomena in Social Media Posts." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 1664–1674.

[14] Antoine Buetti-Dinh, Vanni Galli, Sören Bellenberg, Olga Ilie, Malte Herold, Stephan Christel, Mariia Boretska, Igor V Pivkin, Paul Wilmes, Wolfgang Sand, et al. "Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition." In: *Biotechnology Reports* 22 (2019), e00321.

[15] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. "Activitynet: A large-scale video benchmark for human activity understanding." In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2015, pp. 961–970.

[16] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." In: *arXiv preprint arXiv:1812.08008*. 2018.

[17] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[18] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild." In: *European Conference on Computer Vision*. Springer. 2016, pp. 52–68.

[19] Xinlei Chen and Abhinav Gupta. "Webly supervised learning of convolutional networks." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1431–1439.

[20] Zhaoxi Chen, Gang Li, Francesco Fioranelli, and Hugh Griffiths. "Personnel recognition and gait classification based on multistatic micro-Doppler signatures using deep convolutional neural networks." In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 669–673.

[21] Shinko Y Cheng and Mohan Manubhai Trivedi. "Vision-based infotainment user determination by hand recognition for driver assistance." In: *Transactions on intelligent transportation systems* 11.3 (2010), pp. 759–764.

[22] Dhivya Chinnappa, Srikala Murugan, and Eduardo Blanco. "Extracting Possessions from Social Media: Images Complement Language." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 663–672.

[23] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. "Why Can't I Dance in the Mall? Learning to Mitigate Scene Bias in Action Recognition." In: *NeurIPS*. 2019, pp. 851–863.

[24] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. "Potion: Pose motion representation for action recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7024–7033.

[25] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. "Scaling egocentric vision: The epic-kitchens dataset." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 720–736.

[26] Cristian Danescu-Niculescu-Mizil and Lillian Lee. "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs." In: *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*. 2011.

[27] Morris H DeGroot and Stephen E Fienberg. "The comparison and evaluation of forecasters." In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.1-2 (1983), pp. 12–22.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. "ImageNet: A large-scale hierarchical image database." In: *Cvpr* (2009), pp. 248–255. ISSN: 1063-6919. DOI: 10.1109/CVPRW.2009.5206848.

[29] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification." In: *CVPR*. 2018, pp. 994–1003.

[30] Nachiket Deo and Mohan M Trivedi. "Looking at the driver/rider in autonomous vehicles to predict take-over readiness." In: *IEEE Transactions on Intelligent Vehicles* 5.1 (2019), pp. 41–52.

[31] Armen Der Kiureghian and Ove Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2 (2009), pp. 105–112.

[32] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[33] Anup Doshi and Mohan Trivedi. "A comparative exploration of eye gaze and head motion cues for lane change intent prediction." In: *IEEE Intelligent Vehicles Symposium, Proceedings* (2008), pp. 49–54. ISSN: 1931-0587. DOI: 10.1109/IVS.2008.4621321.

[34] Anup Doshi and Mohan Trivedi. "A comparative exploration of eye gaze and head motion cues for lane change intent prediction." In: *2008 IEEE Intelligent Vehicles Symposium*. IEEE. 2008, pp. 49–54.

[35] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. "FlowNet: Learning optical flow with convolutional networks." In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 2758–2766. ISSN: 15505499. DOI: 10.1109/ICCV.2015.316. arXiv: 1504.06852.

[36] Jiali Duan, Jun Wan, Shuai Zhou, Xiaoyuan Guo, and Stan Z Li. "A unified framework for multi-modal isolated gesture recognition." In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.1s (2018), p. 21.

[37] Cynthia D'Angelo, Daisy Rutstein, Christopher Harris, Robert Bernard, Evgueni Borokhovski, and Geneva Haertel. "Simulations for STEM learning: Systematic review and meta-analysis." In: *Menlo Park: SRI International* (2014).

[38] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. "What have we learned from deep representations for action recognition?" In: *CVPR*. 2018, pp. 7844–7853.

[39] Michael Flad et al. "Personalisation and Control Transition between Automation and Driver in Highly Automated Cars." In: *Smart Automotive Mobility*. ISBN: 978-3-030-45130-1. Book chapter under final review, but already listed. Springer, 2020.

[40] Laura Fontanari, Michel Gonzalez, Giorgio Vallortigara, and Vittorio Girotto. "Probabilistic cognition in two indigenous Mayan groups." In: *Proceedings of the National Academy of Sciences* 111.48 (2014).

[41] Jared A Frank and Vikram Kapila. "Mixed-reality learning environments: Integrating mobile interfaces with laboratory test-beds." In: *Computers & Education* 110 (2017), pp. 88–104.

[42] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. "Devise: A deep visual-semantic embedding model." In: *Advances in neural information processing systems*. 2013, pp. 2121–2129.

[43] Yarin Gal and Zoubin Ghahramani. "Bayesian convolutional neural networks with Bernoulli approximate variational inference." In: *arXiv preprint arXiv:1506.02158* (2015).

[44] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." In: *international conference on machine learning*. 2016, pp. 1050–1059.

[45] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep Bayesian Active Learning with Image Data." In: *Proceedings of the 34th International Conference on Machine Learning (ICML-17)*. 2017.

[46] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 409–419.

[47] Patrick Gebert*, Alina Roitberg*, Monica Haurilet, and Rainer Stiefelhagen. "End-to-end Prediction of Driver Intention using 3D Convolutional Neural Networks." In: *Intelligent Vehicles Symposium (IV)*. Paris, France: IEEE, 2019.

[48] Marcelo C Ghilardi, Gabriel Simoes, Jônatas Wehrmann, Isabel H Manssour, and Rodrigo C Barros. "Real-Time Detection of Pedestrian Traffic Lights for Visually-Impaired People." In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[49] Brian Gleeson, Karon MacLean, Amir Haddadi, Elizabeth Croft, and Javier Alcazar. "Gestures for industry intuitive human-robot communication from human observation." In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2013, pp. 349–356.

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[51] Craig Gordon. "Driver distraction: An initial examination of the 'attention diverted by'contributory factor codes from crash reports and focus group research on perceived risks." In: (2005).

[52] Ilaria Gori, JK Aggarwal, Larry Matthies, and Michael S Ryoo. "Multitype activity recognition in robot-centric scenarios." In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 593–600.

[53] Alex Graves. "Practical variational inference for neural networks." In: *Advances in neural information processing systems*. 2011, pp. 2348–2356.

[54] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On Calibration of Modern Neural Networks." In: *International Conference on Machine Learning (ICML)*. 2017. arXiv: 1706.04599. URL: http://arxiv.org/abs/1706.04599.

[55] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. "On calibration of modern neural networks." In: *International Conference on Machine Learning (ICML)*. 2017.

[56] Zongbo Hao, Qianni Zhang, Ebroul Ezquierdo, and Nan Sang. "Human action recognition by fast dense trajectories." In: *Proceedings of the 21st ACM international conference on Multimedia*. 2013, pp. 377–380.

[57] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: (2017). arXiv: 1711.09577.

[58] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*. 2018, pp. 18–22.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[61] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. "Women also snowboard: Overcoming bias in captioning models." In: *ECCV*. Springer. 2018, pp. 793–811.

[62] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." In: *Proceedings of International Conference on Learning Representations*. 2017.

[63] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural Computation*. 1997.

[64] Sepp Hochreiter and J Urgen Schmidhuber. "Long Short-Term Memory." In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. arXiv: 1206.2944.

[65] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. "Cycada: Cycle-consistent adversarial domain adaptation." In: *arXiv preprint arXiv:1711.03213* (2017).

[66] Yingxiang Huang, Wentao Li, Fima Macheret, Rodney A Gabriel, and Lucila Ohno-Machado. "A tutorial on calibration measurements and calibration models for clinical prediction models." In: *Journal of the American Medical Informatics Association* 27.4 (2020), pp. 621–633.

[67] Anja Katharina Huemer and Mark Vollrath. "Ablenkung durch fahrfremde Tätigkeiten" Machbarkeitsstudie." In: (2012).

[68] SAE International. *Automated driving: levels of driving automation are defined in new SAE international standard J3016*. 2014.

[69] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In: *arXiv preprint arXiv:1502.03167* (2015).

[70] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks." In: *Advances in neural information processing systems*. 2015, pp. 2017–2025.

[71] Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. "Car that knows before you do: Anticipating maneuvers via learning temporal driving models." In: *Proceedings of the Conference on Computer Vision*. 2015, pp. 3182–3190.

[72] Ashesh Jain, Hema S. Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. "Car that knows before you do: Anticipating maneuvers via learning temporal driving models." In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 3182–3190. ISSN: 15505499. DOI: 10.1109/ICCV.2015.364. arXiv: 1504.02789.

[73] Ashesh Jain, Hema S Koppula, Shane Soh, Bharad Raghavan, Avi Singh, and Ashutosh Saxena. "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture." In: *arXiv preprint arXiv:1601.00740* (2016).

[74] Ashesh Jain, Avi Singh, Hema S. Koppula, Shane Soh, and Ashutosh Saxena. "Recurrent Neural Networks for driver activity anticipation via sensory-fusion architecture." In: *Proceedings - IEEE International Conference on Robotics and Automation* 2016-June.Figure 2 (2016), pp. 3118–3125. ISSN: 10504729. DOI: 10.1109/ICRA.2016.7487478. arXiv: 1509.05016.

[75] Imen Jegham, Anouar Ben Khalifa, Ihsen Alouani, and Mohamed Ali Mahjoub. "Safe driving: driver action recognition using SURF keypoints." In: *2018 30th International Conference on Microelectronics (ICM)*. IEEE. 2018, pp. 60–63.

[76] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. "3D convolutional neural networks for human action recognition." In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2013), pp. 221–231.

[77] Michael Karg and Alexandra Kirsch. "A human morning routine dataset." In: *Proceedings of the international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2014, pp. 1351–1352.

[78] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale Video Classification with Convolutional Neural Networks." In: *CVPR*. 2014.

[79] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732.

[80] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2017.

[81] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in Neural Information Processing Systems*. 2017, pp. 5580–5590.

[82] Huda Khayrallah and Philipp Koehn. "On the Impact of Various Types of Noise on Neural Machine Translation." In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. 2018, pp. 74–83.

[83] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[84] Sheila G Klauer, Thomas A Dingus, Vicki L Neale, Jeremy D Sudweeks, David J Ramsey, et al. "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data." In: (2006).

[85] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. "Resource efficient 3d convolutional neural networks." In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 1910–1919.

[86] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances In Neural Information Processing Systems* (2012), pp. 1–9. ISSN: 10495258. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007. arXiv: 1102.0183.

[87] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[88] Hilde Kuehne, Ali Arslan, and Thomas Serre. "The language of actions: Recovering the syntax and semantics of goal-directed human activities." In: *Proceedings of the conference on computer vision and pattern recognition*. 2014, pp. 780–787.

[89] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. "Hmdb51: A large video database for human motion recognition." In: *High Performance Computing in Science and Engineering '12*. Springer, 2013, pp. 571–582.

[90] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. "HMDB: a large video database for human motion recognition." In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2011.

[91] Aviral Kumar and Sunita Sarawagi. "Calibration of encoder decoder models for neural machine translation." In: *arXiv preprint arXiv:1903.00802* (2019).

[92] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. "Trainable calibration measures for neural networks from kernel mean embeddings." In: *International Conference on Machine Learning*. 2018, pp. 2805–2814.

[93] P Kumar, Mathias Perrollaz, Christian Laugier, P Kumar, Mathias Perrollaz, Puneet Kumar, and Mathias Perrollaz. "Learning-based approach for online lane change intention prediction To cite this version : Learning-Based Approach for Online Lane Change Intention Prediction." In: Iv (2013), pp. 1–6.

[94] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." In: *Advances in neural information processing systems*. 2017.

[95] Ivan Laptev. "On space-time interest points." In: *International journal of computer vision* 64.2-3 (2005), pp. 107–123.

[96] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. "Learning realistic human actions from movies." In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[97] Stéphanie Lefèvre, Christian Laugier, and Javier Ibañez-Guzmán. "Exploiting map information for driver intention estimation at road intersections." In: *Intelligent Vehicles Symposium*. IEEE. 2011, pp. 583–588.

[98] Peng Li, Meiqi Lu, Zhiwei Zhang, Donghui Shan, and Yang Yang. "A Novel Spatial-Temporal Graph for Skeleton-based Driver Action Recognition." In: *2019 IEEE Intelligent Transportation Systems Conference*. IEEE. 2019.

[99] Yunan Li, Qiguang Miao, Kuan Tian, Yingying Fan, Xin Xu, Rui Li, and Jianfeng Song. "Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model." In: *International Conference on Pattern Recognition (ICPR)*. IEEE. 2016, pp. 25–30.

[100] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. "Enhancing the reliability of out-of-distribution image detection in neural networks." In: *arXiv preprint arXiv:1706.02690* (2017).

[101] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. "Spatio-temporal LSTM with trust gates for 3D human action recognition." In: *European Conference on Computer Vision*. Springer. 2016, pp. 816–833.

[102] Juncheng Liu, Zhouhui Lian, Yi Wang, and Jianguo Xiao. "Incremental Kernel Null Space Discriminant Analysis for Novelty Detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 792–800.

[103] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. "Unsupervised image-to-image translation networks." In: *Advances in neural information processing systems*. 2017, pp. 700–708.

[104] Ming-Yu Liu and Oncel Tuzel. "Coupled generative adversarial networks." In: *NIPS*. 2016, pp. 469–477.

[105] Daryl Lloyd, David Wilson, David Mais, Wilmah Deda, and Anil Bhagat. "Reported Road Casualties Great Britain: 2014 Annual Report." In: (2015).

[106] Julina Ludwig, Manuel Martin, Matthias Horne, Michael Flad, Michael Voit, Rainer Stiefelhagen, and Sören Hohmann. "Driver observation and shared vehicle control: supporting the driver on the way back into the control loop." In: *at - Automatisierungstechnik* 66(2) (2018), pp. 146 –159.

[107] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[108] David JC MacKay. "A practical Bayesian framework for backpropagation networks." In: *Neural computation* 4.3 (1992), pp. 448–472.

[109] Andrey Malinin. "Uncertainty Estimation in Deep Learning with Application to Spoken Language Assessment." PhD thesis. University of Cambridge, 2019.

[110] Bappaditya Mandal, Liyuan Li, Gang Sam Wang, and Jie Lin. "Towards detection of bus driver fatigue based on robust visual analysis of eye state." In: *IEEE Transactions on Intelligent Transportation Systems* 18.3 (2016), pp. 545–557.

[111] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. "Out-of-distribution detection for generalized zero-shot action recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9985–9993.

[112] Markos Markou and Sameer Singh. "A neural network-based novelty detector for image sequence analysis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.10 (2006), pp. 1664–1677.

[113] Manuel Martin, Johannes Popp, Mathias Anneken, Michael Voit, and Rainer Stiefelhagen. "Body pose and context information for driver secondary task detection." In: *Intelligent Vehicles Symposium*. IEEE. 2018, pp. 2015–2021.

[114] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. "Drive&Act: A Multi-modal Dataset for Fine-grained Driver Behavior Recognition in Autonomous Vehicles." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 2801–2810.

[115] Manuel Martin, Stephan Stuehmer, Michael Voit, and Rainer Stiefelhagen. "Real Time Driver Body Pose Estimation for Novel Assistance Systems." In: *International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 1738–1744.

[116] Sujitha Martin, Eshed Ohn-Bar, Ashish Tawari, and Mohan Manubhai Trivedi. "Understanding head and hand activities and coordination in naturalistic driving videos." In: *Intelligent Vehicles Symposium*. 2014, pp. 884–889.

[117] Sujitha Martin, Sourabh Vora, Kevan Yuen, and Mohan Manubhai Trivedi. "Dynamics of driver's gaze: Explorations in behavior modeling and maneuver prediction." In: *IEEE Transactions on Intelligent Vehicles* 3.2 (2018), pp. 141–150.

[118] Clara Marina Martinez, Mira Heucke, Fei-Yue Wang, Bo Gao, and Dongpu Cao. "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey." In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (2017), pp. 666–676.

[119] Manuel Martinez, Angela Constantinescu, Boris Schauerte, Daniel Koester, and Rainer Stiefelhagen. "Cognitive evaluation of haptic and audio feedback in short range navigation tasks." In: *International Conference on Computers for Handicapped Persons*. Springer. 2014, pp. 128–135.

[120] Manuel Martinez, Alina Roitberg, Daniel Koester, Rainer Stiefelhagen, and Boris Schauerte. "Using technology developed for autonomous cars to help navigate blind people." In: *International Conference on Computer Vision Workshop on Assistive Computer Vision and Robotics (ACVR)*. 2017.

[121] Manuel Martin*, Alina Roitberg*, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. "Drive&Act: A Multi-modal Dataset for Fine-grained Driver Behavior Recognition in Autonomous Vehicles." In: *International Conference on Computer Vision (ICCV)*. Seoul, South Korea: IEEE, 2019.

[122] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4040–4048.

[123] Qiguang Miao, Yunan Li, Wanli Ouyang, Zhenxin Ma, Xin Xu, Weikang Shi, and Xiaochun Cao. "Multimodal gesture recognition based on the resc3d network." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3047–3055.

[124] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. "Howto100M: Learning a text-video embedding by watching hundred million narrated video clips." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 2630–2640.

[125] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." In: 2013.

[126] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[127] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. "Cross-stitch networks for multi-task learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3994–4003.

[128] Reza Mohammadi-Ghazi, Youssef M Marzouk, and Oral Büyüköztürk. "Conditional classifiers and boosted conditional Gaussian mixture model for novelty detection." In: *Pattern Recognition* (2018).

[129] Brian Mok, Mishel Johns, Key Jung Lee, David Miller, David Sirkin, Page Ive, and Wendy Ju. "Emergency, automation off: Unstructured transition timing for distracted drivers of automated vehicles." In: *2015 IEEE 18th international conference on intelligent transportation systems*. IEEE. 2015, pp. 2458–2464.

[130] Rai Munoz, Xuejian Rong, and Yingli Tian. "Depth-aware indoor staircase detection and recognition for the visually impaired." In: *2016 IEEE international conference on multimedia & expo workshops (ICMEW)*. IEEE. 2016, pp. 1–6.

[131] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. "Image to image translation for domain adaptation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4500–4509.

[132] Roozbeh Nabiei, Manish Parekh, Emilie Jean-Baptiste, Peter Jancovic, and Martin Russell. "Object-centred recognition of human activity." In: *2015 International Conference on Healthcare Informatics*. IEEE. 2015, pp. 63–68.

[133] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. "Obtaining well calibrated probabilities using bayesian binning." In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[134] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. "Seeing voices and hearing faces: Cross-modal biometric matching." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8427–8436.

[135] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[136] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[137] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 427–436.

[138] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting good probabilities with supervised learning." In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 625–632.

[139] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. "Zero-shot learning by convex combination of semantic embeddings." In: *arXiv preprint arXiv:1312.5650* (2013).

[140] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition." In: *Journal of Visual Communication and Image Representation* 25.1 (2014), pp. 24–38.

[141] Eshed Ohn-Bar, Sujitha Martin, Ashish Tawari, and Mohan Manubhai Trivedi. "Head, eye, and hand patterns for driver activity recognition." In: *International Conference on Pattern Recognition*. 2014, pp. 660–665.

[142] Eshed Ohn-Bar and Mohan Manubhai Trivedi. "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations." In: *Transactions on intelligent transportation systems* 15.6 (2014), pp. 2368–2377.

[143] Eshed Ohn-Bar and Mohan Manubhai Trivedi. "Looking at humans in the age of self-driving and highly automated vehicles." In: *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pp. 90–104.

[144] Yuichi Ohta and Hideyuki Tamura. *Mixed reality: merging real and virtual worlds*. Springer Publishing Company, Incorporated, 2014.

[145] Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. "Analyzing uncertainty in neural machine translation." In: *arXiv preprint arXiv:1803.00047* (2018).

[146] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift." In: *Advances in Neural Information Processing Systems*. 2019, pp. 13991–14002.

[147] Pau Panareda Busto and Juergen Gall. "Open set domain adaptation." In: *International Conference on Computer Vision*. IEEE, 2017.

[148] Alexander Perzylo, Markus Rickert, Bjoern Kahl, Nikhil Somani, Christian Lehmann, Alexander Kuss, Stefan Profanter, Anders Billeso Beck, Mathias Haage, Mikkel Rath Hansen, et al. "SMErobotics: Smart robots for flexible manufacturing." In: *IEEE Robotics & Automation Magazine* 26.1 (2019), pp. 78–90.

[149] Alexander Perzylo, Nikhil Somani, Stefan Profanter, Ingmar Kessler, Markus Rickert, and Alois Knoll. "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production." In: *2016 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE. 2016, pp. 2293–2300.

[150] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. "A review of novelty detection." In: *Signal Processing* 99 (2014), pp. 215–249.

[151] John Platt et al. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." In: *Advances in large margin classifiers* 10.3 (1999), pp. 61–74.

[152] Jie Qin, Li Liu, Ling Shao, Fumin Shen, Bingbing Ni, Jiaxin Chen, and Yunhong Wang. "Zero-shot action recognition with error-correcting output codes." In: *Proc. CVPR*. 2017.

[153] Zhaofan Qiu, Ting Yao, and Tao Mei. "Learning spatio-temporal representation with pseudo-3d residual networks." In: *IEEE International Conference on Computer Vision (ICCV)*. 2017.

[154] Jonas Radlmayr, Christian Gold, Lutz Lorenz, Mehdi Farid, and Klaus Bengler. "How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 58. 1. 2014, pp. 2063–2067.

[155] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. "Data distillation: Towards omni-supervised learning." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4119–4128.

[156] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling." In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE. 2017, pp. 1025–1032.

[157] Akshay Rangesh, Bowen Zhang, and Mohan M Trivedi. "Driver Gaze Estimation in the Real World: Overcoming the Eyeglass Challenge." In: *Intelligent Vehicles Symposium* (2020).

[158] Carl Edward Rasmussen. "Gaussian processes in machine learning." In: *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[159] Fitsum Reda, Robert Pottorff, Jon Barker, and Bryan Catanzaro. *flownet2-pytorch: Pytorch implementation of FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*. 2017.

[160] Simon Reiß*, Alina Roitberg*, Monica Haurilet, and Rainer Stiefelhagen. "Activity-aware Attributes for Zero-Shot Driver Behavior Recognition." In: *International Conference on Computer Vision and Pattern Recognition Workshop on Visual Learning with Limited Labels (VL-LL)*. IEEE, 2020.

[161] Simon Reiß*, Alina Roitberg*, Monica Haurilet, and Rainer Stiefelhagen. "Deep Classification-driven Domain Adaptation for Cross-Modal Driver Behavior Recognition." In: *Intelligent Vehicles Symposium (IV)*. IEEE, 2020.

[162] Stephen Reynolds, Matthew Tranter, Paul Baden, David Mais, Amardeep Dhani, Elizabeth Wolch, and Anil Bhagat. *Reported Road Casualties Great Britain: 2016 Annual Report*. Tech. rep. September. UK Department for Transport, 2017, p. 361.

[163] Charles Richter and Nicholas Roy. "Safe visual navigation via deep learning and novelty detection." In: *Proc. of the Robotics: Science and Systems Conference*. 2017.

[164] Stephen E Robertson and K Sparck Jones. "Relevance weighting of search terms." In: *Journal of the American Society for Information science* 27.3 (1976), pp. 129–146.

[165] Stephen Robertson. "Understanding inverse document frequency: on theoretical arguments for IDF." In: *Journal of documentation* (2004).

[166] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. "Action Mach: a spatio-temporal maximum average correlation height filter for action recognition." In: *Conference on Computer Vision and Pattern Recognition*. 2008.

[167] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. "A database for fine grained activity detection of cooking activities." In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1194–1201.

[168] Alina Roitberg, Ziad Al-Halah, and Rainer Stiefelhagen. "Informed Democracy: Voting-based Novelty Detection for Action Recognition." In: *British Machine Vision Conference (BMVC)*. Newcastle upon Tyne, UK, 2018.

[169] Alina Roitberg, Monica Haurilet, Manuel Martinez, and Rainer Stiefelhagen. "Uncertainty-sensitive Activity Recognition: a Reliability Benchmark and CARING models." In: *International Conference on Pattern Recognition (ICPR)*. IEEE, 2021.

[170] Alina Roitberg, Monica Haurilet, Simon Reiß, and Rainer Stiefelhagen. "CNN-based Driver Activity Understanding - Shedding Light on Deep Spatiotemporal Representations." In: *Intelligent Transportation Systems Confrence (ITSC)*. 2020.

[171] Alina Roitberg, Monica Haurilet, Simon Reiß, and Rainer Stiefelhagen. "From Driver *Talk* To Future *Action*: Vehicle Maneuver Prediction by Learning from Driving Exam Dialogs." In: 2021.

[172] Alina Roitberg, Chaoxiang Ma, Monica Haurilet, and Rainer Stiefelhagen. "Open Set Driver Activity Recognition." In: *Intelligent Vehicles Symposium (IV)*. IEEE, 2020.

[173] Alina Roitberg, Manuel Martinez, Monica Haurilet, and Rainer Stiefelhagen. "Towards a Fair Evaluation of Zero-Shot Action Recognition using External Data." In: *European Conference on Computer Vision Workshop on Shortcomings in Vision and Language (SiVL)*. Springer.

[174] Alina Roitberg, Alexander Perzylo, Nikhil Somani, Manuel Giuliani, Markus Rickert, and Alois Knoll. "Human activity recognition in the context of industrial human-robot interaction." In: *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE. 2014, pp. 1–10.

[175] Alina Roitberg, Tim Pollert, Monica Haurilet, Manuel Martin, and Rainer Stiefelhagen. "Analysis of Deep Fusion Strategies for Multi-Modal Gesture Recognition." In: *International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 2019.

[176] Alina Roitberg, Nikhil Somani, Alexander Perzylo, Markus Rickert, and Alois Knoll. "Multimodal human activity recognition for industrial manufacturing processes in robotic workcells." In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM. 2015.

[177] Lukas Rybok, Simon Friedberger, Uwe D. Hanebeck, and Rainer Stiefelhagen. "The KIT Robo-Kitchen Data set for the Evaluation of View-based Activity Recognition Systems." In: *IEEE-RAS International Conference on Humanoid Robots*. 2011.

[178] Lukas Rybok, Boris Schauerte, Ziad Al-Halah, and Rainer Stiefelhagen. ""Important stuff, everywhere!" Activity recognition with salient proto-objects as context." In: *IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2014, pp. 646–651.

[179] Allah Bux Sargano, Plamen Angelov, and Zulfiqar Habib. "A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition." In: *applied sciences* 7.1 (2017), p. 110.

[180] Boris Schauerte, Daniel Koester, Manel Martinez, and Rainer Stiefelhagen. "Way to go! Detecting open areas ahead of a walking person." In: *European Conference on Computer Vision*. Springer. 2014, pp. 349–360.

[181] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. "Estimating the support of a high-dimensional distribution." In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[182] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. "Support vector method for novelty detection." In: *Advances in neural information processing systems*. 2000, pp. 582–588.

[183] Christian Schuldt, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. IEEE. 2004, pp. 32–36.

[184] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization." In: *ICCV*. 2017, pp. 618–626.

[185] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[186] Karen Simonyan and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In: *Advances in neural information processing systems*. 2014, pp. 568–576.

[187] Santokh Singh. "Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey." In: *National Highway Traffic Safety Administration* February (2015), pp. 1–2. DOI: https://doi.org/10.1016/j.trf.2015.04.014.

[188] Santokh Singh. *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*. Tech. rep. Stats (National Highway Traffic Safety Administration, Washington, DC), Report No. DOT HS 812 115., 2015.

[189] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. "Zero-shot learning through cross-modal transfer." In: *Advances in neural information processing systems*. 2013, pp. 935–943.

[190] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." In: *arXiv preprint arXiv:1212.0402* (2012).

[191] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[192] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. "The limits and potentials of deep learning for robotics." In: *The International Journal of Robotics Research* 37.4-5 (2018).

[193] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks." In: *arXiv preprint arXiv:1312.6199* (2013).

[194] Ashish Tawari, Sayanan Sivaraman, Mohan Manubhai Trivedi, Trevor Shannon, and Mario Tippelhofer. "Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking." In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 115–120.

[195] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles SAE J 3016*. Society of Automotive Engineers (SAE), 2016.

[196] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3d convolutional networks." In: *ICCV*. 2015, pp. 4489–4497.

[197] Gul Varol, Ivan Laptev, and Cordelia Schmid. "Long-term temporal convolutions for action recognition." In: *IEEE transactions on pattern analysis and machine intelligence* (2017).

[198] Jun Wan, Sergio Escalera, Gholamreza Anbarjafari, Hugo Jair Escalante, Xavier Baró, Isabelle Guyon, Meysam Madadi, Juri Allik, Jelena Gorbova, Chi Lin, et al. "Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3189–3197.

[199] Jun Wan, Yibing Zhao, Shuai Zhou, Isabelle Guyon, Sergio Escalera, and Stan Z Li. "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 56–64.

[200] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. "Action recognition by dense trajectories." In: *CVPR 2011*. IEEE. 2011, pp. 3169–3176.

[201] Heng Wang and Cordelia Schmid. "Action recognition with improved trajectories." In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE. 2013, pp. 3551–3558.

[202] Hongsong Wang and Liang Wang. "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[203] Huogen Wang, Pichao Wang, Zhanjie Song, and Wanqing Li. "Large-scale multimodal gesture recognition using heterogeneous networks." In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 3129–3137.

[204] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal segment networks: Towards good practices for deep action recognition." In: *European Conference on Computer Vision*. Springer. 2016, pp. 20–36.

[205] Qian Wang and Ke Chen. "Zero-shot visual recognition via bidirectional latent embedding." In: *International Journal of Computer Vision* 124.3 (2017), pp. 356–383.

[206] Shuihua Wang and Yingli Tian. "Detecting stairs and pedestrian crosswalks for the blind by RGBD camera." In: *2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops*. IEEE. 2012, pp. 732–739.

[207] Joe H Ward Jr. "Hierarchical grouping to optimize an objective function." In: *Journal of the American statistical association* 58 (1963).

[208] Patrick Weyers, David Schiebener, and Anton Kummert. "Action and Object Interaction Recognition for Driver Activity Classification." In: *2019 IEEE Intelligent Transportation Systems Conference*. IEEE. 2019.

[209] Graham Williams, Rohan Baxter, Hongxing He, Simon Hawkins, and Lifang Gu. "A comparative study of RNN for outlier detection in data mining." In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE. 2002, pp. 709–712.

[210] Elena Wolf, Manuel Martinez, Alina Roitberg, Rainer Stiefelhagen, and Barbara Deml. "Estimating mental load in passive and active tasks from pupil and gaze changes using bayesian surprise." In: *ICMI-MCPMD*. 2018.

[211] Peter Wozniak. "Range imaging based obstacle detection for virtual environment systems and interactive metaphor based signalization." PhD thesis. 2019.

[212] Yongqin Xian, Bernt Schiele, and Zeynep Akata. "Zero-shot learning-the good, the bad and the ugly." In: *arXiv preprint arXiv:1703.04394* (2017).

[213] Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, and Fei-Yue Wang. "Driver activity recognition for intelligent vehicles: A deep learning approach." In: *IEEE Transactions on Vehicular Technology* 68.6 (2019), pp. 5379–5390.

[214] Lijie Xu and Kikuo Fujimura. "Real-time driver activity recognition with random forests." In: *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM. 2014.

[215] Xun Xu, Timothy Hospedales, and Shaogang Gong. "Semantic embedding space for zero-shot action recognition." In: *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 63–67.

[216] Xun Xu, Timothy Hospedales, and Shaogang Gong. "Transductive Zero-Shot Action Recognition by Word-Vector Embedding." In: *International Journal of Computer Vision* (2017), pp. 1–25.

[217] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. "PA3D: Pose-action 3D machine for video recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7922–7931.

[218] Chao Yan, Frans Coenen, and Bailing Zhang. "Driving posture recognition by convolutional neural networks." In: *IET Computer Vision* (2016).

[219] Fei Yan, Mark Eilers, Lars Weber, and Martin Baumann. "Investigating Initial Driver Intention on Overtaking on Rural Roads." In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 4354–4359.

[220] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "Spatial temporal graph convolutional networks for skeleton-based action recognition." In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[221] Kailun Yang, Kaiwei Wang, Weijian Hu, and Jian Bai. "Expanding the detection of traversable area with RealSense for the visually impaired." In: *Sensors* 16.11 (2016), p. 1954.

[222] Jie Ying, Jin Tian, and Lei Lei. "Traffic light detection based on similar shapes searching for visually impaired person." In: *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*. IEEE. 2015, pp. 376–380.

[223] Bianca Zadrozny and Charles Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers." In: *Icml*. Vol. 1. Citeseer. 2001, pp. 609–616.

[224] Bianca Zadrozny and Charles Elkan. "Transforming classifier scores into accurate multiclass probability estimates." In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 694–699.

[225] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. "View Adaptive Neural Networks for High Performance Skeleton-based Human Action Recognition." In: *Transactions on Pattern Analysis and Machine Intelligence* (2019). ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2896631.

[226] CH Zhao, BL Zhang, Jianhong He, and J Lian. "Recognition of driving postures by contourlet transform and random forests." In: *IET Intelligent Transport Systems* 6.2 (2012), pp. 161–168.

[227] Chun Zhu and Weihua Sheng. "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living." In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41.3 (2011), pp. 569–573.

[228] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[229] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. "Towards Universal Representation for Unseen Action Recognition." In: (2018).

[230] Farbod Zorriassatine, Amin Al-Habaibeh, RM Parkin, MR Jackson, and J Coy. "Novelty detection for practical pattern recognition in condition monitoring of multivariate processes: a case study." In: *The International Journal of Advanced Manufacturing Technology* 25.9-10 (2005), pp. 954–963.