LEUPHANA
UNIVERSITY LÜNEBURG

Detecting and Assessing Road Damages for Autonomous Driving Utilizing Conventional Vehicle Sensors

Faculty of Business and Economics at Leuphana University Lüneburg,
submitted as a requirement for the award of the title of

Doctor of Natural Sciences
– Dr. rer. nat. –

Approved thesis by
Felix Kortmann

born April 21, 1991 in Hamm, Germany

Submitted on: Tuesday, 26 October 2021

Thesis defence on: Wednesday, 22 December 2021

Main supervisor and reviewer: Prof. Dr. Paul Drews
Second supervisor and reviewer: Prof. Dr. Burkhardt Funk
Third reviewer: Prof. Dr. Yoshihide Sekimoto

The individual items in the cumulative thesis are or will be published as follows, if applicable with the accompanying paper:

Publication I - Data & Knowledge Engineering 2021 - Submitted Journal Paper (under review):
Felix Kortmann, Pascal Fassmeyer, Burkhardt Funk and Paul Drews, "Watch out, pothole! Featuring Road Damage Detection in an End-to-end System for Autonomous Driving",
Preprint DOI: 10.13140/RG.2.2.29894.19527.

Publication II - IEEE Sensors 2019 - Published Conference Paper:
Felix Kortmann, Henning Peitzmeier, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk and Paul Drews, "Enabling Road Condition Monitoring with an on-board Vehicle Sensor Setup", IEEE Sensors,
DOI: 10.1109/SENSORS43011.2019.8956699.

Publication III - IEEE Vehicular Technology Conference 2020 Fall - Published Conference Paper:
Felix Kortmann, Malte Rodeheger, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk and Paul Drews, "Applying Quarter-Vehicle Model Simulation for Road Elevation Measurements Utilizing the Vehicle Level Sensor", DOI: 10.1109/VTC2020-Fall49728.2020.9348664.

Publication IV - IEEE Sensors 2020 - Published Conference Paper:
Felix Kortmann, Julin Horstkötter, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk and Paul Drews, "Live Demonstration: Passive Sensor Setup for Road Condition Monitoring",
DOI: 10.1109/SENSORS47125.2020.9278776.

Publication V - IEEE Sensors Journal 2021 - Published Journal Paper:
Felix Kortmann, Julin Horstkötter, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, Paul Drews, "Modeling the Quarter-Vehicle: Use of Passive Sensor Data for Road Condition Monitoring",
DOI: 10.1109/JSEN.2020.3042620.

Publication VI - IEEE Intelligent Transportation Systems Conference 2020 - Published Conference Paper:
Felix Kortmann, Yi-Chen Hsu, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk and Paul Drews, "Creating Value from in-Vehicle Data: Detecting Road Surfaces and Road Hazards",
DOI: 10.1109/ITSC45102.2020.9294684.

Publication VII - IEEE International Conference on Big Data 2020 - Published Conference Paper:
Felix Kortmann, Kevin Talits, Pascal Fassmeyer, Alexander Warnecke, Nicolas Meier, Jens Heger, Paul Drews and Burkhardt Funk, "Detecting Various Road Damage Types in Global Countries Utilizing Faster RCNN",
DOI: 10.1109/BigData50022.2020.9378245.

Publication VIII - IEEE Vehicular Technology Conference 2021 Fall - Accepted Conference Paper:
Pascal Fassmeyer, Felix Kortmann, Paul Drews and Burkhardt Funk, "Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN",
Preprint DOI: 10.13140/RG.2.2.20745.36964.

Publication IX - IEEE Transactions on Intelligent Vehicles 2021 - Submitted Journal Paper (under review):
Felix Kortmann, Kevin Talits, Aalexander Warnecke, Nicolas Meier, Jens Heger, Paul Drews, and Burkhardt Funk, "The Value of Deep Learning Tools in Object Detection: YOLOv5 in a Road Damage Use Case",
Preprint DOI: 10.13140/RG.2.2.23183.30880.

Publication X - IEIE/IEEE International Conference on Electronics, Information, and Communication 2020 - Published Conference Paper:
Felix Kortmann, Daniel Brieske, Pia Piekarek, Julian Eckstein, Alexander Warnecke, Paul Drews and Dirk Uwe Sauer, "A Cloud State Modeling System for Lead-Acid Batteries: Theory and Prototyping",
DOI: 10.1109/ICEIC51217.2021.9369785.

Year of publication: 2021

Abstract

Motivation: Environmental perception is one of the biggest challenges in autonomous driving to move inside complex traffic situations properly. Perceiving the road's condition is necessary to calculate the drivable space; in manual driving, this is realized by the human visual cortex. Enabling the vehicle to detect road conditions is a critical and complex task from many perspectives. The complexity lies on the one hand in the development of tools for detecting damage, ideally using sensors already installed in the vehicle, and on the other hand, in integrating detected damages into the autonomous driving task and thus into the subsystems of autonomous driving. High Definition Feature Maps, for instance, should be prepared for mapping road damages, which includes online and in-vehicle implementation. Furthermore, the motion planning system should react based on the detected damages to increase driving comfort and safety actively. Road damage detection is essential, especially in areas with poor infrastructure, and should be integrated as early as possible to enable even less developed countries to reap the benefits of autonomous driving systems. Besides the application in autonomous driving, an up-to-date solution on assessing road conditions is likewise desirable for the infrastructure planning of municipalities and federal states to make optimal use of the limited resources available for maintaining infrastructure quality.

Research Approach: Addressing the challenges mentioned above, the research approach of this work is pragmatic and problem-solving. In designing technical solutions for road damage detection, we conduct applied research methods in engineering, including modeling, prototyping, and field studies. We utilize design science research to integrate road damages in an end-to-end concept for autonomous driving while drawing on previous knowledge, the application domain requirements, and expert workshops.

Contribution: This thesis provides various contributions to theory and practice. We design two individual solutions to assess road conditions with existing vehicle sensor technology. The first solution is based on calculating the quarter-vehicle model utilizing the vehicle level sensor and an acceleration sensor. The novel model-based calculation measures the road elevation under the tires, enabling common vehicles to assess road conditions with standard hardware. The second solution utilizes images from front-facing vehicle cameras to detect road damages with deep neural networks. Despite other research in this area, our algorithms are designed to be applicable on edge devices in autonomous vehicles with limited computational resources while still delivering cutting-edge performance. In addition, our analyses of deep learning tools and the introduction of new data into training provide valuable opportunities for researchers in other application areas to develop deep learning algorithms to optimize detection performance and runtime. Besides detecting road damages, we provide novel algorithms for classifying the severity of road damages to deliver additional information for improved motion planning. Alongside the technical solutions, we address the lack of an end-to-end solution for road damages in autonomous driving by providing a concept that starts from data generation and ends with servicing the vehicle motion planning. This includes solutions for detecting road damages, assessing their severity, aggregating the data in the vehicle and a cloud platform, and making the data available via that platform to other vehicles.

Limitations: Fundamental limitations in this dissertation are due to boundaries in modeling. Our pragmatic approach simplifies reality, which always distorts the degree of truth in the result. This affects the model building of the quarter-vehicle and deep learning. Further limitations occur in the end-to-end concept. This represents the integration of road damages in the autonomous driving task but does not detail the aggregation modules and interfaces of the subsystems.

Future Research: The completion of this work does not conclude the topic of road damage detection and assessment in autonomous driving. Research must continue to optimize the proposed solutions and test them on a widespread basis in the real world. Furthermore, the sensor fusion of different approaches is fascinating in order to combine the advantages of individual systems. Integrating the end-to-end concept into the ecosystem of an autonomous vehicle is another fascinating field, taking interfaces and cloud platforms into account.

Keywords

Road damage detection, Road damage severity, modeling, quarter-vehicle, vehicle dynamic sensor, computer vision, deep learning, autonomous driving architecture

Acknowledgement

My first and special thanks goes to my thesis supervisors. I would like to thank Professor Paul Drews for his trust and the great degree of freedom he allowed me in my research, his highly differentiated criticism, and the broad insight he gave me into a wide range of topics in Information Systems.

I would also like to thank Professor Burkhardt Funk, he has had a great influence on the topics of the dissertation. I especially thank him for the excellent scientific teamwork, the insights into teaching, and the great support of my DAAD-funded research stay at Dartmouth College (USA) following the doctorate.

I would like to thank Professor Jens Heger for his continued support, even in difficult times during the pandemic, and his contribution to the publications. My thanks also include his input from an engineering perspective, which was indispensable for this highly interdisciplinary research topic. The friendly, appreciative atmosphere with all three supervisors has been incredibly motivating and already serves as a role model for me in supervising my students.

Furthermore, I would like to thank Professor Yoshihide Sekimoto for reviewing and I am grateful for his expertise in the dissertations' domain.

Another special thanks goes to my laboratory manager, Dr. Alexander Warnecke, for the great conditions in the laboratory. He always supported me to fully concentrate on research. Furthermore, his advice and experience were invaluable in getting me through difficult periods of the dissertation. The same counts for Dr. Nicolas Meier. I am glad to have always found an open ear with him and thank him not only for the scientific exchange but also for his support in organizational matters.

Thank you very much to my fellow doctoral researchers, my students and the staff of the advanced engineering department at HELLA GmbH & Co. KGaA for the great job they did working with me. Many thanks also to Brigitte Krause for her reliance and support over many years, it has always been a pleasure.

Lastly, I would like to thank my parents, Heidi and Peter, my brother Florian and my girlfriend Anna for their support. I am invaluably grateful for their patience and understanding.

. . . .

Author's signature

Contents

Preamble

Publications

Appendix

List of Tables

List of Figures

List of Abbreviations

AD Autonomous Driving . . . 21

Adam Adaptive Moment Estimation . . . 50

ADAS Advanced Driver Assistance System . . . 85

ADS Automotive Dataset . . . 11

AI Artificial Intelligence . . . 61

ANN Artificial Neural Network . . . 49

AP Average Precision . . . 257

AS Acceleration Sensor . . . 11

AV Autonomous Vehicle . . . 22

AWS Amazon Web Services . . . 184

BCE Binary Cross-Entropy . . . 251

BMS Battery Management System . . . 274

CAN-Bus Controller Area Network Bus 182

CAV Connected and Autonomous Vehicle 24

CDS Challenge Dataset 107

CIPOS Contactless Inductive Position Sensor 47

CIoU Complete Intersection over Union 101

CNN Convolutional Neural Network 49

CSPN Cross Stage Partial Network 249

COCO Common Objects in Context 250

CV Computer Vision 27

DA Data Augmentation 62

DL Deep Learning 24

DNN Deep Neural Network 29

DSR Design Science Research 29

EEC Electrical Equivalent Circuit 274

EIS Electrochemical Impedance Spectroscopy 275

E2E End-to-end ... 27

FLOPS Floating Point Operations Per Second ... 249

FN False Negatives ... 205

FP False Positives ... 205

FPN Feature Pyramid Network ... 55

FP16 half-precision floating-point format ... 253

FP32 single-precision floating-point format ... 253

FRCNN Faster Region-based Convolutional Neural Network ... 29

GCM Golden Car Model ... 142

GIoU Generalized Intersection over Union ... 101

GPS Global Positioning System ... 88

GPU Graphics Processing Unit ... 55

GRDDC Global Road Damage Detection Challenge ... 29

HE Hyperparameter Evolution ... 62

HD Feature Map High Definition Feature Map ... 24

IBS	Intelligent Battery Sensor	273
IoU	Intersection over Union	52
IRI	International Roughness Index	44
IS	Information System	85
LAB	Lead-Acid Battery	272
LIDAR	Laser Detection and Ranging	38
LIN bus	Local Interconnect Network-Bus	274
LPWAN	Low Power Wide Area Network	272
LSTM	Long Short-Term Memory	68
mAP	mean Average Precision	34
MEMS	Micro Electro Mechanical System	48
ML	Machine Learning	24
NN	Neural Network	39
OSM	Open Street Map	184
OMS	Online Mapping System	25

PANet Path Aggregation Network 249

PCB Printed Circuit Board 47

PSD Power Spectral Density 137

PWM Pulse-Width Modulation 47

QVM Quarter Vehicle Model 27

RADAR Radio Detection and Ranging 38

RBF Radial Basis Function 185

RDD Road Damage Detection 23

RDDS Road Damage Dataset 29

RDS Road Damage Severity 29

ReLU Rectified Linear Unit 105

RMSE Root Mean Square Error 144

RoI Region of Interest 51

RPN Region Proposal Network 51

R-CNN Region Based Convolutional Neural Network 11

SAE	Society of Automotive Engineers	22
SGD	Stochastic Gradient Descent	50
SiLU	Sigmoid Linear Units	249
SoC	State of Charge	274
SoF	State of Function	274
SoH	State of Health	274
SVM	Support Vector Machine	92
TL	Transfer Learning	68
TN	True Negatives	205
TP	True Positives	205
TTA	Test Time Augmentation	62
VAE	Variational Autoencoder	104
VLS	Vehicle Level Sensor	27
VMPS	Vehicle Motion Planning System	22
VPS	Vehicle Perception System	25
YOLO	You Only Look Once	29

Preamble

Chapter 1

Introduction

1.1 Motivation

Autonomous Driving (AD) will fundamentally change mobility as we know it today. This applies to the labor sectors in public transportation and logistics, which are undergoing a major transformation and facing new challenges [1]. Private transportation is also transforming through improved on-demand shared services and the creation of leisure time on the way to work and on vacation. To master the AD task, major research efforts are required for the development of AD technology. Due to its diverse impact on numerous areas of society and the economy, autonomous driving is considered a key technology for social change with far-reaching consequences [2].

Media coverage over the last decade reveals that AD was expected to be realized more quickly. The Guardian proposed in 2015 that we will be "permanent backseat drivers" by 2020 [3], and a Forbes article from 2017 predicted that "10 million self-driving cars will hit the road by 2020" [4]. Despite significant advances in technology, the automotive industry (e.g., General Motors Company, Waymo LLC, Tesla Inc., Honda Motor Company, Toyota Motor Corporation) has not been able to keep its promises to make AD a serial feature by 2020.

> *"Google is working on self-driving cars, and they seem to work. People are so bad at driving cars that computers don't have to be that good to be much better."*
>
> *Marc Andreessen, Venture Capitalist, 2011*

Andreessen's quote exposes that just being better than manual driving is not enough to spur widespread application of AD, which is already largely safer than manual driving [5]. However, the extent of the social and legal hurdles that must be overcome in order to make AD possible has been underestimated, despite the high degree of maturity of the technology. Thus, there is a large technological gap between being better than manual driving and being good enough for social and legal acceptance [6].

It is necessary to examine this topic from an ethical point of view because the use of AD technology requires regulation and ethical guidelines for successful deployment. Recent research has investigated moral decision making in various studies, e.g., Greene et al. 2016 [7], Conitzer et al. 2017[8], and Awad et al. 2018 [9]. Ethics committees in most industrialized countries are also dealing with the matter. According to the Society of Automotive Engineers (SAE) [10], the necessity for regulation and ethical guidelines is especially marked for high levels of automation (levels 4 and 5, cf. Section 2.1 and Appendix A). In Germany, for instance, this implies that AD must embody the following characteristics: a defensive and anticipatory driving style, a preference for property damage over personal injury, no qualification of people characteristics, and an overall strongly positive risk balance considering the number of accidents and fatalities [6].

One strategy employed by automotive manufacturers is to initially apply AD in scenarios of limited complexity. This means, for instance, that the early series-produced vehicles will drive autonomously on freeways and highways and will transition to driver's control when exiting onto rural roads or into urban traffic. The more complex the traffic situation, the more demanding it is for any Autonomous Vehicle (AV) to navigate reliably. AD involves a recurring perception-planning-action process executed by different technologies and components in the vehicle [11]. In practical terms, this means that the environment is perceived, the trajectory is planned on the basis of the environment, and the motion is executed in an iterative process by the vehicle's dynamic system.

The vehicle perception system is responsible for perceiving the environment, including road lanes, road signage, traffic lights, other vehicles, and many more objects. An essential part of the Vehicle Motion Planning System (VMPS) is the calculation of the "drivable space" based on the perceived environment [12, 13, 14]. This requires the detection of road quality and any significant damage to the road [15, 16]. Without this feature, AD would only be feasible in areas with rather good road quality. To overcome the technical challenge of autonomously driving the vehicle in poor road conditions, these conditions must be properly recorded. This technical challenges also have social consequences, as AD should not only be realized in industrialized countries but also in developing countries, which tend to have infrastructure of worse quality.

The recognition of road conditions has not yet been sufficiently implemented in the perception system of AV. We determine this from the literature (e.g., Koch et al. 2015 [17]) and from the expert workshops we conducted, as described in Publication I. Even Tesla, Inc., famously an early adopter of camera-based object detection, announced as recently as 2020 via a tweet from CEO Elon Musk that they are labeling road damages in their database in order to react to them proactively and increase the safety and comfort of vehicle occupants.

> *"Yes! We're labeling bumps & potholes, so the car can slow down or steer around them when safe to do so."*
>
> *Elon Musk, CEO of Tesla, 2020*

The implementation of Road Damage Detection (RDD) algorithms in series-produced vehicles has been delayed for many reasons. One is undoubtedly the aging electronic and electric vehicle architecture and the current efforts of traditional automotive manufacturers to facilitate centralized computing capacities. The days of a fragmented, control unit-based vehicle architecture seem to be numbered. As a consequence, developers and researchers are still struggling to find a solution for RDD.

A potential solution is the resource-saving use of already existing vehicle sensor systems. We find this particularly significant, as from a philosophical point of view it serves a pragmatic approach and leans on critical inquiry and the principle of abduction, which encourages looking for immediate solutions first in order to solve problems [18]. Compared to commercial solutions that record road quality, which cities and municipalities spend a lot of money on in order to monitor their infrastructure [19, 20], our technology proposed in this dissertation can record data in real-time. Thus, degradation of the road quality can be recorded transparently and accelerated changes in quality due to weather conditions can be quickly tracked to improve vehicle navigation, which also benefits the infrastructural planning of municipalities.

Although this dissertation focuses primarily on new road condition monitoring technologies, it also examines the lack of an end-to-end system for using such road condition data to support AD. Consequently, we address both the challenges specific to vehicle architecture and the technological problems related to the application of RDD.

1.2 Problem Statements

The growing number of components is to some extent a required practice for car manufacturers; the addition of new features increases safety. However, this trend has fallen out of favor with manufacturers, as it is a significant cost driver [21]. How, then, can we effectively integrate new features, in our case explicit monitoring of road conditions in real-time, into the vehicle without having to install additional hardware?

Solutions for the real-time detection of road conditions in AD that leverage existing vehicle sensor technology are missing.

Problem 1

Studies on the detection of road damage have been conducted in the past, especially camera-based research, but the solutions have so far been geared towards detection performance. As a result, they ignore the limited computing resources in vehicles. Lightweight Machine Learning (ML) solutions with hand-crafted features cannot keep up with Deep Learning (DL) approaches and therefore fall out of consideration. A low-cost and lightweight approach becomes essential when considering how many tasks AVs must perform in parallel. In the overall context of the AD task, RDD is only a tiny part of successful implementation, so a solution should only take up an incremental share of the total computing power. According to Koch et al. [17], past research did not satisfactorily achieve road condition and defects assessment in 2015. The workshops with experts in AD that are described in Publication I confirm that this statement still holds today.

No satisfactory high-performance and lightweight solution exists for sensing road conditions with conventional front-facing vehicle camera sensors.

Problem 2

Connected and Autonomous Vehicles (CAVs) draw information from their on-board sensor systems. High Definition Feature Maps (HD Feature Maps) are critical to validation of local detections. Many modern vehicles are capable of sensing the environment continuously. These multiple sources lead to redundant information in the online HD Feature Map and therefore increase the confidence about the presence of objects in road traffic. Yet, no end-to-end concept that incorporates from data generation to delivery of information for vehicle motion planning is currently available. The technologies for detecting road damage are likewise neither clearly defined nor embedded in a concept. If road damage and possible hazards are to be successfully addressed in AD, the new features must be integrated into the existing AD pipeline (cf. Section 2.1.3).

No end-to end CAVs concept that addresses road damage in the on-board and online integration in HD Feature Maps is currently available.

Problem 3

A lightweight concept that integrates road damage features into the architecture of modern AVs, especially regarding the VMPS, the Vehicle Perception System (VPS), and the Online Mapping System (OMS), has not yet been formulated.

1.3 Research Questions

Based on the three problems discussed in the previous section, we generate research questions that we address through our research. Our research questions are designed according to Thuan et al. [22]. The publications listed in this dissertation each contribute to the answers of the research questions.

Research question 1 (RQ1) is drawn from Problem 1 and addresses the lack of solutions for RDD that utilize existing vehicle sensors. Novel approaches are to be explored to record road damage with existing vehicle sensors, only by using additional software.

How can built-in vehicle sensors be utilized to detect road damages?

Research Question 1 (RQ1)

Research question 2 (RQ2) is drawn from Problem 2 and addresses the current focus on detection performance in DL. Due to the limited computational resources in AVs, DL algorithms must be designed in a resource-saving manner. We strive to develop DL algorithms that maintain cutting-edge detection performance while being lightweight enough to be applicable in AVs. RQ2 focuses on the training methods of DL algorithms and the impact of the training data.

How and upon what data must a road damage DL algorithm be trained to be applicable in AVs?

Research Question 2 (RQ2)

Research Question 3 (RQ3) is respectively drawn from Problem 3 and focusing on how an end-to-end concept must be designed to support the AV motion planning. The question addresses the process from data generation of on-board sensors, via calculation of road damage features to the actual utilization of the features in motion planning.

How can sensor data be used in an end-to-end concept to deliver valuable AVs motion planning information?

Research Question 3 (RQ3)

1.4 Research Structure

Each of the topics of this dissertation contributes to our ultimate goal of teaching the autonomous vehicle how to sense road conditions, correctly assess and process them, and incorporate them into the vehicle motion planning. The goal is to allow AVs to drive comfortably and safely, even in poor road conditions. The technical implementations of individual components differ greatly, which is why we structure the dissertation following the technological solutions and the description of an end-to-end concept from sensors to service.

Table 1.1: Overview of the publications of the dissertation

No.	Outlet
I	Data & Knowledge Engineering (under review)
II	IEEE Sensors Conference 2019
III	IEEE Vehicular Technology Conference 2020 Fall
IV	IEEE Sensors Conference 2020
V	IEEE Sensors Journal
VI	IEEE Intelligent Transportation Systems Conference 2020
VII	IEEE International Conference on Big Data 2020
VIII	IEEE Vehicular Technology Conference 2021 Fall
IX	IEEE Transactions on Intelligent Vehicles (under review)
X	IEEE Int. Conf. on Electr., Information and Commu. 2021

Table 1.1 displays the outlets of the ten publications in this dissertation. The table does not represent the chronological ordering as in many research projects (and reinforced by the applied setting of the publication in HELLA's

research laboratory for automotive electronics E-LAB), the publication contents follow the competence development of the researchers and new ideas throughout the research process. For this reason, the contents of the ten publications were usually not processed in strict succession but partly in parallel and influenced by each other. Publication X in Table 1.1 is out of the scope of this dissertation and will not be addressed in the preamble; however, it has been included for the sake of completeness.

The following explains the new, thematically appropriate structure, the research methodology, and the sensor technology that was used. In order to explain the research contributions, limitations, and future work in the preamble, we group Publications I to IX in three subject areas: utilization of on-board sensors in the Quarter Vehicle Model (QVM) to assess road elevation, Computer Vision (CV) application of front-facing camera sensors to detect and classify road damages ahead of the vehicle, and End-to-end (E2E) concept that integrates road damage features into the modern architecture of AVs, from sensor data generation to support of vehicle motion planning.

Table 1.2 outlines our QVM-based research on assessment of road conditions. The table illustrates, for Publications I to VI, the sensors that were utilized, the research approach and methodology, and the relevant contributions.

QVM: The first approach uses sensors in the vehicle dynamics to compute the elevation profile of the road under the tires. It is a model-based calculation that utilizes the Vehicle Level Sensor (VLS) and an AS in the QVM. The output is an accurate elevation profile that can be processed into different road quality indices (c.f. Section 2.2). Publications II to VI deal with the evolutionary stages of the novel method for measuring the road surface (see Table 1.2). Publication II delivers first results in a research-in-progress paper, introducing the design of a laboratory setup explicitly developed for the proof of concept. The setup reflects a quarter of a vehicle and isolates the vertical forces, which allows for evaluation of the sensor setup for the model-based calculation. Publication III describes the exact modeling process, tests different parameter sets of the QVM in the laboratory setup, and demonstrates the setup's functionality. It represents a new level of maturity compared to Publication I and includes the evaluation of the research. Publication IV is a live demonstration paper. At IEEE Sensors (conference), demonstrations of research were covered in a dedicated track and were open for discussion over the duration of the conference. We were able to present our laboratory setup live in front of an expert audience. Publication V is an extension of Publication III and represents the transfer of the functional principle to a real vehicle. Dealing with vehicle interfaces, raw sensor data, and data processing on edge devices was a significant part of the extension and prerequisite for the field study. Evaluation of the technology in a test vehicle is a major step toward application.

Table 1.2: Publications related to QVM-based road condition monitoring

	Publication II & Publication III
Sensors	Pre-installed Vehicle Level Sensor & Acceleration Sensor
Approach	Model building & Prototyping
Contribution	Novel road assessment utilizing the quarter-vehicle
	Publication IV
Sensors	Pre-installed Vehicle Level Sensor & Acceleration Sensor
Approach	Live demonstration at conference including discussion
Contribution	Road assessment in laboratory setup
	Publication V
Sensors	Pre-installed Vehicle Level Sensor & Acceleration Sensor
Approach	Prototyping & field study
Contribution	Laboratory setup & field study in a real vehicle setup
	Publication VI
Sensors	Pre-installed Vehicle Level Sensor & Acceleration Sensor
Approach	Prototyping & field study
Contribution	ML-based detection of road pavements and damages

Publication VI indicates the transition in our research from the first to the second approach. We pair our use of the VLS and the AS in the vehicle dynamics with classic ML approaches. The sequence of data is used to detect road coatings (e.g., concrete, asphalt, gravel) and damage. Already transversed road damages are transmitted to a cloud platform to be used for subsequent vehicles. For this publication we have performed measurements in Toulouse, France with our test vehicle. Furthermore, the publication includes the architecture of the cloud prototype and the visualization of the results in an online map.

In addition to these contributions, we acknowledge that the recognition of poor road conditions happens when the vehicle has already passed them and that the measurements are limited to the surface under the tires. Despite the application in the cloud scenario, we would like to enable the vehicle to detect damage in a predictive and board-autonomous way in order to move more comfortably and safely, even without an internet connection. This led us to the use of camera sensors for RDD. A considerable advantage of this technology is that camera sensors face the direction of travel and can assess road damages in advance. In this scenario, road damages can lead to adjustments in driving behavior within the vehicle and do not have to be obtained via a cloud platform.

CV: By utilizing the camera in the second approach, we apply CV technology. The Publications VII to IX are displayed in Table 1.3, including the sensor technology, the research approach, and the contributions. CV-related contributions also answer RQ1, as front-facing cameras are standard hardware in modern vehicles. Additionally, the approach contributes to RQ2 through its focus on applicability regarding computational resources. From Publication VII to IX, we increasingly focus on designing DL algorithms for edge devices in vehicles. Detecting road damages rather than processing an accurate profile of the road has advantages and disadvantages. These are explained in detail in the publications and are included in the contributions of the preamble (c.f. Contributions).

Publication VII represents our DL approach to RDD, Faster Region-based Convolutional Neural Network (FRCNN) (cf. Section 2.4.1). With this solution, we were able to place twelfth of over 200 submissions in the IEEE Global Road Damage Detection Challenge (GRDDC) 2020 [23, 24], which led to an invitation to publish our results at the IEEE International Conference on Big Data. All teams were provided with the Road Damage Dataset (RDDS), including over 31,000 damage annotations (cf. Datasets and Labeling).

Building on these successes and the competing solutions in the GRDDC, we investigate You Only Look Once (YOLO) algorithms (cf. Section 2.4.2) in Publications VIII and IX. Although the first investigates YOLOv4 and the second YOLOv5, both focus on designing a sweet-spot for efficient performance of a Deep Neural Network (DNN) while maintaining real-time capability in AVs. We generalize our analyses and application of the tools to facilitate research of object detection tasks independent of the use case. In addition to RDD, Road Damage Severity (RDS) classification is of great relevance for the motion planning of AVs. Minor damages should not be avoided, even if they are detected. Avoidance of significant damage, which is crucial for vehicle comfort and, above all, for safety, should be a priority.

Publications I and VIII provide approaches on classifying the severity of potholes. However, the subject is not conclusively addressed in this dissertation for other damages classes than potholes (cf. Limitations). For the two publications, we self-labeled over 10,000 automotive-grade camera images, which we use alongside the RDDS from the challenge. The ADS is not public, but it is described in detail in Publications VIII and IX, as well as in Appendix A.

E2E: In Publication I, we conduct Design Science Research (DSR) and propose an end-to-end concept for the AD system that incorporates road dam-

Table 1.3: Publications related to the CV-based road condition monitoring

Publication VII	
Sensors	Front-facing camera: RDDS from GRDDC
Approach	DL modeling & evaluation
Contribution	FRCNN DL algorithm for road damage detection
Publication VIII	
Sensors	Front-facing camera: self-labeled ADS & RDDS
Approach	Prototyping
Contribution	Lightweight DL algorithm for RDD & RDS detection
Publication IX	
Sensors	Front-facing camera: self-labeled ADS & RDDS
Approach	Prototyping & structured impact analysis of tools in DL
Contribution	Impact of DL tools and guidelines for DL model design

ages and improves the high- and low-level path planning of AVs. This utilization of both technologies in this dissertation (QVM and CV) in an end-to-end concept to be implemented in the AD architecture is the first of its kind. The concept combines the two approaches to both enable predictive detections of road damages and also to validate the detections with the QVM-based calculation of the road. E2E in this context refers to the process that begins with the generation of sensor data in the vehicle and ends with the provision of road damage information to the AV's motion planning system. We have specifically designed algorithms for RDD and RDS that are in accordance with the limited computation resources.

Table 1.4: Publication related to the (E2E) concept for road damage detection and severity classification in autonomous vehicles

Publication I	
Sensors	Front-facing camera: self-labeled ADS & RDDS
Approach	Design Science Research incl. expert workshops & modeling
Contribution	End-to-end concept on road damage detection in AVs

1.5 Research Methodology

To answer the RQs raised in Section 1.3, we have chosen a bottom-up approach to pay respect to previous knowledge and the research setting in the laboratory for automotive electronics (E-LAB) at HELLA GmbH & Co. KGaA. Bottom-up in this dissertation refers to first focusing on developing technical solutions (QVM and CV) that are capable of assessing road conditions. We follow up with a concept of how to utilize them within the existing vehicle architecture. In this section, we provide an overview of the research methodologies applied in this dissertation. First, we explain our approach to the Literature Analysis, which is necessary for all research content. We then explain the DSR paradigm that was employed in Publications I and IX. Finally, we discuss the Applied Quantitative Research to the development of our technical solutions. We define the methods, modeling, prototyping, and, field study commonly applied in engineering that appear repeatedly in the dissertation's publications.

1.5.1 Literature Analysis

Reviewing the relevant research literature is crucial to understanding the relevant domain contents, issues, and results [25]. Familiarity with related research expands the knowledge base and prevents previously explored topics from being researched again. As none of the publications include a full literature review, an in-depth heuristic literature analysis has been performed for each publication.

We identify and analyze relevant publications for our field, using scholarly search engines, such as Google Scholar and IEEE Xplore. The selection of keywords is based on the related literature and is refined during the analysis as we explore which wording is primarily utilized in our domain. The sources of particularly suitable publications are examined to ensure that no relevant publications have been omitted. The analysis also includes grouping the publications by specific properties. For example, we were able to cluster research results on road condition monitoring according to the sensor technology used.

1.5.2 Design Science Research

We have holistically applied DSR only in Publication I and as a research paradigm in Publication IX. However, studying the research method also helped us to problematize, iterate, and adapt to the needs of the application domain in the other publications (especially VIII). We are drawn to DSR's epistemological stance, especially its pragmatism and preference for utility over truth [26, 27, 28].

A characteristic feature of DSR is the recurring examination of the knowledge base and the needs and requirements of the application domain. The framework for this iterative process is displayed in Figure 1.1. It entails the relevance cycle, which deals with the application domain, and the rigor cycle, which deals with the research foundations. The design cycle is between the other two cycles and represents the actual build process of the determined artifact(s), as well as its evaluation.

Building artifacts in DSR can follow very different paths depending on which artifact type is being designed. Artifact types include but are not limited to algorithms, model designs, guidelines, frameworks, and processes [29]. The research in Publication I is special and unusual, as two different artifact types and three artifacts are designed in total (one model design, two algorithms) in response to the application domain context. The evaluations can and must be as varied as the artifacts. This allows for both qualitative testing of their processes and concepts (via workshops and interviews) and quantitative testing of their algorithms (via mathematical evaluation metrics). The three cycles and our take on each are fully explained in Publication I

Transparency regarding the applied tools is especially important in the execution of DSR due to the high degree of freedom of the researchers. For example, we supply a step-by-step description of our approach and implementation in inputs, methods, steps and results in the respective cycles.

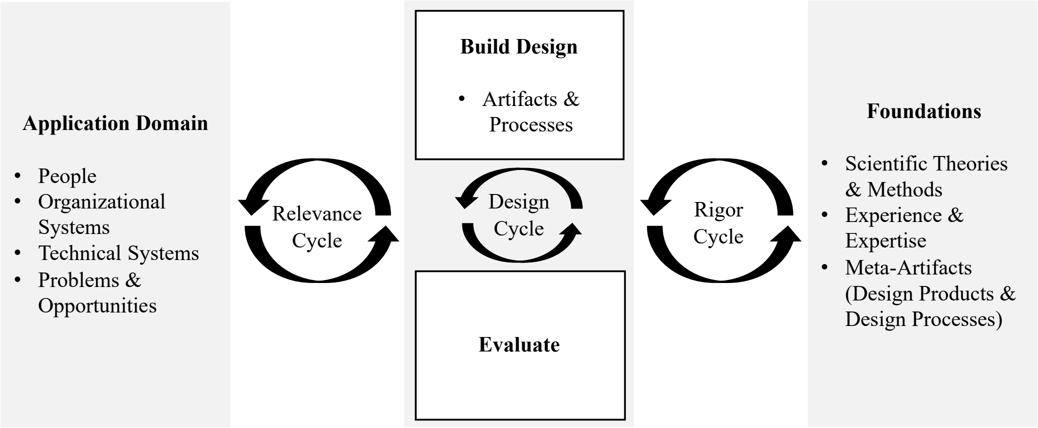


Figure 1.1: Cycles of Design Science Research according to Hevner 2007 [30]

1.5.3 Applied Quantitative Research

Our research approach in building the technical QVM-based and CV-based solutions is applied and quantitative. Applied research is designed to solve practical problems, rather than to acquire knowledge for knowledge's sake (similar to DSR focusing on pragmatism over truth). One can conclude that the goal of the applied scientist is to improve the human condition in the modern world [31]. It employs theories, knowledge, and techniques for a specific purpose. Due to its application oriented nature, applied research is mostly empirical. The process of gathering and evaluating numerical data is known as quantitative research. It may be used to look for patterns, make predictions, evaluate causal relationships, and extrapolate results [32]. Results are obtained on the basis of quantitative evaluations.

Smith 2003 describes the engineers take on problem solving as "the use of heuristics to cause the best change in a poorly understood situation within the available resources" [33], which is very familiar to the approach taken in this work. Regarding Smith's description, we explain below the applied and quantitative methods, modeling, prototyping, and field study utilized in our research.

Modeling: In modeling, researchers strive to build models that expresses only those properties of the real-world artifact that are considered important for the investigation. The simplification aims at a manageable and mathematically computable model or a model suitable for experimental investigations. Models tackle different purposes which is why they can be designed to represent a structure, a behavior, or a process. Model building can be conducted systematically and/or creative [34]

Figure 1.2 displays an iterative process of model building according to Nollau 2009 [34]. The original system and the model cannot be directly compared with each other, as Figure 1.2 illustrates. Based on the observed behavior, a initial modeling approach results in a starting model. The model behavior can then be compared with the observed behavior of the original system to derive behavioral differences. Design iterations are executed to improve the model design.

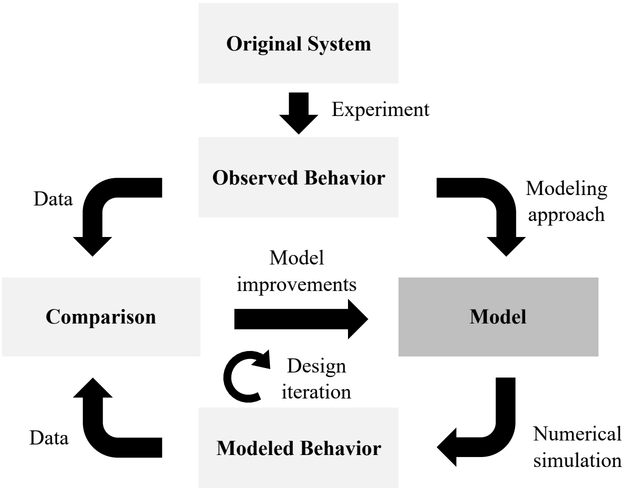


Figure 1.2: Process of model building [34]

The QVM utilized in Publications II to V is a simplification of the reality. We use this model to calculate the road elevation and quantitatively evaluated the results by comparing the real-world behavior with the model behavior. Modeling in DL, especially in object detection, is also an iterative optimization task. In this case, we want to realize the task of object recognition as precisely as possible. The detection performance evaluated by computer vision metrics (e.g., mean Average Precision (mAP), F1-score) and visualized in confusion matrices, for example.

Prototyping: Major content of this work is realized in prototypes. Prototyping is commonly applied in engineering and, like DSR, focuses on pragmatism. A prototype is a tangible representation of a specific artifact, often in an interactive system. It is the opposite of an abstract description that must be interpreted. Designers, as well as managers, developers, customers, and end users, can use the prototype to visualize and troubleshoot the final system, as it provides limited exhibition of the main functional capabilities of the proposed system. This method is often used due to its pragmatic and solution-oriented nature, which allows for easy testing of principles and functions [35].

Regarding the QVM-based solution (cf. Table 1.2), we first design a laboratory setup (a miniaturized prototype of a quarter-vehicle) to rapidly investigate the VLS and the AS in the QVM. The design of the laboratory setup is based on a morphological analysis, a creativity method for the systematic analysis of complex artifacts. The method structures and investigates the total set of relationships contained in multidimensional, non-quantifiable problems [36]. Our goal was to build miniaturizations of a quarter-vehicle and of road segments that represent various road coatings and qualities. We structured the laboratory setup into assembly parts, for which potential solutions are listed regarding each part. By combining parts in a variety of ways, innovative, goal-oriented approaches can be identified [36]. The result is the design of the laboratory setup published in Publications II to V. We followed with the development of the model-based calculation of the laboratory setup's QVMs. This includes the acquisition of the spring and damper constants and of the sprung and unsprung masses. Subsequently, we integrated the VLS and the AS signal processing on a microcontroller to gather the motion data from the spring-damper system of the laboratory setup. Based on the successful implementation of the model-based calculation in miniaturization, we proceed with the implementation in a test vehicle to evaluate the concept.

Field Study: Field studies are the extension of laboratory experiments into real-life situations and provide a more realistic take on the phenomena or artifact under investigation [37]. The projection from a laboratory setup into an application system can entail massive time and cost expenses. It often involves dealing with subsystems and interfaces to other systems. The goal of a field study is to evaluate the behavior of the object of investigation in a real-world environment. In contrast to field studies in business science, field studies in engineering science can be evaluated quantitatively regarding a measurand, if the ground truth is available [37].

In this dissertation, the QVM-based solution on assessing the road surface is evaluated in a field study. To get to this point, integration into the vehicle must have taken place. The evaluation of CV solutions, on the other hand, is debatably a field study, as we have not implemented the system in a test vehicle, though, the ADS (cf. Section 2.4.3) provides images from real front-facing vehicle cameras. We furthermore evaluated the runtime of the algorithms on an edge device built for lightweight application.

Chapter 2

Research Background

This chapter addresses a cross-section of the fundamental research background of Publications I to IX. This sets the stage for a smooth transition to the publications' content, as each publication contains its specific research background. Hence, this chapter is not redundant but complementary to the publications at a higher level of abstraction. Following the introduction to the fundamental principles of AD, the Perception-Planning-Action Pipeline (Section 2.1.3) is introduced, focusing particularly on the Environmental Perception (Section 2.1.1), and HD Feature Maps (Section 2.1.2). Inspired by DSR, this mirrors the application domain. The contextualization increases the relevance of the publications and is therefore invaluable.

The same applies more specifically to the fundamentals of Vehicle Dynamics (Section 2.3), in particular the Quarter-Vehicle Model (Section 2.3.1) and the sensors installed therein (Section 2.3.2). These sections are particularly relevant for the context and content of Publications II to VI.

The last part of the research background is about CV, especially DNNs, the basis of Publications VII to IX. Following an introduction to CV, discussion focuses upon object recognition, basic parameters, and settings for training DNNs. Next, we explain the two DL frameworks, FRCNN (Section 2.4.1) and YOLO (Section 2.4.2), which are essential to this research.

2.1 Autonomous Driving

With fully autonomous driving, vehicle occupants no longer have to perform any driving tasks. To achieve this goal driver assistance systems have already relieved the driver of a considerable portion of vehicle control. This includes the engine, the brakes, and the steering. In the automotive industry, autonomous driving is seen as a mega-trend with considerable impact on the industry itself and society as a whole [2].

In 2014, the SAE published what became the global standard for defining levels of vehicular automation [10]:

- Level 0: No automation.
- Level 1: The vehicle operates with support systems, such as antilock braking system or electronic stability control, that intervene automatically in appropriate situations.
- Level 2: Automated systems take over partial tasks (e.g., adaptive cruise control, lane change assistant, automatic emergency braking). However, the driver retains control of and responsibility for the vehicle.
- Level 3: The car can conditionally accelerate and can break and steer independently. If necessary, the system prompts the driver to take control.
- Level 4: In standard operation, the vehicle can be fully autonomous. However, the driver has the option to intervene and drive manually.
- Level 5: Fully automated, autonomous operation of the vehicle with no possibility of or necessity for driver intervention.

A more detailed explanation of the different levels of automation can be found in Appendix A, Table A.1. There we further distinguish the levels between *Steering & Acceleration*, *Monitoring of Driving Environment*, *Fallback Performance Driving Task*, and *Drive Modes*.

AD requires various types of support systems that in higher levels of automation must interact to achieve truly autonomous behavior. Among them are braking, adaptive speed, traffic jam, lane departure, turning, and parking systems. In addition, there are various sensor systems. AVs must be equipped with systems for environment recognition. Besides cameras, other sensor systems include Radio Detection and Ranging (RADAR) and Laser Detection and Ranging (LIDAR) sensors for measuring distances and processing environmental data.

AD level 5 will entail far-reaching changes to the vehicle interior, as it will no longer be possible or necessary for the vehicle occupants to intervene in the driving task. In Germany, in contrast to other highly developed countries, the legal framework for the highest level of automation is still unclear because the law currently states that the driver must be able to "overrule" the vehicle [38]. Accidents, even if in reduced numbers, should still be expected. Liability in such events remains a partly open issue because the vehicle itself can hardly be held accountable, and the automotive companies will want to prevent liability.

2.1.1 Environmental Perception

Starting at level 2 and increasing dramatically in subsequent levels, the vehicle must understand the scene in which it is located by perceiving various objects in the environment. In order to gain a sufficient understanding of the traffic situation to be able to move reliably in it, various tasks must be carried out in parallel to capture the scene. In the following, we will therefore describe the subsystems *Localizer*, *Mapper*, *Road Mapper*, *Moving Objects Tracker*, and *Traffic Signalization Detector*, which are required to reliably detect the traffic situation and the vehicle's own role in that situation [39]. It is critical that the system run in real-time in order to detect changes as quickly as possible.

The *Localizer* subsystem is responsible for pose estimation of the AV, which consists of its position and orientation. Initially, it might appear promising to use GPS sensors for pose detection, but these are not reliable in urban areas, do not work in tunnel systems, or between tall buildings, and would be unsuitable for use even in rural areas due to the inaccuracy of the position information. In the literature, LIDAR and camera systems or a combination of both are primarily used for localization. While LIDAR sensors remain expensive in mass production, the pose estimation they enable is very reliable and accurate. [40, 41, 42]. A combination of both primarily utilizes LIDAR-data to build a map of the environment and camera images to estimate the pose within that environment [43, 44]. Solutions solely based on camera are less expensive, though also less precise [45, 46]; Some also utilize Neural Networks (NNs), though with poorer performance [47, 48].

The *Mapper* is a subsystem that operates both online and offline and computes the operating environment of the vehicle. It helps to prevent collisions but also to navigate within the environment according to the relevant traffic rules. Topological representations of the environment often describe it in graphs of nodes and edges. Nodes are features/objects of the map, such as traffic lights or other important objects, that are relevant for AD. Edges describe relationships between nodes with position, orientation, and distance between the nodes. Google Maps and Open Street Map [49] utilize similar approaches. The resolution of a topographic map can be very low in rural areas with very straight highways, for example, and very high in cities with many roads, intersections, and possibly even tunnels. In-depth information about HD Feature Maps are given in Section 2.1.2. Typically, an offline (static) map is merged with local data from environmental detection sensors (a combination of LIDAR, RADAR, and camera) and the vehicle's state [13, 39].

Another essential task consists of tracking moving objects and estimating their position in relation to the vehicle. This task is performed by the *Moving Objects Tracker*, which detects and tracks the velocity, location, and orientation of pedestrians, other vehicles, and other moving objects [50].

Some features of the *Traffic Signalization Detector* are already installed in mass-produced vehicles in the form of a traffic signage recognition. This is an example of a assistance system (level 1) that supports the driver with information, such as speed limits. To succeed with higher levels of AD, horizontal traffic signage such as road markings are of great relevance for decision marking and must be present in the vehicle map. The detector needs to either recognize various signage as well as identify the exact position, class, and status [51, 39].

The described components can be summarized as the VPS. In the next sections, we explain the remaining systems that are required to drive autonomously.

2.1.2 HD Feature Maps

Humans can successfully apply maps with low information density to navigate complex environments because of their experience and depth perception. However, the multitude of information from the sensor systems of the CAV cannot be accommodated in a structured way on a conventional map. As a consequence, AVs demand a new map structure from which they can obtain information that is required to perform the driving task.

On-board sensor systems (e.g., Camera, LIDAR, RADAR) perceive traffic objects which need to be stored in a local environment consisting of a semantic structure, enabling AVs to store environmental information accurately, which allows for the construction of HD Feature Maps directly from 3D sensory data. Such a map is called an HD Feature Map and is both the endpoint of the AV's environmental perception system and the starting point for the motion planning task. Rather than a full 3D map of the environment, HD Feature Maps include all the key geometric features of the road network and semantic information about, for example, the lanes of a road, speed limits, traffic signs, and traffic lights. The condition of the road can be precisely stored via a grid or via markings, which are two different approaches that are discussed later in Section 2.2 [52, 53, 54].

HD Feature Maps commonly compose three types of map features: static objects, traffic control devices, and roadway geometry. Static objects represent objects of interests for the vehicle to be considered in motion planning due to their size or position. Static objects can be trees, buildings, barriers, walls, or, relating to this dissertation, significant road damages.

Traffic control devices guide the vehicle on their supposed behavior by establishing traffic rules for a designated location or area. This include but is not limited to road surface markings, traffic signs, and traffic lights. The roadway geometry is represented by polylines, polynomial curves, or splines, crucial to understand the designated path to be followed for the AVs. Section 2.1.2 displays a picture of a road course including a highway and a highway exit. The below polylines describe the road curse. The lines are often considered as edges, while the fixed points are often stated as nodes. Both can be applied with specific properties. AVs must perform their self-localization in the environment to utilize the information provided by the HD Feature Map. This was previously introduced as the localizer (cf. Section 2.1.1).

Besides the local HD Feature Map, representations of the environment are stored in online HD Feature Maps. The combined use of both local and online maps entails many opportunities. The online HD Feature Map is a joint map, generated from a high number of individual vehicles through aggregation processes. Individual vehicles contribute to an online map while being able to gain information from the map for preceding road segments [54, 55]. In our expert workshops conducted in Publication I, the experts considered the online HD Feature Map as an additional sensor system to increase confidence for static objects, traffic control devices and the roadway geometry.

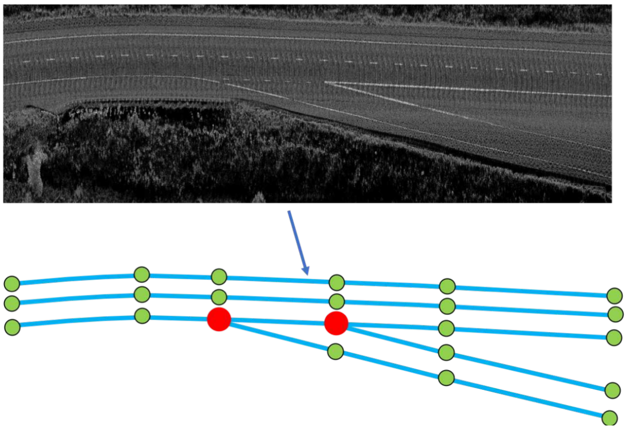


Figure 2.1: A road course including nodes and edges [55]

Further details on HD Feature Maps are included in Publication I. Liu et al. 2020 [56] perform a comprehensive overview of HD Feature Maps in AD, while Jo et al. 2018 [57] propose a concept on updating the online map based on crowd sourcing from AVs.

2.1.3 Perception-Planning-Action Pipeline

The VPS represents the first part of the perception-planning-action pipeline in AD and is explained in detail above (cf. Section 2.1.1). The VPS perceives the environment and enables the planning tasks based on the structured data gathered in the HD Feature Maps. The remaining planning and action tasks are explained in the following.

Motion planning can be split into high- and low-level path planning. High-level path planning, commonly called route planning, is performed through graph search algorithms in a road network. The calculated route is a sequence of waypoints represented by coordinates. Suppose the road network is represented by a weighted directed graph, whose vertices are waypoints. In that case, edges connect pairs of waypoints, and edge weights denote the cost of traversing a road segment defined by two waypoints. Computing the best route can then be reduced to finding the shortest path in a weighted directed graph [13, 39]. The performance of algorithms for route planning in road networks has improved significantly in recent years, resulting in more complex calculations than those explained above. Even for routes that cross continents, newly developed algorithms can compute driving directions in milliseconds or less. Bast et al. (2016) provide a review of route planning algorithms in road networks that are suitable for self-driving cars [58]. The second task within high-level path planning is the actual path planner subsystem that finds the proper path in the selected route. Input values for the planning task are the AV's state, the perceived environment, and the traffic rules. A path is composed of a sequence of poses (the vehicle's position and orientation). The path planning is typically executed for several tens to hundreds of meters ahead [59].

Low-level path planning also consists of two subsystems, the behavior planner and the motion planner. Selecting the proper behavior in a given road traffic situation is mandatory to successfully perform AD. The subsystem is responsible for selecting the right behavior in the perceived environment by, for example, keeping the lane, changing the lane, stopping the vehicle, or accelerating to pass another vehicle. Within the allowed decision time frame, the behavior planner selects a goal based on current driving behavior and input from the collision avoidance system regarding static and moving objects in the environment. From this step onwards, not solely technical issues are of relevance for the subsystems. As 100% safety can probably never be achieved, moral decision making is a research topic under investigation [7, 9]; however, that research is beyond the scope of this dissertation. Classic solutions for selecting a behavior are heuristics, decision trees, and finite state machines [60], however single techniques perform poorly in complex traffic scenarios. Combinations of finite state machines and other techniques are applied to improve the decision making (e.g., Aeberhard et al. 2015 [61], Okumura et al. 2016 [62]) while also focusing on lowering the uncertainty of moving objects in the environment by utilizing partially-observable Markov decision processes (e.g., Galceran et al. 2017 [63]).

The motion planner calculates the vehicle trajectory in the selected path with the chosen behavior, constrained by the vehicle kinematic and dynamic. Two different approaches are discussed in the literature and practice, one using a sequence of commands that includes a velocity, a steering angle, and a command duration or sequence of states (poses to be achieved in a specified time frame). The ultimate goal of the motion planner is to comfortably and safely move the vehicle from its origin to the desired destination. Several methods for motion planning are discussed in González et al. 2015 [59] and Paden et al. 2016 [64].

The action part of the perception-planning-action pipeline is responsible for the execution of the planning results. This includes commands to the respective actuators of the vehicle, the steering wheel, throttle, and brakes of the AV. As this is less relevant to the content of this dissertation, we refer to Ziegler et al. 2014 [51] and Paden et al. 2016 [64].

The influence of road damage considerations on the perception-planning-action pipeline is explained in detail in Publication I, as our goal is to actively contribute to these planning tasks in order to improve passenger comfort and safety.

2.2 Road Condition Metrics

Analyses of road conditions arise from different motivations: (1) to assess the quality of the infrastructure for road authorities and (2) to record road damages to avoid dangerous and uncomfortable situations based on road evaluations. In order to successfully implement both use cases, metrics are needed that can be applied to compare road analyses with a defined standard. The literature offers two different types of metrics, the first of which is the detection and evaluation of a single instance of road damage at an exact geo-position. The second is an evaluation of road sections via representative grading. The left side of Figure 2.2 visualizes the evaluation of single damages, while the right side represents possible gradings of road sections.

Case (I) focuses on detecting and evaluating the type, severity, and location of single damages. In our research, we apply the type classification of the Japanese Road Maintenance and Repair Guidebook 2013 [65], which was also utilized by Maeda et al. 2018 [66]. The relevant damage classifications for our research are D00 for longitudinal crack, D10 for lateral crack, D20 for alligator crack, and D40 for pothole. Further classifications are listed in Table 2.1 in Section 2.4.3.

In case (II), various options exist for the analysis of road segments. In 1973 a method evolved to interpret the elevation profile of a road. If we consider the road profile as a curve, each curve can be broken down into a set of sinusoids [67]. The analysis results in wave numbers that indicate whether a road is more likely to have roughness due to ground waves or high-frequency damages [68]. A second standard was introduced with the International Roughness Index (IRI) by Sayers et al. 1982 [69]. Sayers used the IRI to measure the longitudinal road profile by accumulating deviations between the sprung and unsprung masses of a vehicle at a velocity of 80 km/h over a specific distance (a section). This and further road condition metrics are explained in the *Little Book of Profiling* [67].

The green color in Figure 2.2 indicates a very good rating for the road segment, while the red color indicates a very poor rating. Case (II) is not further adopted in our research, though the QVM-based calculation, subject to Publications II to V, delivers a an elevation profile of the road, which can be used to calculate the described metrics.

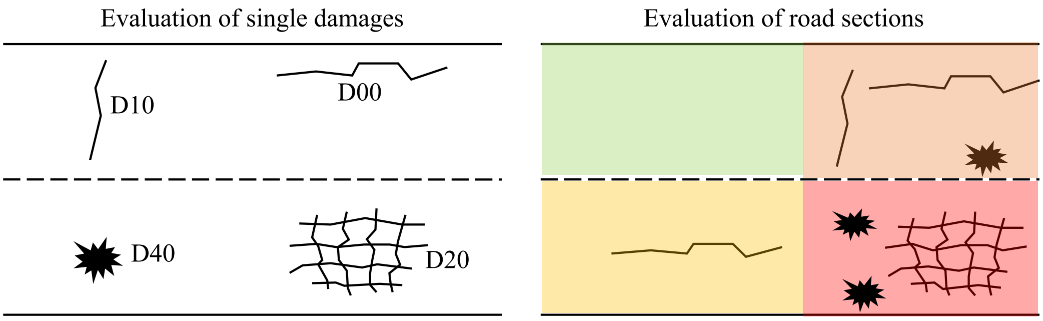


Figure 2.2: Damage and section classifications of a road's condition

2.3 Modeling in Vehicle Dynamics

The vehicle dynamics describes the behavior of a vehicle under the influence of forces, for example, those caused by cornering, acceleration, and braking processes. Road unevenness also acts on the vehicle via forces and affects its behavior. Predicting the motion of a vehicle, which consists of about 10,000 individual parts, is a challenging task. In many cases, it helps to combine a limited number of components into subsystems. Simplifying a mathematical model facilitates understanding in many situations or can even enable statements to be made with sufficient accuracy in the first place. However, these models are subject to limitations that are defined by the model boundaries. The factors affecting vehicle dynamics can be grouped into five categories: drivetrain and braking, suspension and steering, distribution of mass, aerodynamics, and tires [70].

In order to describe the motion of vehicles, various substitute vehicle systems focus on the target variables that must be simulated. This includes, for example, the full vehicle model (e.g., Kruczek and Stribrsky 2004 [71]), the single track model (e.g., Schramm et al. 2014 [72]), and the QVM [73].

As our target is to assess road condition, we build a novel modeling approach calculating the road elevation that utilizes the QVM. This model is particularly suitable due to the isolated analysis of the vertical forces of a single wheel mass and the respective quarter of the chassis mass. The model is described in detail below.

Another reason to model the quarter vehicle is the presence of the sensors introduced in Section 2.3.2. These are able to record physical quantities within the model and therefore operate within the model boundaries.

2.3.1 Quarter-Vehicle Model

Two different kinds of QVMs are reported in the literature, a multi-body suspension model [74] and a simpler model with two degrees of freedom [75, 76]. Figure 2.3 displays each of these models. The multi-body suspension model expresses the non-linear geometry of the mechanical system as well as the dynamics in a more realistic manner. The second kind of QVM is particularly suitable for examining vertical forces in the system. The full suspension system is represented by the sprung(/chassis) mass m_s, the unsprung(/wheel) mass m_u, the spring-damper system, and the replacement system for a wheel, which is also a spring-damper system. Hence, there are a total of two spring constants (c_s, c_u) and two damper constants (d_s, d_u). Due to the very small damping influence of the wheel, d_u is neglected in most research [77, 78]. The motion of the chassis mass $z_s(t)$ results from the underlying motions of the wheel mass $z_u(t)$ and the elevation of the road profile $z_r(t)$. Publications II to VI contain all steps for the realization of the calculation, including the intersection of the masses and the differential equations. We conduct the parameterization of a representative design of a QVM in a laboratory setup and in a real test vehicle (cf. Publication V).

Our QVM approach (cf. Table 1.2) is distinguished from the commonly conducted research on the QVM in which given forces from the road profile $z_r(t)$ are used to calculate the behavior of the wheel and chassis masses from bottom to top. In contrast, we turn the computed path upside down by acquiring sensor data for the wheel and chassis masses and back-calculating to the road profile. In combination with the sensor technology used in the subsequent section, we can calculate the model and obtain the road elevation $z_r(t)$. The calculation of the QVM is described in Publications II to V. In the following, we introduce the utilized sensor technologies, which are located within these model boundaries.

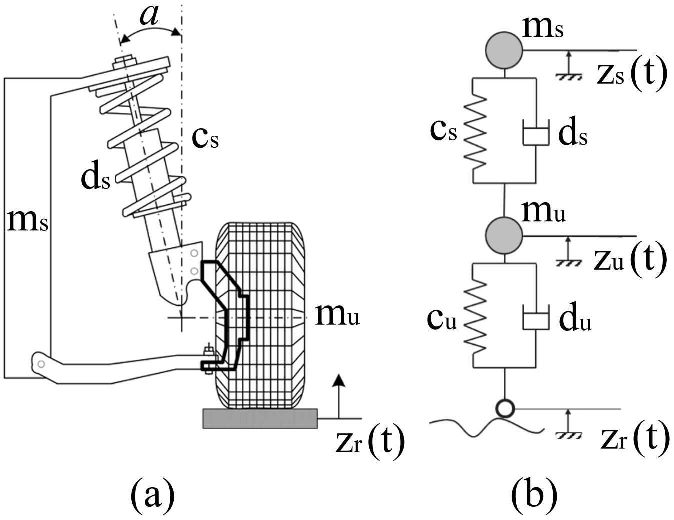


Figure 2.3: (a) Multi-body QVM [79], (b) simplified QVM

2.3.2 Sensors in Vehicle Dynamics

To enable the back-calculation of the QVM, data regarding the motion of the wheel and chassis masses are required. Here we rely on the use of pre-installed sensors in common vehicles: the Vehicle Level Sensor (VLS) and the Acceleration Sensor (AS). The combined use of both sensors allows us to calculate the elevation profile of the road using the parameterized QVM. Despite the fact that our research only uses the data from these sensors, we will briefly discuss their operating principles so that the VLS and the AS do not appear to be black boxes in the further course of the work.

Vehicle Level Sensor

The sensor is widely used to measure the front to rear tilt of vehicles, due to the mandatory implementation of headlight range control for xenon- and LED-based headlamps in many countries, especially those in the European

Union [80]. The headlight range control enables adaptive adjustments of the luminosity based on the vehicle's tilt angle in order to avoid blinding oncoming traffic and to maintain the best possible vision. A further application requiring the VLS sensor data is the adaptive chassis and suspension system that enables vehicles to adjust their suspension system to current road conditions [81, 82].

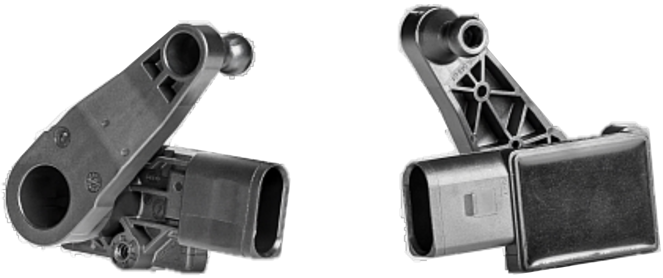


Figure 2.4: Vehicle Level Sensor: Rotatable lever and fixed base housing

The housing of the VLS contains a Contactless Inductive Position Sensor (CIPOS) that consists of two essential parts, the stator and the rotor. The stator consists of the Printed Circuit Board (PCB) and an electronic assembly in which the exciter and receiver coils are shaped as PCB traces. The electronics for evaluation as well as the interface to the control unit are located in the electronic assembly. The rotor is a trace formed in a specific geometry, as illustrated in Figure 2.5. An alternating current flows permanently through the excitation coils and thus generates an electromagnetic field which reaches the traces of the rotor. The induced alternating current in the rotor then induces a second electromagnetic field, and both electromagnetic fields act on the three receiver coils located on the stator and induce an alternating current that is dependent on the rotor position. Due to the excitation coil's geometry, the rotor's induction is independent of the position angle inside the case. In contrast, induction in the receiver coils depends on the rotor distance and thus on the angle of the sensor level position (cf. Figure 2.4). The electronic assembly is responsible for the calculation as well as for the signal processing of different interfaces (e.g., PSI5, Pulse-Width Modulation (PWM)).

The VLS is mounted on the chassis mass and the wheel mass with one fix point at the body of the VLS and one at the lever. The mounting geometry allows the offset of the chassis and wheel masses to be determined based on the lever position. Thus, the VLS measures the displacement between wheel and chassis mass in real-time based on the angle generated in the CIPOS.

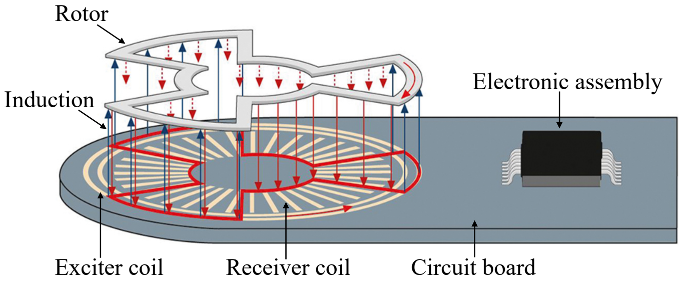


Figure 2.5: Design of the Contactless Inductive Position Sensor (CIPOS) [83]

Acceleration Sensor

Micro Electro Mechanical System (MEMS) ASs are commonly used in the spring-damper system of modern vehicles. MEMS techniques create mechanical sensing micro structures that are often based on silicon. When paired with microelectronic circuits, MEMS sensors can measure physical quantities such as acceleration [84]. ASs are usually mounted at the wheel mass and frequently at the chassis mass and are utilized for active chassis control systems. They generate data for active shock absorbs which allows the driver to utilize various driving styles (sport, urban traffic, etc.). The ASs used in vehicle dynamics must operate in low-power with a reasonable accuracy while being robust in harsh environments [85]. Publication V delivers further information on the AS used in our QVM-based calculation of the road elevation.

In our test vehicle, the AS fits in a 3D printed housing, mounted with a pipe clamp on the axle rod of the wheel mass (Figure 2.6). The inner attachment in the housing compensates the angle of the axle rod to enable one direction of the AS to fully measure vertical acceleration.

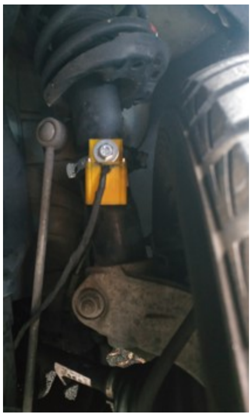


Figure 2.6: Mounting of the AS's housing (yellow box) at the wheel mass

2.4 Deep Learning in Object Detection

This section is intended to provide the research background regarding the adopted DL notations, concepts, and algorithms. The central theme is the exploratory study of DL models and their performance in the detection of road damages and their severity. We outline the building blocks of Convolutional Neural Networks (CNNs) for object detection, which are primarily explained in CV. Section 2.4.1 and 2.4.2 lead over to our application, the detection of road damages. Here we explain the DL algorithms FRCNN and YOLO, which are applied in Publications I, VII, VIII, and IX.

DNNs are a specific class of Artificial Neural Networks (ANNs) that are inspired by the visual cortex of humans and consist of a collection of connected cells [86, 87]. These cells are called neurons, and each of them processes the sum of incoming signals using a defined, often non-linear function before passing on the output (e.g., Figure 2.7). Equation 2.1 displays the heavyside step function that transforms to zero any incoming signals that are less than or equal to zero and transforms to one any that are above zero:

$$\Theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leqslant 0 \end{cases} \tag{2.1}$$

Neurons of the same depth are aggregated in layers that perform different transformations on the input signals and thus are responsible for fulfilling different tasks according to the use case. The connections between the neurons are called weights and represent the strength of the signal. A higher strength leads to a bigger impact of the connection in the output of the DNN.

DL is a subset of ML and is largely based on ANN. The success of ANN in object detection is due to the rapid development since the introduction of AlexNet in 2012 where the CNN solution surpassed hand-crafted ML solutions in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [88]. Improvements in computation power have also enabled the application of this demand-heavy technology [89]. ANN utilizes a large amount of data for training rather than expert knowledge of rule-based solutions, which is why transparency is not always granted and has increasingly become a subject of investigation [90].

DNNs are a specific kind of ANNs with multiple layers between input and output layers, for which reason they are also referred to as “deep”. The layers in between are called hidden layers. Every neuron has a bias that is added to the sum of inputs and acts as a sort of offset. As training proceeds, these weights increase or decrease the strength of signals, leading to adjustment of the biases.

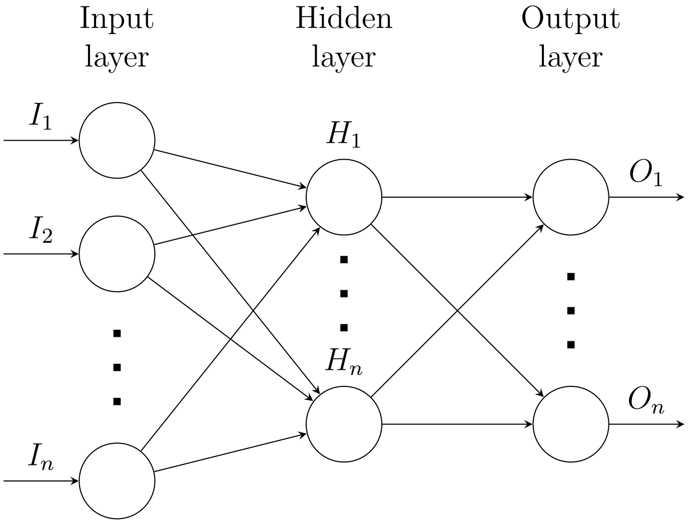


Figure 2.7: Example structure of a simple ANN with input layer, hidden layer, and output layer

The network has to be trained over time from training data to learn the suitable weights and biases for a given task. The model presents a subset of the available data as a training set, and the rest is kept separate for validation and testing purposes. In the beginning, all weights and biases are initialized randomly, and with each training step, they are corrected to better fit the desired output. The different weights and biases influence the outputs, and the errors are calculated using the gradient of the cost function, which incorporates the weights and biases. This method is called backpropagation [91]. This gradient is also used for the gradient descent. Various optimization algorithms can be used, each with its own set of advantages and disadvantages [92]. However, on high-dimensional data, common gradient descent with no modifications struggles to converge; therefore, further adaptations have been introduced to develop new techniques as Stochastic Gradient Descent (SGD) with momentum [91] or Adaptive Moment Estimation (Adam) [93].

CNNs are a subcategory of ANNs that are used to create most modern object detectors [94]. With the introduction of AlexNet [95], the authors achieved a considerable performance boost in the image classification challenge ImageNet [96]. As a result of this achievement, CNNs may now replace hand-crafted feature extractors in image classification and object detection processes. Advances in computational power and data availability [97] have further enabled CNNs to improve in recent years.

The essential parts of the CNN are the convolutional layers. An input, in our case an image, becomes increasingly abstracted in deeper layers, resulting in a features map that represents the input image. These layers perform the binary operator called Frobenius inner product, that takes two matrices and returns a number as illustrated in Equation 2.2 of the convolutional kernal B and part of its input matrix A.

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_{ij} \overline{A_{ij}} B_{ij} \tag{2.2}$$

The size of the kernel defines the input of a neuron by its restricting area (called receptive field). These shared weights of one kernel across a layer of neurons result in a translation equivariant feature map [98]. The stride s defines the step size of the convolutional kernel, which is used to slide over the image to create a feature map. A stride $s = 1$ preserves the input dimensions, while $s = 2$ would downsample the input to 0.5.

Pooling layers are commonly used to reduce the data dimensions and are frequently located in CNNs. Pooling layers use clusters to combine the inner values to one neuron. In practice, max pooling, in which the largest number of the cluster is transferred to the subsequent neuron, and average pooling, in which the average of the cluster is transferred, are most commonly applied.

Hereafter we detail two popular and cutting-edge CNN frameworks, the multi-stage FRCNN (Section 2.4.1) and the single-stage detector YOLO (Section 2.4.2).

2.4.1 Faster R-CNN Algorithm

The FRCNN algorithm contains two individual modules. The first module, Region Proposal Network (RPN), is a CNN with the task to generate region proposals with an objectness score. Each proposal is recognized as a Region of Interest (RoI). A Fast R-CNN [99] is the second module and investigates the RoIs. Both modules share computation via a common set of convolutional layers. FRCNN is considered as a DNN with attention [100], with the RPN suggesting the Fast R-CNN to investigate the RoIs.

The shared convolutional layers are responsible for the feature map of the original input, and utilized by the RPN. It creates k, anchor boxes each position, while a classification layer derives $2 \cdot k$ scores including the presence of an object and a regression layer calculates $4 \cdot k$ as the bounding box coordinates. The anchor is placed at the center of the sliding window, going over the feature map. Every anchor is associated with a scale and an aspect ratio. This method ensures translation invariance for the proposals and detections, and multi-scale anchors eliminates the need for feature maps of different sizes and aspect ratios [101, 102] to identify objects of different scales and proportions.

Only anchor boxes with a high enough overlap with the ground-truth boxes are counted as correct predictions. To measure this, the Intersection over Union (IoU) is needed, which is the division of area of overlap by the area of union. With predicted box A and ground-truth box B, the IoU is:

$$\mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{2.3}$$

Anchors achieve a positive label if they have the highest IoU with a ground-truth box or if the IoU is greater than 0.7. They achieve a negative label with an IoU of 0.3 or lower. Every other anchor is not considered for the calculation of the loss. The Loss function for the RPN is defined as:

$$L\left(\{p_i\}, \{t_i\}\right) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right), \tag{2.4}$$

where i denotes the index for anchors in a mini-batch and p_i the probability for being an object. Ground-truth labels p_i^* are 1 if the anchor is positive and 0 if negative. With these parameters a log loss over two classes L_{cls} is computed and normalized with the mini-batch size N_{cls}. Bounding box coordinates are represented in the vector t_i, with ground-truth coordinates for positive anchor as t_i^*. The regression loss L_{reg} from [99] is only activated for a positive anchor ($p_i^* = 1$) and is normalized with the number of anchor locations. To roughly equal out the impact of both loss functions, $\lambda = 10$ is chosen, because of the different values for the normalizing terms.

Mini-batches are created from single images to train the RPN. Each mini-batch consists of 256 randomly sampled anchors (using all anchors would bias toward negative samples, which are dominant). The ratio of positive to negative samples is up to $1:1$.

A Fast R-CNN takes an image and a set of object proposals as an input. With the assistance of a CNN, the image gets processed and a feature map is created. This step is performed by the shared convolutional layer in the Faster R-CNN architecture. The ZF model from Zeiler and Fergus [94], which has five shareable layers, and the VGG16 model from Simonyan and Zisserman [103] with 15 shareable layers were originally tested and used for this purpose.

The object proposals are generated by the RPN, and for every proposal a RoI pooling layer extracts a feature vector from the feature map. RoI pooling is a special case of the spatial pyramid pooling layer [101] that includes sub-window calculation. The proposals are fed forward into a sequence of fully connected layers that branch into two sibling output layers. One computes a softmax probability estimate over K objects plus a background class,

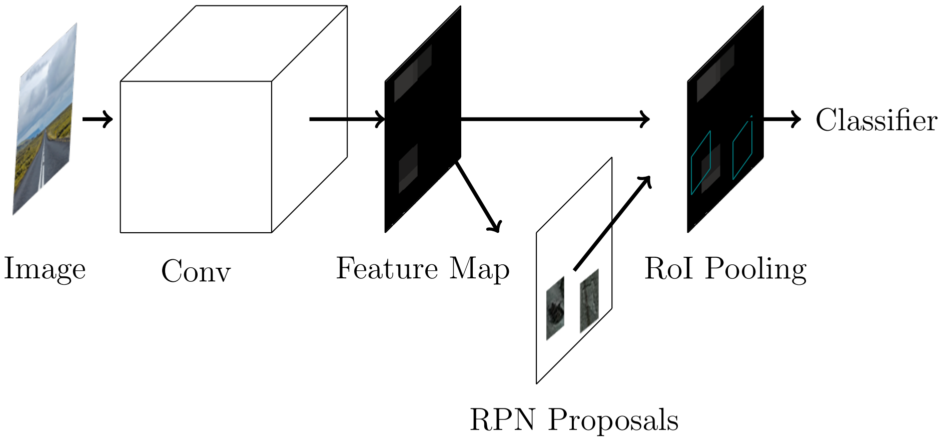


Figure 2.8: Faster R-CNN as a single network for object detection with an attention mechanism [99]

generating a confidence score for every possible object. The other outputs four coordinates for the location of the object.

A softmax function σ transforms all components of a M-dimensional vector z into the value range $(0, 1)$, which allows them to be summed.

$$\sigma : \mathbb{R} \rightarrow \left\{ z \in \mathbb{R} \mid z_i \geqslant 0 \, , \sum_{i=1}^{M} z_i = 1 \right\} \tag{2.5}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{M} e^{z_k}} \quad , \quad \text{for} j = 1, ..., M \tag{2.6}$$

A combination of the two modules works and detects like one single network structure. The structure of a Faster R-CNN can be seen in Figure 2.8.

2.4.2 YOLO Algorithm

Contrasting the two-stage FRCNN framework, YOLO operates in a single stage. YOLO considers object detection as a regression problem of bounding boxes and associated class probabilities. YOLO has been evolving since its invention in 2016 [104], and due to its great success, both official and unofficial versions have since then been released. In this dissertation we use YOLOv4 [105, 106] in Publications I and VIII and YOLOv5, an unofficial version by Glenn Jocher[1], in Publication IX. The latter v5 version is a parallel development to YOLOv4 with marginal differences in detection performance and runtime.

The great attention paid by other scientists to the YOLO framework is due to the fact that, in contrast to previous state-of-the-art frameworks, no RPN is required (cf. FRCNN). As detailed above, RoIs in the image previously had to first be proposed and then followed by a classifier that runs over these regions. Subsequently, duplicates had to be removed via postprocessing. Instead of iterating the process of classifying different regions of the image, the YOLO framework computes all the image features. By making predictions for all objects at the same time with a single NN of multiple CNNs, YOLO predicts vectors corresponding to each object in the image.

The processing steps of YOLO are illustrated in Figure 2.9. Entire images are divided into a $S \times S$ grid (default: 7×7). If an object is located in parts or in a grid cell, it is responsible for its detection. Bounding boxes are predicted simultaneously over all classes for each grid in the image, including a confidence score. The confidence score reflects the probability of an object being in the calculated bounding box, as shown in Equation 2.7.

$$\text{confidence score} = Pr(\text{obj}) \cdot \text{IoU}_{\text{pred}}^{\text{true}}. \tag{2.7}$$

$Pr(\text{obj})$ reflects the probability of an object being in the cell, while $\text{IoU}_{\text{pred}}^{\text{true}}$ is the intersection over union of the predicted box. When it is necessary to detect more classes of objects, an $S \times S \times (B \cdot 5 + C)$ tensor of class specific confidences $Pr(\text{Class}_i \,|\, \text{obj})$ is created from each cell.

$$Pr(\text{Class}_i \,|\, \text{obj}) \cdot Pr(\text{obj}) \cdot \text{IoU}_{\text{pred}}^{\text{true}} = Pr(\text{Class}_i) \cdot \text{IoU}_{\text{pred}}^{\text{true}} \tag{2.8}$$

Besides the confidence scores, each bounding box additionally is defined by the center coordinates (x, y) of the box as well as the height h and width w. In total, five parameters correspond to each bounding box [104].

The YOLOv1 framework essentially consists of the Darknet architecture[2] and two following fully connected layers. The Darknet architecture can be either 24 convolutional layers for normal-YOLO or 9 convolutional layers in fast-YOLO depending on the complexity of the object detection task.

The loss function L_{yolo} includes three parts, being the bounding box loss L_{bb}, the objectness loss L_{obj}, and the classification loss L_{cls} (cf. [104]). For more information on the individual losses, see Redomn et al. 2016 [104].

$$L_{yolo} = L_{bb} + L_{obj} + L_{cls} \tag{2.9}$$

With the introduction of YOLOv2 [107], batch normalization was established as a feature improve the training of DNNs by stabilizing the distribu-

[2] Darknet and Scaled-YOLOv4 Framework: https://pjreddie.com/darknet/

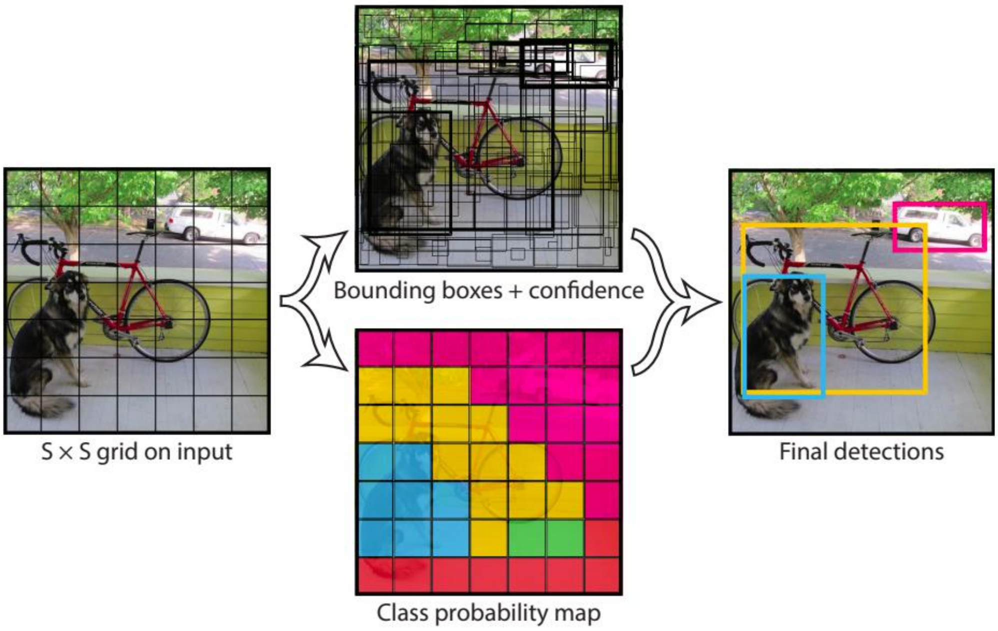


Figure 2.9: YOLO decision making process, including 7 × 7 grid cells [104]

tion of the input layers [108]. Batch normalization reduces the training time, increases generalization and make dropouts to prevent overfitting unnecessary. A new Darknet architecture was introduced with Darknet-19 including 19 concolutional layers and five max pooling layers for improved detection performance and runtime.

YOLOv3 [109] establishes a new network part between the backbone and head, named the neck (cf. Publication IX). The neck is responsible for feature extraction of different scales from the backbone, similar to a Feature Pyramid Network (FPN) [110]. Darknet-19 is replaced by Darknet-53 which consists of 53 convolutional layers and shortcut connections, delivering improved results. Graphics Processing Units (GPUs) can be better utilized with the novel network structure [109]. YOLOv4 and YOLOv5 are detailed in Publication VIII, and Publication IX respectively.

2.4.3 Datasets and Labeling

In this section, we illustrate the two datasets used in this dissertation, road images and any damages. We address the labels, the characteristics of the images, and the placement of the camera in the vehicle as well as the differences in the datasets.

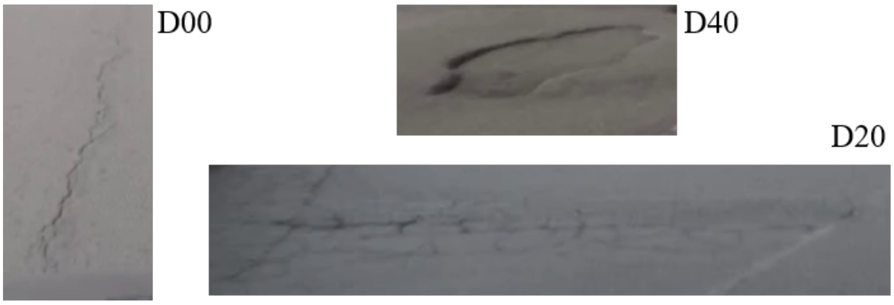


Figure 2.10: Road damages drawn from the example images in Appendix A

Road Damage Dataset 2020

The RDDS was the data basis of the IEEE GRDDC[3] 2020 and builds on previous datasets from Japan [66, 111]. It includes images of roads from Japan, India, and the Czech Republic. The GRDDC divided the data into three groups, one for training and two iterations of test sets to evaluate the results of the challenge participants. Only the training data was released with damage annotations according to the Japanese Road Maintenance and Repair Guidebook 2013 [65], also proposed in Maeda et al. 2018 [66]. Table 2.1 displays the annotation classes of the guidebook, although the GRDDC includes only four classes D00 (longitudinal cracks), D10 (lateral cracks), D20 (alligator cracks, and D40 (potholes). The other annotation classes are not considered to be significant road damage and differ greatly from country to country, which makes them less applicable in DL [24, 24]. Section 2.4.3 displays an aligator crack, a longitudinal crack and a pothole drawn from the RDS. In Appendix A full scene of different damages are displayed in various countries representative for the ADS.

Figure 2.11 shows the number of damages in the training set for each class. With a total of 8381 damages, aligator cracks are the most common, followed by longitudinal cracks (6592), potholes (5627), and lateral cracks (4446).

A smartphone is mounted on the windshield in the inner cabin of the vehicle to capture images of the road ahead. The majority of the images have a resolution of 600 px × 600 px at a frame rate of 1 fps. The images in India are captured in 720 px × 960 px and resized to 720 px × 720 px to gain a squared aspect ratio. About 50% of the damages are located in Japan, 37% in India, and 13% in the Czech Republic.

The data annotation enables a dataset to be used in CV. Bounding boxes

[3] Official IEEE GRDDC website: https://rdd2020.sekilab.global/

Table 2.1: Road damage classes according to the Japanese Road Maintenance and Repair Guidebook 2013 utilized in Maeda et al. 2018 [66]

Damage Type			Detail	Class Name
Crack	Linear Crack	Longitudinal	Wheel-marked part	D00
			Construction joint	D01
		Lateral	Equal interval	D10
			Construction joint	D11
	Alligator Crack		Partial pavement	D20
Other Damage			Pothole	D40
			Manhole Cover	D50
			Storm Drain	D51

over individual damages mark their location in the image. A damage is fully annotated when defined with a class label and the bounding box coordinates, being x_{min}, x_{max}, y_{min}, and y_{max}.

The RDDS is employed in the Publications VII, VIII, and IX of this dissertation. In Publications VIII and IX, it is applied in conjunction and comparatively with the ADS explained in the following.

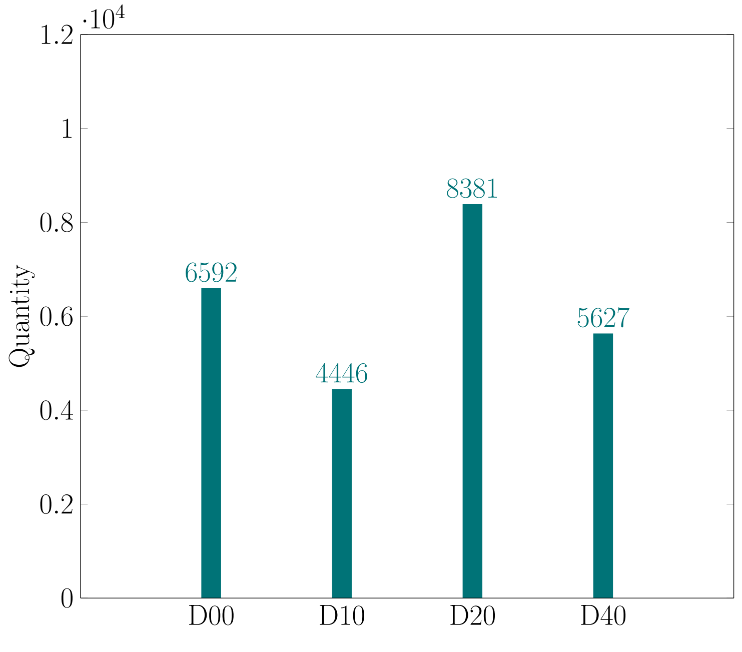


Figure 2.11: Distribution of road damage classes in the Road Damage Dataset

Automotive Dataset

The videos from which the images for ADS are derived were captured by HELLA Aglaia Mobile Vision GmbH and were entrusted to us for the purpose of RDD. The frame rate of the videos is 30 fps, and we extract three images a second from the videos for labeling. Due to the research setting of the work in the E-LAB of HELLA GmbH & Co. KGaA, the dataset is not publicly available.

It contains 10,421 images from Germany (40%), the USA (28%), the United Kingdom (10%), South Korea (10%), Poland (4%), France (3%), Latvia (3%), and Finland (2%). An automotive-grade camera from Sony Corporation was used to capture the images in 3504 px × 1072 px resolution. The camera is installed flush with the windshield and oriented at the scene in the direction of travel.

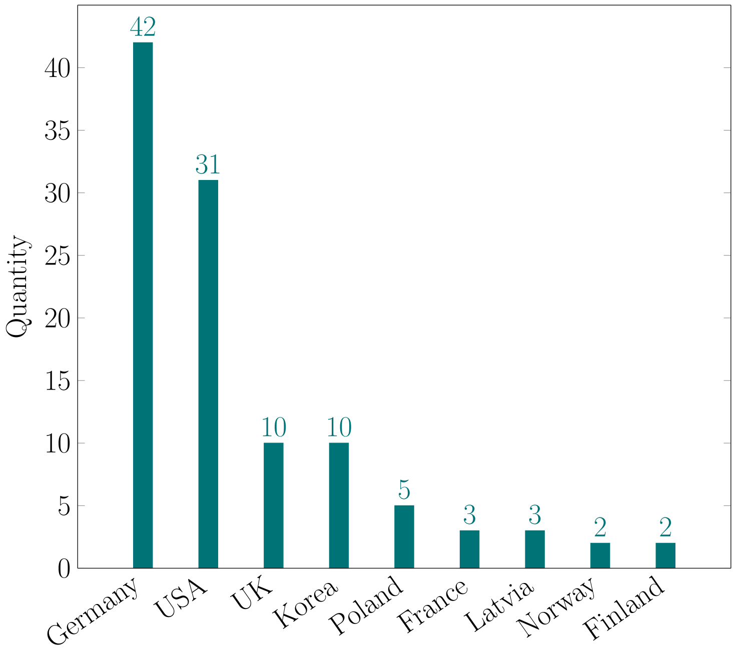


Figure 2.12: Video sequences from which the images of the ADS are derived with regard to their country of origin

We completed the image labeling ourselves. Based on our previous research and work with the RDD dataset in Publication VII, we were able to label subjectively better in comparison, but we did not find a way to quantify this statement. Besides the above mentioned classes of the GRDDC, we additionally labeled longitudinal construction joints (D01), lateral construction joints (D11), manhole covers (D50), and storm drains (D51). Figure 2.13 displays the quantity of all classes used in our research in cyan and the classes left for future work in red (cf. Future Research). Construction joints (D01, D11) are the classes that occur most often. Of the annotations we use, longitudinal cracks occur most frequently with a number of 3693. The other three classes, D10, D20, and, D40 range between 1500 and 2000 occurrences.

In addition to the higher resolution, these images are of much better quality than those in the other dataset. No dirty windshields are recorded and the angle to the road is also improved. The data set is of critical importance to the development of AV-tailored DL algorithms that are capable of detecting road damage.

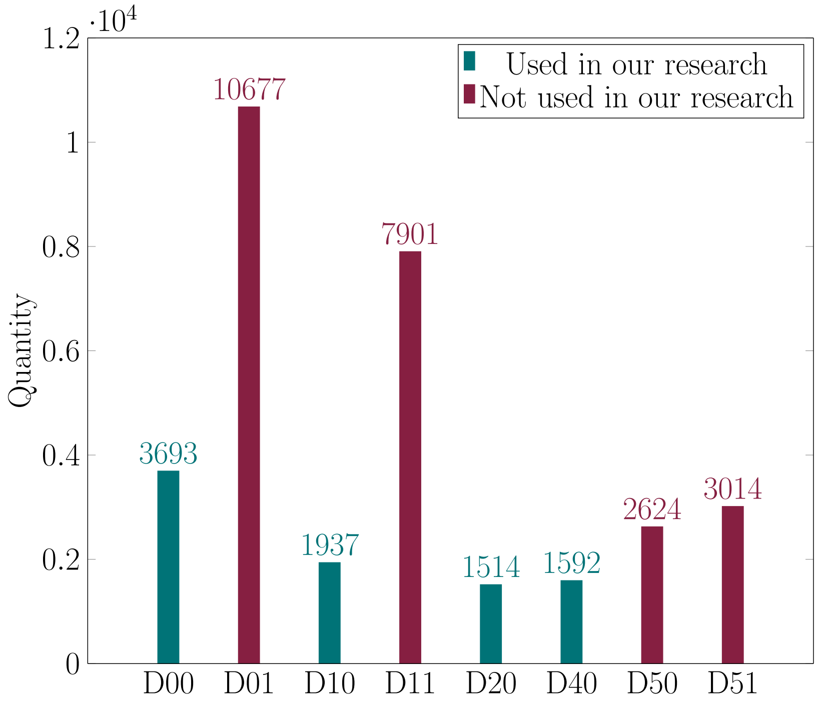


Figure 2.13: Quantity of all damage classes in our self-labeled ADS, including those used in our research (cyan) and those for future work (red)

Chapter 3

Contributions

Regarding a fatality in context with Tesla's autopilot in 2016, Mary Cummings, professor at Duke University and director of the Humans and Autonomy Laboratory, stated that AVs are "absolutely not ready for widespread deployment." To date, many fields of application for autonomous driving systems have been introduced, but we still agree with the statement made five years ago, as a widespread use is given when AVs have a lasting impact on people's mobility in their daily lives. This has not yet been achieved. Instead, AD is gradually being introduced in scenarios of limited complexity.

> *"These accidents are inevitable because the technology really has not been tested across the wide span of people and road conditions."*
>
> *Mary Cummings, Professor at Duke University, 2016*

All contributions of this dissertation share the common goal to improve AD regarding safety and comfort by appropriately detecting road damages and consider them in vehicle motion planning. As a result of the applied research topic, the theoretical contributions lead to practical implications in most cases. We highlight specific areas where this deviates. In the following, we describe our contributions in detail to answer the RQs raised in Section 1.3.

The first contribution of this work is the end-to-end concept that utilizes on-board sensor technology to detect road damages, which can be put to immediate use in the VMPS. Our approach, detailed in Publication I, represents what we believe to be a necessary level of integration, through the sensor fusion of Artificial Intelligence (AI) and analytical methods. Often, time-to-market is much more important for the success of products and features than is the quality of the solution, which is a danger in our use case if we, for example, solely rely on AI to be quickly applied [112]. Our end-to-end concept therefore offers a lightweight solution that requires no new hardware components to be implemented and, through utilization of the QVM, delivers more safety by validating detected damages. Proposing the concept, we provide a contribution to theory with direct implications for practice.

The second part of the contributions deal with our technology inventions. We propose technology to successfully detect road damages from a moving vehicle. We answer RQ1 by developing a novel model-based calculation of the QVM, utilizing the VLS and an AS. Both sensors are pre-installed in modern vehicles and are, in combination, capable of detecting road conditions. Only additional software is required. The model-based calculation of the QVM that is used to measure road elevation is described in Publications II to V. Additionally, Publication VI elaborates on the use of the VLS and AS to detect potholes, speed bumps and distinguishes between different road coatings, such as concrete, asphalt, and gravel, by utilizing classic ML. The novel approach is a contribution to the theory, since based on the quarter vehicle no road heights have been calculated using the VLS. The use of the VLS allows engineers to measure road conditions by vehicles in daily use. The practical implications for vehicle safety and infrastructure planning are manifold.

For the second technology to detect road damages, we, like many others, apply CV using DL with front-facing automotive cameras in the vehicle, answering RQ1 through a different approach than those used in previous studies. We were able to improve the training of the DNNs by effectively applying modern DL tools to the use case. The influences of individual tools (e.g., Data Augmentation (DA), Test Time Augmentation (TTA), and Hyperparameter Evolution (HE)) are analyzed and evaluated with regard to the impact on detection performance and on the runtime of the models based on the results reported in Publications VII to IX.

AVs require lightweight detection algorithms because a large number of tasks have to be performed in parallel in the VPS and powerful hardware is a strong energy consumption and cost driver. Our proposed algorithms evolved to become increasingly applicable between Publications VII to IX and are valuable contributions to RQ2, focusing on the applicability of AI on edge devices. Our contributions to theory can be applied to other application domains. Finding a sweet-spot between performance and runtime is of great relevance for many applications, which is why we have given a comprehensive overview of how to execute DL training in Publication IX, which brings numerous advantages for practitioners. Our expectation is that this analysis will assist developers in other domains with optimizing their models.

While research focuses on detecting road damages, the classification of the severity have not been addressed in DL. We contribute two approaches on how to assess damage severity utilizing DL in Publications I and VIII.

The use of the ADS, our self-labeled dataset of more than 10,000 images (cf. Appendix A), in DL offers many benefits when applied in AD. Despite the significant effort involved in labeling such a high number of images, the object detection performance of DNNs is closely related to the quality of the training data. We present the notable impact of new data in Publications VIII and IX. The selection of different traffic scenarios, daytime, and weather conditions in the ADS further increases the generalizability. In this way, we contribute to research by demonstrating the great influence of the dataset on the performance of object detection.

Chapter 4

Limitations

Designing technology, proposing concepts, or building frameworks imposes limitations that are always related to the utilized research method and design itself. An obvious but nevertheless major limitation of all the publications in this thesis was the lack of time. The topics that would have been desirable to investigate further are listed in the next section, Future Research.

All models are always simplifications of the reality. Limitations result from the model boundaries. Although model boundaries are desired, often to constrain the complexity of a technology or issue and thereby make it calculable, the limitations must be made transparent.

Below, we separately discuss the limitations for the QVM-based calculation and those for CV approaches. Afterwards, we explain the limitations of the end-to-end (E2E) concept proposed in Publication I.

QVM.1. The isolated analysis of the quarter-vehicle is a limitation of the approach, as the influences of the remaining three quarters of the vehicle on the sprung mass are not taken into account.

QVM.2. In our modeling, the test vehicle and the laboratory demonstrator are parameterized. This includes the vehicle and wheel mass, as well as spring and damper constants. As vehicles age, these parameters change, sometimes considerably (e.g., tires). We do not take this into account.

Despite the acknowledged limitations, the elevation profile qualitatively and quantitatively (including errors) represents the profile and strives for pragmatism over truth [26].

Object detection in CV includes limitations of a general nature and specific, framework-dependent limitations. We acknowledge these in the following:

CV.1. In principle, DNNs can only be as good as the database on which they are trained. This applies especially to our use case, as two-thirds of the training data consists of the RDDS provided from the GRDDC. Though, we are grateful for the public dataset, the image quality suffers from rather poor viewing angles, dirty windshields, missing variance of weather conditions, and in parts, poor label quality. In addition, the resolutions of the smartphone cameras used are very different from our automotive-grade cameras. We acknowledge that these preconditions are not ideal, which negatively influences and limits the results. By adding our ADS (cf. Section 2.4.3 and Appendix A), we were able to achieve improved results and lay the foundation for further optimization.

CV.2. Despite significant research efforts to improve object recognition, it is assumed that 100% object detection cannot be realized. This is especially true when developing DL models that have to run under limited resources and therefore require trade-offs between detection performance and runtime.

The proposed end-to-end concept in Publication I is a valuable contribution because it combines CV-based RDD and QVM-based RDD to be applied for increased effectiveness. However, as the word "concept" implies, it is still a first approach that needs to be shaped. Further, the interfaces of the components need to be defined. This results in the following three limitations:

E2E.1. Despite the proposed concept being end-to-end, the details given for implementation are quite low regarding the aggregation components (cf. C3 and C4 in Publication I). Our pragmatic approach of DSR resulted in a T-shaped research structure: a high-level end-to-end concept and in-depth algorithm designs for C1.

E2E.2. The implementation of the road damage features designed for the HD Feature Map remains unclear regarding the exact semantic structure of the HD Feature Map in the vehicle and online.

E2E.3. The sensor fusion of the QVM and CV approaches to calculate the damage type, severity, and location is not designed to the end. This includes whether a detected damage is passed through to be evaluated by the QVM approach.

Chapter 5

Future Research

Many different adjustments, tests, and experiments were left for future research, as the experiments carried out in this work are very time-consuming. This affects both approaches for detecting road conditions with the model-based calculation (Publications I-VI) requiring vehicle conversion, signal processing, sensor data filtering, and real-world testing. In our CV-related research (Publications VII-IX), high time expenditures occurred in training and the labelling process of new data in particular.

With regard to the QVM approach, investigation of the following topics will lead to an improvement of the approach and an optimized application in vehicles.

QVM.1. An automatic detection of road damage from the obtained elevation profile of the QVM-based calculation. In this context, it is necessary to investigate how crowd sourcing can compensate for measurement errors, as each drive will pass through a damage differently (e.g., centered, touched).

QVM.2. A comparison of analyses of other vehicles with different QVM parameterizations using the model-based calculation should be performed to prove similar outcomes. Deviating outcomes could be assessed to improve the model design by disclosing influences that lead to measurement errors.

QVM.3. A segment-by-segment evaluation of road sections on the basis of the calculated elevation profile. It would be particularly exciting to exchange ideas with and receive input from infrastructure planners in order to define a target for a platform that is fed from our QVM-based calculation of road conditions (this also includes dealing with crowd data).

In the field of CV technology, especially in object detection of road damages using DL, the following topics are of particular interest to us:

CV.1. The addition of further damage classes or other classes often associated with misclassification of damages (e.g. construction joints, fading road markings) in training. This could potentially avoid misclassifications of such elements when they are labeled as *background* in the training data. If a DNN has a higher confidence that a manhole cover is a pothole as opposed to *background*, adding the class *manhole cover* could make the confidence for this class even higher than for potholes, thus preventing misclassification (diversification of the feature space).

CV.2. At present, we use individual images, but in the vehicle we do have sequences of images that are related to each other, because we are driving toward damages. It would be exciting to use Long Short-Term Memory (LSTM) networks to predict future positions of initially detected damages. If damages recurrently occur in the expected area, the presence of the damage should be marked with a higher confidence, thus increasing the likelihood of immediate use in the VMPS.

CV.3. Currently, the application of the YOLO frameworks for our use case leads to impressive results. Transfer Learning (TL) seems to be very powerful. Despite our failed attempts, not discussed in this dissertation, to develop a specifically designed DNN for the use case, a DL model redesigned from scratch should still be a future topic of investigation.

Apart from the two RDD approaches, potential research topics open up at a higher level of abstraction. The application domain demands an end-to-end concept including solutions for RDD and RDS classification:

E2E.1. Lightweight semantics that can be implemented with sufficient granularity to account for road damage in the HD Feature Map should be discussed and developed as a standard.

E2E.2. The aggregation of road damages in the cloud, transmitted by a large number of vehicles, as “many to one” algorithms will be studied and researched. Here, special attention must be paid to the case of damage repairs by infrastructure works.

E2E.3. At the edge level in the AVs, the aggregation of predictions from the CV algorithms and actual QVM-model calculations must be realized in a central processing unit (C3 in Publication I).

E2E.4. For a successful integration of the features, the cost functions in the VMPS of the high-level (route) and low-level (behavior, motion) have to be adjusted. The influence on the planning tasks has to be evaluated depending on the type of damage and its severity and tested in field studies.

Bibliography

[1] C. Pakusch, G. Stevens, A. Boden, and P. Bossauer, "Unintended effects of autonomous driving: A study on mobility preferences in the future," *Sustainability*, vol. 10, no. 7, p. 2404, 2018.

[2] S. D. Gleave, R. Frisoni, F. Dionori, L. Casullo, C. Vollath, L. Devenish, F. Spano, T. Sawicki, S. Carl, R. Lidia, J. Neri, R. Silaghi, and A. Stanghellini, "Eu road surfaces: Economic and safety impact of the lack of regular road maintenance–study," *Transport and Tourism*, pp. 1–5, 2014.

[3] T. Adams, "Self-driving cars: From 2020 you will become a permanent backseat driver," Sep 2015.

[4] O. Garret, "10 million self-driving cars will hit the road by 2020 – here's how to profit," Mar 2017.

[5] L. P. Robert, "Are automated vehicles safer than manually driven cars?," *AI & SOCIETY*, vol. 34, no. 3, pp. 687–688, 2019.

[6] U. Di Fabio, M. Broy, R. Brüngger, U. Eichhorn, A. Grunwald, and D. Heckmann, "Ethik-kommission automatisiertes und vernetztes fahren," 2017.

[7] J. Greene, F. Rossi, J. Tasioulas, K. B. Venable, and B. Williams, "Embedding ethical principles in collective decision support systems," in *Thirtieth AAAI Conf. on artificial intelligence*, 2016.

[8] V. Conitzer, W. Sinnott-Armstrong, J. S. Borg, Y. Deng, and M. Kramer, "Moral decision making frameworks for artificial intelligence," in *Thirty-first aaai Conf. on artificial intelligence*, 2017.

[9] E. Awad, S. Dsouza, R. Kim, J. Schulz, J. Henrich, A. Shariff, J.-F. Bonnefon, and I. Rahwan, "The moral machine experiment," *Nature*, vol. 563, no. 7729, pp. 59–64, 2018.

[10] SAE, "Levels of driving automation are defined in new sae international standard j3016: 2014," *SAE Int.: Warrendale, PA, USA*, vol. 1, 2014.

[11] M. Berk, O. Schubert, B. Buschardt, H.-M. Kroll, and D. Straub, "Assessing the safety of environment perception in automated driving vehicles," *SAE Int. J. Trans. Safety*, 2020.

[12] M. Hoss, M. Scholtes, and L. Eckstein, "A review of testing object-based environment perception for safe automated driving," *arXiv preprint arXiv:2102.08460*, 2021.

[13] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[14] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of driver assistance systems.* Springer Int. Publishing Amsterdam, The Netherlands:, 2014.

[15] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous driving*, pp. 425–449, Springer, 2016.

[16] M. Berk, *Safety Assessment of Environment Perception in Automated Driving Vehicles.* PhD thesis, Technische Universität München, 2019.

[17] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[18] M. Bunge, *Chasing reality.* University of Toronto Press, 2016.

[19] G. Maerschalk, A. Ueckermann, and S. Heller, "Längsebenheitsauswerteverfahren," *Bewertetes Längsprofil": Weiterentwicklung der Längsebenheitsbewertung der Zustandserfassung und–Bewertung,"[Analysis procedure for longitudinal road roughness "Weighted Longitudinal Profile"], Report of the Federal Highway Research Institute (bast)*, no. S73, 2011.

[20] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[21] M. Broy, I. H. Kruger, A. Pretschner, and C. Salzmann, "Engineering automotive software," *Proc. of the IEEE*, vol. 95, no. 2, pp. 356–373, 2007.

[22] N. H. Thuan, A. Drechsler, and P. Antunes, "Construction of design science research questions," *Communications of the Association for Information Systems*, vol. 44, no. 1, p. 20, 2019.

[23] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv preprint arXiv:2008.13101*, 2020.

[24] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," in *2020 IEEE Int. Conf. on Big Data (Big Data)*, pp. 5533–5539, IEEE, 2020.

[25] J. v. Brocke, A. Simons, B. Niehaves, B. Niehaves, K. Reimer, R. Plattfaut, and A. Cleven, "Reconstructing the giant: On the importance of rigour in documenting the literature search process," 2009.

[26] P. J. Ågerfalk, "Getting pragmatic," *European J. of Information Systems*, vol. 19, no. 3, p. 251–256, 2010.

[27] G. Goldkuhl, "Pragmatism vs interpretivism in qualitative information systems research," *European J. of information systems*, vol. 21, no. 2, pp. 135–146, 2012.

[28] J. Iivari, "Distinguishing and contrasting two strategies for design science research," *European J. of Information Systems*, vol. 24, no. 1, pp. 107–115, 2015.

[29] P. Offermann, S. Blom, M. Schönherr, and U. Bub, "Artifact types in information systems design science–a literature review," in *Int. Conf. on Design Science Research in Information Systems*, pp. 77–92, Springer, 2010.

[30] A. R. Hevner, "A three cycle view of design science research," *Scandinavian J. of Information Systems*, vol. 19, no. 2, p. 4, 2007.

[31] R. A. Krueger, *Focus groups: A practical guide for applied research.* Sage publications, 2014.

[32] R. Plonsey, R. C. Barr, and A. Bioelectricity, *Quantitative Approach.* Springer, 2007.

[33] K. A. Smith, "Discussion of the method: Conducting the engineer's approach to problem solving," *J. of Engineering Education*, vol. 92, no. 3, p. 205, 2003.

[34] R. Nollau, *Modellierung und Simulation technischer Systeme: eine praxisnahe Einführung.* Springer Science & Business Media, 2009.

[35] M. Beaudouin-Lafon and W. E. Mackay, "Prototyping tools and techniques," in *Human-Computer Interaction*, pp. 137–160, CRC Press, 2009.

[36] T. Ritchey, "General morphological analysis (gma)," in *Wicked problems–Social messes*, pp. 7–18, Springer, 2011.

[37] A. T. Mısırlı, A. Bener, B. Çaglayan, G. Çalıklı, and B. Turhan, "A methodology for construction and evaluation of recommendation systems in software engineering," *Recommendation Systems in Software Engineering*, p. 329, 2014.

[38] C. Luetge, "The german ethics code for automated and connected driving," *Philosophy & Technology*, vol. 30, no. 4, pp. 547–558, 2017.

[39] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, p. 113816, 2020.

[40] R. W. Wolcott and R. M. Eustice, "Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving," *The Int. J. of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.

[41] L. D. P. Veronese, J. Guivant, F. A. A. Cheein, T. Oliveira-Santos, F. Mutz, E. de Aguiar, C. Badue, and A. F. De Souza, "A light-weight yet accurate localization system for autonomous cars in large-scale and complex environments," in *2016 IEEE 19th Int. Conf. on intelligent transportation systems (ITSC)*, pp. 520–525, IEEE, 2016.

[42] J. Rohde, I. Jatzkowski, H. Mielenz, and J. M. Zöllner, "Vehicle pose estimation in cluttered urban environments using multilayer adaptive monte carlo localization," in *2016 19th Int. Conf. on information fusion (FUSION)*, pp. 1774–1779, IEEE, 2016.

[43] Y. Xu, V. John, S. Mita, H. Tehrani, K. Ishimaru, and S. Nishino, "3d point cloud map based vehicle localization using stereo camera," in *2017 IEEE intelligent vehicles symposium (IV)*, pp. 487–492, IEEE, 2017.

[44] A. Viswanathan, B. R. Pires, and D. Huber, "Vision-based robot localization across seasons and in remote locations," in *2016 IEEE Int. Conf. on robotics and automation (ICRA)*, pp. 4815–4821, IEEE, 2016.

[45] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 652–665, 2015.

[46] J. Ziegler, H. Lategahn, M. Schreiber, C. G. Keller, C. Knöppel, J. Hipp, M. Haueis, and C. Stiller, "Video based localization for bertha," in *2014 IEEE intelligent vehicles symposium*, pp. 1231–1238, IEEE, 2014.

[47] L. J. Lyrio, T. Oliveira-Santos, C. Badue, and A. F. De Souza, "Image-based mapping, global localization and position tracking using vg-ram weightless neural networks," in *2015 IEEE Int. Conf. on robotics and automation (ICRA)*, pp. 3603–3610, IEEE, 2015.

[48] G. L. Oliveira, N. Radwan, W. Burgard, and T. Brox, "Topometric localization with deep learning," in *Robotics Research*, pp. 505–520, Springer, 2020.

[49] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[50] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual slam by tracking moving objects," in *2009 IEEE Int. Conf. on robotics and automation*, pp. 375–380, IEEE, 2009.

[51] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[52] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 19–29, 2015.

[53] H. G. Seif and X. Hu, "Autonomous driving in the icity—hd maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, 2016.

[54] S. Zheng and J. Wang, "High definition map-based vehicle localization for highly automated driving: Geometric analysis," in *2017 Int. Conf. on Localization and GNSS (ICL-GNSS)*, pp. 1–8, IEEE, 2017.

[55] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun, "Dagmapper: Learning to map by discovering lane topology," in *IEEE/CVF Int. Conf. on Computer Vision*, pp. 2911–2920, 2019.

[56] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *The J. of Navigation*, vol. 73, no. 2, pp. 324–341, 2020.

[57] K. Jo, C. Kim, and M. Sunwoo, "Simultaneous localization and map change update for the high definition map-based autonomous driving car," *Sensors*, vol. 18, no. 9, p. 3145, 2018.

[58] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm engineering*, pp. 19–80, Springer, 2016.

[59] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[60] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*, vol. 56. Springer, 2009.

[61] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, "Experience, results and lessons learned from automated driving on germany's highways," *IEEE Intelligent transportation systems magazine*, vol. 7, no. 1, pp. 42–57, 2015.

[62] B. Okumura, M. R. James, Y. Kanzawa, M. Derry, K. Sakai, T. Nishi, and D. Prokhorov, "Challenges in perception and decision making for intelligent automotive vehicles: A case study," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 20–32, 2016.

[63] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Autonomous Robots*, vol. 41, no. 6, pp. 1367–1382, 2017.

[64] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[65] JRA, *Maintenance and Repair Guide Book of the Pavement 2013.* Japan Road Association, 2013.

[66] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.

[67] M. W. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," 1998.

[68] C. Dodds and J. Robson, "The description of road surface roughness," *J. of sound and vibration*, vol. 31, no. 2, pp. 175–183, 1973.

[69] M. W. Sayers, "Guidelines for the conduct and calibration of road roughness measurements," tech. rep., 1984.

[70] D. Schramm, M. Hiller, and R. Bardini, "Vehicle dynamics," *Modeling and Simulation. Berlin, Heidelberg*, vol. 151, 2014.

[71] A. Kruczek and A. Stribrsky, "A full-car model for active suspension-some practical aspects," in *IEEE Int. Conf. on Mechatronics*, pp. 41–45, IEEE, 2004.

[72] D. Schramm, M. Hiller, and R. Bardini, "Single track models," in *Vehicle Dynamics*, pp. 223–253, Springer, 2014.

[73] R. Rajamani, *Vehicle dynamics and control.* Springer Science & Business Media, 2011.

[74] M. F. Ismail, Y. Sam, S. Sudin, K. Peng, and M. K. Aripin, "Modeling and control of a nonlinear active suspension using multi-body dynamics system software," 2014.

[75] D. Karnopp, M. J. Crosby, and R. Harwood, "Vibration control using semi-active force generators," 1974.

[76] E. Guglielmino, T. Sireteanu, C. W. Stammers, G. Ghita, and M. Giuclea, *Semi-active suspension control: improved vehicle ride and road friendliness.* Springer Science & Business Media, 2008.

[77] D. Hrovat, “Survey of advanced suspension developments and related optimal control applications,” *Automatica*, vol. 33, no. 10, pp. 1781–1817, 1997.

[78] J. Levitt and N. Zorka, “The influence of tire damping in quarter car active suspension models,” *ASME J. Dynamic Systems, Measurement and Control*, vol. 113, pp. 134–137, 1991.

[79] S. Nie, Y. Zhuang, F. Chen, and J. Xie, “Invariant points of semi-active suspensions,” *Advances in Mechanical Engineering*, vol. 10, no. 7, p. 1687814018777316, 2018.

[80] U. ECE, “Uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signaling devices,” *Regulation*, no. 48, 2010.

[81] H. Duda and S. Berkner, “Integrated chassis control using active suspension and braking,” in *7th Int. Symposium on Advenced Vehicle Control (AVEC)*, vol. 347, p. 352, 2004.

[82] M. Münster, U. Mair, H.-J. Gilsdorf, A. Thomä, C. Müller, M. Hippe, and J. Hoffmann, “Electromechanical active body control,” *ATZautotechnology*, vol. 9, no. 3, pp. 24–29, 2009.

[83] H. Weber, H. Baran, F. Utermöhlen, and C. Schuster, “Macromodeling of mutual inductance for displaced coils based on laplace’s equation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[84] M. Gad-el Hak, *The MEMS handbook.* CRC press, 2001.

[85] J. W. Judy, “Microelectromechanical systems (mems): fabrication, design and applications,” *Smart materials and Structures*, vol. 10, no. 6, p. 1115, 2001.

[86] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The J. of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[87] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.

[88] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[89] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[90] G. Bologna and Y. Hayashi, "Characterization of symbolic rules embedded in deep dimlp networks: a challenge to transparency of deep learning," *J. of Artificial Intelligence and Soft Computing Research*, vol. 7, 2017.

[91] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93, Elsevier, 1992.

[92] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[93] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[94] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conf. on computer vision*, pp. 818–833, Springer, 2014.

[95] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[96] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conf. on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[97] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[98] W. Zhang, K. Itoh, J. Tanida, and Y. Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," *Applied optics*, vol. 29, no. 32, pp. 4790–4797, 1990.

[99] R. Girshick, "Fast r-cnn," in *IEEE Int. Conf. on computer vision*, pp. 1440–1448, 2015.

[100] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[101] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[102] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[103] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[104] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on computer vision and pattern recognition*, pp. 779–788, 2016.

[105] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[106] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 13029–13038, 2021.

[107] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *IEEE Conf. on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[108] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, pp. 448–456, PMLR, 2015.

[109] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[110] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[111] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2021.

[112] J. Chen, R. R. Reilly, and G. S. Lynn, "The impacts of speed-to-market on new product success: the moderating effects of uncertainty," *IEEE Transactions on engineering management*, vol. 52, no. 2, pp. 199–212, 2005.

Publications

Chapter I

Watch out, pothole! Featuring Road Damage Detection in an End-to-end System for Autonomous Driving

Outline

Bibliographic Information

Felix Kortmann, Pascal Fassmeyer, Burkhardt Funk, and Paul Drews, (**under review**, 2021, October). “Watch out, pothole! Featuring Road Damage Detection in an End-to-end System for Autonomous Driving”. In Data & Knowledge Engineering. Elsevier.
Preprint DOI: 10.13140/RG.2.2.29894.19527.

Author’s contribution

The author’s share of the publication is 65%. Table B.1 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

While autonomous driving technology made significant progress in the last decade, road damage detection as a relevant challenge for ensuring safety and comfort is still under development. This paper addresses the lack of algorithms for detecting road damages that meet autonomous driving systems’ requirements. We investigate the environmental perception systems’ architecture and current algorithm designs for road damage detection. Based on the autonomous driving architecture, we develop an end-to-end concept that leverages data from low-cost pre-installed sensors for real-time road damage and damage severity detection as well as cloud- and crowd-based HD Feature Maps to share information across vehicles. In a design science research approach, we develop three artifacts in three iterations of expert workshops and design cycles: the end-to-end concept featuring road damages in the system architecture and two lightweight deep neural networks, one for detecting road damages and another for detecting their severity as the central components of the system. The research design draws on new self-labeled automotive-grade images from front-facing cameras in the vehicle and interdisciplinary literature regarding autonomous driving architecture and the design of deep neural networks. The road damage detection algorithm delivers cutting-edge

performance while being lightweight compared to the winners of the IEEE Global Road Damage Detection Challenge 2020, which makes it applicable in autonomous vehicles. The road damage severity algorithm is a promising approach, delivering superior results compared to a baseline model. The end-to-end concept is developed and evaluated in and with experts of the application domain.

I.1 Introduction

AVs are on the verge of being introduced into our daily lives. While many steps have been taken towards the goal of full automation in recent decades, several technical problems related to functional safety [1] and socio-economic issues remain unsolved, primarily in the field of AI [2].

AI claims to have a transformational influence in Information Systems (ISs) across sectors and industries, accelerating ”as people get used to the reduced human element in all levels of society and the increased use of automation” [3]. As a result, the development of ISs is increasingly intertwined with the development of AI algorithms.

Leading car manufacturers already offer systems involving AI that are far ahead of human-guided driving in terms of safety (S). “People are so bad at driving cars that computers don’t have to be that good to be much better”, stated Andreessen in 2011. However, despite AVs being safer, AD systems systems are not yet widely available on the market, as much of the intelligence is based on learned knowledge and thus cannot easily deal with unexpected situations — situations not present in the training data [4].

The AD task can be divided into two sub-areas: recognition of the environment (perception system) and the motion of the vehicle in this environment (decision-making system, [5]). Levels of AD range from no automation, level 0, to full automation, level 5 [6]. As the level increases, the involvement of the human decreases, while responsibility of the AD system increases. AD functions of various automation levels require representations of the real environment to navigate based on the situation. Perception algorithms are used to create an environment model, represented by HD Feature Maps.

According to [7], the detection of the road surface has already been an essential part of environment detection in the development of Advanced Driver Assistance Systems (ADASs), automation level 1–3, and is even more important with increasing automation [8]. The condition of the road plays a decisive role in this regard. Subjects of investigation include the type of road coating (e.g., concrete, cobblestone, asphalt), the weather conditions on the road (e.g., dry, wet, icy), and significant road damages such as cracks and potholes [7, 9].

As in manual driving, information about road conditions will be required, especially considering poor infrastructure in developing countries and even high-tech countries such as the USA. Claims amounting to $217 billion per year alone are connected to poor road conditions in the USA. 42,000 deaths occur each year, with road conditions being the cause or at least a factor in 52% of the cases [10]. The vehicle's mechanical components are subject to greater wear and tear in bad conditions, which accounts for a significant portion of the costs. AVs have to deal with aging and dilapidated road networks, because the global infrastructure is unlikely to improve in the coming years. To ensure that countries with poor infrastructure also experience the positive effects of AD on society, the vehicle must actively respond to poor conditions, especially significant road damages.

Car manufacturers seem not to be concerned with poor road conditions yet as early mass-produced AVs will offer AD only in scenarios of low complexity such as highways with a tendency to good road quality. However, in response to a Twitter user, Elon Musk, the CEO of Tesla, replied in 2020, "We're labeling bumps & potholes, so the car can slow down or steer around them when safe to do so". Despite early solutions with a low level of integration into the AD task such as the Volvo's Hazard Light Alert (2016) and Slippery Road Alert (2019) [11], no end-to-end solution yet exists in research and industry to tackle road damages in AD. In addition to the impact on AD, the monitoring of road conditions for road maintenance is of great importance due to its high economic impact on society [12].

Previous IS research in the domain of road damages dealt with decision support systems for road agencies [13]. In contrast to systems that support people, we are engaged in systems for systems in order to improve automation and thereby increase the still missing acceptance [3].

Academia though primarily discusses single technical solutions for a limited problem (e.g., object detection, segmentation) rather than a holistic approach to how road conditions can be utilized to actively support AD. The impact of RDD algorithms proposed in research remains mostly unexplained on AD (cf. section Theoretical Review). Original equipment manufacturers are less and less willing to add new hardware components for new features in the vehicle as this development has already been a major cost driver in recent decades. A low-cost solution is therefore being sought to reliably detect road conditions and classify their severity (RDS), if possible using existing sensor systems.

We strive to develop an end-to-end and low-cost concept that enables AVs to detect road conditions and properly act according to them. Due to our ambition for a practical contribution, we refer to the DSR paradigm.

Supported and validated by expert workshops, we address the following research objectives with the focus on creating added value in a targeted and application-oriented manner:

1. We strive to propose an end-to-end concept that enables AVs to account for road damages in their motion planning task while making use of built-in sensor technology.

2. We strive to develop novel modules for RDD and RDS classification applicable in the end-to-end concept for AD.

I.2 Theoretical Review

Approaching the objectives from an information system's perspective is of great benefit since an end-to-end solution must be embedded in the actual process of dealing with environmental data in AD. As a solution consisting of many components with application in the AD architecture, we must define an "as it is" architecture based on leading research so that we can explain the effect of the components regarding the architecture. AD requires complex and high-tech modules that interact with each other to succeed driving autonomously. Due to their great relevance for our concept of an end-to-end solution of utilizing road damages for AD, we will first discuss the Autonomous Driving System and especially the Environment Modeling in the next subsections.

Subsequently, we will cover two approaches of recognizing road damages, one to predict road damages ahead of the vehicle and another to validate the damages according to the vehicle feedback caused by the damage. Following our motivation, the technical concepts of both approaches merely require the use of pre-installed vehicle sensors.

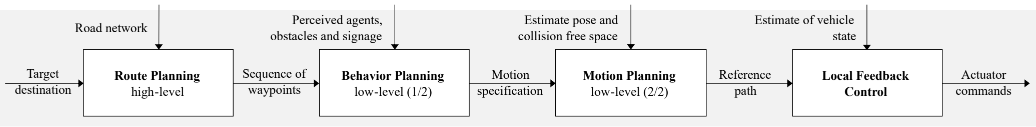

Road network
Perceived agents, obstacles and signage
Estimate pose and collision free space
Estimate of vehicle state
Target destination
Route Planning high-level
Sequence of waypoints
Behavior Planning low-level (1/2)
Motion specification
Motion Planning low-level (2/2)
Reference path
Local Feedback Control
Actuator commands

Figure I.1: Vehicle Planning System for AD according to [14]

I.2.1 Autonomous Driving System

AD has undergone many developments in the last 30 years. Although the first AVs were developed in the 1980s [15], vehicle automation continues to be a major area of research. Most driving scenarios are relatively simple and have already been realized by early prototypes, but solutions to borderline cases in an arbitrarily complex world have proven to be elusive. A 99.9% safety rate is not sufficient in an application that is critical to one's life and the lives of other road users. Hence, research in this area is expected to continue even if the products are already on the market. AD systems in academia and industry differ in design and complexity but all share a division into perception and decision-making systems [5].

The perception system is responsible for self-localization and recognition of the environment. Typically, on-board sensors, including camera, LIDAR, RADAR, and Global Positioning System (GPS) sensors create an ad-hoc model of the surrounding environment, including but not limited to the road network, the traffic situation, pedestrians, and traffic signage/lights.

The decision-making system is responsible for the navigation of the vehicle within the environment. It manages the navigation task of the passenger, starting from the initial position and considering established traffic rules and the safety of all traffic participants while providing a comfortable driving experience [5]. The decision-making system executes multiple planning tasks to appropriately evaluate the vehicle's route, path, behavior, and motion. As AVs advance toward realistic road traffic, they will be confronted with road scenarios in which the dynamics of other road users must be explicitly considered. These situations include everyday driving maneuvers, such as merging into the flow of traffic, overtaking oncoming traffic, changing lanes, or avoiding other vehicles.

Cost functions are ideally suited to solve this problem. With the help of a cost function understandable and representative for humans, the forecasted traffic scenarios can be evaluated and the best trajectory can be selected based on the lowest cost [16]. In most cases, safety, compliance with traffic regulations, comfort, and route efficiency play a decisive role in the order of listing [17]. In their survey, [14] define the hierarchy of decision-making processes.

Route Planning, also called high-level path planning, performs the selection of a route through the road network, from the initial position to the desired destination. Optimal route planning is formulated as the search for a minimum-cost path on a road network, while the crossing of a road is always charged with a specific cost based on the road's properties. The optimization problem is addressed in many scientific studies, distinguishing between human-driven and self-driving vehicles [18]. The architecture of the road network is described in detail in the Environment Modeling section. A prerequisite for route planning is up-to-date information about the road network and the capacity to utilize this information.

Behavior Planning is the first part of the low-level path planning and takes place within the selected route. The general task involves executing maneuvers according to traffic rules and behavior conventions to pass through the route. The selection of maneuvers is based on the underlying road and lane network and the current traffic situation [19], considering other road users, obstacles, and infrastructure signage. The real-time data must be obtained from the vehicle's perception system. Possible maneuvers include maintaining a lane, changing lanes, turning, or stopping the vehicle.

In the second part of the low-level planning, Motion Planning, the selected maneuver is planned in such a way that the vehicle is guided safely and comfortably to the desired position with a dynamically feasible trajectory. In this processing stage, standard methods of motion planning are drawn from the robotic literature. State-of-the-art path planning methods include but are not limited to visibility graph-search methods [20] and incremental search [21].

The Local Feedback Controller is used to select appropriate actuator inputs to execute the planned motion and correct tracking errors. The tracking errors generated during the execution of a planned motion are partly due to the inaccuracies of the vehicle model. Therefore, great emphasis is placed on the robustness and stability of the control loop.

The described AD system is needed to understand where road damage features can potentially add value for AD. The section End-to-end Concept closes with how road damage features will be utilized to improve the route, behavior, and motion planning.

I.2.2 Environment Modeling

The perception system of AVs is responsible for understanding the vehicle's environment, which involves estimating its pose (position and orientation) in the 3D world and the perception of its local surroundings. In contrast to standard GPS navigation system maps, HD Feature Maps usually contain much richer environmental information, such as roadside objects (e.g., signs, barriers) and lane-level information (e.g., markings, widths). Exploitation of these features enhances the perception system, enabling precise localization with centimeter-level accuracy. [22] provide a comprehensive overview of HD Feature Maps for automated driving.

Vector HD Feature Maps contain precise geometric descriptions of road elements and retain higher-level semantic information, such as road topology, lane type, and speed limit, allowing them to have a significantly smaller data size and to scale to more vehicles and larger regions [23]. According to Ovum's Location Platform Index, the leading service vendor in the location platform market is HERE, followed by Google, Mapbox, and TomTom. Most of them are based on OpenDRIVE [24]. To be more specific, we adopt the HERE HD Live Map structure and terminology in this section.

The HERE HD Map is a multiple-layered hierarchical arrangement composed of the road model, the HD lane model, and an HD localization model. The road model network contains the road topology and can be represented as a directed graph consisting of a set of intersections and a set of road segments [19]. The nodes represent the start and end of road segments (i.e., intersections or dead ends), and the edges depict the course of the road [22]. Different road-related information and semantic information can be encoded, such as travel direction, type, speed limits, elevation, slope, and curvature. An optimal road-level route (i.e., a sequence of road segments from start to goal) can be calculated using edge weights that correspond to the cost of traversing a road segment and then applying a shortest path algorithm [14]. While the road model's smallest modeling unit is the road segment, the lane model provides a more precise representation of the environment and contains the lane topology [19], including lane-level features such as lane lines, lane width, and lane markings.

As the real-world environment around vehicles is constantly changing and emerging, a once-built offline HD Feature Map is not sufficient for accurate environment perception — road conditions deteriorate over time, speed limits and road markings change, construction sites and road hazards come and go. Thus, online updates to the map are necessary to account for the dynamic environment changes. HD Feature Maps are transitioning from the use of systematic industrial capture with LIDAR-equipped vehicles to a crowd-sourced model that can perform map maintenance in near real-time. As the participating vehicles (i.e., the crowd) differ in size, travel path, and sensor setups, the aggregation algorithms must incorporate sensor-fusion and DL approaches that accommodate the different sensor data. The resulting HD Feature Map can be viewed as an extended vehicle sensor and complement the local perception setup [22]. While the road model alone can only support high-level route planning by evaluating road-related information, behavior planning requires powerful real-time perception and decision systems inside the vehicle. However, an updated lane-level map with precise localization provides an exact trajectory and reduces the in-vehicle computational burden [23], as the perception system is also able to utilize the crowd-based map knowledge to initiate appropriate maneuvers.

I.2.3 Camera-based Detection of Road Damages

Recognizing road damages ahead of the vehicle is crucial for AD to actively execute low-level path planning according to the situation. This is especially important if a damage is not yet present in the online HD Feature Map. But even if it is already featured, a local perception increases the probability of existence.

Primarily, camera and LIDAR-based approaches are reported in research to assess road conditions in advance. Recent LIDAR solutions of assessing the road surface focus on grading road damages [25], detecting cracks [26, 27], assessing the flatness of the road [28], estimating roughness [29] and detecting potholes [30].

However, as environment recognition in AD is performed using a combination of camera, LIDAR and RADAR sensors, different sensor combinations always include front-facing cameras but not always LIDAR sensors, which is our reasoning to focus on camera-based RDD only.

The utilization of camera sensors for RDD has been subject of considerable research, primarily in engineering. A substantial amount of research has been conducted on RDD utilizing rule-based algorithms and ML methods. [31], [13], [32] summarize ML- and rule-based approaches, though we focus on RDD with DL methods due to their general superiority in object detection.

From an academic perspective, camera-based detection of road damages reached its preliminary peak in the GRDDC 2020 [33, 34]. The RDDS 2020 was shared among researchers with solely focusing on detection performance, not accounting for speed/applicability. The provided datase consists of 21,041 labeled images from India, Japan, and the Czech Republic [34]. The research content and the techniques used are shown in Table I.1.

In total 121 teams submitted their RDD approaches to the GRDDC which were then rated by an evaluation server through the F1 metric. The test data set includes 5,295 images of longitudinal, transverse, and alligator cracks, as well as potholes. The top-12 ranked teams are displayed in Table I.1 including their solutions, all applying transfer learning with established DL object detection frameworks. Since the only metric of the challenge was detection performance over the F1 score, most of the better teams applied ensemble models, which includes #1, #2, #3 and #6. DA is a common tool to extend the database which is successfully applied by most teams, while the application of TTA leads to varying results, with #1 resulting in an upgrade, but #11 in a downgrade. Likewise, different results occur when applying road segmentation to focus on the relevant image area (#3, #6 reporting an upgrade, #10 a downgrade).

YOLO-based models reach superior results compared to FRCNN models

(#9-#12), regardless of the YOLO version utilized (#1, #2, #4-#6 and #8). Another finding of the challenge is that expert models from different countries can help to improve detection performance. However, this requires a prior country classifier (#4, #10, #12).

DL solutions proposed in research still lack real-time capability, as most research focuses on performance over speed, which is not sufficient for a lightweight application in the vehicle [31].

Table I.1: GRDDC 2020 Literature and proposed DL techniques

Articles	Techniques
Hegde et al. [35]	YOLO-v5, Data & TTA, Ensembles
Doshi et al. [36]	YOLO-v5, Data & TTA, Ensembles
Pei et al. [37]	Cascade R-CNN, Data Aug., Ensembles
Mandal et al. [38]	YOLO-v5, Data Augmentation
Dongjun [39]	YOLO-v5, Data Augmentation
Liu et al. [40]	YOLO-v4 & Faster RCNN, Data Aug.
Naddaf-sh et al. [41]	EfficientDet
Xiaoguang et al. [23]	YOLO-v4, Data Augmentation
Hascoet et al. [42]	Faster R-CNN, two stage detection
Vishwakarma et al. [43]	Faster R-CNN, multi-stage
Pham et al. [44]	Detectron2 and Faster R-CNN
Kortmann et al. [45]	Faster R-CNN, regional experts

The detection of road damages is just one step of the CV task for AVs. Another critical aspect is the severity of a damage as not all damages are relevant for the vehicle dynamic and we seriously want to prevent vehicles from avoiding all non-relevant damages. [32] and [42] report a lack of severity algorithms to be applied in practice. [46] proposed a close range method to measure a potholes size and depth, not applicable while driving and for longer range. [47] also captures 2D images for 3D reconstruction, using visual and spatial characteristics of potholes. [48] whereas applies Support Vector Machine (SVM) and random forest ML approaches to assess the severity, including just 65 miles of road. All approaches have not been tested on automotive grade front-facing cameras in a driving scenario.

Despite the many strengths of the camera, images do not give feedback on how AVs are influenced by damages and what actual impact damages have on the vehicle dynamics.

I.2.4 Road Damage Detection with Vehicle Feedback

Besides predicting the presence of a damage, the influence on the vehicle needs to be measured to validate whether the damage is of importance for motion planning of AVs. For this reason, we present technical solutions of measuring road damages from on-board sensors located within the vehicle dynamics. Although the information about the damage is then only available after driving through them, this analysis provides the true impact on the vehicle. The gathered in-depth information can be shared via the Online HD Feature Map to impact motion planning of vehicles driving the same route.

[49] utilize retrofitted ASs to access the IRI, a metric to describe the roads' condition over a specified distance. [50] employs similar sensors, but takes a different approach in detecting potholes from the sequential sensor data. Other attempts involve the use of smartphones and their ASs in the driver's cabin for detecting potholes [51, 52]. Due to the strong suspension and damping of the driver's cabin and that for different types of vehicles, [53] and [54] apply the evaluations of different smartphones in a crowd sensing scenario to compensate for uncertainties of individual measurements. [55] estimate road damage severity by making use of the relation between vertical acceleration and relative vertical displacement of the vehicle.

[56] employ the VLS, pre-installed in modern vehicles as part of the automatic headlamp levelling, and an AS to consistently measure road elevation by modelling the QVM. The output is a accurate elevation profile of the road from which damage can be detected and assessed.

I.3 Research Approach

In this chapter, we first describe the underlying DSR approach, followed our iterative description of the research process (cf. Table I.2) for designing the end-to-end concept. Besides the end-to-end concept, the Section I.3.2 represents the process for revealing research gaps in RDD and RDS classification, being our second and third objective to be designed.

I.3.1 Design Science Research

This chapter describes our epistemological position on DSR as a research paradigm. Despite partly contrary views on DSR in the literature [57, 58], we support the understanding of DSR being pragmatic in delivering utility in a defined application domain rather than focusing on truth [59, 58, 60]. Actions are motivated by a purpose (relevance) and are enabled by knowledge (rigor) to achieve environmental change [58]. Pragmatism focuses on how new artifacts designed in DSR can drive outcomes, given limited resources.

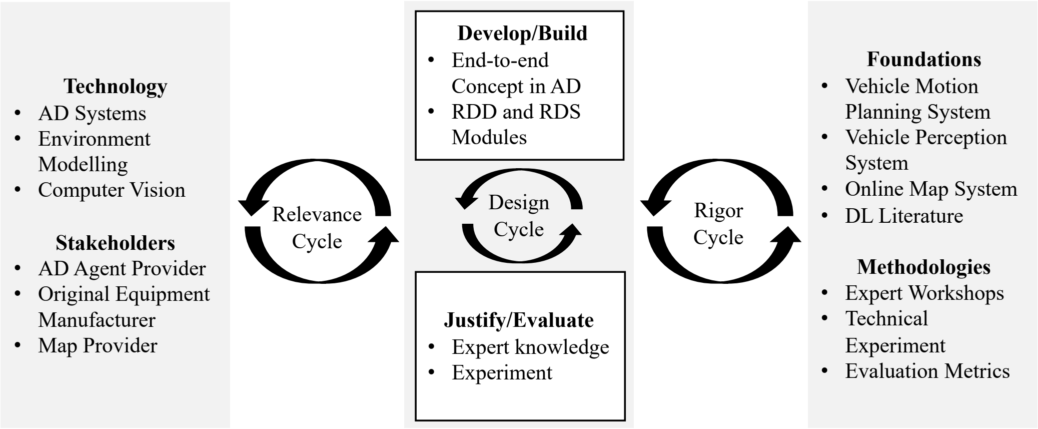


Figure I.2: Three cycles of DSR according to [61]

We utilize the DSR framework of [61] for our investigation, which is adapted to our use case in Figure I.2. It comprises three research cycles to effectively conduct DSR: relevance, rigor and design.

The relevance cycle connects the design activity with the occupied application domain, thus with the real-world problems drawn from the environment. The origin of the problems can be people, organizations or technology. The results of the research should be novel to the application domain and pragmatic and valuable in nature.

The contrasting rigor cycle deals with knowledge from research of the occupied field. In DSR, researchers simultaneously draw from the existing knowledge, as well as extending it by their designed artifacts. The rigor cycle consists of both, technical foundations as well as methodologies used in research.

The actual design takes place in the design cycle, influenced by the relevance and rigor cycle. The design cycle includes an iterative process of building artifacts of different kind to solve the problems of the environment.

I.3.2 Research Implementation

We conduct three iterations, following the DSR paradigm. Each iteration includes a relevance, a rigor, and a design cycle. Table I.2 describes the iterations' inputs, methods, steps and results respectively for all cycles.

An iteration, thus the discussion with the experts via a workshop (relevance cycle), the investigation of the knowledge base (rigor cycle) and the design phase (design cycle), spans four weeks each.

Table I.2: Overview regarding the implementation of DSR including three iterations over three cycles

	Design science research implementation		
	Relevance Cycle	Rigor Cycle	Design Cycle
		Iteration 1-3	
Input	• Application scenario • Expert demands, knowledge and feedback	• Literature on AD • Literature on RDD: • Analytical solutions • GRDDC solutions	• Literature • Expert experience in AD • Experience in DL
		Iterations 1-3	
Method	• Literature review • Expert workshop	• Literature review • Design science research	• Literature review • Prototyping & field study
		Iteration 1	
Step	• Conduct workshop 1: • Review AD architectures • Propose AD subsystems drawn from literature • Discuss the modules of the end-to-end solution • Discuss on the lack of lightweight RDD modules	• Gather relevant literature: • Vehicle perception system • Decision-making system • RDD techniques • DL state-of-the-art in CV • Analyze literature • Determine design cycle inputs for end-to-end solution and DL	• Designing the "as it is" architecture • Examine required tools for the end-to-end concept • Fit tools in the AD architecture and explain impact on motion planning • Design final end-to-end architecture
		Iteration 2	
	• Conduct workshop 2: • Propose end-to-end solution • Discuss the sub-modules • Propose our RDD module • Discuss RDS importance	• Select YOLO DL Framework • Select DL tools for training • Conclude a lack of RDS modules in research	• Analyze DL frameworks and tools for RDD • Analyze DL tools to keep real-time capability • Design final base and expert models
		Iteration 3	
	• Conduct workshop 3: • Propose our RDS module • Discuss the RDS module	• Document end-to-end solution • Document RDD and RDS • Publication writing	• Analyze DL domain for RDS • Testing RDS concepts • Design final RDS prototype
		Iteration 1	
Result	• Agreement on an AD architecture • Requirements for a cost-efficient end-to-end solution on RDD • A research gap in applicable RDD modules	• Knowledge on AD subsystems: • Vehicle Perception System • Vehicle Motion Planning • Online Map System • Knowledge on RDD techniques • Knowledge on HD Feature Maps	• Low-cost end-to-end concept on RDD and RDS classification
		Iteration 2	
	• Validation of the sub-modules of the end-to-end concept • Validation of the lightweight RDD module • A research gap in RDS	• Knowledge on DL tools • Identification of useful tools in DL to reduce runtime	• Lightweight and performant RDD algorithm
		Iteration 3	
	• Validation of the RDS module as a promising approach	• This paper	• Lightweight RDS prototype

We consult three experts in our workshops. One expert is dealing with environmental perception of autonomous vehicles in his daily business, primarily utilizing LIDAR sensors. The second expert is responsible for the appropriate use of the environment by developing functions for AD in the decision-making system, while the third expert manages projects in this very area.

As illustrated in Table I.2, we were able to develop the concept on how to utilize road damages in AD in the first two iterations leveraging the knowledge base and the domain know-how of the experts. Beforehand, we had to define an architecture of AD, consisting of three subsystems (cf. Section I.4.1) due to the high complexity of the topic.

Based on the workshops and the evaluation of existing research, we identified a research gap in the field of RDD. Existing approaches are mostly focused exclusively on detection performance in CV for RDD (cf. Section I.2). The GRDDC focused on improving road management systems, while we learned that RDD is still hardly considered in AVs. Though, the CV solutions lack applicability due to high computational demands of the DL models. When applying such models in AVs, it is of great importance to consider the limited resources in the vehicle. RDD is just an incremental part of succeeding with AD in harsh environments. The AV must perform multiple object detection and tracking tasks solely in the VPS being just one component to succeed with AD. Thus, RDD can just take an incremental part of the total computing power available. Another, yet insufficiently examined topic, is RDS classification as depicted in Section I.2.4. The research gap is spotted in workshop 2 and evaluated in workshop 3.

I.4 Results

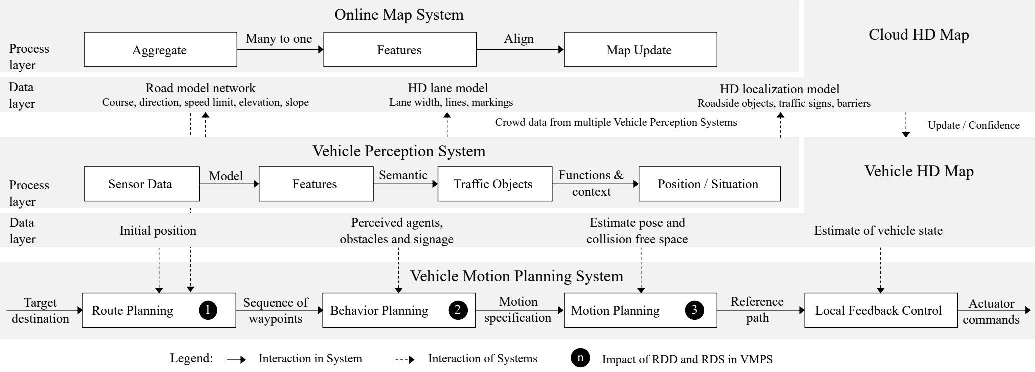


Figure I.3: The subsystems VMPS, VPS and OMS in a joint architecture of AD

I.4.1 Common Research Ground

Due to the complexity of AD, it became necessary to establish common ground with the experts and problematize the lack of architectural vision. Figure I.3 shows the AD architecture "as it is", drawn from the expert workshops/knowledge and authors perspective. We distinguish and divide the AD system into three subsystems, being the VMPS, the VPS and the OMS. The VPS and the OMS consist of a process and data layer to distinguish between the inherent operations of the system to create valuable data for the VMPS as well as defining data objects required for motion planning.

The first of two vehicle sub-systems is the VMPS, explained in Figure I.1. The second sub-system VPS located within the vehicle is responsible for gathering environmental information to create a fundamental database for motion planning [62]. The process layer depicts the procedure in the VPS on how the sensor data finally process to traffic objects with a determined position. The data layer whereas shows the deliverables drawn from the process for the below VMPS. The traffic objects are stored in the local Vehicle HD Map, the entry point for motion planning.

The third system, the OMS, is stored on an online server accessible via the vehicle's telecommunications unit. The OMS must aggregate all the information gathered from the local VPSs by performing "many to one" aggregation, with increasing amounts of redundant detections increasing the probability of existence of objects. When detecting new features, a map update must align the semantic layer with the new data. The Online HD Feature Map is thus capable of self-healing. It can be considered as an additional sensor to validate local perceptions faster. The process layer depicts the procedure from aggregation to the map update.

I.4.2 End-to-end Concept

This subsection presents our concept for factoring road damages and their severity in AD. As we cannot rely on the front-facing camera only, especially in high-speed driving situations, we adopt two divergent technical solutions for RDS and RDD and apply them in a crowd sensing scenario. The use case we deal with is visualized in Figure I.4. It shows a vehicle that detects road damages and their severity using on-board sensors. We distinguish between the camera-based approach that detect road damages ahead (C1) and the QVM-based road analysis (C2) (cf. [56]) that detect the area under the tires that has already been traversed. Both analysis are merged in C3, the Edge Aggregation, in the local HD Feature Map and updated with the OMS.

The component C1 is responsible for RDS classification and RDD. Both information can be utilized immediately in the VMPS, though, both consisting of predictions $\hat{y}$ with a specified probability for each damage class and respectively each severity class.

$$\hat{y} = \begin{pmatrix} \text{predicted type} \\ \text{predicted severity} \\ \text{predicted location} \end{pmatrix} = \begin{pmatrix} \hat{t} \\ \hat{s} \\ \hat{l} \end{pmatrix} \tag{I.1}$$

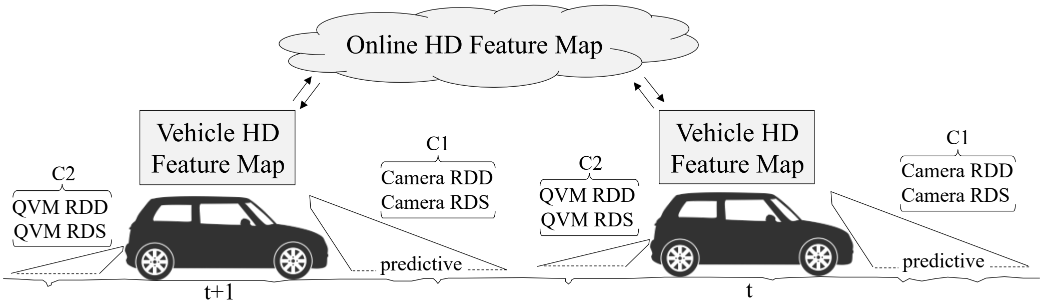


Figure I.4: Use case for camera- and QVM-based RDD and RDS classification

Most damages are detected by a single pass through with the camera system including an initial confidence value based on the detection algorithm C1. The synchronization with the OMS is of great importance in this case, as damages should have been recognized by other vehicles before. Furthermore, damages that were initially recognized only by object recognition (via C1) can be validated by passing through the damage via component C2, utilizing the QVM-based analysis. The aggregation of both analyses generated in C1 and C2 takes place in C3, Edge Aggregation. Predicted damages $\hat{y}$ that are passed through are validated utilizing C2 to become y, a much more reliable representation of the damage calculated from the elevation profile generated from the QVM response. This leads to some damages being transmitted as $\hat{y}$ and some as y to the OMS.

The OMS is responsible for "many to one" Aggregation (C4) as lots of local VPSs feed the OMS continuously with data (many $\hat{y}$ and y) to specify a single $\dot{y}$ including a fixed type, severity and location.

The end-to-end concept proposed is an information system for the AD system, similar to decision support systems for humans (e.g., [13]). The unique feature of this concept is the description of the required modules for integration into the existing system (cf. Figure I.3). The complexity is higher compared to the human-supporting systems, since the interfaces have to be defined precisely.

The design and evaluation of the concept with the experts revealed two research gaps. While the edge aggregation (C3) and the cloud aggregation (C4) can be similarly implemented as for example traffic lights or road signage, we uncover (1) a lack of lightweight RDD algorithms and (2) a lack of lightweight RDS classification algorithms. Utilizing the freedom given in DSR, we consistently and pragmatically develop algorithms to enhance the knowledge base and deliver value for the application domain. To accomplish this, we labeled supplemental data in addition to the RDDS data, as described below.

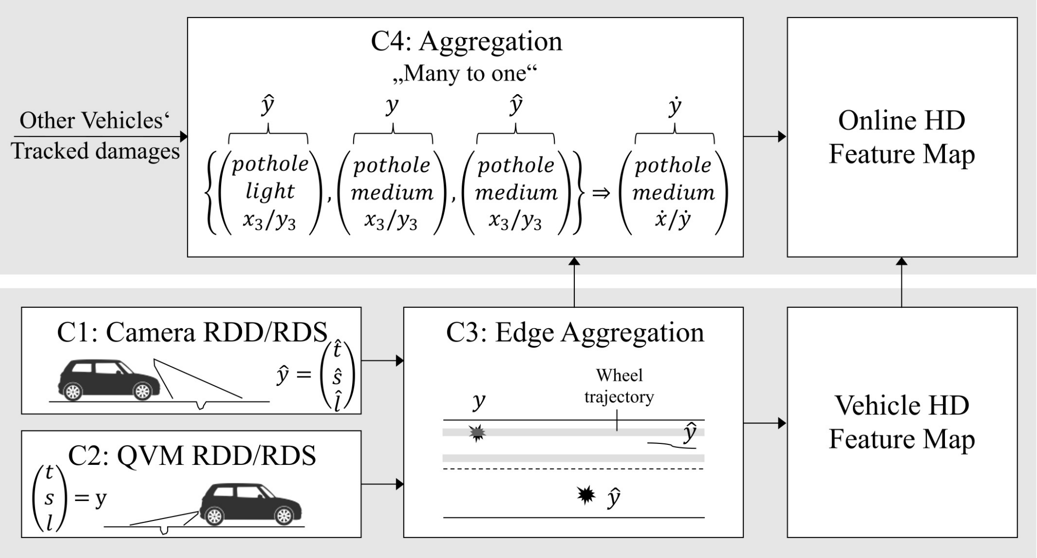


Figure I.5: End-to-end concept for the utilization of road damages in AD

I.4.3 Extended Data Set

About three-quarters of our database consists of the RDDS 2020 [34], the foundation for the GRDDC 2020 (cf. Section I.2.3). The RDDS considers four damage categories: longitudinal, horizontal and alligator cracks, and potholes.

As an addition to the RDDS, we labeled 10,260 images from automotive-grade front-facing cameras recorded in Germany (40%), the USA (28%), the United Kingdom (10%), South Korea (10%), Poland (4%), France (3%), Latvia (3%), and Finland (2%) applying the same labels. To build the ADS, we select different regions to enhance the generalization of the algorithms reflect different road conditions in various countries. The images also contain night, snow, and rain situations. The images are collected by HELLA Aglaia Mobile Vision GmbH across countries and continents, using identical measurement vehicles and an automotive-grade camera from Sony Corporation.

The addition of ADS is particularly important as the RDDS only uses images from smartphone camera sensors. The images differ greatly in terms of quality, aspect ratio and viewing angle. While the smartphone is mounted on the windshield of the inner cabin, the automotive camera is properly installed flush with the windshield. The image size of the RDDS is 600 px × 600 px, respectively 720 px × 720 px for India, whereas the ADS contains images of 3504 px × 1072 px. Incorporating the self-labeled data, we aim to tailor the algorithms optimally for application in the AVs, as good results

Table I.3: Tools for increasing RDD performance

Method	Description
Training Data	
Photometric Transforms	Random hue, saturation, brightness values
Geometric Transforms	Random perspective transform, left/right flip
Moasic Augmentation [63]	Mixing4 different images as 1 training sample
Synthetic Samples [64, 65]	New data with class-conditional VAE-GAN
Classification Loss	
Class Weighting [66]	Weighting according to inverse frequency of classes
Smooth Class Labels [67]	Turning hard one-hot encoded into soft labels
Weight Initialization	
Transfer Learning/ Fine Tuning	Using pre-trained weights/features learned on another problem as starting point
Anchors	
Optimized Anchors [68]	Generation of custom anchors with k-means
Training Strategy	
Image Weighting [69]	Weighting data sampling according to classes' inverse performance of previous epoch testing
Multi-Scale Training [68]	Varying image size randomly by +/- 50%

are more likely with using similar images in training and application.

I.4.4 Road Damage Detection

Given a frame captured by a front-facing vehicle camera, the damage detection system should decide:

1. Does that frame contain any road damages?
2. If yes, localize each damage on that image with a bounding box.
3. Determine the damage type(s) by classifying each detected damage.

The GRDDC indicated the superiority of YOLO architectures in this domain. Meanwhile, the state-of-the-art object detection framework for fast and accurate object detectors Scaled-YOLOv4 [70] was released. The authors proposed a compound network scaling approach along with five network architectures that optimally trade-off speed and accuracy. We picked the two smallest neural network structures from the Scaled-YOLOv4 framework: YOLOv4-Tiny and YOLOv4-CSP. YOLOv4-Tiny achieves real-time performance on embedded devices like the Jetson NANO, and YOLOv4-CSP is a medium-sized architecture, which is still fast, with an inference speed of 93FPS on a GPU V100. We used the Mish activation function [71] for both network designs. The three-part loss functions are composed of a bounding box regression loss (Generalized Intersection over Union (GIoU) [72] for YOLOv4-Tiny, Complete Intersection over Union (CIoU) [73] for YOLOv4-CSP), and cross-entropy losses for objectness and classification. The actual choice of model depends on the specific hardware setup and the desired performance (e.g., YOLOv4-CSP has higher accuracy but is slower than YOLOv4-Tiny). Yang et al. [74] address the challenge of having embedded hardware constraints in autonomous vehicles and conduct an industrial design study to improve throughput for CNN-based systems.

Numerous design options exist to increase the accuracy of object detection. However, many of the methods that achieve the highest accuracy in challenges (e.g., remarkable model upscaling, TTA, and ensemble models) are accompanied by significantly increased inference speed. While increasing the accuracy, such strategies typically fall short of the vehicle's processing speed requirements. Nevertheless, we can still resort to an extensive collection of best practices that can improve test accuracy without increasing the inference time in practical deployment by influencing the training process. Such methods include standard tricks like image transformations and transfer learning, but also more complex techniques like the artificial generation of additional data (cf. Table I.3).

We considered the four classes from the damage detection challenge: longitudinal cracks, transverse cracks, alligator cracks, and potholes. Our end goal is to maximize the predictive performance of damage detection with front-facing automotive cameras. We split the ADS into 70% training (7,287 images), 15% validation (1,561 images), and 15% test samples (1,561 images). In addition to the ADS, we used the challenge dataset to improve the feature learning process and to provide a comparable test score. Our approach for both the YOLOv4-CSP and YOLOv4-Tiny architectures is given as below:

1. Build a **base model** using TL from MS COCO pre-trained weights using a combined training set composed of the challenge and ADS.

2. Build an **AD expert model** by fine-tuning the base model solely using the ADS.

We determined the hyperparameters (e.g., the learning rate, amount of data augmentation, and objective function parameters) using a random search [75] followed by a genetic algorithm [76] utilizing the validation set of the first step. The ADS validation set was employed to determine the optimal number of epochs and the non-maximum suppression threshold for step two, and was added to the training set for final training. We used Stochastic Gradient Descent (SGD) with Nesterov momentum [77] as our optimizer. The YOLOv4-CSP base model was trained for 30 epochs with a learning rate of 0.016. We reduced the learning rate by a factor of 0.1 to fine-tune the expert for 80 epochs. The Tiny base model was trained for 90 epochs with a learning rate of 0.006, followed by optimizing the expert for 150 epochs with a decreased learning rate of 0.001.

RDD Performance Evaluation

Table I.4: RDD performance and runtime of the base models on the RDDS

Rank	Model	mAP	F1	Speed
#9	Tiny	42.1	54.4	5.5ms
#4	CSP (512px)	51.1	58.4	26.0ms
#4	CSP (640px)	51.0	58.8	29.9ms

Table I.4 shows the results of the base models, while Table I.6 depicts the autonomous driving expert models. We measured the inference speed on a GTX 1080 Ti GPU for a batch size of one.

The base models are evaluated on the GRDDC evaluation server results (cf. Table I.4, [33]). The YOLOv4-CSP model is the best non-ensemble solution with #4 on the leaderboard, utilizing the tools applied in Table I.3 with a runtime of 26ms. The YOLOv4-Tiny base model reaches #9 with a runtime of only 5.5ms.

High-cost base models are displayed for comparison in Table I.5. Applying TTA in a YOLOv4-CSP model enhances the F1-score to 63 (+4.6 compared to no TTA), while also the runtime increases significantly to 71.5ms (+175%). Applying ensembles as a single tool increases the runtime even more to 135.4ms while delivering poorer results than just applying TTA, which indicates, that ensembles are not at all useful when searching for a sweet-spot between detection performance and runtime.

The GRDDC's high-ranked solutions applied ensembles with large YOLO architectures such as L or XL, which exorbitantly increases the runtime and decreases the likelihood of application in AVs even further compared to the models displayed in Table I.5.

The evaluation of our AD expert models, optimized for the automotive-grade images is displayed in Table I.6. As the F1-score was utilized in the

Table I.5: High-cost base models utilizing TTA and ensembles

Rank	**Model**	**F1**	**Speed**
#2	CSP (TTA)	63.0	71.5ms
#4	CSP (512px) Ensemble	60.3	135.4ms
#2	CSP (512px) Ensemble & TTA	64.1	370.6ms

Table I.6: RDD performance and runtime of the AD expert and base models on the automotive dataset

Model	**mAP**	**Speed**
AD expert Tiny	31.1	5.3ms
Base Tiny	24.5	5.5ms
AD expert CSP (512px)	48.7	24ms
Base CSP (512px)	0.418	26.0ms
AD expert CSP (640px)	50.7	25.1ms
Base CSP (640px)	0.442	29.9ms

GRDDC, we considered this metric. Though, in object detection, the mAP has become increasingly important over the last years, as it mirrors the entirety of the precision-recall curve much better than the F1 score, which only provides information on a defined confidence score. The YOLOv4-Tiny expert model performs much poorer than the YOLOv4-CSP expert models. The YOLOv4-Tiny model suffers from a lower depth. We examined that primarily smaller images are not recognized by the algorithm. If we assume that small damages have less influence on the vehicle dynamics, it remains to be judged whether these very damages must not be taken into account at all.

The performance of the latter two algorithms differs only slightly in performance (+4%) an runtime (+4.5%). The use of a larger input size should therefore be weighed up according to the available resources.

The improvement in detection performance using our self-labeled dataset (cf. Section I.4.3) is impressive. The base models perform significantly worse on the validation dataset of the automotive data (cf. Table I.6), which justifies the enormous effort to add further data, much closer related to the application field.

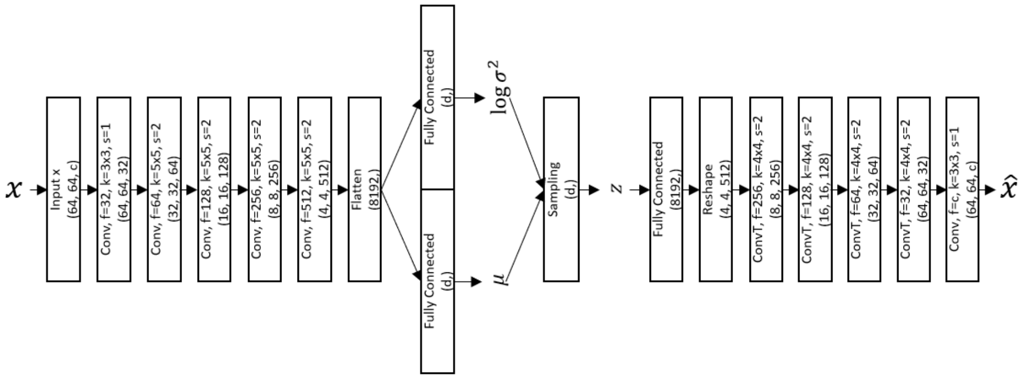

x
Input x (64, 64, c)
Conv, f=32, k=3x3, s=1 (64, 64, 32)
Conv, f=64, k=5x5, s=2 (32, 32, 64)
Conv, f=128, k=5x5, s=2 (16, 16, 128)
Conv, f=256, k=5x5, s=2 (8, 8, 256)
Conv, f=512, k=5x5, s=2 (4, 4, 512)
Flatten (8192,)
Fully Connected (d,)
Fully Connected (d,)
$\log \sigma^2$
μ
Sampling (d,)
z
Fully Connected (8192,)
Reshape (4, 4, 512)
ConvT, f=256, k=4x4, s=2 (8, 8, 256)
ConvT, f=128, k=4x4, s=2 (16, 16, 128)
ConvT, f=64, k=4x4, s=2 (32, 32, 64)
ConvT, f=32, k=4x4, s=2 (64, 64, 32)
Conv, f=c, k=3x3, s=1 (64, 64, c)
$\hat{x}$

Figure I.6: Applied Variational Autoencoder (VAE) architecture

I.4.5 Road Damage Severity Classification

Besides localization and classification (i.e., damage detection), assessing damage severity is a relevant factor to accomplish the AD task, as confirmed by the expert interviews. However, we do not possess any severity labels for the damages. Therefore, we leveraged unsupervised learning and then subjectively labeled 120 potholes from the challenge and ADS to train a classifier on top of the learned features, making this a semi-supervised approach. The reasoning for being able to train a generalizable classifier even with a small subset of supervised examples is that we also include information from the large surplus of unlabeled data in the learning process, thus already obtaining a highly nonlinear feature extractor for the damages. Similar observations (e.g., in terms of severity) will be encoded at a similar location in the feature space, facilitating the process of finding an accurate discriminator thereafter. We confine ourselves to potholes, as they are the most critical type of damage, and used VAEs [78] for feature learning. Note that the proposed approaches can be easily extended to more damage classes by additionally conditioning the VAE encoder and decoder on the damage type.

Autoencoders are unsupervised approaches to uncover latent feature representations from data. Typically, they are composed of a neural network encoder $e(x) = z$, responsible for encoding the original image data x into lower-dimensional features z and a neural network decoder $d(e(x)) = d(z) = \hat{x}$, which maps from the latent space back to the image space. The dimensionality of the z-space is typically chosen to be much smaller than the image space, forcing the model to learn a meaningful encoding. By minimizing a reconstruction error (i.e., making $\hat{x}$ close to x), we implicitly maximize the information contained in the feature representation z, as it needs to be sufficiently helpful to approximately reconstruct the original data point.

However, standard autoencoders often fail at learning continuous and interpretable latent spaces, as they rely on deterministic mappings.

VAEs are a probabilistic version of autoencoders that regularize the latent variables and thus learn a valid space of z. Instead of compressing the input image into a fixed embedding with a deterministic encoder $e(x) = z$, VAEs turn a single image into the mean and variance parameters of a diagonal Gaussian, which is forced to be close to a standard normal distribution. The resulting latent variables typically encode continuous, independent, and interpretable factors of variation in the data x, and meaningful interpolation based on the resulting z-space is feasible. For example, if x represents the image of a pothole, z should encode factors of variation in potholes, such as shape, color, surface characteristics, or depth.

Formally, the encoder represents an approximate posterior distribution $q(z|x)$ over the latent variables z, and the forward propagation of an image x is given as follows:

1. Encode x through $e(x) = (\mu, \sigma)$ to obtain $q(z|x) = \mathcal{N}(z; \mu, diag(\sigma))$.

2. Sample a latent vector z from $q(z|x)$ through $z = \mu + \sigma \circ \varepsilon$, where $\varepsilon \sim N(0, I)$ and $\circ$ is an element-wise product.

3. Decode the feature vector z through $d(z) = \hat{x}$.

The sampling step (step 2.) is carried out by applying the reparameterization trick [78] to render backpropagation through the normal distribution tractable.

Our encoder–decoder VAE structure is illustrated in Figure I.6, where c depicts the number of image channels, d is the dimensionality of the latent space, and $ConvT$ denotes a transpose convolutional layer. We used the Rectified Linear Unit (ReLU) LeakyReLU activation function and batch normalization [79] for all hidden convolutional layers and identity activation for the fully connected layers (log σ^2 ensures $\sigma^2 \geqslant 0$), and the last layer applies a sigmoid activation function, as pixels are normalized to the 0–1 range.

The objective of VAEs is the maximization of a lower bound on the intractable data likelihood given by $\mathbb{E}_z[\log p(x|z)] - KL[q_\phi(z|x)||p(z)]$, where KL denotes the Kullback–Leibler divergence and x is one image sample. Suppose we use the analytical solution of the KL divergence for two Gaussians and assume a pixel-wise Bernoulli decoder. In that case, we can equivalently minimize the following two-part loss function consisting of reconstruction and regularization loss:

$$L(x,\hat{x}) = -\sum_{i=1}^{s^2} x_i \log \hat{x}_i + (1-x_i)\log(1-\hat{x}_i) + \frac{1}{2}\sum_{j=1}^{d} \mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1 \tag{I.2}$$

where s denotes the height and width of the input image (in our case $s = 64$), d is the dimensionality of the z-space, i is the pixel index, and j denotes the element index of the mean and variance vector. The first term represents the decoding part from the latent space to the image space and aims to reconstruct the original image as closely as possible (i.e., maximizing the likelihood of the reconstruction). The second part is a regularization term to avoid ill-formed latent spaces and overfitting by making the approximate Posterior $q(z|x)$ resemble the standard normal Prior $p(z) = \mathcal{N}(0, I)$ (i.e., minimizing the KL divergence between the two distributions).

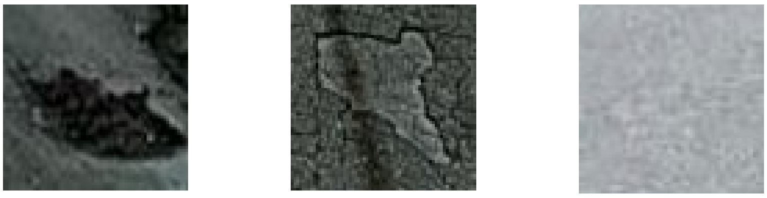


Figure I.7: Examples of high, medium and low severity potholes

We created the training set for unsupervised feature learning by cropping out the bounding boxes of the potholes and resizing them to 64 × 64 pixels, resulting in a dataset of 6,895 samples. We transformed the samples from RGB into grayscale ($c = 1$), used a two-dimensional embedding space ($d = 2$), and trained the VAE for 5,000 epochs using the RMSProp optimizer. Figure I.8 shows the resulting 2D data manifold created by linear interpolation through the latent space from 2 to -2 in steps of 0.2 and decoding the respective z-vectors. We can interpret z_1 as a color feature, with lighter potholes having values above zero. The feature z_2 can be loosely interpreted as a depth feature, with less severe potholes near $z_2 = 0$, while the corners of the space represent serious damage. We then subjectively selected 15 low severity potholes and 15 high severity potholes and visualized their encoded mean z-vector in latent space (cf. Figure I.9), which confirmed the above interpretation.

Given the promising results with only two dimensions, we wanted to build a classifier that distinguishes between three severity levels. Hence, we assigned low, medium, and high labels to 120 potholes (cf. Figure I.7).

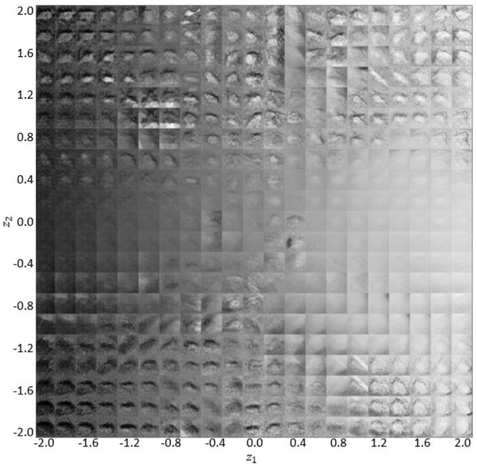


Figure I.8: Decoded potholes by interpolation through 2D latent space

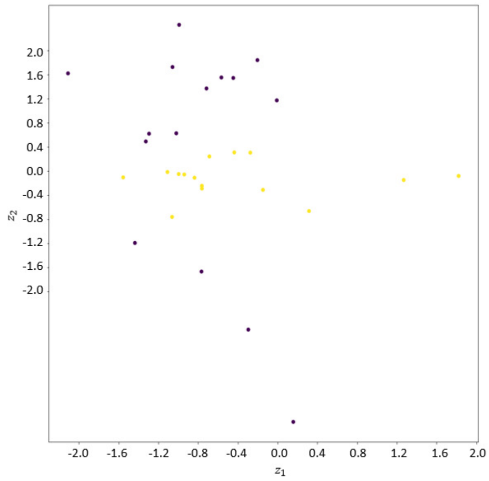


Figure I.9: Encoded severe (yellow) & non-severe (purple) potholes

Approximately half of the samples originated from the Challenge Dataset (CDS) and half from the ADS. We ensured high inter-class heterogeneity in terms of shape, color, and distance and split the 120 samples into 90 training samples (30 per class) and 30 validation samples (10 per class). This time, we trained the VAE with $c = 3$ and $d = 16$ for 5,000 epochs to learn a higher dimensional feature representation accounting for the increased complexity of the task. The features for the classifier were obtained by sampling the approximate Posterior $z^i \sim q(z^i|x^i)\forall x^i, i = 1, ..., 90$ utilizing the trained encoder. The classifier comprises two hidden fully connected layers with 16 units and ReLU activation and a softmax output layer. It was trained with the Adam optimizer [80] for 1,500 epochs using a cross-entropy loss function. Note that the approach of first training a VAE to learn a feature representation and then training a classifier on it is described as the M1 model by [80]. To evaluate this semi-supervised scheme, we also train a baseline classifier that takes the damage x instead of the features z as input.

RDS Performance Evaluation

The accuracy of the classification in three severity levels on 30 validation examples amounts 80% and just 53% with our baseline model. This indicates that our VAE learns a viable latent space facilitating the classification task.

Table I.7: Results of the severity classification

Model	**Accuracy**
Baseline (x)	16/30 (53%)
VAE (z)	24/30 (80%)

I.5 Discussion

This paper addresses the lack of AVs in dealing with road damages. The application of the DSR paradigm allows us to follow an iterative process to develop the end-to-end concept, execute gap spotting, and proceed in designing the RDS and RDD algorithms, answering the revealed research gaps.

Referring to research objective 1 (cf. Section I.1), we provide researchers and practitioners with an end-to-end concept on how to utilize built-in sensor technology in AVs to actively respond to road damages in the vehicle's decision-making-system. The concept enables the implementation of various use cases. The first consists of the real-time processing of road damages for the immediate adjustment of the trajectory. The second use case allows detected damages to be matched with the local HD Feature Map via the online HD Feature Map. The preventive information can help to consider damages that have not been detected locally, as well as to increase the confidence of the environment model in case of redundant local detection. Further, but outside the scope of our paper, the information is usable for road agencies seeking greater real-time transparency of their road conditions and to optimize action planning.

An essential step in the development of the concept is the definition of the "as it is" AD architecture in collaboration with the experts, which is necessary to create a common basis for the export workshops due to the highly complex topic of AD. Our proposed concept is a contribution to theory and practice. The topic have not been investigated in research before and offers a novel combination of modules for an effective use of available sensor technology for RDD and RDS classification in the complex AD architecture. The DSR approach is thereby very pragmatic and application-oriented, which is why the end-to-end concept also provides a valuable contribution for developers to implement such a solution and to effectively integrate the obtained features into the motion planning task of the VMPS.

The end-to-end concept proposed enables the AV to enhance the VMPS (cf. Figure I.4) in the following way:

1. Impact on High-Level Path Planning: In route planning, cost functions that prioritize different properties are applied. By default, routes can be optimized according to the shortest possible travel time and the shortest possible distance. Among other things, research is also concerned with considering the air quality in the areas to be traversed. In addition to these considerations, road segments with severe road damage can be factored in. The cost of passing the road segment increases or could decrease if damages are repaired. The importance of this feature will increase with automation level, as the interior of the vehicle can be used for leisure time as well as sleeping time. Thus, the features should help to increase driving comfort through improved route planning in areas with good road quality. To grade road segments, the feature should include the type of damage, the damage size, and its location and severity.

2. Impact on Behavior Planning: In behavior planning, the AV decides on the maneuvers to be executed based on the current traffic situation. Road damages do not yet have a cost in this segment, although it could affect maneuver planning if the road damage is severe. In extreme cases, severe damage over a wide area of the road could trigger a full stop. It is also conceivable that the lane needs to be changed, if the traffic situation permits, in order to avoid damage leading to a loss of comfort or safety. A lane-level map can also guide the vehicle to initiate proactive maneuvers and thus avoid dangerous situations due to damages.

3. Impact on Motion Planning: Road damages are not yet considered in the cost function when calculating the to be executed trajectory. Although, for small damages, it is conceivable that, within a maneuver, the trajectory could be planned in such a way that the damage is not passed through (e.g., avoid the damage to the right or left or drive over it in such a way that the tires do not pass through it). Many AD systems also include collision avoidance systems. Severe damages that are critical threats to driving safety can be considered and treated in a similarly to an object on the road that must not be hit.

Contributing to research objective 2, we deliver a mature RDD algorithm with significant improvements in runtime and cutting-edge road damage detection performance compared to the top-ranked solutions proposed at the GRDDC 2020. This is a major step towards successful in-vehicle application considering limited computation resources. The results show that it was worth the effort to label the automotive-grade camera data ADS. We were able to achieve considerable improvements from this and through the intelligent use of DL tools in training (cf. Table I.3).

The RDS classification algorithm is a promising first approach. The initial results show a significant improvement over the baseline model. An improvement in classification performance would be likely when introducing more severity-labeled damages due to a greater generalization. However, it must be further investigated whether this approach also works for other damage types.

I.5.1 Limitations

The limitations must be split in analogy to the three artifacts designed in this work.

We make a valuable contribution introducing the end-to-end concept which features road damages for AVs' motion planning (high- and low-level, cf. Section I.4.2), though the exact implementation details remain unaddressed. These include the aggregation steps on edge (C3) and the many-to-one aggregation in the OMS (C4). In addition, we do not provide a detailed description of the semantics of the HD Feature Map to account for road damages, what must be discusses regarding what level of severity damages should be considered and data granularity. Thus, it remains to be judged whether these are distributed to exact positions or, e.g., to segments to save data volume.

Furthermore, due to the great change in the electronic and electrical architecture of modern vehicles from distributed control units to centralized computers, it has to be problematized as to where the algorithms (C1 and C2) as well as the aggregation of the data (C3) run and how they are included in the local HD Feature Map.

Despite the achievements regarding an RDD algorithm that delivers great detection while being as much lightweight as possible, it remains to be evaluated which resources are actually free in the VPS of AVs to accomplish this task. This would make it possible to optimize a model exactly for the intended resources. Optimization can also be carried out with regard to the installed hardware.

Our proposed Road Damage Severity Classification algorithm is a promising approach, however, these results must be viewed with caution when compared to our RDD algorithm, as the results are based on only 30 validation images of potholes. More potholes labeled according to their severity would be desirable for a more meaningful assessment. Ideally, severity labeling should not be based on subjective criteria, but on a recognized standard for road damage assessment. However, this would require us to measure each pothole considered, which was not achievable in our research approach. Furthermore, it is of particular interest whether the same approach can be successfully applied to other damage classes (e.g., cracks).

I.6 Conclusions & Future Work

Regarding the research objectives formulated in the Introduction, we contribute to 1. in proposing an end-to-end concept that utilizes two sensors technologies for RDD and RDS classification. Moreover, we describe modules to enable AVs to actively respond to road damages while paying respect to the modern architecture of AVs (cf. "as it is" Architecture, Figure I.3). The algorithms developed represent a novel and in-depth contribution to the knowledge base. The RDS algorithms delivers cutting-edge performance while being as lightweight as possible, paying respect to limited computation resources in AVs, and the RDS algorithm being a promising approach on how to assess the severity with images from front-facing cameras.

Many different adjustments, tests and experiments were left for the future due to lack of time. This applies to the end-to-end concept as well to the RDD and RDS algorithms.

The biggest and most exciting challenge for us lies in the complete design of the **end-to-end concept**, taking into account all the requirements of a vehicle with centralized computing capabilities. The implementation would require going one level deeper in the aggregation components C3 and C4 and implementing a solution. To accomplish this, existing aggregation procedures for other features of the HD Feature Map should be applied. The RDD and RDS algorithms should be adapted based on the assigned computational capacities. The design of the road damage layer of the HD Feature Map needs to be discussed regarding information depth needed and data volume available to provide a reasonable layout.

It is also advisable to examine the potential of this approach for improving the infrastructure planning of cities and states. What is certain is that the end-to-end solution provides transparency about road conditions in real-time. This could be used to react more quickly and depending on the degree of severity. [13] proposed a decision support system, though utilizing feature-based ML approaches.

The camera-based RDD and RDS algorithms designed in this work are yet not aligned with the QVM-based calculation of the road elevation. This is a sensor fusion topic, which needs to be examined in detail to enable the predicted damages $\hat{y}$ to become validated by a vehicle feedback-based system. Furthermore, the RDS approach represents an exciting approach, which should be further investigated. However, this requires a much larger database with damages not only labeled in terms of damage type, but also in terms of severity.

Bibliography

[1] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[2] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[3] C. Collins, D. Dennehy, K. Conboy, and P. Mikalef, "Artificial intelligence in information systems research: A systematic literature review and research agenda," *Int. J. of Information Management*, vol. 60, p. 102383, 2021.

[4] I. Y. Noy, D. Shinar, and W. J. Horrey, "Automated driving: Safety blind spots," *Safety science*, vol. 102, pp. 68–78, 2018.

[5] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, p. 113816, 2020.

[6] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *On-Road Automated Driving committee*, pp. 1–35, 2018.

[7] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *Proceedings of FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[8] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of driver assistance systems.* Springer Int. Publishing Amsterdam, The Netherlands:, 2014.

[9] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous driving*, pp. 425–449, Springer, 2016.

[10] T. R. Miller and E. Zaloshnja, "On a crash course: The dangers and health costs of deficient roadways," 2009.

[11] E. Uhlemann, "Save the butterflies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 4, pp. 94–98, 2019.

[12] S. D. Gleave, R. Frisoni, F. Dionori, L. Casullo, C. Vollath, L. Devenish, F. Spano, T. Sawicki, S. Carl, R. Lidia, J. Neri, R. Silaghi, and A. Stanghellini, "Eu road surfaces: Economic and safety impact of the lack of regular road maintenance–study," *Transport and Tourism*, pp. 1–5, 2014.

[13] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, "Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection," *Int. Conf. on Information Systems*, pp. 1–16, 2018.

[14] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[15] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," *Machine vision and applications*, vol. 1, no. 4, pp. 223–240, 1988.

[16] J. Wei, J. M. Dolan, and B. Litkouhi, "A prediction-and cost function-based algorithm for robust autonomous freeway driving," in *IEEE Intelligent Vehicles Symposium*, pp. 512–517, IEEE, 2010.

[17] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium*, pp. 163–168, IEEE, 2011.

[18] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm engineering*, pp. 19–80, Springer, 2016.

[19] L. Zheng, B. Li, H. Zhang, Y. Shan, and J. Zhou, "A high-definition road-network model for self-driving vehicles," *ISPRS Int. J. of Geo-Information*, vol. 7, no. 11, p. 417, 2018.

[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The Int. J. of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[21] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The Int. J. of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[22] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *The J. of Navigation*, vol. 73, no. 2, pp. 324–341, 2020.

[23] Z. Xiao, D. Yang, T. Wen, K. Jiang, and R. Yan, "Monocular localization with vector hd map (mlvhm): A low-cost method for commercial ivs," *Sensors*, vol. 20, no. 7, p. 1870, 2020.

[24] M. Dupuis, M. Strobl, and H. Grezlikowski, "Opendrive 2010 and beyond–status and future of the de facto standard for the description of road networks," in *Driving Simulation Conf. Europe*, pp. 231–242, 2010.

[25] S. Seydi and H. Rastiveis, "A deep learning framework for roads network damage assessment using post-earthquake lidar data.," *Int. Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019.

[26] X. Chen and J. Li, "A feasibility study on use of generic mobile laser scanning system for detecting asphalt pavement cracks," *ISPRS Arch*, vol. 41, 2016.

[27] Y.-C. J. Tsai and F. Li, "Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3d laser technology," *J. of Transportation Engineering*, vol. 138, no. 5, pp. 649–656, 2012.

[28] P. Tang, D. Huber, and B. Akinci, "Characterization of laser scanners and algorithms for detecting flatness defects on concrete surfaces," *J. of Computing in Civil Engineering*, vol. 25, no. 1, pp. 31–42, 2011.

[29] M. R. De Blasiis, A. Di Benedetto, M. Fiani, and M. Garozzo, "Assessing of the road pavement roughness by means of lidar technology," *Coatings*, vol. 11, no. 1, p. 17, 2021.

[30] R. Ravi, A. Habib, and D. Bullock, "Pothole mapping and patching quantity estimates using lidar-based mobile mapping systems," *Transportation Research Record*, vol. 2674, no. 9, pp. 124–134, 2020.

[31] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020.

[32] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[33] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," in *IEEE Int. Conf. on Big Data*, pp. 5533–5539, IEEE, 2020.

[34] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv:2008.13101*, 2020.

[35] V. Hegde, D. Trivedi, A. Alfarrarjeh, A. Deepak, S. H. Kim, and C. Shahabi, "Yet another deep learning approach for road damage detection using ensemble learning," in *IEEE Int. Conf. on Big Data*, pp. 5553–5558, IEEE, 2020.

[36] K. Doshi, Y. Yilmaz, *et al.*, "Road damage detection using deep ensemble learning," in *IEEE Int. Conf. on Big Data*, pp. 5540–5544, IEEE, 2020.

[37] Z. Pei, R. Lin, *et al.*, "Cfm: A consistency filtering mechanism for road damage detection," *IEEE Int. Conf. on Big Data*, pp. 5584–5591, 2020.

[38] V. Mandal, A. R. Mussah, and Y. Adu-Gyamfi, "Deep learning frameworks for pavement distress classification: A comparative analysis," in *IEEE Int. Conf. on Big Data*, pp. 5577–5583, IEEE, 2020.

[39] J. Dongjun, "Road damage detection using yolo with smartphone images," in *IEEE Int. Conf. on Big Data*, pp. 5559–5562, IEEE, 2020.

[40] Y. Liu, X. Zhang, B. Zhang, and Z. Chen, "Deep network for road damage detection," in *IEEE Int. Conf. on Big Data*, pp. 5572–5576, IEEE, 2020.

[41] S. Naddaf-Sh, M.-M. Naddaf-Sh, A. R. Kashani, and H. Zargarzadeh, "An efficient and scalable deep learning approach for road damage detection," in *IEEE Int. Conf. on Big Data*, pp. 5602–5608, IEEE, 2020.

[42] T. Hascoet, Y. Zhang, A. Persch, R. Takashima, T. Takiguchi, and Y. Ariki, "Fasterrcnn monitoring of road damages: Competition and deployment," in *IEEE Int. Conf. on Big Data*, pp. 5545–5552, IEEE, 2020.

[43] R. Vishwakarma, R. Vennelakanti, *et al.*, "Cnn model & tuning for global road damage detection," in *IEEE Int. Conf. on Big Data*, pp. 5609–5615, IEEE, 2020.

[44] V. Pham, C. Pham, and T. Dang, "Road damage detection and classification with detectron2 and faster r-cnn," in *IEEE Int. Conf. on Big Data*, pp. 5592–5601, IEEE, 2020.

[45] F. Kortmann, K. Talits, P. Fassmeyer, A. Warnecke, N. Meier, J. Heger, P. Drews, and B. Funk, "Detecting various road damage types in global countries utilizing faster r-cnn," in *IEEE Int. Conf. on Big Data*, pp. 5563–5571, IEEE, 2020.

[46] P. L. Y. Tiong, M. Mustaffar, and M. R. Hainin, "Road surface assessment of pothole severity by close range digital photogrammetry method," *World Applied Sciences J.*, vol. 19, no. 6, pp. 867–873, 2012.

[47] G. Jog, C. Koch, M. Golparvar-Fard, and I. Brilakis, "Pothole properties measurement through visual 2d recognition and 3d reconstruction," in *Computing in Civil Engineering*, pp. 553–560, 2012.

[48] Y.-C. J. Tsai, Y. Zhao, B. Pop-Stefanov, and A. Chatterjee, "Automatically detect and classify asphalt pavement raveling severity using 3d technology and machine learning," *Int. J. of Pavement Research and Technology*, vol. 14, no. 4, pp. 487–495, 2021.

[49] M. W. Sayers, "On the calculation of international roughness index from longitudinal road profile," *Transportation Research Record*, no. 1501, 1995.

[50] H. Bello-Salau, A. Aibinu, A. Onumanyi, E. Onwuka, J. Dukiya, and H. Ohize, "New road anomaly detection and characterization algorithm for autonomous vehicles," *Applied Computing and Informatics*, 2020.

[51] H. Song, K. Baek, and Y. Byun, "Pothole detection using machine learning," *Advanced Science and Technology*, pp. 151–155, 2018.

[52] F. Seraj, B. J. Van Der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big data analytics in the social and ubiquitous context*, pp. 128–146, Springer, 2015.

[53] C.-W. Yi, Y.-T. Chuang, and C.-S. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1905–1917, 2015.

[54] G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, "SmartRoadSense: Collaborative road surface condition monitoring," in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 210–215, IARIA, 2014.

[55] P. Harikrishnan and V. P. Gopi, "Vehicle vibration signal processing for road surface monitoring," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5192–5197, 2017.

[56] F. Kortmann, J. Horstkötter, A. Warnecke, N. Meier, J. Heger, B. Funk, and P. Drews, "Modeling the quarter-vehicle: Use of passive sensor data for road condition monitoring," *IEEE Sensors J.*, 2020.

[57] A. S. Lee and J. V. Nickerson, "Theory as a case of design: lessons for design from the philosophy of science," in *Hawaii Int. Conf. on System Sciences*, pp. 1–8, IEEE, 2010.

[58] G. Goldkuhl, "Pragmatism vs interpretivism in qualitative information systems research," *European J. of information systems*, vol. 21, no. 2, pp. 135–146, 2012.

[59] P. J. Ågerfalk, "Getting pragmatic," *European J. of Information Systems*, vol. 19, no. 3, p. 251–256, 2010.

[60] J. Iivari, "Distinguishing and contrasting two strategies for design science research," *European J. of Information Systems*, vol. 24, no. 1, pp. 107–115, 2015.

[61] A. R. Hevner, "A three cycle view of design science research," *Scandinavian J. of Information Systems*, vol. 19, no. 2, p. 4, 2007.

[62] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2584–2601, 2017.

[63] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020.

[64] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2021.

[65] X. Zhang, X. Xia, N. Li, M. Lin, J. Song, and N. Ding, "Exploring the tricks for road damage detection with a one-stage detector," in *IEEE Int. Conf. on Big Data*, pp. 5616–5621, IEEE, 2020.

[66] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," in *Int. Conf. on Machine Learning*, pp. 872–881, PMLR, 2019.

[67] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[68] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.

[69] I. Loshchilov and F. Hutter, "Online batch selection for faster training of neural networks," *arXiv:1511.06343*, 2015.

[70] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 13029–13038, 2021.

[71] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv:1908.08681*, vol. 4, p. 2, 2019.

[72] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.

[73] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *AAAI Conf. on Artificial Intelligence*, vol. 34, pp. 12993–13000, 2020.

[74] M. Yang, S. Wang, J. Bakita, T. Vu, F. D. Smith, J. H. Anderson, and J.-M. Frahm, "Re-thinking cnn frameworks for time-sensitive autonomous-driving applications: Addressing an industrial challenge," in *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 305–317, IEEE, 2019.

[75] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *J. of machine learning research*, vol. 13, no. 2, 2012.

[76] G. Jocher, "Hyperparameter evolution." `https://github.com/ultralytics/yolov5/issues/607`, 2020. Accessed: 2021-05-17.

[77] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Int. Conf. on machine learning*, pp. 1139–1147, PMLR, 2013.

[78] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Int. Conf. on Learning Representations*, pp. 1–14, 2013.

[79] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, pp. 448–456, PMLR, 2015.

[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

Chapter II

Enabling Road Condition Monitoring with an on-board Vehicle Sensor Setup

Outline

Bibliographic Information

Felix Kortmann, Julin Horstkötter, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, and Paul Drews, (2019, October). “Enabling Road Condition Monitoring with an on-board Vehicle Sensor Setup”. In 2019 IEEE Sensors (pp.1-4). IEEE. DOI: 10.1109/SENSORS43011.2019.8956699. Preprint on Researchgate: 338598671.

Author’s contribution

The author’s share of the publication is 80%. Table B.2 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

This paper reports a novel approach for assessing the surface profile of roads utilizing the vehicle level sensor. The sensor is already legally binding installed in modern vehicles with headlights including LED and Xenon lighting sources in Europe due to the automatic luminaire width control. The effective application of the presented sensor setup is validated within a laboratory setup displaying and simulating the quarter-vehicle-model for known surface profiles while comparing the simulated results with measured values from the sensor setup. The results show that the measured data are in accordance to the general characteristic of the signals frequency and slope. The amplitudes deviate slightly due to inertia in the laboratory setup. This paper shows that the approach of utilizing the vehicle level sensor for road condition monitoring works in principle. It is the first step towards a real-time modelling of road conditions from common vehicles utilizing given sensors.

II.1 Introduction

Huge amounts of various sensor data are already existent within the E/E architecture (electric and electronic architecture) of vehicles. The data is transferred via communicating bus systems to steer and control most functionalities of the vehicle. At present, the human controls the velocity and steers the vehicle by his control loop using his eyes, hands and feet. When investigating automated driving, environmental influences such as current traffic situation, road conditions and the road network are regarded as control variables. A novel road condition sensor setup is presented to preventatively adjust the vehicle velocity and control the active chassis [1, 2].

Various solutions of determining road conditions are already known within the literature. Utilizing smartphone ASs inside vehicles [3, 4], retrofitting ASs between wheel and spring-damper system [5] or even camera-based road condition detection methods have been published in the past [6, 7, 8]. Instead of a cost-intensive on-board solution for road condition monitoring with additional hardware components or a cost-effective determination by mobile phones in a largely unknown position in the driver's cab, the vehicle could fall back on data from built-in automotive level sensors.

In this paper, we investigate the effectiveness of simulating the quarter-vehicle-model to characterize the surface profile below the tires. Although, the measuring vehicle cannot benefit from the computed information as it already passed through the road segment, it could benefit from a centralized platform approach where every vehicle in its common use contributes data for a road condition cloud platform.

II.2 Experimental Setup

The experimental setup subdivides into three Parts: (A) The Vehicle Level Sensor as it is part of the Sensor Setup; (B) The Quarter-Vehicle-Model as it is the substitute model for the vehicle and utilized in the simulation; (C) The Laboratory Setup to explain the functioning of the sensor setup in the constructed replica of the Quarter-Vehicle-Model.

II.2.1 Vehicle Level Sensor

The VLS is a commonly used sensor in the automotive industry to measure the tilt from the vehicle front to the rear. Its first application occurred due to the automatic luminaire width control for headlamp systems to ensure proper illumination and prevent glare for preceding and oncoming vehicles.

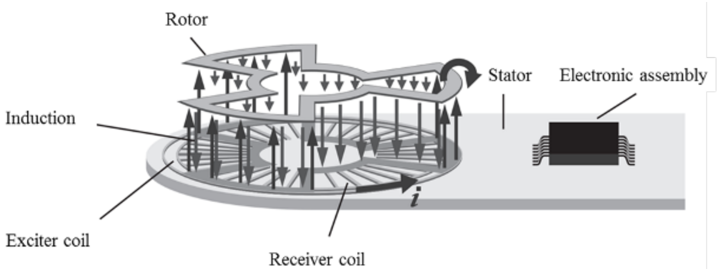


Figure II.1: Contactless inductive position sensor in the VLS

This function is mandatory in the EU for all newly registered vehicles [9]. The sensor data is furthermore used within the adaptive chassis and suspension system which enables vehicles to adjust to heterogenous road conditions with respect to safety and comfort [10, 11]. The contactless inductive position sensor inside the VLS displays in Figure II.1 The sensor measures the rotation angle and essentially consists of the moving rotor and the fixed stator. The stator implies an exciter coil operated by alternating current. The resulting electromagnetic field induces current into the rotor, which in turn generates another second electromagnetic alternating field. Both electromagnetic fields act on the three receiver coils also located on the stator and induce an alternating current dependent on the rotor position. Due to the geometry of the excitation coil, the induction in the rotor is independent of the position angle inside the case. In contrast, the induction in the receiver coils depends on the distance to the rotor and thus on the angle position of the sensor lever. The electronic assembly performs the induction to angle data provided via pulse width modulation. The rotor is connected to the actuating lever and rotates when the VLS moves. The displacement in the spring damper system can be calculated from the length and angle.

II.2.2 Quarter-Vehicle-Model

To describe the vehicle dynamics, various in its complexity differentiating substitute models ranging from a single-mass oscillator to a full-vehicle-model are considered. The quarter-vehicle-model is a modelling concept used to describe the dynamic behavior as vertical oscillation, wheel stroke and chassis stroke of one wheel including the spring-damper system and a quarter of the vehicle mass [12]. This model provides sufficient accuracy for our sensor setup. Figure II.2 shows the physical relationship in the quarter-vehicle-model on the left side and the constructed replica in component form on the right side. The yellow vehicle cage and the weights represent the sprung mass m_{sprung} (m_s) while the wheel and suspension system represent the unsprung

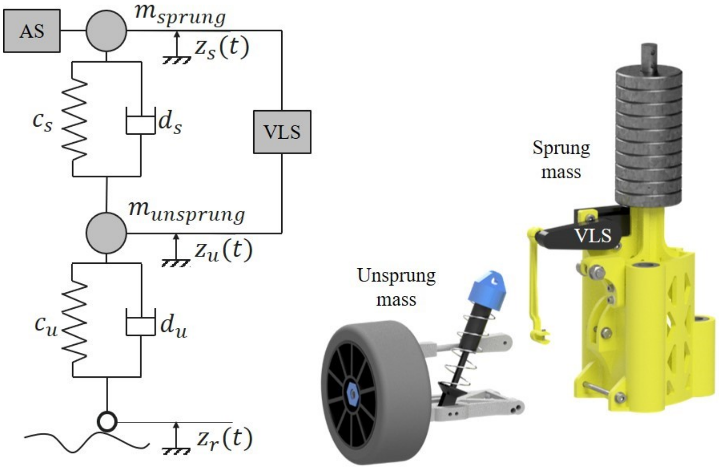


Figure II.2: Substitute model and replica of the Quarter-Vehicle-Model

m_s	592.7 g	c_s	530.5 N/m	d_s	115.46 kg/s
m_u	305.7 g	c_u	6573.8 N/m	d_u	0, negligible [13]

Table II.1: Laboratory Setup Parameters

mass $m_{unsprung}$ (m_u). The parameters are as shown in Section II.2.2.

From the VLS angle data, the displacement in the spring-damper-system between ms and mu can be calculated. Additional to the use of the VLS, an AS is used within the sensor setup to solve the differential equations following in the next chapter.

II.2.3 Laboratory Setup

The laboratory setup contains of two replicas of the quarter-vehicle-model which are fixed in width and depth and movable in height by linear ball bearings (Figure II.3). An electronic drive rotates the 3D-printed road elements on a slewing ring. The rotary motion results in a rotation of the wheels of the replicas. The circumference of the full surface profile is 1.5 m. The yellow vehicle cages, representing the chassis, have a sufficient clearance of +/-30 mm in vertical direction. Limited by the damper travel of +/-20 mm, the vehicle model can swing freely without hitting the upper end plate or the bottom of the box.

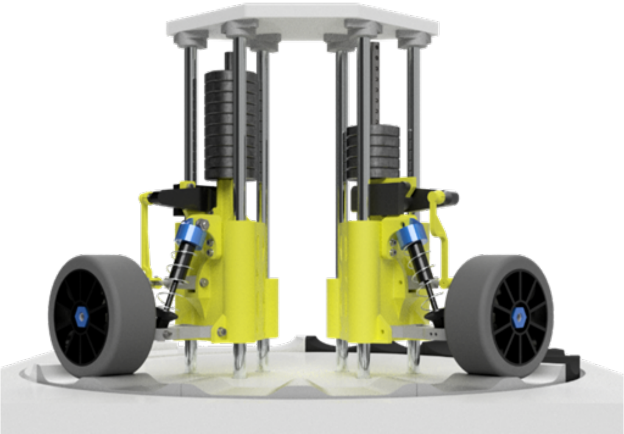


Figure II.3: Vizualization of the laboratory setup including the surface profile

II.3 Surface Profile Calculation

First and foremost, the ability of the laboratory setup to represent the quarter-vehicle-model must be demonstrated. Forces arise when springs are compressed or stretched from their rest position Equation (II.1), when dampers move Equation (II.2) and when masses accelerate Equation (II.3).

$$F_c = c \cdot z \tag{II.1}$$

$$F_d = d \cdot \dot{z} \tag{II.2}$$

$$F_a = m \cdot \ddot{z} \tag{II.3}$$

For the application of the intersection principle, the forces within the previously introduced Quarter-Vehicle-Model are listed in Figure II.4. According to Newton, the following differential equations result for the chassis mass Equation (II.4) and wheel mass Equation (II.5).

$$\begin{aligned} 0 = -m_u\ddot{z_u} + d_u(\dot{z_r} - \dot{z_u}) + c_u(z_r - z_u) \\ +d_s(\dot{z_s} - \dot{z_u}) + c_s(z_s - z_u) \end{aligned} \tag{II.4}$$

$$0 = -m_s\ddot{z_s} + d_s(\dot{z_u} - \dot{z_s}) + c_s(z_u - z_s) \tag{II.5}$$

The double derivation of the displacement z_s of the chassis m_s displays the acceleration of m_s Equation (II.6). The VLS however measures the absolute distance within the spring-damper system Equation (II.7). Consequently, the displacement of m_u calculates from the double-integrated acceleration data as subtracted by the VLS data Δs Equation (II.8).

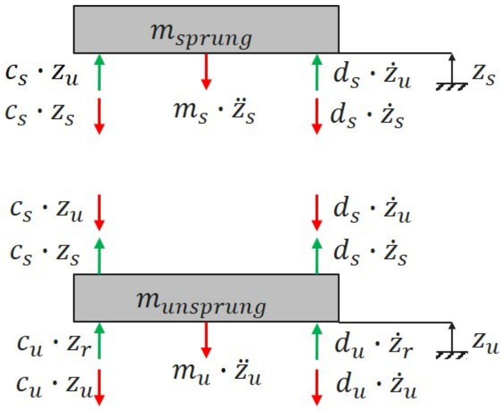


Figure II.4: Application of the intersection principle for the quarter-vehicle

$$a_s = \ddot{z_s} \tag{II.6}$$

$$\Delta s = z_s - z_u \tag{II.7}$$

$$z_u = \iint a_s - \Delta s \tag{II.8}$$

The simulation results show the excitation of the model using the known 3D-printed surface profile. The simulated values are compared to the measured values from the sensor setup.

II.4 Results

In this chapter, the displacement the wheel mass mu is simulated and compared with the measured and calculated values from the sensors of the laboratory setup. The experimentally collected vehicle parameters for the simulation are listed in Section II.2.2. Figure II.5 shows the surface profile of the 3D-printed road element extracted from the CAD-environment in red color.

The tire always hits the unevenness orthogonally in the direction of rolling. The surface profile is extracted in the center of the tire width. The blue curve displays the simulated displacement of the wheel z_u. The green curve displays z_u in the laboratory setup calculated from $\ddot{a_s}$ and Δz. As the velocity of the laboratory setup is not controlled, there are temporal deviations in x-axis. However, the obstacles within the red surface profile lead to changes in the altitude for the measured displacement of mu. Compared to the simulated results, the measured displacement in maximum amplitude match with an exception at 2 s and from 3.3 s to 3.8 s. The frequency however matches despite the temporal deviations.

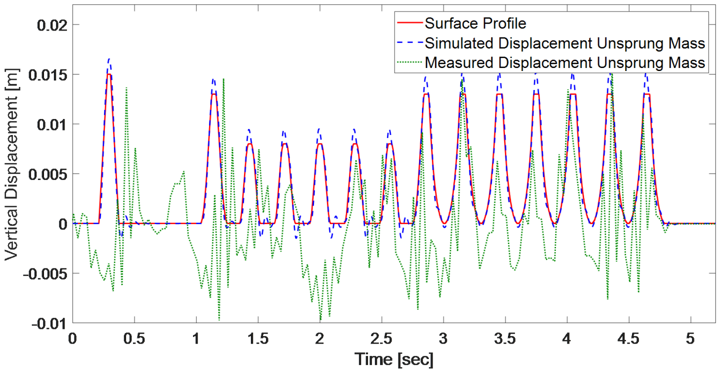


Figure II.5: Calculated displacement of the sprung mass on the road

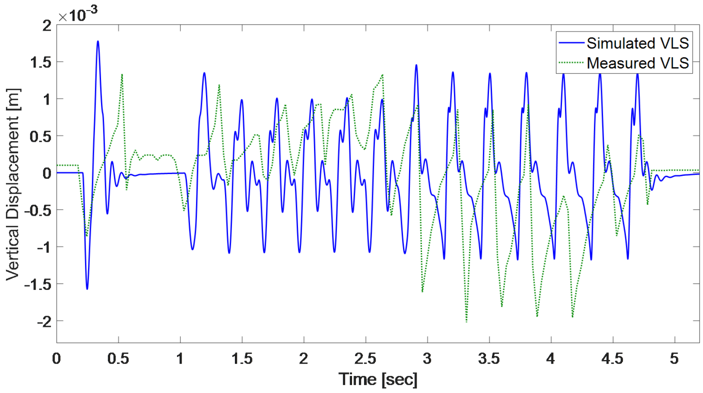


Figure II.6: Calculated displacement from sprung to unsprung mass

The displacement of the sprung mass z_s is not displayed in this paper. The spring-damper system is quite hard compared to the masses used, which leads to just a slight difference to the unsprung mass. This must be adjusted in future investigations. Furthermore, the absolute distance in the spring damper system Δz of the simulation, calculated from z_s and z_u (6), and the real measured values of the VLS are compared. The characteristic shape of the simulated curve in Figure II.6 can be found again in the measured values. The frequency and amplitudes of the curves do also match but the amplitudes from 2.8 s do turn which need to be further investigated. The measured VLS data also show a slight upswing from 1.5 s to 2.6 s, which can be reduced by filters.

Figure II.7 performs a comparison of the simulated and measured accelerations of the chassis mass ms. The simulated displacement of the chassis is derived twice and compared with the measured acceleration values from the laboratory setup.

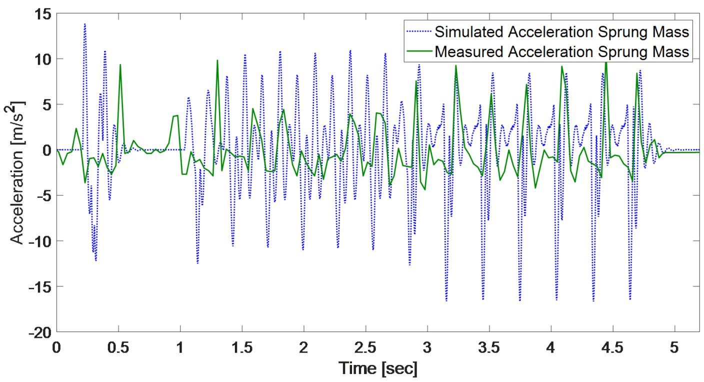


Figure II.7: Calculated acceleration of the sprung mass

The results show that the characteristic shape of the simulations mostly agree with the measured data. The amplitudes show differences which could be caused by inertia in the laboratory demonstrator and the sampling frequency. For example, at 1 s the cut curve shows that the sampling frequency is not always able to measure the maximum amplitudes. Even though, in average it can be seen, that the bigger the obstacle, the bigger the acceleration. The frequencies of the simulated and measured results do also match for Figure II.7.

Furthermore, in the later application of the simulation, the transfer function must be inverted so that an unknown surface profile can be calculated from the sensor data. Therefore, the improper differential equation must be inverted which will be executed in future works.

II.5 Conclusion and Future Works

The purpose of this paper is the validation of the described sensor setup to calculate the surface profile. In contrast to previous methods of determining the surface profile, the demonstrated solution represents a lean and automotive-grade approach without additional hardware efforts. Despite the advantages of camera-based methods to be forward-looking, our approach of enabling common vehicles to measure road quality gains importance with the rise of connected cars where the data can be joined to ensure a proper penetration rate of calculated road conditions in a platform.

Between the capability of recovering the surface profile in its height, the calculated profile can be used to determine national and international road condition indicators as the International Roughness Index [14, 15] or the Power Spectral Density [14, 16] which will be calculated in future works utilizing the presented sensor system in the laboratory setup and in a test vehicle on the road.

While the measurement area in the setup is limited to the contact surface of the tire, the capability of the sensor setup needs to be evaluated for more realistic road imitation as the 3D-printed synthetic road profile currently includes a homogenous surface profile in direction of the tire width.

By the simulation of the improper inverse differential equation, future works need to evaluate whether different vehicle parameterizations simulate the same surface profile.

Bibliography

[1] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA World Automotive Congress*, vol. 22, p. 27, Citeseer, 2006.

[2] F. O. Flemisch, K. Bengler, H. Bubb, H. Winner, and R. Bruder, "Towards cooperative guidance and control of highly automated vehicles: H-mode and conduct-by-wire," *Ergonomics*, vol. 57, no. 3, pp. 343–360, 2014.

[3] G. Alessandroni, A. Bogliolo, A. Carini, S. Delpriori, V. Freschi, L. Klopfenstein, E. Lattanzi, G. Luchetti, B. Paolini, and A. Seraghiti, "Mobile crowdsensing of road surface roughness," in *Int. Conf. on Mobile Systems, Applications, and Services*, pp. 439–439, 2015.

[4] K. Laubis, V. Simko, A. Schuller, and C. Weinhardt, "Road condition estimation based on heterogeneous extended floating car data," in *Hawaii Int. Conf. on System Sciences*, 2017.

[5] Y. Du, C. Liu, D. Wu, and S. Jiang, "Measurement of international roughness index by using-axis accelerometers and gps," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[6] P. Jonsson, "Classification of road conditions: From camera images and weather data," in *IEEE Int. Conf. on Computational Intelligence for Measurement Systems and Applications*, pp. 1–6, IEEE, 2011.

[7] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, "Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection," *Int. Conf. on Information Systems*, 2018.

[8] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. M. Kolbe, "Intelligent road maintenance: a machine learning approach for surface defect detection.," in *European Conf. on Information Systems*, p. 194, 2018.

[9] R. No, "48. uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signalling devices," 2011.

[10] H. Duda and S. Berkner, "Integrated chassis control using active suspension and braking," in *Int. Symposium on Advenced Vehicle Control*, vol. 347, p. 352, 2004.

[11] M. Münster, U. Mair, H.-J. Gilsdorf, A. Thomä, C. Müller, M. Hippe, and J. Hoffmann, "Electromechanical active body control," *ATZautotechnology*, vol. 9, no. 3, pp. 24–29, 2009.

[12] L. Eckstein, "Vertikal-und querdynamik von kraftfahrzeugen," *Schriftenreihe Automobiltechnik, Fka*, 2014.

[13] M. Mitschke and H. Wallentowitz, *Dynamik der kraftfahrzeuge*, vol. 4. Springer, 1972.

[14] M. W. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 1998.

[15] M. W. Sayers, "Two quarter-car models for defining road roughness: Iri and hri," *Transportation Research Record*, no. 1215, 1989.

[16] P. Andren, "Power spectral density approximations of longitudinal road profiles," *Int. J. of Vehicle Design*, vol. 40, no. 1-3, pp. 2–14, 2006.

Chapter III

Applying Quarter-Vehicle Model Simulation for Road Elevation Measurements Utilizing the Vehicle Level Sensor

Outline

Bibliographic Information

Felix Kortmann, Malte Rodeheger, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, and Paul Drews, (2020, November). “Applying Quarter-Vehicle Model Simulation for Road Elevation Measurements Utilizing the Vehicle Level Sensor”. In 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall) (pp.1-6). IEEE. DOI: 10.1109/VTC2020-Fall49728.2020.9348664. Preprint on ResearchGate: 348566303.

Author’s contribution

The author’s share of the publication is 80%. Table B.3 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

In the past years, automated driving has become one of the most important research fields in the automotive industry. A key component for a successful substitution of human driving by vehicles is a real-time model of the current environment including the traffic situation, the guide-way, and the road itself. Although, most of the information for the environment model are provided via in-vehicle generated data based on camera, LIDAR, and RADAR sensors, we propose a solution of classifying road quality within the spring-damper system of the vehicle. In this paper, we utilize the VLS, which is a standard component in modern vehicles, for road condition assessment. We present a simulation of the QVM for road elevation measurement to enable each connected vehicle to provide valid data for a potential crowd sensing approach where every vehicle contributes data for past and consumes data for upcoming segments. The generated data is capable of providing the environment model with real-time data of upcoming road segments. The simulation results are validated on a test bench including a review of the errors.

III.1 Introduction

The increasing automation of vehicles offers many opportunities, such as reducing traction congestion, improving comfort for vehicle occupants and increasing safety, as the vast majority (about 90 %) of accidents are caused by human error [1]. Recent technological advances in the automotive industry have contributed to a holistic environment model for CAVs to reach automation (level 3-5, SAE J3016 [2]). While LIDAR, RADAR and camera-based environmental recognition has been explored extensively and has also already been applied in practice [3, 4, 5], digital services play a subordinate role in current prototypes of automated driving [6]. With the further improvement of internet connectivity and the evolution of 5G, digital services offer great potential to provide reliable information for the automated driving task and environment model in the future.

The Vienna Conventions on Road Traffic [7] in 1968 define behavioral rules for drivers in 80 countries around the world. Article 13 formulates that the vehicle speed must be adjusted to road and weather conditions, speed limits, and the presence of other vehicles. In recent research, the required environmental model for automated driving does consider the road surface as one influencing factor for vehicle control. Based on Donges [8] chassis support system model for ADASs, Winner and Hakuli [9] (cf. Figure III.1) describe a modified version of a hierarchy model for automated vehicles. The hierarchy model, visualized in Figure III.1, consists of the vehicle and the environment. While in automated driving, the remaining task of the driver consists only in the transfer of the transport task to the vehicle, the vehicle takes over the navigation, guidance and stabilization. The road surface is part of the environment and is necessary for the actual trajectory and speed of the vehicle motion.

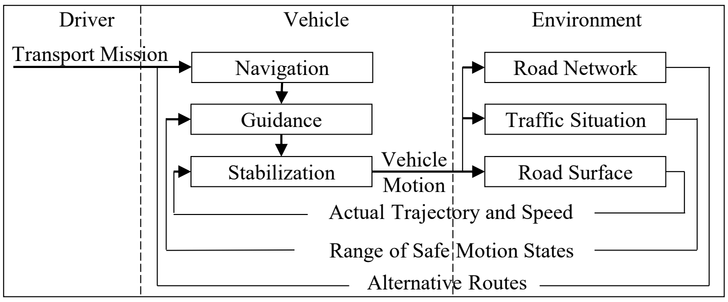


Figure III.1: Hierarchy model for automated driving according to Winner [9]

Despite the numerous possibilities of measuring road surfaces, which are going to be explained in detail in the Related Work Chapter (Section III.2), our proposed approach of measuring road roughness makes use of already build-in hardware and enables a crowd sensing solution in the future. The employed VLS measures the displacement of the vehicle chassis mass and the wheel mass. This enables the automatic luminary width control for headlamp systems to ensure proper illumination and prevent glare for preceding and oncoming vehicles. The penetration rate of the sensor on the market is very high due to a 2013 regulation of the United Nations Economic Commission in which an automated adjustment is marked as mandatory for Xenon and LED-based lighting systems with a certain intensity [10]. Between the automatic luminary width control, the sensor data is furthermore used within the adaptive chassis and suspension system which enables vehicles to adjust to heterogeneous road conditions with respect to safety and comfort [11, 12].

Figure III.2 shows a road with different road hazards. The turquoise vehicle only measures road areas it has driven on and only those areas immediately under the tires. This is visualized with transparent turquoise lines behind the vehicle. The road height profile can be properly calculated by our sensor setup. The red vehicle also moves over the road in longitudinal direction and measures the height, though for different road damages as it is driving on a different horizontal position. The yellow vehicle in turn stands for n drives through the same road segment. Due to the large number of drives, almost all uneven areas can be identified. Instead of a topology along and across the road, international and national road quality indices (cf. Section III.2) are to be calculated over route discrete segments (1^{st} Seg., 2^{nd} Seg.) and transferred to a cloud. In the cloud, it is essential to ensure that the results of all vehicles are not averaged, as vehicles tend to dodge hazards.

Using existing data for the application prevents further growth of components and additional sensors in the vehicle for the purpose of road condition assessment. Furthermore, a crowd sensing approach allows for a large amount of data to be used to implement an integrated digital service solution. Another feature is the measurement in the spring damper system as it is also related to ride quality and applicable to vehicle dynamics in contrast to vision-based methods.
Our contributions can be summarized as follows:

- We prove the applicability of the VLS for measuring road conditions.
- We present a viable test bench for road quality assessment including the VLS.
- We introduce a simulation for the successful calculation of a road height profile for different QVM configurations.

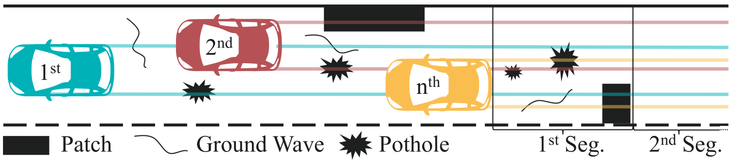


Figure III.2: Schematic overview describing the measuring surfaces of single vehicles and data merging opportunities

III.2 Fundamentals and Related Work

This chapter will first present the procedure of calculating common indicators for road roughness based on a road height profile. Secondly, we show the state of the art for individual technical implementations of determining the road height profile while dissociating from our solution.

As an important part of the environment model, it is crucial to understand how road surfaces are calculated in detail. The aim of road surface measurements is the extraction of the surface elevation profile and the classification in discrete roughness sections.

In 1982, the World Bank introduced the IRI as a standard [13]. Sayers used the IRI for measuring the longitudinal road profile [14] by accumulating deviations measured in vertical inches per mile or millimeter per meter. In his 1989 publication, Sayers considers the IRI as an accumulation of the motion between sprung and unsprung masses of a vehicle at a velocity of 80 km/h over a specific distance [15].

Another method of describing road roughness was introduced by Dodds and Robson [16] in 1973 on the basis of the Power Spectral Density (PSD) of the elevation profile. It displays the importance of various wave numbers of the elevation profile as each road profile can be composed of a set of sinusoids [14]. Regardless of the index used for classification, various solutions has already been described in the literature for measuring road elevation, which is the first step for all classification indexes.

The common road classification technique in Germany was standardized in 1991 by the Federal Ministry of Transport and Digital Infrastructure (German: Bundesministerium für Verkehr und digitale Infrastruktur - BMVI) and is based on four triangulation laser sensors. The lasers are mounted vertically pointing downward on a grid measuring beam and process the longitudinal road profile in 10 cm intervals [17, 18]. This process is very stable and highly accurate as it is carried out by special measuring vehicles. Federal roads and motorways are supposed to be measured in a four-year interval which makes it non-real-time. Furthermore, the hardware is expensive and not applicable for common vehicles.

Camera-based approaches primarily focus on defect detection (e.g crack, patch) on road surfaces. Chatterjee describes a visual decision system which processes 2-D images acquired by front-facing cameras utilizing machine learning [19, 20]. Rajab et al. [21] examined the applicability of camera-based road defect detection for cracking and potholes compared to traditional road survey methods. While camera-based solutions could potentially also be applied without extensive new hardware investments as front-facing cameras will be standard hardware for CAVs, the applications struggle with real depth information to calculate the elevation profile. Its strengths occur primarily in detecting defects as patches, cracks and potholes, which could make it a great opportunity for sensor fusion with our proposed approach.

Crowd sensing solutions make use of smartphones in vehicles to measure road roughness and defects. The research project SmartRoadSense analyzes smartphone acceleration data in the inner cabin of vehicles. Between the actual calculation of road roughness [22], the project addresses speed dependencies for sensor signals [23, 24] and geospatial aggregation techniques of crowd-sensing data [25]. Despite of the elegance of using smartphones, which comes with a great penetration rate on the roads, this approach makes use of the collaborative approach as the data of a single device is subject to uncertainty in the position and cushioning (e.g. seats, beverage storage area) of the smartphone in the inner cabin and the intermediate spring damper system of different vehicles.

In contrast to the aforementioned solutions, our proposed approach measures within the spring damper system of vehicles, utilizing the QVM. Kanjanavapastit [26] propose a simulation of the elevation profile with two ASs for just one excitation of the system by an artificial speed bump. Based on Kanjanavapastit [26], we demonstrate the practical suitability in a real test bench with a varying sensor setup including the VLS while we build on a similar simulation of the QVM.

III.3 Measurement Setup

In this paper, the results obtained from the simulation are validated in a test bench. The test bench reproduces the QVM, which reflects the dynamic behavior of a quarter of the vehicle. This model is particularly suitable for road condition evaluation, as it just absorbs the vertical forces and our sensor setup is located within these system limits, between the sprung mass and the road profile. Figure III.3 shows two quarter vehicle models with different specifications. Varying parameters are the spring and damper coefficient and the substitute mass for the vehicle body. The detailed specifications can

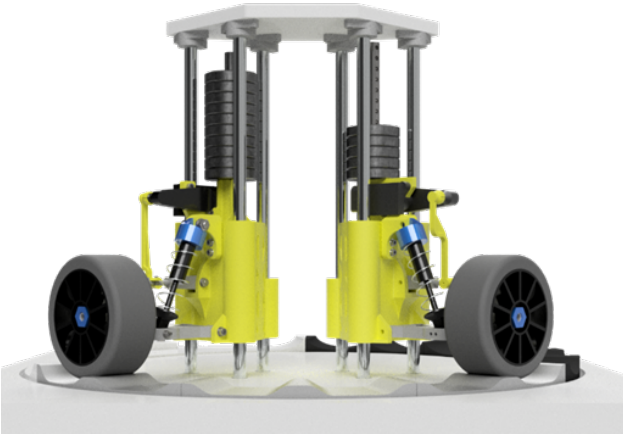


Figure III.3: Visualization of the laboratory test bench [27]

be found in Table III.1. Two models were implemented to show that similar results can be generated by modeling under different specifications. In reality, this is equivalent to different vehicles and their capability of measuring road quality.

III.3.1 Sensor and Simulation Setup

This section describes the implementation of the quarter vehicle simulation – thus, the calculation of the elevation profile of the road. The sensor system consists of a common MEMS acceleration sensor and a VLS which is a contactless inductive position sensor measuring tilt from the vehicle front to rear. It measures displacement within the spring-damper system, which will be explained in more detail in [27]. We sample both sensors every 6 ms which results in a measuring frequency of $166,\overline{6}$ Hz. Figure III.4 displays the quarter vehicle model, including the sensor setup and a visualization of it in the test bench. The masses of the vehicle are split into two parts - the sprung mass m_s including the vehicle body and everything inside and the unsprung mass m_u, also called wheel mass, including the wheel, bearing and brake. The tyre is associated with the spring coefficient c_u and the damper coefficient d_u. The actual suspension system of the vehicle is represented by the spring coefficient c_s and damper coefficient d_s. The road height z_r is the indicator that is to be measured. Finally, z_u and z_s represent the displacements of the wheel mass m_u and chassis mass m_s resulting from the suspension system and road profile.

To understand the measurement of the elevation profile utilizing VLS and an additional acceleration sensor (AS), the intersection principle of Newton must be applied for the quarter vehicle model, cf. [27]. The forces applied to the unsprung mass and sprung mass are detailed in Equation (III.1) and Equation (III.2).

$$\begin{aligned} 0 = -m_u\ddot{z}_u &+ d_u(\dot{z}_r - \dot{z}_u) + c_u(z_r - z_u) \\ &+ d_s(\dot{z}_s - \dot{z}_u) + c_s(z_s - z_u) \end{aligned} \tag{III.1}$$

$$0 = -m_s\ddot{z}_s + d_s(\dot{z}_u - \dot{z}_s) + c_s(z_u - z_s) \tag{III.2}$$

The additional acceleration sensor is mounted at the sprung mass. The displacement of the sprung mass z_s is calculated based on the double integrated acceleration data Equation (III.3). Whereas z_u must be calculated utilizing the VLS data which is named as Δz and measures the distance between unsprung mass z_u and sprung mass z_s Equation (III.4).

$$z_s = \iint \ddot{z}_s dt^2 = \iint a_s dt^2 \tag{III.3}$$

$$z_u = z_s - \Delta z = \iint a_s dt^2 - \Delta z \tag{III.4}$$

To calculate the road height z_r, Equation (III.1) and Equation (III.2) are rearranged and the displacements z_s and z_u are exchanged with terms measurable with our sensor system. Thus, z_s is represented by the double integrated acceleration sensor data $\iint a_s dt^2$ Equation (III.3) and z_u by the double integrated acceleration data minus the VLS data Δz Equation (III.4). Equation (III.5) represents the solvable differential equation by the sensor system used, converted to z_r.

$$\begin{aligned} z_r = \frac{1}{c_u}\Bigg[m_u\Big(a_s - \ddot{\Delta} z\Big) &+ m_s a_s + d_u\Big(\int a_s dt - \dot{\Delta} z\Big) \\ &+ c_u\Big(\iint a_s dt^2 - \Delta z\Big) - d_u\dot{z}_r\Bigg] \end{aligned} \tag{III.5}$$

The simulation was implemented in Matlab following the example of Kanjanavapastit [26]. Figure III.5 displays the implementation in Matlab Simulink adjusted to our sensor input representing Equation (III.5). The input values from the VLS and the AS are displayed on the left side. In the middle, the derivations and integrations for the realization of Equation (III.5) are performed. The calculation results in the road elevation.

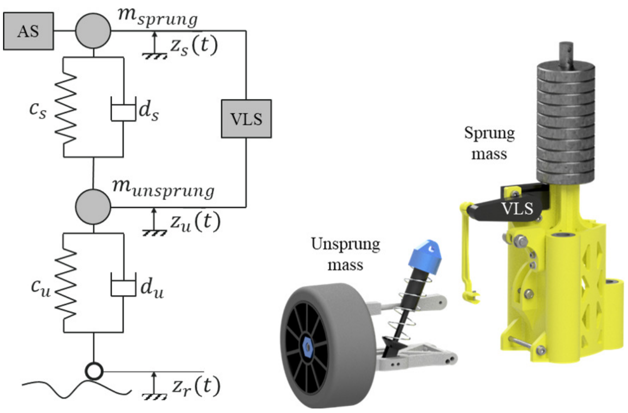


Figure III.4: QVM as a substitute model and in components [27]

III.3.2 Quarter Vehicle Configuration

The configuration of the quarter vehicle models examined are essential for the simulation. Based on the quarter vehicle configuration given to Equation (III.5), the current elevation in road profile is measured. To obtain a valid configuration for our test bench, where model building components are used in a scale of one-eighth to conventional vehicles and road anomalies, the dampers and springs are adjusted to guarantee proper parameters. In its original state, the relative proportions of the masses and the damper/spring coefficients were far from those of the GCM [14, 28].

For adjusting the damper coefficient, various damper oils with low viscosity are used to trim the damper coefficient. The damper and spring coefficient are determined experimentally. The parameter evaluation of the two model ts result in progressive spring characteristic curves that can be simplified as linear in the efficient spring range from one to two millimeters. The damper

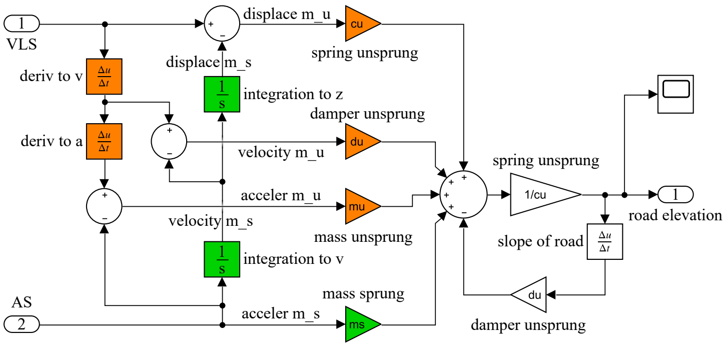


Figure III.5: Matlab Simulink model for calculating road elevation profile

Table III.1: Parameterization of the consulted QVMs in comparison with the Golden Car Model (GCM)

Param. / Model	m_s/kg	m_u/kg	$\frac{m_u}{m_s}$	$c_s/\frac{\mathrm{N}}{\mathrm{m}}$	$\frac{c_s}{m_s}/\frac{\mathrm{N}}{\mathrm{mkg}}$	$c_u/\frac{\mathrm{N}}{\mathrm{m}}$	$\frac{c_u}{m_s}/\frac{\mathrm{Nkg}}{\mathrm{m}}$	$d_s/\frac{\mathrm{kg}}{\mathrm{s}}$	$\frac{d_s}{m_s}/\frac{\mathrm{kg}^2}{\mathrm{s}}$
GCM	–	–	0.15	–	63.3	–	653	–	6
C1	1.444	0.307	0.213	530.5	367,382	2517	1743.075	46.376	32,116
C2	1.184	0.283	0.239	530.5	448,057	2265	1913,007	26.66	22.517

coefficient of the tires are assumed to be zero according to [14]. For the adjustment of the sprung mass (chassis mass), weight plates as illustrated in Figure III.3 and Figure III.4 are applied.

Table III.1 displays the configuration of both QVMs C1 and C2. The relative values in Table III.1, m_u/m_s, c_s/m_s, c_u/m_s and d_s/m_s, can be consulted to perform a comparison to the GCM. The spring coefficients of both C1 and C2 models are the only parameters that have not changed. C1 is more damped, has a higher sprung substitute mass and a tire with a lower spring coefficient compared to C2. The unsprung masses deviate only very slightly. While the relation of the unsprung mass m_u to the sprung mass m_s lies close to the reference GCM, the relative values including spring and damper coefficient to the sprung mass factor trowel differences from three to seven. The deviations are the result, despite adaptations, of damper and spring coefficients that are too high. In spite of the differences, realistic results can still be expected due to the physically solvable simulation model, which will be shown in Section III.4.

III.3.3 Road Elements

The detachable 3D-printed road elements are shown in Figure III.6. Nine different road elements and road anomalies are mounted onto the moving ring. The blue curve of Figure III.7 displays the elevation of all road elements in one diagram. From left to right in Figure III.7 and from the bottom counterclockwise in Figure III.6, a speed bump, a flat element without anomalies, a rail crossing in two elements, another flat element without anomalies, a pothole, cobblestone road, a raised round manhole cover and a lowered square manhole cover are shown in scale one-eight. Each segment is a miniaturization of existing street conditions in the city of Lippstadt in Germany. The scale is identical to the miniaturization of the tires at one-eight.

The tire-adjusted elevation profile, however, reflects the comparative value

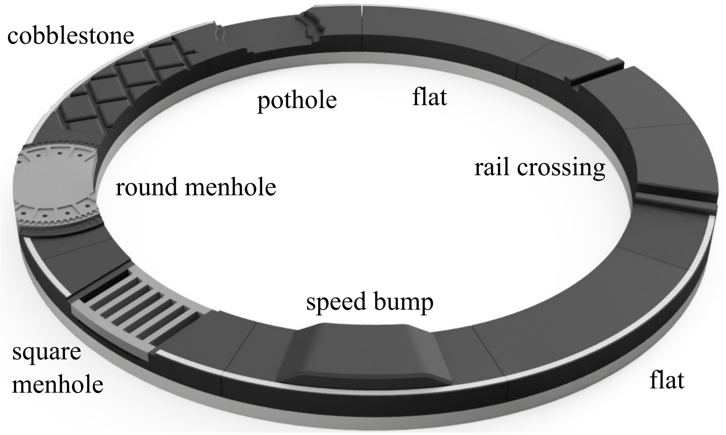


Figure III.6: The nine mounted road elements on the slewing ring

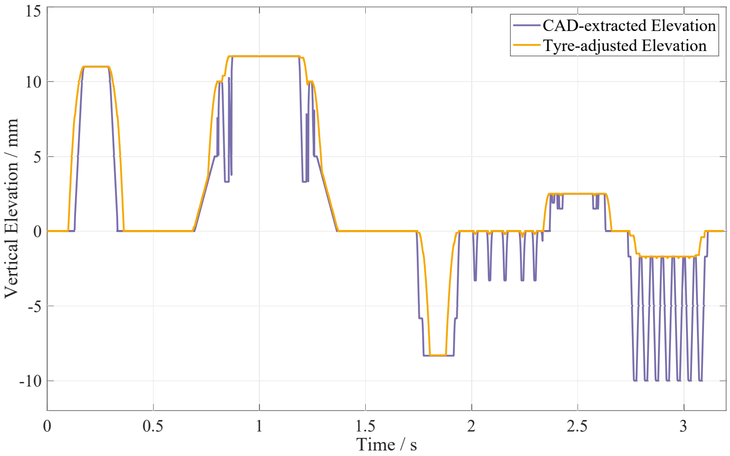
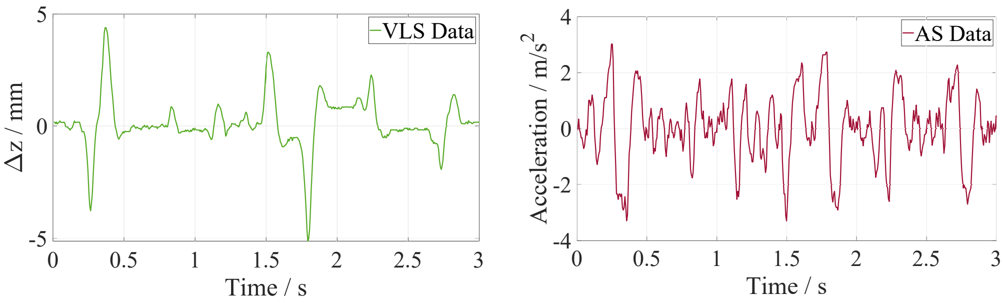


Figure III.7: Elevation of the 3D-elements compared to the tire-adjusted elevation

for our measurements, as the profile is not measured by a point excitation on z_r, but must be adjusted by the circular arc of the tire. The orange curve displays the tire-adjusted road profile.

Between the mechanical design, a PWM-based speed control of the electric motor is implemented within the laboratory test bench to regulate the desired speed at different loads on the ring resulting from the two configurable QVMs.



(a) Input VLS data

(b) Input AS data

Figure III.8: Input data from the sensor setup for the MATLAB Simulink model

III.4 Evaluation

This chapter discusses the results of the calculation from the test bench with the input values from the VLS and the AS. Section III.4 displays the input signal for the MATLAB Simulink model shown in Figure III.5. The VLS measures the absolute distance between m_s and m_u.

Section III.4, however, pictures the acceleration at m_s. The standard deviation of the AS in idle mode is 0.052 m/s^2. With the engine for rotating the road ring in operation and jacked-up QVM, the standard deviation amounts to 0.121 m/s^2.

Applying the simulation for C1, Figure III.9 displays the calculation result of the road height in petrol, while the reference profile remains in orange. The actual elevations of the anomalies are calculated well by the simulation model. The first three anomalies, the speedbump, the railway crossing and the pothole, can be recognized accurately. The manhole covers from 2.2 s to 3 s can also be represented well, but the transitions of the anomalies are comparatively worse, which may be due to the fact that there is no smooth road element between the elements. The calculated height of the anomalies differs in a range of about one to two millimeters maximum. The Root Mean Square Error (RMSE) of one turnaround amounts to 1.485 mm with a standard deviation of 1.4739 mm and a mean deviation of the error of 1.002 mm. The higher frequency share in the calculated elevation is brought about by the sensor noise of the acceleration sensor.

The calculation results of C2 are shown in Figure III.10 and are not that different to the results of C1. This is very important because we want to show that we are able to determine the height differences in the road profile despite different vehicle models. The maximum elevation do not exceed two millimeters and the sensor noise was smoothed by the slightly increased

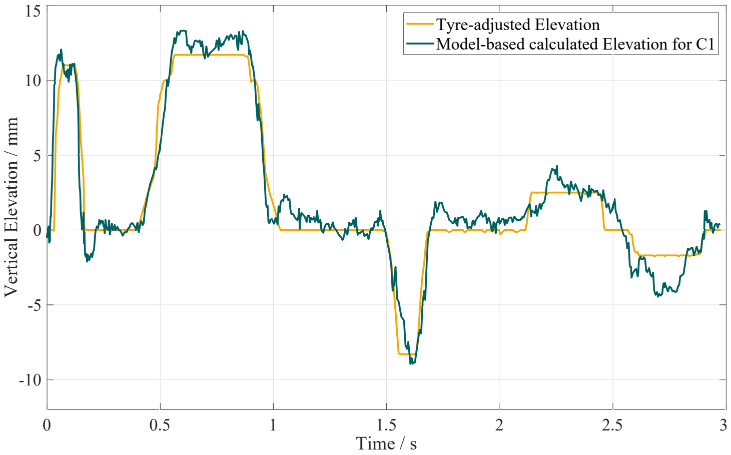


Figure III.9: The tire-adjusted elevation in comparison to the result for C1

spring coefficient of the tire. In comparison, the error analysis of C2 is slightly better, but in a comparable range with the RMSE being 1.354 mm, the standard deviation being 1.344 mm and the mean deviation of the error being 0.861 mm.

The error analysis leads to similar values for both simulation results, which supports the applicability of the modeling performed. Despite the miniaturization of the QVMs, the calculation result in a proper representation of the road conditions. As the sensor accuracy is not scaled to miniaturization, a significantly lower error percentage can be expected for real vehicle conditions, as the road anomalies are much more significantly recorded by the sensor system.

While we already introduced the tire-adjusted elevation profile to be the reference, it is still worth mentioning that the results shown in Figure III.9 (from 1.75 s) and Figure III.10 (from 2 s) do not allow to distinguish between sensor noise, cobblestone, and manhole covers due to the suspension system and tires.

Whether the manhole covers are recessed or protrude can again be verified via the elevation profile. However, new detection methods need to be examined to differentiate between various road coatings as cobblestone, concrete or asphalt and anomalies that contain smaller holes, slightly protruding elements or fine perfection.

III.5 Conclusion

In this paper, we studied the capability of the VLS and an additional AS of measuring the elevation on road surfaces. The utilized sensor system is

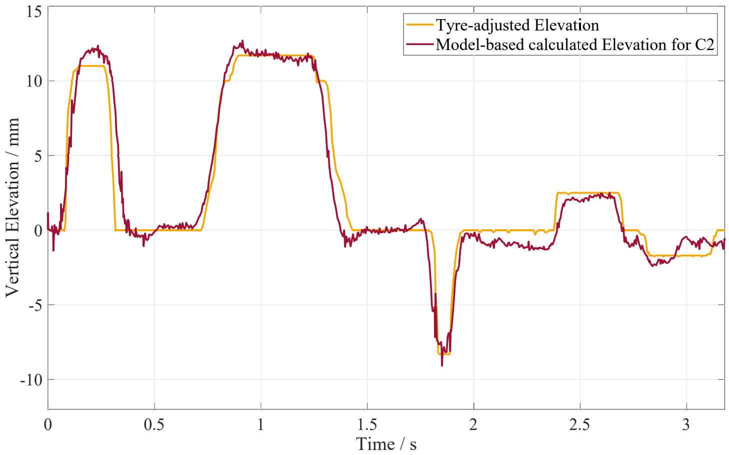


Figure III.10: The tire-adjusted elevation in comparison to the result for C2

used in a laboratory setup reflecting a quarter of a vehicle. In a true to scale setup, we showed that the elevation profile of the road can be reconstructed by a physical model in MATLAB Simulink. Based on existing car data and components, we successfully measured road elevation. The applicability in a crowd sensing scenario was achieved by comparing the two configurations C1 and C2 representing to different types of vehicles.

Besides the advantages of the application, a limitation occurs due to the excitation via the tires, road anomalies with large differences in height over a small distance cannot be optimally recorded. It is questionable, however, whether these unevennesses are highly relevant in automated driving tasks. The measurement in the spring damper system seems to be extremely useful here, as the measurement is performed directly in the relevant safety and comfort system of the vehicle and the hardware utilized is commonly used in current vehicle generations.

In future works, a test scenario with a vehicle is to be set up to prove the applicability for vehicles during conventional driving.

Furthermore, in comparison to road data from the laser-based measurement method used by road authorities in Germany, a bench-marking with the approach we propose would be interesting.

Between the actual height information, varying road coatings, the presence of moisture, water and ice also result in varying friction values. Thus, it must be investigated weather our sensor system is capable of detecting these influences.

Bibliography

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," tech. rep., 2015.

[2] S. SAE J3016, "J3016," *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, vol. 4, pp. 593–598, 2014.

[3] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE transactions on intelligent transportation systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[4] W. Shi, M. B. Alawieh, X. Li, and H. Yu, "Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey," *Integration*, vol. 59, pp. 148–156, 2017.

[5] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[6] B. für Verkehr und digitale Infrastruktur, "Bericht zum stand der umsetzung der strategie automatisiertes und vernetztes fahren," November 2017.

[7] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems*, pp. 1455–1462, IEEE, 2015.

[8] E. Donges, "Supporting drivers by chassis control systems," *Smart vehicles*, 1995.

[9] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[10] U. ECE, "Uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signaling devices," *Regulation*, no. 48, 2010.

[11] H. Duda and S. Berkner, "Integrated chassis control using active suspension and braking," in *7th Int. Symposium on Advenced Vehicle Control (AVEC)*, vol. 347, p. 352, 2004.

[12] M. Münster, U. Mair, H.-J. Gilsdorf, A. Thomä, C. Müller, M. Hippe, and J. Hoffmann, "Electromechanical active body control," *ATZautotechnology*, vol. 9, no. 3, pp. 24–29, 2009.

[13] M. W. Sayers, "Guidelines for the conduct and calibration of road roughness measurements," tech. rep., 1984.

[14] M. W. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," 1998.

[15] M. W. Sayers, "Two quarter-car models for defining road roughness: Iri and hri," *Transportation Research Record*, no. 1215, 1989.

[16] C. Dodds and J. Robson, "The description of road surface roughness," *J. of sound and vibration*, vol. 31, no. 2, pp. 175–183, 1973.

[17] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[18] G. Maerschalk, A. Ueckermann, and S. Heller, "Längsebenheitsauswerteverfahren," *Bewertetes Längsprofil": Weiterentwicklung der Längsebenheitsbewertung der Zustandserfassung und–Bewertung,"[Analysis procedure for longitudinal road roughness "Weighted Longitudinal Profile"], Berichte der Bundesanstalt für Straßenwesen, Reihe S: Straßenbau*, no. S73, 2011.

[19] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, "Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection," 2018.

[20] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. Kolbe, "Intelligent road maintenance: a machine learning approach for surface defect detection.," in *ECIS*, p. 194, 2018.

[21] M. I. Rajab, M. H. Alawi, and M. A. Saif, "Application of image processing to measure road distresses," *WSEAS Transactions on Information Science & Applications*, vol. 5, no. 1, pp. 1–7, 2008.

[22] G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, “SmartRoadSense: Collaborative road surface condition monitoring,” in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 210–215, IARIA, 2014.

[23] G. Alessandroni, A. Carini, E. Lattanzi, and A. Bogliolo, “Sensing road roughness via mobile devices: a study on speed influence,” in *Int. Symposium on Image and Signal Processing and Analysis*, pp. 272–277, IEEE, 2015.

[24] G. Alessandroni, A. Carini, E. Lattanzi, V. Freschi, and A. Bogliolo, “A study on the influence of speed on road roughness sensing: The SmartRoadSense case,” *Sensors*, vol. 17, no. 2, 2017.

[25] V. Freschi, S. Delpriori, L. C. Klopfenstein, E. Lattanzi, G. Luchetti, and A. Bogliolo, “Geospatial data aggregation and reduction in vehicular sensing applications: the case of road surface monitoring,” in *Int. Conf. on Connected Vehicles and Expo*, IEEE, 2014.

[26] A. Kanjanavapastit and A. Thitinaruemit, “Estimation of a speed hump profile using quarter car model,” *Procedia-Social and Behavioral Sciences*, vol. 88, pp. 265–273, 2013.

[27] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, “Enabling road condition monitoring with an on-board vehicle sensor setup,” in *IEEE Sensors*, IEEE, 2019.

[28] M. W. Sayers, “On the calculation of international roughness index from longitudinal road profile,” *Transportation Research Record*, no. 1501, 1995.

Chapter IV

Live Demonstration: Passive Sensor Setup for Road Condition Monitoring

Note: This publication is a brief description of our live demonstration session at IEEE Sensors 2020. It is included in the proceedings of the conference.

Outline

Bibliographic Information

Felix Kortmann, Julin Horstkötter, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, and Paul Drews, (2020, October). “Live Demonstration: Passive Sensor Setup for Road Condition Monitoring”. In 2020 IEEE SENSORS (p.1). IEEE. DOI: 10.1109/SENSORS47125.2020.9278776. Preprint on ResearchGate: 347545157.

Author’s contribution

The author’s share of the publication is 90%. Table B.4 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

IV.1 Introduction

The automation of vehicles is a major challenge and subject to many publications focusing on sensor systems, capable of receiving environmental information. As stated in the Society of Automotive Engineers (SAE) Vehicle Standard [1], automated driving level four to five requires observations of the environment to establish a continuous environment model, including road conditions. In our paper presented at IEEE Sensors 2019 [2], we propose a solution of measuring road conditions with vehicle on-board data utilizing a VLS and an AS. The VLS is a standard component for vehicles with an automatic luminary width control for headlamp systems to ensure proper illumination and prevent glare for preceding and oncoming vehicles. Along with internet connectivity, the sensor system enables every vehicle of measuring road condition to potentially provide it to a cloud platform.

IV.2 Demonstration Setup

The demonstration setup incorporates a laboratory test bench including two QVMs [3] with different specifications (Named C1 and C2) of the tyres (spring coefficient wheel c_u, damper coefficient wheel d_u) and the spring-damper system (spring coefficient chassis c_s, damper coefficient chassis d_s). The variable chassis mass is represented by aluminium weight plates while the wheel mass consists of the weight of the spring-damper system, linear bearings for vertical guidance and a 3D printed frame. The VLS is a contactless inductive position sensor and measures the displacement between the chassis and wheel masses. The dimensions of the test bench are 60 cm x 60 cm with an inner ring driven by an electric motor. 3D-printed road elements with different road surfaces and anomalies are mounted onto the ring, which are to be calculated by our sensor system. In contrast to reality, the road surface moves under the tyres allowing to isolate the force exerted by the road profile on the vertical axis to realize the application of QVM for road condition monitoring. A photography of the test bench can be seen in Figure IV.1 with a pothole starting from the left, cobblestone and two different manhole covers closing on the right.

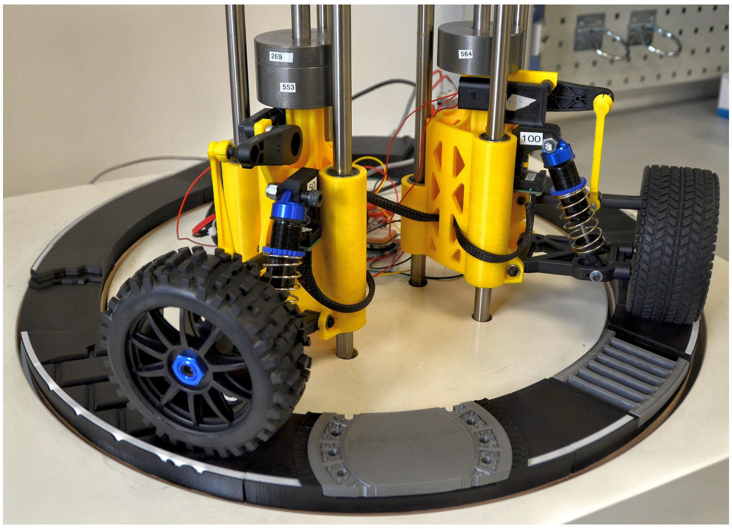


Figure IV.1: Visualization of the laboratory test bench [2]

IV.3 Visitor Experience

- The observation of a QVM, stimulated by the road profile and the road height output if the calculation.
- The application of our model-based calculation [2] and the comparison of the calculated road height with the reality (cf. Figure IV.2).
- The calculation of international road quality indices for the considered road profile.

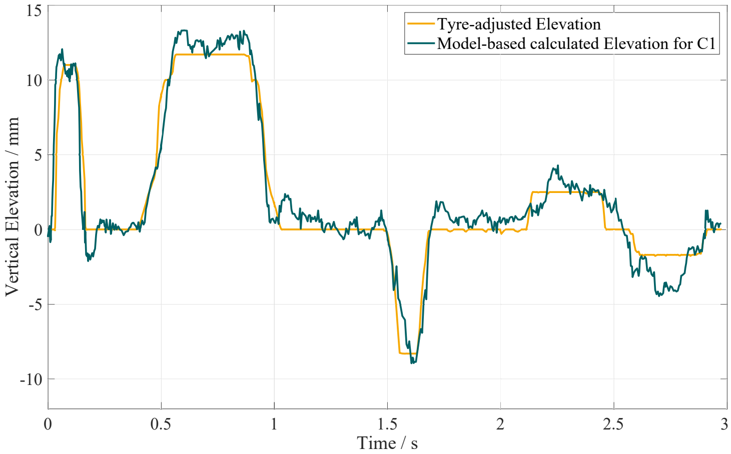


Figure IV.2: Profile of 3D-printed elements compared to our calculation

Bibliography

[1] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (sae standard j3016 201401)," *Warrendale, PA: SAE International*, 2014.

[2] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, "Enabling road condition monitoring with an on-board vehicle sensor setup," in *IEEE Sensors*, pp. 1–4, IEEE, 2019.

[3] R. N. Jazar, "Vehicle roll dynamics," in *Vehicle Dynamics: Theory and Application*, pp. 665–725, Springer, 2008.

Chapter V

Modeling the Quarter-Vehicle: Use of Passive Sensor Data for Road Condition Monitoring

Note: Publication V is an extension of Publication III.

Outline

Bibliographic Information

Felix Kortmann, Julin Horstkötter, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, and Paul Drews, (2021, July). “Modeling the Quarter-Vehicle: Use of Passive Sensor Data for Road Condition Monitoring”. in IEEE Sensors Journal, vol. 21, no. 14, (pp. 15535-15543). IEEE. DOI: 10.1109/JSEN.2020.3042620. Preprint on ResearchGate:347760870.

Author’s contribution

The author’s share of the publication is 75%. Table B.5 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

In recent years, automated driving has become one of the most important research fields in the automotive industry. A key component for a successful substitution of human driving by vehicles is a real-time model of the current environment including the traffic situation, the guide-way, and the road itself. We propose a solution for measuring road conditions within the spring-damper system of the vehicle. In this paper, we utilize a VLS and an AS, both of which are standard components in modern vehicles, for road condition monitoring. Our model-based approach therefore consists purely of additional software. We present a calculation of the QVM for road elevation measurements to enable each connected vehicle to provide valid data for a potential crowd-sensing approach, where every vehicle contributes past data and consumes data for upcoming segments. The generated data are capable of providing the environment model with real-time data. Our calculations are first validated in a laboratory setup, representing a down-scaled Quarter-Vehicle. The knowledge gained it then applied to a real vehicle. For this

purpose, the measurement setup is explained, the model-based calculation and the parameters are adjusted, and the results are compared.

V.1 Introduction

The increasing automation of vehicles offers many opportunities, such as reducing traction congestion, improving comfort for vehicle occupants, and increasing safety, as the vast majority (about 90 %) of accidents are caused by human error [1]. Recent technological advances in the automotive industry have contributed to a holistic environment model for CAVs to achieve automation (level 3-5, SAE J3016 [2]). While LIDAR, RADAR, and camera-based environmental recognition have been explored extensively and have also been applied in practice [3, 4, 5], digital services play a subordinate role in current automated driving prototypes [6]. With the further improvement of internet connectivity and the evolution of 5G, digital services offer great potential for providing reliable information for automated driving and the environment model in the future.

The Vienna Conventions on Road Traffic [7] in 1968 defined behavioral rules for drivers in 80 countries around the world. According to Article 13, the vehicle speed must be adjusted to road and weather conditions, speed limits, and the presence of other vehicles. Indeed, in recent research the environmental model needed for automated driving considers the road surface as one factor influencing vehicle control. Based on Donges' chassis support system model for ADASs [8], Winner and Hakuli [9] (cf. Figure V.1) describe a modified version of a hierarchy model for automated vehicles. The hierarchy model consists of the vehicle and the environment. In automated driving, the task of the driver is only related to the transfer of the transport task to the vehicle, as the vehicle takes over navigation, guidance, and stabilization. The road surface is part of the environment and is necessary for determining the actual trajectory and speed of the vehicle.

Despite the numerous possibilities for measuring road surfaces, which will be explained in detail in the related work chapter (Section V.2), our proposed approach of measuring road roughness makes use of built-in hardware and will enable a crowd-sensing solution in the future. We utilize a combination of an AS, at either the wheel or the chassis mass, and a VLS, measuring the displacement of the vehicle chassis mass and the wheel mass. The VLS is primarily used to enable automatic luminary width control for headlamp systems to ensure proper illumination and to prevent glare for preceding and oncoming vehicles. The penetration rate of the sensors on the market is very high due to a 2013 regulation of the United Nations Economic Commission, which made automated adjustment mandatory for Xenon- and LED-based lighting systems with a certain intensity [10]. In addition to automatic luminary width control, the sensor data are used within the adaptive chassis and suspension system, which enables vehicles to adjust to heterogeneous road conditions, improving safety and comfort [11, 12].

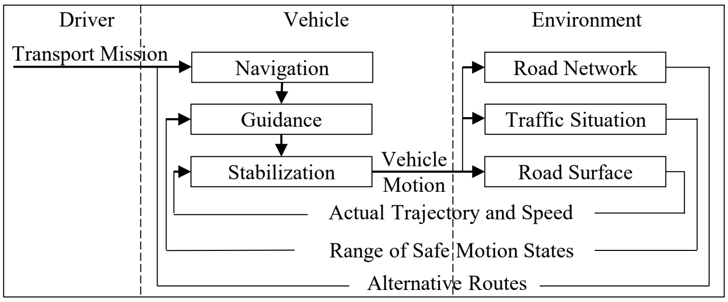


Figure V.1: Hierarchy model for automated driving according to Winner [9]

Figure V.2 shows a road with different road hazards. The turquoise vehicle only measures road areas it has driven on, and only those areas immediately under the tires. This is visualized with transparent turquoise lines behind the vehicle. The road height profile can be properly calculated by our sensor setup. The red vehicle also moves over the road in a longitudinal direction and measures the height, although for different road areas, as it is driving on a different horizontal position. The yellow vehicle in turn represents the nth drive through the same road segment. Due to the large number of drives, almost all uneven areas can be identified. Instead of measuring the topology along and across the road, international and national road quality indices (cf. Section V.2) are to be calculated over discrete route segments (1st Seg., 2nd Seg.) and transferred to a cloud. In the cloud, it is essential to ensure that the results of all vehicles are not averaged, as vehicles tend to dodge hazards.

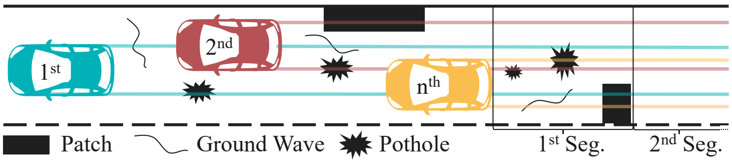


Figure V.2: Schematic overview: Measuring surface and data merging

Using existing data for the application reduces the need for additional components and sensors in the vehicle for road condition assessment. Furthermore, a crowd-sensing approach allows a large amount of data to be used to implement an integrated digital service solution. Another feature is the measurement in the spring-damper system, which is also related to ride quality and is applicable to vehicle dynamics, in contrast to vision-based methods. Our contributions can be summarized as follows:

- We present a laboratory setup for road quality assessment including the VLS and an AS.
- We prove the adaptability of the model-based calculation to measure road elevation in different QVM configurations.
- We apply the model in a real vehicle setup and validate the results.

V.2 Fundamentals and Related Work

This chapter will first present common indicators for road roughness. Second, we will show the state-of-the art individual technical implementations for determining the condition of the road.

As an important part of the environment model, it is crucial to understand how road surfaces are calculated in detail. The aim of road surface measurements is the extraction of the surface elevation profile and subsequent classification into discrete roughness sections.

In 1982, the World Bank introduced the IRI as a standard [13]. Sayers used the IRI for measuring the longitudinal road profile [14] by accumulating deviations measured in vertical inches per mile or millimeter per meter. In his 1989 publication, Sayers considered the IRI as an accumulation of the motion between sprung and unsprung masses of a vehicle at a velocity of 80 km/h over a specific distance [15].

Another method of describing road roughness was introduced by Dodds and Robson [16] in 1973 based on the PSD of the elevation profile. It emphasizes the importance of various wave numbers of the elevation profile, as each road profile can be composed of a set of sinusoids [14]. Regardless of the index used for classification, various solutions have been described in the literature for measuring road elevation, which is the first step for all classification indexes.

The common road classification technique in Germany was standardized in 1991 by the Federal Ministry of Transport and Digital Infrastructure (German: Bundesministerium für Verkehr und digitale Infrastruktur - BMVI) and is based on four triangulation laser sensors. The lasers are mounted vertically, pointing downward on a grid-measuring beam and processing the longitudinal road profile in 10-cm intervals [17, 18]. This process is very stable and highly accurate, as it is carried out by special measuring vehicles. Federal roads and motorways are supposed to be measured in four-year intervals, which results in non-real-time data. Furthermore, the hardware is expensive and not applicable for passenger vehicles.

Camera-based approaches primarily focus on defect detection (e.g, crack, patch) on road surfaces. Chatterjee is a visual decision system that processes 2-D images acquired by front-facing cameras utilizing machine learning [19, 20]. Rajab et al.[21] examined the applicability of camera-based road defect detection for cracking and potholes compared to traditional road survey methods. In 2020, the IEEE introduced the 'Global Road Damage Detection Challenge 2020', with further deep learning solutions for classifying road damages to be published by December 2020 at the IEEE BigData 2020 Conference utilizing data from Japan, India, and Czech [22]. While camera-based solutions could potentially also be applied without extensive new hardware investments since front-facing cameras will be standard hardware for CAVs, the applications struggle with real depth information to calculate the elevation profile. The strengths primarily lie in detecting defects, such as patches, cracks, and potholes, which could be useful for sensor fusion with our proposed approach.

Crowd-sensing solutions make use of smartphones in vehicles to measure road roughness and defects. The research project SmartRoadSense analyzes smartphone acceleration data in the inner cabin of vehicles. In addition to the actual calculation of road roughness [23], the project addresses speed dependencies for sensor signals [24, 25] and geospatial aggregation techniques of crowd-sensing data [26]. Despite the elegance of using smartphones, which have a large penetration rate on roads, this approach is collaborative, as data from a single device are subject to uncertainty with regard to the position and cushioning (e.g., seats, beverage storage area) of the smartphone in the inner cabin and the intermediate spring-damper system of different vehicles. Sayers described these approaches as invalid profilers that obtain a 'wiggly line' over the road profile without an established physical relationship [14].

In contrast to the aforementioned solutions, our proposed approach measures within the spring-damper system of vehicles, utilizing the QVM. Based on the results of Kanjanavapastit and Thitinaruemit [27], who proposed a solution of calculating the elevation profile with two ASs, we demonstrate its practical suitability in a laboratory setup with a different array of sensors, including the VLS and a single AS. In addition, we present real-world measurements from our Volkswagen Passat passenger vehicle for significant road anomalies.

Table V.1: Parameterization of the laboratory setup QVMs and the GCM

Param. Model	m_s/kg	m_u/kg	$\frac{m_u}{m_s}$	$c_s/\frac{\text{N}}{\text{m}}$	$\frac{c_s}{m_s}/\frac{\text{N}}{\text{mkg}}$	$c_u/\frac{\text{N}}{\text{m}}$	$\frac{c_u}{m_s}/\frac{\text{N}}{\text{mkg}}$	$d_s/\frac{\text{kg}}{\text{s}}$	$\frac{d_s}{m_s}/\frac{1}{\text{s}}$
GCM	–	–	0.15	–	63.3	–	653	–	6
C1	1.444	0.307	0.213	530.5	367.382	2517	1743.075	46.376	32,116
C2	1.184	0.283	0.239	530.5	448.057	2265	1913.007	26.66	22.517
VW [28]	495	45	0.091	78000	157.576	260000	525.253	2276.5	4.599

V.3 Measurement Setup

In this paper, the results obtained from a model-based calculation are validated in a laboratory setup and in a real passenger vehicle. Both the laboratory setup and the passenger car are described as QVMs, which reflects the dynamic behavior of a quarter of the vehicle. This model is particularly suitable for road condition evaluation, as it absorbs the vertical forces, and our sensor setup is located within these system limits, between the sprung mass and the road profile. Varying parameters include the spring and damper coefficient and the substitute mass for the vehicle body. The detailed specifications of the two laboratory QVMs as well as the passenger vehicle QVM can be found in Table V.1. Two models were implemented in the laboratory setup to show that similar results can be obtained by modeling under different specifications. In reality, this is equivalent to different vehicles and their varying capabilities of measuring road quality.

As a passenger vehicle, we used a Volkswagen Passat to check the results from the laboratory setup in reality. The configuration of the quarter vehicle models examined is essential for model-based calculation. The current elevation in the road profile was measured based on the quarter vehicle configuration used in our model-based calculation.

Following, the measurement systems of the laboratory setup (Figure V.3) and the passenger vehicle setup (Section V.3.2) are described. The commonalities and differences in the modeling are described, and the corresponding differential equations are established. Furthermore, the configurations of the QVMs from Table V.1 are discussed. Subsequently, the measured road elements and the respective scaled-down 3D printed elements in the laboratory setup are listed. [28]

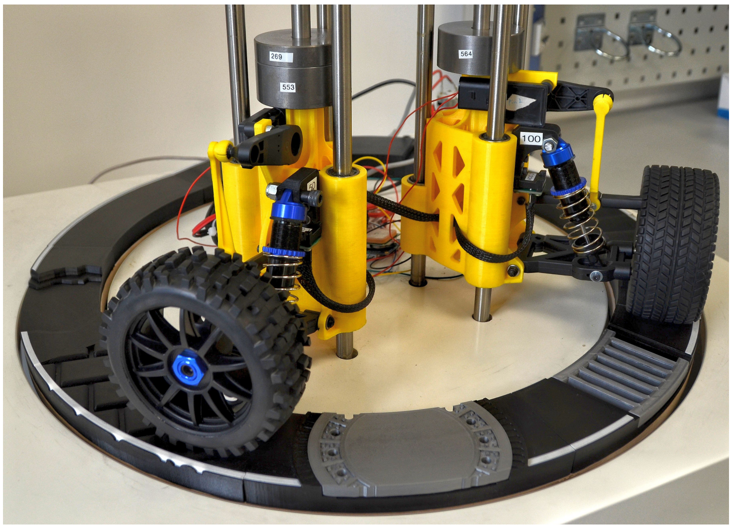


Figure V.3: Picture of the laboratory setup, including two QVMs

V.3.1 Laboratory Setup

This section describes the implementation of the Sensor Setup and Modeling; hence, the calculation of the elevation profile of the road from our sensor system. It consists of a common MEMS AS and a VLS, which is a contactless inductive position sensor measuring tilt from the vehicle front to rear. It measures displacement within the spring-damper system, which is explained in more detail in Kortmann et al. [29].

We sample both sensors every 6 ms, which results in a measuring frequency of $166,\overline{6}$ Hz. Figure V.4 displays the quarter vehicle model, including the sensor setup and a corresponding visualization of the laboratory setup. The masses of the vehicle are split into two parts – the sprung mass m_s, including the vehicle body and everything inside, and the unsprung mass m_u, also called the wheel mass, including the wheel, bearing, and brake. The tire is associated with the spring coefficient c_u and the damper coefficient d_u. The actual suspension system of the vehicle is represented by the spring coefficient c_s and damper coefficient d_s. The road height z_r is the indicator to be measured. Finally, z_u and z_s represent the displacements of the wheel mass m_u and chassis mass m_s resulting from the suspension system and road profile.

To understand the measurement of the elevation profile utilizing VLS and an additional acceleration sensor (AS), the intersection principle of Newton must be applied to the quarter vehicle model, cf. [29]. The forces applied to the sprung and unsprung mass are detailed in Equation (V.1) and Equation (V.2).

$$0 = -m_u\ddot{z_u} + d_u(\dot{z_r} - \dot{z_u}) + c_u(z_r - z_u) \ + d_s(\dot{z_s} - \dot{z_u}) + c_s(z_s - z_u) \tag{V.1}$$

$$0 = -m_s\ddot{z_s} + d_s(\dot{z_u} - \dot{z_s}) + c_s(z_u - z_s) \tag{V.2}$$

The additional acceleration sensor is mounted at the sprung mass. The displacement of the sprung mass z_s is calculated based on the twice integrated acceleration data (Equation (V.3)). Meanwhile, z_u must be calculated utilizing the VLS data, which is named Δz and measures the distance between the unsprung mass z_u and sprung mass z_s (Equation (V.4)).

$$z_s = \iint \ddot{z_s}dt^2 = \iint a_s dt^2 \tag{V.3}$$

$$z_u = z_s - \Delta z = \iint a_s dt^2 - \Delta z \tag{V.4}$$

To calculate the road height z_r, Equation (V.1) and Equation (V.2) are rearranged, and the displacements z_s and z_u are exchanged with terms that are measurable with our sensor system. Thus, z_s is represented by the double-integrated acceleration sensor data $\iint a_s dt^2$ (Equation (V.3)) and z_u by the double-integrated acceleration data minus the VLS data Δz (Equation (V.4)). Equation (V.5) is the solvable differential equation used by the sensor system, converted to z_r.

$$z_r = \frac{1}{c_u}\Big[m_s a_s + m_u(a_s - \ddot{\Delta z}) + c_u\Big(\iint a_s dt^2 - \Delta z\Big) + d_u\Big(\int a_s dt - \dot{\Delta z}\Big) - d_u\dot{z_r}\Big] \tag{V.5}$$

The model was implemented in Matlab following the research of Kanjanavapastit [27]. Figure V.5 displays the Matlab Simulink implementation adjusted to our sensor input, representing Equation (V.5). The input values from the VLS and AS are displayed on the left side. In the middle, the derivations and integrations for the realization of Equation (V.5) are shown. The calculation results in the road elevation.

To obtain valid Quarter Vehicle Configurations for our laboratory setup, where model building components are used at a scale of one-eighth of conventional vehicles and road anomalies, the dampers and springs are adjusted to guarantee proper parameters. In its original state, the relative proportions

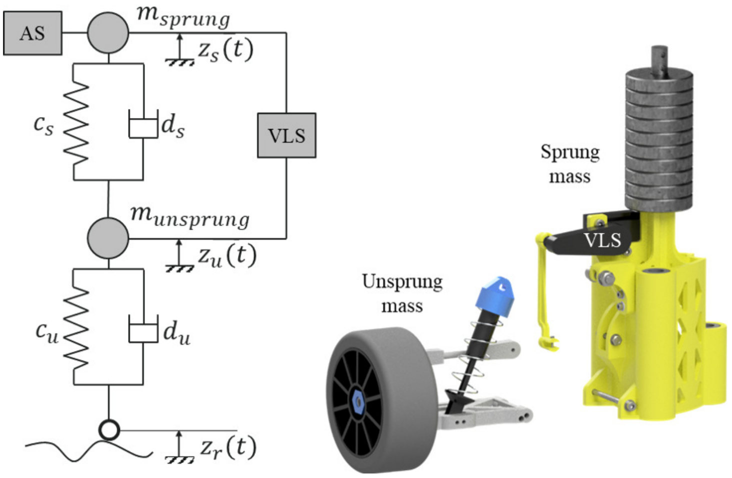


Figure V.4: QVM as a substitute model and in components [29]

of the masses and the damper/spring coefficients were far from those of the GCM [14, 30].

To adjust the damper coefficient, various damper oils with low viscosity are used to trim the damper coefficient. The damper and spring coefficients are determined experimentally. The parameter evaluation of the two models results in progressive spring characteristic curves, which can be simplified as linear in the efficient spring range of one to two centimeters. For the adjustment of the sprung mass (chassis mass), weight plates are applied, as illustrated in Figure V.3 and Figure V.4.

Table V.1 displays the configurations of both QVMs C1 and C2. The relative values in Table V.1, m_u/m_s, c_s/m_s, c_u/m_s, and d_s/m_s can be used to perform a comparison to the GCM. The spring coefficients of both the C1 and C2 models are the only parameters that have not changed. C1 is more damped, has a higher sprung substitute mass, and has a tire with a higher spring coefficient compared to C2. The unsprung masses deviate only slightly. While the relation of the unsprung mass m_u to the sprung mass m_s is close to the reference GCM, the relative values, including spring and damper coefficients to the sprung mass factor trowel, deviate from three to seven. The deviations are the result, despite adaptations, of damper and spring coefficients that are too high. In spite of these differences, realistic results can still be expected due to the physically solvable model, which will be shown in Section V.4.

The detachable 3D printed Road Elements are shown in Figure V.6. Nine different road elements and road anomalies are mounted onto the moving ring. The blue curve of Figure V.7 displays the elevation of all road elements

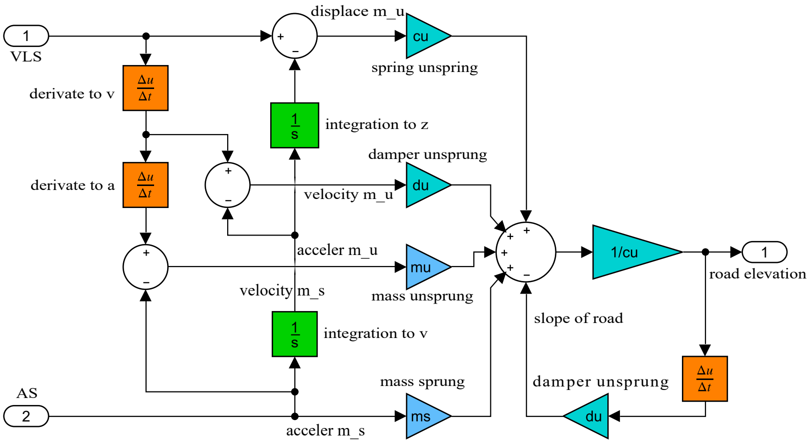


Figure V.5: MATLAB Simulink model for the laboratory setup

in one diagram. From left to right in Figure V.7 and from the bottom counterclockwise in Figure V.6, a speed-bump, a straight element without anomalies, a rail crossing in two elements, another element without anomalies, a pothole, a cobblestone road, a raised round manhole cover, and a lowered square manhole cover are shown at a one-eighth scale. Each segment is a miniaturization of existing street conditions in the city of Lippstadt in Germany. The scale is identical to the miniaturization of the tires (one-eighth).

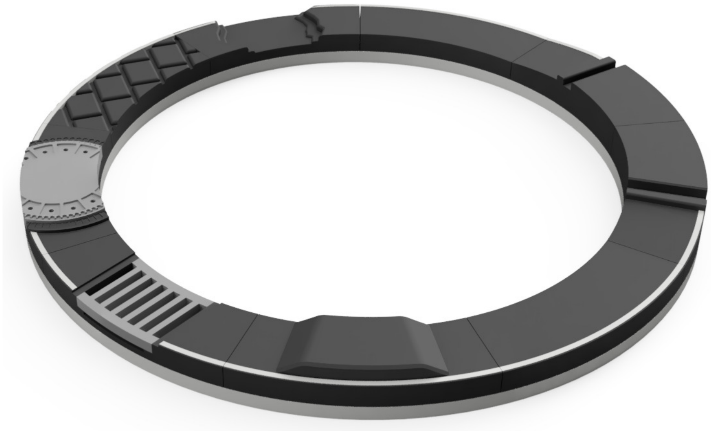


Figure V.6: The nine mounted road elements on the slewing ring

The tire-adjusted elevation profile, however, reflects the comparative value for our measurements, as the profile is not measured by a point excitation on z_r but must be adjusted by the circular arc of the tire. The orange curve displays the tire-adjusted road profile.

In addition to the mechanical design, a PWM-based speed control is used with the electric motor in the laboratory setup to regulate the desired speed at different ring loads resulting from the two configurable QVMs.

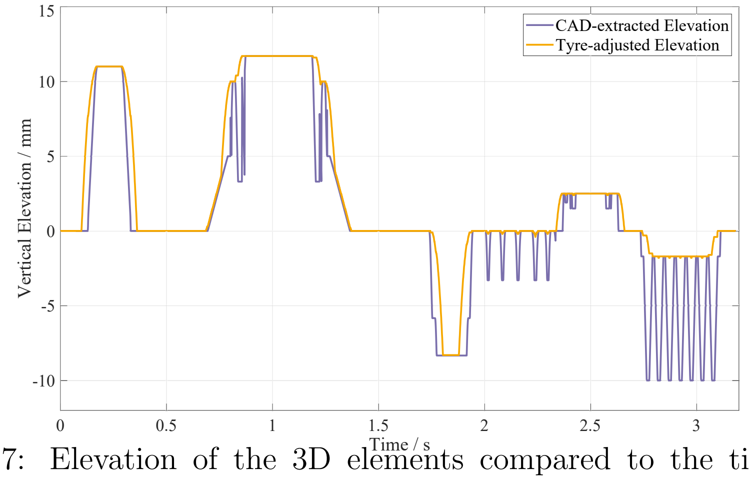


Figure V.7: Elevation of the 3D elements compared to the tire-adjusted elevation

Figure V.8 shows the measurement setup in the vehicle. The Arduino is responsible for the signal processing of the sensor data from VLS and AS and the CAN Shield on it, which reads the CAN Network of the vehicle to determine the current speed. Furthermore, the GPS sensor is shown, which takes over the localization of the vehicle, and a labeling box, which allows the marking of road sections to be viewed. The notebook is used to evaluate the data from the micro-controller and to store the raw data and measurement results.

V.3.2 Passenger Vehicle Setup

In the passenger vehicle, the Sensor Setup and Modelling needs to be slightly adjusted. The factory-fitted VLS is used, which is also accessed every 6 ms via PWM before the control unit in the vehicle. We attached the AS to the wheel mass using a pipe clamp and a 3D printed housing, which compensates for the tilt of the axle rod. By changing the positioning of the AS from the vehicle mass to wheel mass, we hope to reduce the influence of the other tires on the vehicle.

Due to the changed positioning of the AS, the displacement of the sprung mass in the vehicle ζ_s and the displacement of the unsprung mass in the vehicle ζ_u are calculated slightly differently. Equation (V.6) calculates ζ_s as the double-integrated AS signal from m_u summed with the output of the

VLS in the vehicle $\Delta\zeta$. ζ_u, however, is calculated by the double integration of the AS $\iint \alpha_u dt^2$, as shown in Equation (V.7).

$$\zeta_s = \zeta_u + \Delta\zeta = \iint \alpha_u dt^2 + \Delta\zeta \tag{V.6}$$

$$\zeta_u = \iint \ddot{\zeta}_u dt^2 = \iint \alpha_u dt^2 \tag{V.7}$$

Despite the unchanged differential equation, this results in a modified calculation of the road elevation ζ_r resulting from Equation (V.6) and Equation (V.7).

$$\zeta_r = \frac{1}{c_u}\Big[m_s(\alpha_u + \ddot{\Delta}\zeta) + m_u\alpha_u + c_u \iint \alpha_u dt^2 + d_u \int \alpha_u dt - d_u\dot{\zeta}_r\Big] \tag{V.8}$$

The Matlab Simulink model must be consistently adapted according to Equation (V.8). In this paper, just the front-left QVM in the passenger vehicle is examined.

In order to validate the measurement results of the laboratory setup, we investigate Road Elements from the real world rather than 3D printed elements. A speed bump displayed in Figure V.12 is measured by hand to obtain a reference profile for the comparison with our model-based calculation results. The height profiles of the speed bump are shown in Section V.4 and Figure V.14, together with the measurement result. We focus on a short-to-medium wavelength anomaly, as long wavelengths in the road profile are of little relevance for comfort and safety in the vehicle.

The Quarter Vehicle Configurations of our Volkswagen Passat are also displayed in Table V.1.

V.4 Evaluation

This chapter discusses the results of the calculation from the laboratory setup with the input values from the VLS and the AS in Section V.4.1. Then, the result of the vehicle in reality is discussed for a speed bump in Section V.4.2. Due to the specific noise and drift characteristics of ASs, filters based on Hofmann [32] are used in the double integration process. This includes, adapted to our application, a bandpass filter for the raw data and a bandpass filter after the first integration. Despite identical interference frequencies, the laboratory demonstrator shows factorially different measurement quantities in frequency and amplitude compared to the real measurement system in

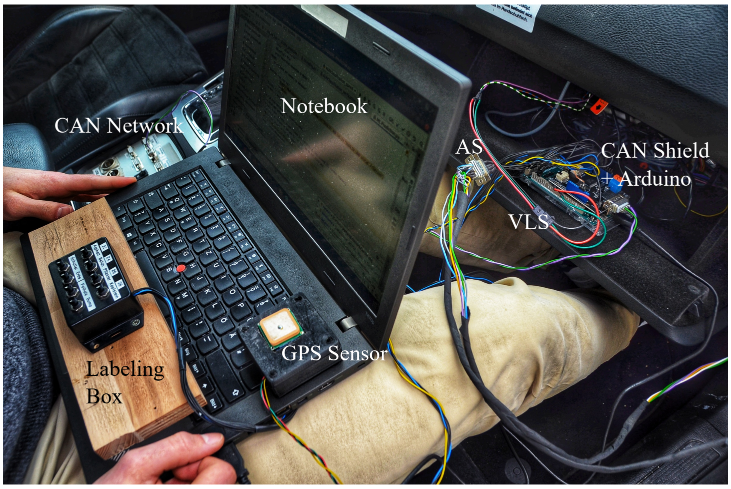


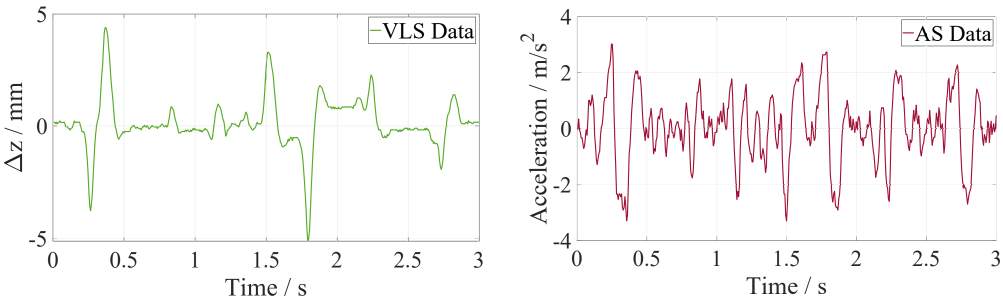
Figure V.8: System Setup in the Vehicle [31]

the vehicle, leading to differences in filter design. We discuss the results of the laboratory setup in Section V.4.1, and then we compare the calculated height of a speed bump with the reference profile to prove qualitatively that our sensor system also works in reality.

V.4.1 Laboratory Setup Results

Figure V.9a displays the input signal for the MATLAB Simulink model shown in Figure V.5. The signal is considered as absolute motion differences between m_s and m_u. Figure V.9b, however, shows the acceleration at m_s. The standard deviation of the AS in idle mode is 0.052 m/s^2. With the engine rotating and the road ring in operation and jacked-up QVM, the standard deviation amounts to 0.121 m/s^2.

Applying the model-based calculation to C1, Figure V.10 displays the calculation result of the road height in petrol color, while the reference profile remains in orange color. The actual elevations of the anomalies are calculated well by the model. The first three anomalies, the speed bump, the railway crossing. and the pothole are all recognized accurately. The manhole covers from 2.2 s to 3 s are also represented well, but the transitions of the anomalies are comparatively worse, which may be due to the fact that there is no smooth road element between the elements. The calculated height of the anomalies differs in a range of about one to two millimeters maximum. The RMSE of one turnaround amounts to 1.485 mm, with a standard deviation of 1.4739 mm and a mean deviation of the error of 1.002 mm. The higher frequency share in the calculated elevation is due to the sensor noise of the



(a) Input VLS data

(b) Input AS data

Figure V.9: Input data from the sensor setup for the MATLAB Simulink model

acceleration sensor.

The C2 calculation results are shown in Figure V.11 and and do not differ considerably from the results of C1. This is important because we want to show that we are able to determine the height differences in the road profile despite different vehicle models. The maximum elevation does not exceed two millimeters, and the sensor noise was smoothed by the slightly increased spring coefficient of the tire. In comparison, the error analysis of C2 is slightly better, but in a comparable range, with an RMSE of 1.354 mm, a standard deviation of 1.344 mm, and a mean deviation of the error of 0.861 mm.

The error analysis leads to similar values for both model-based calculation results, which supports the applicability of the modeling performed. Despite the miniaturization of the QVMs, the calculation results in a proper representation of the road conditions. Since the sensor accuracy is not scaled to miniaturization, a significantly lower error percentage can be expected in real vehicle conditions, as the road anomalies are much more significantly recorded by the sensor system.

While we already introduced the tire-adjusted elevation profile as a reference, it is still worth mentioning that the results shown in Figure V.10 (from 1.75 s) and Figure V.11 (from 2 s) do not allow us to distinguish between sensor noise, cobblestone, and manhole covers due to the suspension system and tires.

Whether the manhole covers are recessed or protruding can be determined by the height profile. However, new detection methods are needed to differentiate between various road coatings, such as cobblestone, concrete, or asphalt, and anomalies that contain smaller holes, slightly protruding elements, or fine imperfections.

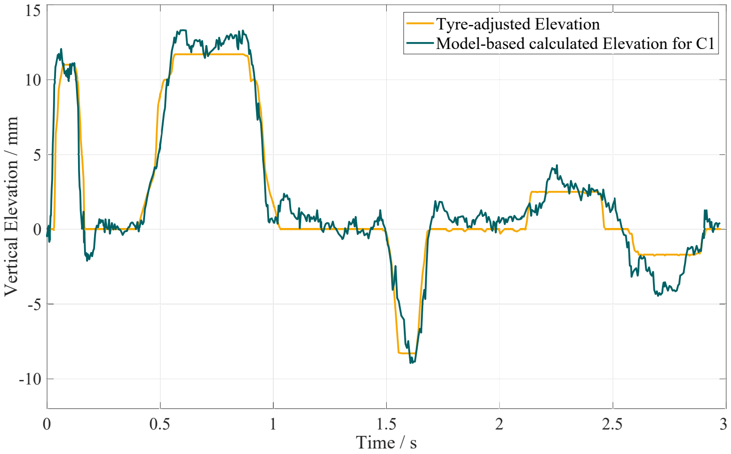


Figure V.10: The tire-adjusted elevation in comparison to the result for C1

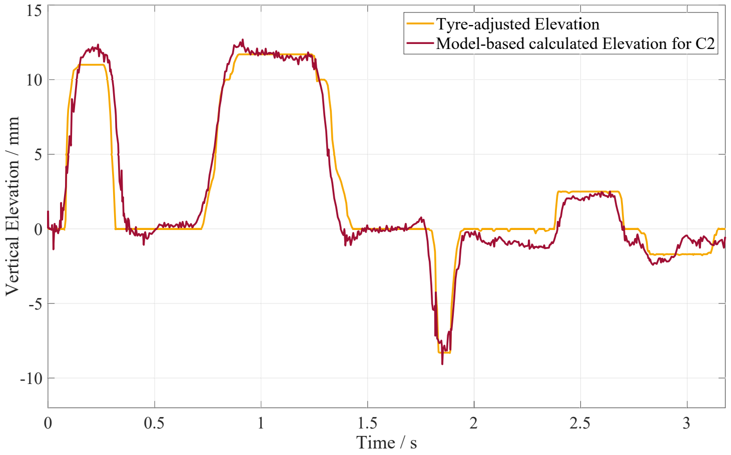


Figure V.11: The tire-adjusted elevation in comparison to the result for C2

V.4.2 Passenger Vehicle Results

To transfer the laboratory setup results to our passenger vehicle, we decided to use a manually measurable speed bump.

The elevation profile of the speed bump shown in Figure V.12 is displayed in orange color in Figure V.13. The length in the longitudinal direction is 1.7 m with a height of 6.5 cm. We passed the speed bump at 25 km/h.

Further graphs presented in Figure V.13 are the double integration of the AS (displacement of the wheel mass ζ_u) and $\Delta\zeta$, the displacement between the wheel mass ζ_u and chassis mass ζ_s, calculated from the VLS signal.

The displacement of the wheel mass ζ_u is displayed in red color. While the general height of the speed bump can be calculated effectively, the oscillating

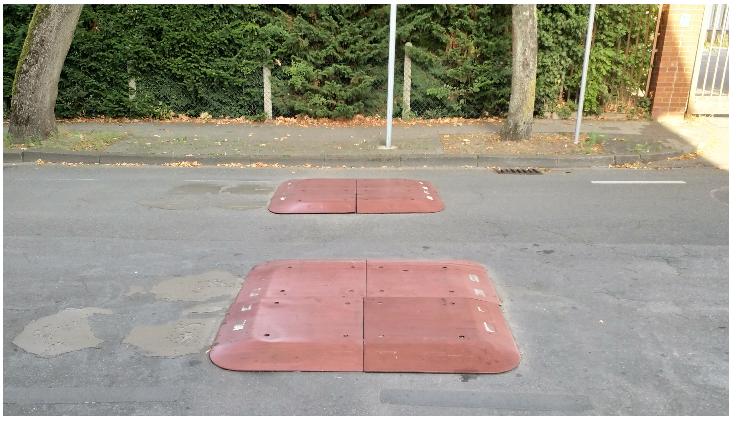


Figure V.12: Validation anomaly: speed-bump

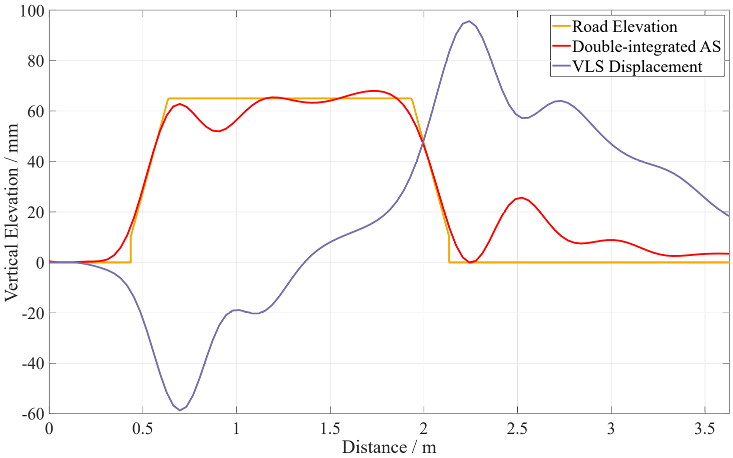


Figure V.13: Road Elevation with the double-integrated AS and VLS

influence of the chassis can be clearly seen when jumping on top of the speed bump at 0.7 m and jumping off the speed bump at 2.2 m. Hence, using double integration for road elevation measurements is not possible with a single sensor setup.

The purple graph represents $\Delta\zeta$, describing the displacement change of the two masses in relation to each other (VLS Displacement). The difference becomes apparent during the speed bump crossing. At 0.7 m, the spring-damper system is compressed -5.9 cm, bringing the wheel and chassis mass an equal amount closer to each other. At 2.2 m, the spring-damper releases the pressure to press the wheel back from the bump to the road, with an absolute length of 9.5 cm.

In Figure V.14, the displacement of m_s (Chassis Displacement) is displayed in blue color, showing a delayed reaction of the chassis mass in time compared to the wheel mass m_u (red color, Figure V.13). The height change per time is than for the wheel mass due to the effect of the spring-damper system.

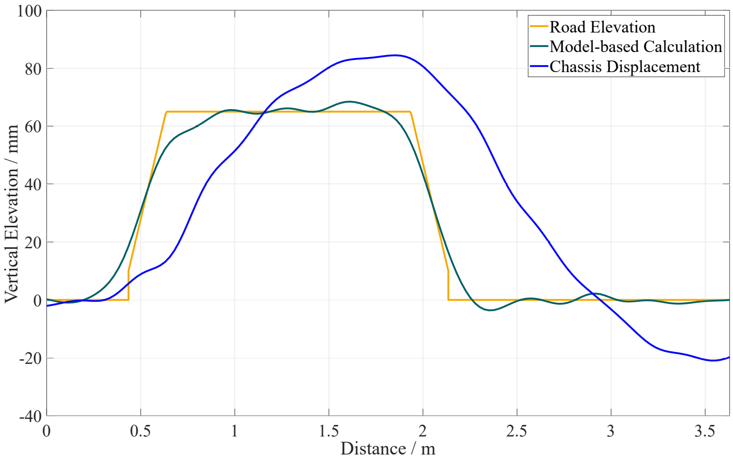


Figure V.14: Model-based calculation result and chassis displacement

The petrol-colored graph in Figure V.14 shows our model-based calculation. By calculating the terms of differential Equation (V.8), a significantly reduced oscillation behavior is obtained compared to the double integration of the acceleration, while the height ratios of the speed bump are measured correctly with 6.5 cm. However, the corners of the speed bump are less pronounced, which could be due to the plastic material of the speed bump. The calculated speed bump occurs from 0.4 m to 2.2 m, which is a total of approximately 1.9 m and quite close to the reference of 1.7 m. The earlier rise and delayed descent of the speed bump in our calculation is due to the circular excitation of the tire.

V.5 Conclusion

In this paper, we studied the capability of the VLS and an additional AS in measuring the elevation on road surfaces. The sensor system utilized was first used in a laboratory setup reflecting a quarter of a vehicle. Then, the modeling in a real vehicle on the road was tested on a significant speed bump to qualitatively verify the results of the down-scaled laboratory setup.

In a true-to-scale setup, we showed that the elevation profile of the road can be reconstructed by a physical model in Matlab Simulink. Based on existing car data and components, we successfully measured road elevation. Applicability in a crowd-sensing scenario was achieved by comparing the two configurations, C1 and C2, representing vehicles with varying properties in terms of masses and spring-damper systems. The calculations can be included in environment models to enable automated driving and other safety and comfort systems.

The measurement in the spring-damper system is useful, as it is performed directly in the relevant safety and comfort system of the vehicle, and the hardware that is utilized is commonly used in current vehicle generations.

Despite the advantages of the application, there is a limitation due to the excitation through the tires. It is questionable, however, whether unevennesses with very high frequencies is relevant to vehicle driving behavior.

The results with the real vehicle highlight the applicability of the sensor system for this task in addition to its actual functionality: headlamp leveling. However, in addition to strong anomalies on different road surfaces (e.g., concrete, asphalt, gravel, brick), investigations are needed to determine whether or not other anomalies, such as alligator cracking and manhole covers, can be detected. For this, a detailed reference profile is required, for example, from road authorities in Germany using laser-based measurements, to enable benchmarking with our proposed approach.

Our work provides a foundation for a cloud platform that makes road quality measurable without additional hardware. The development of such a platform could be implemented and investigated in the future.

In addition to the actual height information, the presence of moisture, water, and ice also results in varying friction values. Thus, it must be investigated whether our sensor system is capable of detecting these influences or should be used in sensor fusion systems.

Bibliography

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," tech. rep., 2015.

[2] S. SAE J3016, "J3016," *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, vol. 4, pp. 593–598, 2014.

[3] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE transactions on intelligent transportation systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[4] W. Shi, M. B. Alawieh, X. Li, and H. Yu, "Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey," *Integration*, vol. 59, pp. 148–156, 2017.

[5] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[6] F. M. of Transport and D. I. (Germany), "Report on the status of implementation of the automated and networked driving strategy," November 2017.

[7] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems*, pp. 1455–1462, IEEE, 2015.

[8] E. Donges, "Supporting drivers by chassis control systems," *Smart vehicles*, 1995.

[9] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[10] U. ECE, "Uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signaling devices," *Regulation*, no. 48, 2010.

[11] H. Duda and S. Berkner, "Integrated chassis control using active suspension and braking," in *Int. Symposium on Advenced Vehicle Control (AVEC)*, vol. 347, p. 352, 2004.

[12] M. Münster, U. Mair, H.-J. Gilsdorf, A. Thomä, C. Müller, M. Hippe, and J. Hoffmann, "Electromechanical active body control," *ATZautotechnology*, vol. 9, no. 3, pp. 24–29, 2009.

[13] M. W. Sayers, "Guidelines for the conduct and calibration of road roughness measurements," tech. rep., 1984.

[14] M. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," 1998.

[15] M. Sayers, "Two quarter-car models for defining road roughness: Iri and hri," *Transportation Research Record*, no. 1215, 1989.

[16] C. Dodds and J. Robson, "The description of road surface roughness," *J. of sound and vibration*, vol. 31, no. 2, pp. 175–183, 1973.

[17] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[18] G. Maerschalk, A. Ueckermann, and S. Heller, "Längsebenheitsauswerteverfahren," *Bewertetes Längsprofil": Weiterentwicklung der Längsebenheitsbewertung der Zustandserfassung und–Bewertung,"[Analysis procedure for longitudinal road roughness "Weighted Longitudinal Profile"], Report of the Federal Highway Research Institute (bast)*, no. S73, 2011.

[19] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, "Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection," 2018.

[20] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. Kolbe, "Intelligent road maintenance: a machine learning approach for surface defect detection.," in *ECIS*, p. 194, 2018.

[21] M. I. Rajab, M. H. Alawi, and M. A. Saif, "Application of image processing to measure road distresses," *WSEAS Transactions on Information Science & Applications*, vol. 5, no. 1, pp. 1–7, 2008.

[22] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv preprint arXiv:2008.13101*, 2020.

[23] G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, "SmartRoadSense: Collaborative road surface condition monitoring," in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 210–215, IARIA, 2014.

[24] G. Alessandroni, A. Carini, E. Lattanzi, and A. Bogliolo, "Sensing road roughness via mobile devices: a study on speed influence," in *Int. Symposium on Image and Signal Processing and Analysis*, pp. 272–277, IEEE, 2015.

[25] G. Alessandroni, A. Carini, E. Lattanzi, V. Freschi, and A. Bogliolo, "A study on the influence of speed on road roughness sensing: The SmartRoadSense case," *Sensors*, vol. 17, no. 2, 2017.

[26] V. Freschi, S. Delpriori, L. C. Klopfenstein, E. Lattanzi, G. Luchetti, and A. Bogliolo, "Geospatial data aggregation and reduction in vehicular sensing applications: the case of road surface monitoring," in *Int. Conf. on Connected Vehicles and Expo*, IEEE, 2014.

[27] A. Kanjanavapastit and A. Thitinaruemit, "Estimation of a speed hump profile using quarter car model," *Procedia-Social and Behavioral Sciences*, vol. 88, pp. 265–273, 2013.

[28] K. Dobaj *et al.*, "Influence of car wheel suspension parameters on improvement of active safety and ride comfort," *Czasopismo Techniczne*, vol. 2015, no. Mechanika Zeszyt 3-M (21) 2015, pp. 13–22, 2015.

[29] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, "Enabling road condition monitoring with an on-board vehicle sensor setup," in *IEEE Sensors*, IEEE, 2019.

[30] M. W. Sayers, "On the calculation of international roughness index from longitudinal road profile," *Transportation Research Record*, no. 1501, 1995.

[31] F. Kortmann, Y.-C. Hsu, A. Warnecke, N. Meier, J. Heger, B. Funk, and P. Drews, "Creating value from in-vehicle data: Detecting road surfaces and road hazards," in *IEEE Vehicular Technology Conf.*, p. Accepted for publication, IEEE, 2020.

[32] S. Hofmann, "Numerische integration von beschleunigungssignalen [numerical integration of acceleration signals]," *Information from the Institute of Mechanical Engineering at Clausthal University of Technology*, vol. 38, pp. 103–114, 2013.

Chapter VI

Creating Value from in-Vehicle Data: Detecting Road Surfaces and Road Hazards

Outline

Bibliographic Information

Felix Kortmann, Yi-Chen Hsu, Alexander Warnecke, Nicolas Meier, Jens Heger, Burkhardt Funk, and Paul Drews, (2020, September). “Creating Value from in-Vehicle Data: Detecting Road Surfaces and Road Hazards”. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) (pp.1-6). IEEE. DOI: 10.1109/ITSC45102.2020.9294684. Preprint on ResearchGate:347957002.

Author’s contribution

The author’s share of the publication is 70%. Table B.6 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

An important component for the realization of the automated driving task is a holistic environment model. CAVs must be capable of detecting other vehicles, road markings, dangerous obstacles and upcoming road conditions. Apart from the comfort dependency on the road condition, friction values are calculated on the basis of road properties, which in turn are relevant for e.g. breaking and safety distances of CAVs. Due to the substitution of the human control task by the machine, this information must in future be detected by the vehicle itself. Based on the existing VLSs and ASs data, which are standard components in modern vehicles, a machine-learning approach of determining road surface materials and road hazards is presented. Our software solution of determining different road surface materials as asphalt, concrete, cobblestone or gravel with a total accuracy of 92.36 % is presented. Furthermore, the results of the road hazards detection as potholes and speed bumps with a total accuracy of 92.39 % is stated. Additionally to the edge calculations in the vehicle, our idea resolves in connected vehicles being capable of classifying road conditions enabling them to provide road analyses

to a cloud platform. The goal is to establish a holistic cloud solution for road conditions to enable CAVs for the consumption of road condition data of upcoming road segments and empower them to adjust to those.

VI.1 Introduction

Automated driving is one of the most researched topics within the automotive industry in recent years. The automated driving task can fundamentally disrupt traffic as we know it today. Besides the creation of new leisure time, working time or rest time, the transfer of the driving task to a computer enables a great improvement in driving comfort and safety. The realization of this task requires a holistic environment model, in which the vehicle is situated, so that it can move adapted to outer conditions. Most environmental information such as the position of other vehicles, road markings and possible obstacles are usually recorded by an on-board sensor system consisting of a combination of LIDAR, RADAR and camera components. In addition to on-board recognition, the improved development of radio technologies (e.g. 5G) allows information from the internet to be provided as digital services. These services can provide previously unknown information, create redundancies for existing on-board data and, in bad weather situations, provide information that may no longer be covered by on-board systems.

The Vienna Convention on Road Traffic [1] of 1968 defines rules of conduct for drivers in 80 countries around the world. Article 13 states that vehicle speed must be adapted to road conditions, among other things. Based on Donges' [2] Chassis Support System Model for ADASs, Winner and Hakuli [3] describe a modified version of a hierarchy model for automated vehicles, as shown in Figure VI.1. The hierarchy model is divided into driver, vehicle and environment. While in automated driving the driver's remaining task consists only of transferring the transport task to the vehicle, the vehicle takes over navigation, guidance and stabilisation. The road surface is part of the environment and is necessary for the actual trajectory and speed of vehicle movement.

In addition to the height profile of the road, which in practice and literature is covered by route-discrete evaluation indices, the road surface materials and strong anomalies are of great interest for the driving task as they lead to potential danger in terms of safety and comfort. Friction values depend on the specific vehicle, the road surface materials, the height profile and any moisture and wetness on the road.

Instead of an additional hardware system for recording preceding road surfaces and anomalies, we rely on the use of data already in the vehicle.

Primarily the data from the VLSs and AS at the wheel mass are used for this purpose. The VLS measures the displacement of the chassis mass and the wheel mass, thus the influence of the road to the spring-damper system [4]. We want to enable each vehicle to contribute to a cloud platform of road conditions, surface materials and anomalies by our software approach. The overall goal is to provide a worldwide online road condition map.

Our contributions can be summarized as follows:

- We investigate in-vehicle data for supervised road surface monitoring and anomaly detection.
- We propose new features and investigate the importance of those for the use case.
- We prove the applicability of machine learning for road surface and hazard recognition.

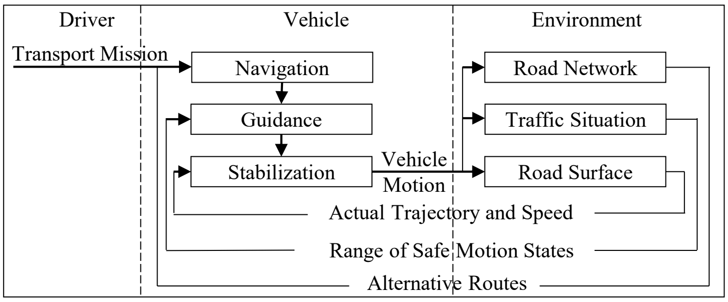


Figure VI.1: Hierarchy model for automated driving according to Winner [3]

VI.2 Fundamentals and Related Work

There are different measurands for the investigation of road conditions. One measurand is the complete recording of road height differences in longitudinal and transverse direction. The information obtained is then broken down into discrete segments and evaluated by a road quality index. In this paper we describe this as road roughness. The second measurand is the detection of the road surface and material such as gravel, cobblestone, asphalt or concrete - road surface material. The latter measurand deals with the detection of anomalies like potholes and speed bumps - road hazards.

For standardizing road quality, specifically road roughness, the World Bank introduced the IRI in 1982 [5]. Sayers used the IRI for measuring the longitudinal road profile [6] by accumulating deviations measured in vertical inches per mile or millimeter per meter. He specifies the IRI as an accumulation of the displacement between sprung and unsprung masses of a vehicle at 80 km/h over a discrete distance [7]. Dodds and Robson [8] also described road roughness in 1973 on the basis of the PSD of the elevation profile. Each elevation profile can be decomposed in a set of sinusoids to gather the importance of different wave numbers [6].

The Federal Ministry of Transport and Digital Infrastructure (German: Bundesministerium für Verkehr und digitale Infrastruktur - BMVI) developed a road roughness classification index by utilizing four triangulation laser sensors. The technique is capable of measuring road elevation in intervals of 10 cm along the longitudinal road profile [9, 10]. The measurement procedure is very reliable and accurate but also expensive due to the high investments in hardware. The system is not applicable for common vehicles. Special measuring vehicles drive over country roads and motorways in a rhythm of about 4 years.

In the literature, camera-based approaches of detecting road hazards are described. Chatterjee et al. [11, 12] utilize frontal cameras and machine learning for the detection of road defects as cracks and spots. Rajab et al. [13] whereas investigated cameras for the detection of crackings and potholes. Islam et al. [14] use images and a histogram analysis for surface detection.

A commonly used crowd sensing approach utilizes smartphones in the inner cabin of the vehicle. In the research project SmartRoadSense [15], the acceleration sensors of smartphones analyze road roughness or defects from the inner cabin. The project focus, between the actual calculation of road roughness [15], on speed dependencies for the sensor signals [16, 17, 18, 19, 20] and geospatial aggregation techniques of crowd-sensing data [21]. The use of smartphones comes with a great penetration rate on the roads. Even tough, a single device is subject to uncertainty in the position and cushioning (e.g. seats, beverage storage area). A collaborative approach is needed to obtain reliable results. Sattar et al. [22] perform a review onto road surface monitoring using smartphones.

Our presented solution includes the machine-learning based detection of road hazards and road surfaces. The paper describes how the road elevation and therefore the road roughness can be calculated using the setup explained in Section VI.3. The solution covers road hazards and road surface material detection.

VI.3 Experimental Setup

This chapter presents the experimental setup of the road surface material and road hazard detection. Between the actual edge architecture of the test vehicle, the cloud architecture of the use case is explained. Furthermore, the pre-processing of the data and the labeling through supervised machine learning are presented. We utilize machine learning based on SVM, as we wish to create our own features, listed in Table VI.1 and Table VI.2. Furthermore, the appealing solution of utilizing neural networks is accompanied with an insufficiently clarified situation to ensure functional safety in automated driving.

VI.3.1 Edge Architecture

The VLSs are pre-built into the vehicle and additional ASs are retrofitted onto the unsprung masses of the two front wheels by a pipe clamp and a 3D-printed housing. The ASs is a common MEMS sensor capable of measuring accelerations in three dimensions. Depending on the vehicle and the actual receiver functions of the VLS data in the vehicle, the data is conventionally distributed in the vehicle network via the Controller Area Network Bus (CAN-Bus) every 20 ms or 40 ms. To achieve an improved data rate, resulting in more survey points in a discrete segments on the road, the VLS data is read directly from the sensor via PWM. Due to the adoption, we realize a data rate of 6 ms.

More information on the functional description of the CIPOS inside the VLS can be found in Bartscht et al. [23] and Hobein et al. [24]. The displacement of the chassis mass (sprung mass) and the wheel mass (unsprung mass) can be derived from the angular position of the VLS. The body of the vehicle sensor is fixed on the chassis and the lever is fixed on the spring strut of the suspension system at the wheel mass. The vertical displacement in the spring-damper system caused by the irregularities of road profiles can be further determined with the measured angle and the lever length through (1). Where d is the displacement of the chassis to the wheel, r is the length of the lever of the sensor and α is the measured angle.

$$d = 2r \cdot sin(\frac{\alpha}{2}) \tag{VI.1}$$

In addition, a GPS sensor which detects the current localization of the vehicle and enables the mapping explained in detail in Section VI.5.3 is installed. A micro-controller takes over the signal processing and sends the data to a Linux computer.

In addition to the already mentioned sensors, a CAN shield mounted on the top of the micro-controller is used to obtain further data from the vehicle CAN network. The current velocity is needed for the later domain-transformation from time to space. The measurement setup explained is shown in Figure VI.2 as a schematic view and in Figure VI.3 in real life. In the real life representation the connections on the micro-controller that receive the raw data from the sensors are described with AS and VLS.

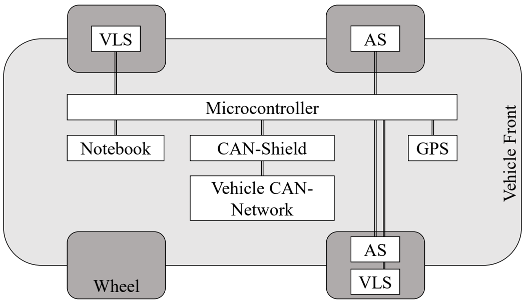


Figure VI.2: Edge architecture including all hardware components

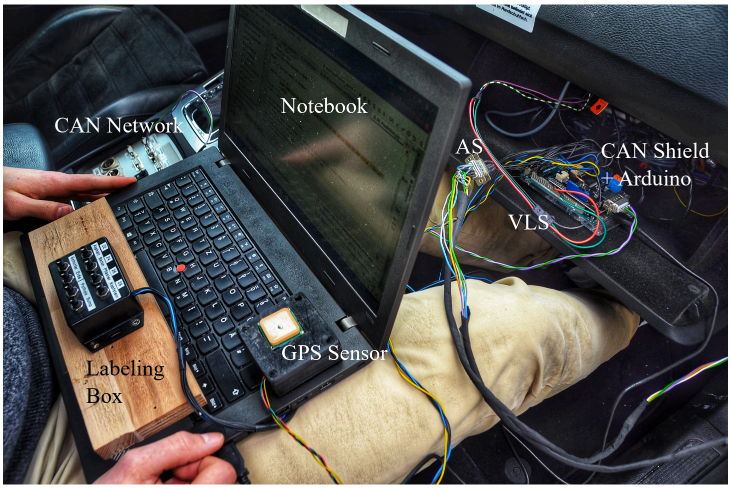


Figure VI.3: Real life system setup in the test vehicle

VI.3.2 Cloud Architecture

Figure VI.4 displays the functional blocks of the prototyping cloud architecture. The data transmitted by the edge device in the connected car ends up in the device gateway (Connection) of the cloud platform. In the integration step, the edge device elevations are mapped to the external source map data of Open Street Map (OSM)[1] by merging the GPS-based elevations to nodes at OSM. As a result, the road data from the edge devices is processed together with the real road network data in a central storage solution. The central storage solution is a relational database, which is particularly suitable for position-based applications.

In the batch layer, the data of the different connected cars and their evaluations are aggregated in order to provide a holistic solution of road conditions. Since different evaluations of vehicles for identical road sections are available, this process will become very important, but is fairly underutilized in its present maturity. The speed layer handles the recent contributed data from one vehicle/edge device to enable users checking their data contribution. Both can be displayed on a web application based on OSM and Leaflet[2], a JavaScript library for mobile-friendly interactive maps.

The cloud architecture, including the device gateway, serverless functions, the object storage and the relational database utilized in our prototype was build with Amazon Web Services (AWS).

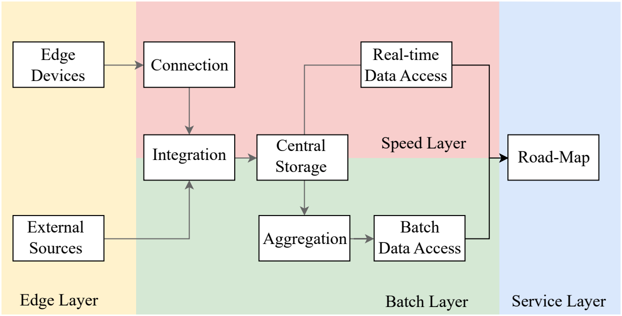


Figure VI.4: Cloud architecture design including its functional blocks

VI.3.3 Pre-processing

Figure VI.5 displays the processing steps for feature creation and the prediction. Starting from the raw data, the speed dependency of the VLS and AS data must be filtered out as well as possible. In order to achieve the most realistic possible speed dependency of the data, we are driving over a 2 km long road with extremely heterogeneous road conditions at varying velocities of 20 km/h to 100 km/h in steps of ten. A linear regression $r(v)$ as a function of velocity v allows us to realize first adjustments (3).

[1] https://www.openstreetmap.org/

[2] https://leafletjs.com/

$$r(v) = a \cdot v + b \tag{VI.2}$$

Furthermore, we extend this to include the re-scaling of all raw senor data y_{raw} to a reference velocity v_{ref}. The adjusted sensor data y_{adj} is calculated by y_{raw} multiplied by the ratio between the linear regression at reference velocity $r(v_{ref})$ and the linear regression at the current velocity from the CAN-Bus $r(v_{CAN})$.

$$y_{adj} = y_{raw} \cdot \frac{r(v_{ref})}{r(v_{CAN})} \tag{VI.3}$$

Following, the time sequential data must be transferred from time to space domain to enable road roughness and road surface materials being displayed on discrete road segments. Thereby the current velocities from the CAN-Bus are used. For road hazard detection, the signals are kept in time domain as road hazards are mapped as GPS points instead of segments. In addition to the purely physical sensor data, further physical and statistical features are created, the process of which is described in detail in Section VI.4 and is named in Figure VI.5 as Feature Engineering. In the following, the trained ML model is applied. The model used was the SVM with the Radial Basis Function (RBF) kernel [25, 26].

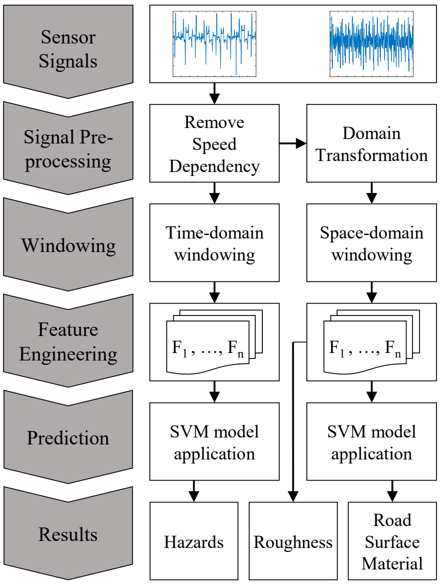


Figure VI.5: Pre-processing steps from raw data via features to predictions

VI.3.4 Labeling

The labeling process was included in the test drives. In order to make the process as accurate and convenient as possible, a labeling box was set up, which can be found in Figure VI.3. To label the road surface materials, the respective buttons are pressed when changing to a new pavement. To label the road hazards, the buttons are pressed for the duration of the anomaly. This is prone to errors, which is why labeling can be checked by recording video data in parallel to adjust the hazard windows.

VI.4 Feature Engineering

Table VI.1: Physical Quantities of the measurement setup

Phys. Quantity	**Description**
vehicle level	Displacement of unsprung and sprung mass
tilt	Inclination between front and rear axle due to VLSs measurements
slope	Derivation of the VLS data
acceleration(z)	Acceleration value of the vertical z axis
roll	Rotation angle around the vehicles heading direction $x = atan(a_x, a_z)$
pitch	Rotation angle around the vehicles transverse direction $y = atan(-a_y, \sqrt{a_x^2 + a_z^2})$

To prepare the data for the classification tasks, the features are extracted for road surface material detection in space and for road hazard detection in time window. The time windows are always of 1 s duration while with half a second overlap to prevent hazards from being found only in the edge areas of a window and therefore not being detected. The individual sensor signal can be used directly as a feature or can be combined with each other through mathematical operations to create new features. The relevant physical quantities derived from the sensor signals are listed in Table VI.1. A fundamental approach for pattern recognition draws on established concepts in statistical decision theory to differentiate data from different classes based on quantitative features of the data. There are a variety of statistical techniques that can be used for feature extraction. The usual statistical functions are applied to each signal channel including mean, range, RMS, standard deviation, skewness and kurtosis. Other functions such as Signal Magnitude Vector (SMV), Correlation of Acceleration along Gravity and Heading Direction (CAGH) and Eigenvalues of Dominant Directions (EVA) are applied on the acceleration signals.

These characteristics are extracted both for road surface classification and for hazard detection. In addition, some features in the frequency domain are extracted from spatial domain windows only, including PSD, PSD in different bandwidths, peak frequency, peak band and spectral entropy. The features are listed in Table VI.2.

Table VI.2: Statistical Features derived from the physical quantities

Features	**Description**
Mean	Average value of the signal in the window
Max, min	Maximal and minimal value of the signal in the window
Range	Difference between the maximum and minumum in the window
RMS	Quadratic mean value
Standard deviation	Mean deviation of the signal compared to average
Skewness	Degree of the asymmetry of the signal distribution; neutral, positive-biased or negative-baised
Kurtosis	Degree of the peak of the signal distribution; the distribution is wide or concentrated
Signal magnitude vector (SMV)	Sum of the euclidean norm over the three axis over the entire window normalized by the window length $\sum\sqrt{a_x^2 + a_y^2 + a_z^2/n}$
Eigenvalues of dominant directions (EVA)	Eigenvalues of the co-variance matrix of the accelerations along x, y, z axis. It refers to the variance along the dominant acceleration directions
CAGH	Correlation of the acceleration vectors in x and z axis, corr(a_x, a_z)
Total power spectral density	Total power of the PSD
Relative band power	Band power at 5 different bandwidths in percentage, bandwidth: [0 - 0.2] , [0.2 - 0.5] , [0.5 - 1] , [1 - 2] , [2 - 5]
Peak frequency	The frequency the peak appears
Peak band	The band the peak appears
Power entropy	The distribution of frequency components

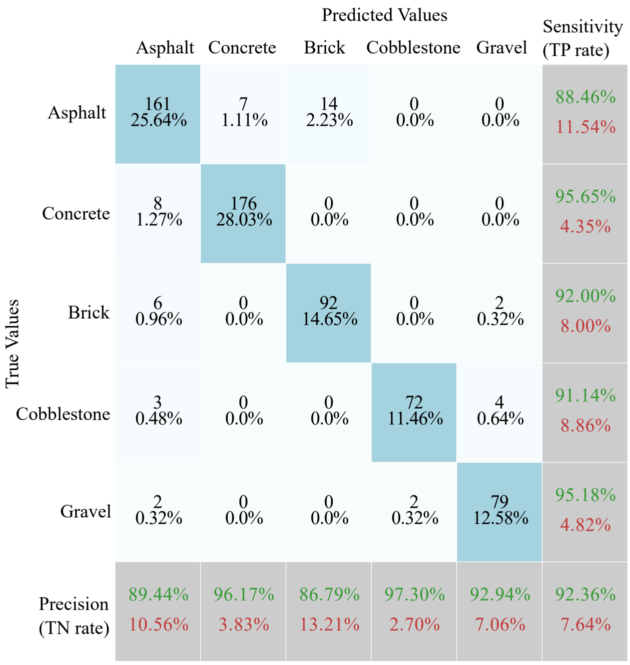
VI.5 Results

VI.5.1 Road Surface Material Prediction

This chapter presents and explains the results of the SVM on the prediction of road surface materials in Figure VI.6. The results are based on the evaluation of 9.8 km of test data from various road surfaces at varying road roughness levels. The segment length is 10 m. From our data set, 60 % are used as training set and 40 % as test set. The total accuracy of the presented model is 92.36 %. The biggest deviation is in the detection of asphalt with just

88.46 % accuracy. In total 40 segments are either wrong classified as asphalt (false positives - 19 segments) or wrong classified as another surface material despite being asphalt (false negative - 21 segments).

It can be assumed that the method of labeling via the labeling box results in inaccuracies in the exact position of the pad changes. To a large extent, there are road surface changes within the segments, which are only classified by one surface material in the labeling, but can be perceived as a different road surface by contracting the features into 10 meter segments.

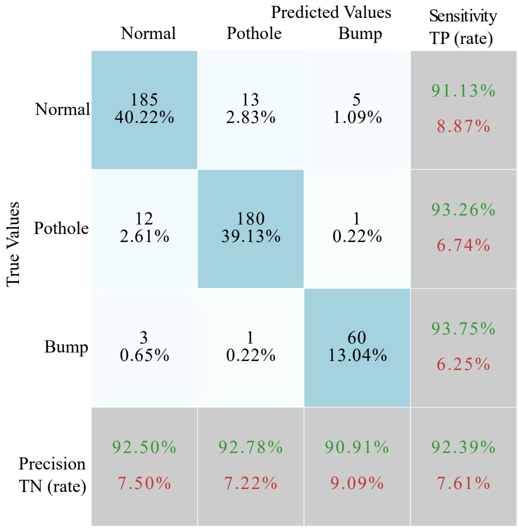


		Predicted Values					
		Asphalt	Concrete	Brick	Cobblestone	Gravel	Sensitivity (TP rate)
True Values	Asphalt	161 25.64%	7 1.11%	14 2.23%	0 0.0%	0 0.0%	88.46% 11.54%
	Concrete	8 1.27%	176 28.03%	0 0.0%	0 0.0%	0 0.0%	95.65% 4.35%
	Brick	6 0.96%	0 0.0%	92 14.65%	0 0.0%	2 0.32%	92.00% 8.00%
	Cobblestone	3 0.48%	0 0.0%	0 0.0%	72 11.46%	4 0.64%	91.14% 8.86%
	Gravel	2 0.32%	0 0.0%	0 0.0%	2 0.32%	79 12.58%	95.18% 4.82%
	Precision (TN rate)	89.44% 10.56%	96.17% 3.83%	86.79% 13.21%	97.30% 2.70%	92.94% 7.06%	92.36% 7.64%

Figure VI.6: Confusion matrix of the road surface material prediction

VI.5.2 Road Hazard Prediction

The results of the SVM prediction for road hazard detection can be found in Figure VI.7. The complete data set is split again in 60 % training and 40 % test data. The data was collected on a parking lot with many potholes and speed bumps. The total precision of the hazard detection is 92.39 % with over 93 % in pothole and bump detection. The confusion matrix also shows that the true positive rate of windows without anomaly is with 91.13 % slightly lower. 13 potholes and 5 speed bumps are detected despite normal road conditions.



		Predicted Values			Sensitivity
		Normal	Pothole	Bump	TP (rate)
True Values	Normal	185 40.22%	13 2.83%	5 1.09%	91.13% 8.87%
	Pothole	12 2.61%	180 39.13%	1 0.22%	93.26% 6.74%
	Bump	3 0.65%	1 0.22%	60 13.04%	93.75% 6.25%
	Precision TN (rate)	92.50% 7.50%	92.78% 7.22%	90.91% 9.09%	92.39% 7.61%

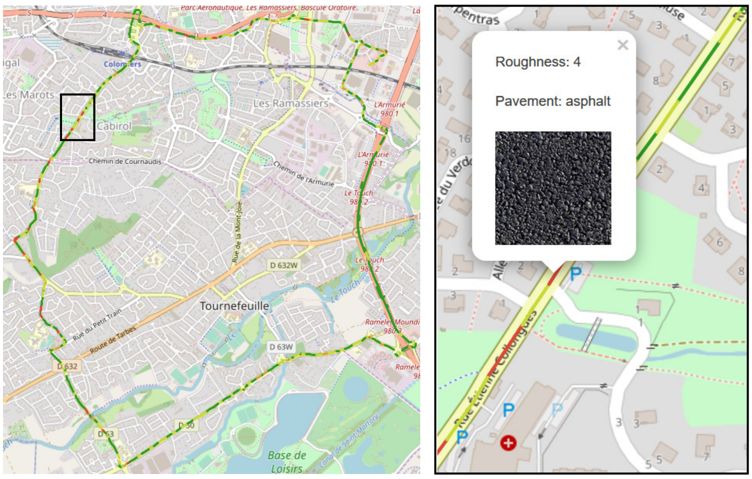
Figure VI.7: Confusion matrix of the road hazard prediction

VI.5.3 Visualization

Table VI.3: Road Roughness levels based on PSD evaluations [6]

Roughness Level		PSD
🟥	very high roughness	PSD $\geqslant 3$
🟧	high roughness	$1.5 \leqslant$ PSD < 3
🟨	low roughness	$0.5 \leqslant$ PSD < 1.5
🟩	very low roughness	PSD < 0.5

By the definition of OSM, a road is a sequence of nodes which form a polyline. The nodes are defined to approximate the curve of a road. A long straight road may only contain two nodes, while a curvy road may contain many nodes aligned narrowly. The length of the road segments varies from several meters to a few kilometers. The road segments are taken as reference storage units, so road segments that are too long are not desirable. In order not to change the geometry of the roads, they are divided into smaller segments by simply inserting additional nodes in between to keep each segment below a certain threshold length. To give an overview, the road conditions are visualized on the map. The visualization is realized with Leaflet, an open source JavaScript Library for the development of mobile friendly interactive maps, which allows different drawing layers to overlap on the map. After all information is matched to the roads, the road quality is displayed with different colors according to its roughness index. The roughness is divided into four levels, as shown in Table VI.3.


Roughness: 4

Pavement: asphalt

Figure VI.8: Map visualization of the road roughness including pop-up windows for surface material predictions

The road surface materials can be found in special pop-up windows by clicking onto the segments. The anomalies are visualized separately, shown in Figure VI.9. The green points represent the irregular points constructed on purpose, such as speed bumps. The red ones show the anomalies that require repair, such as potholes.

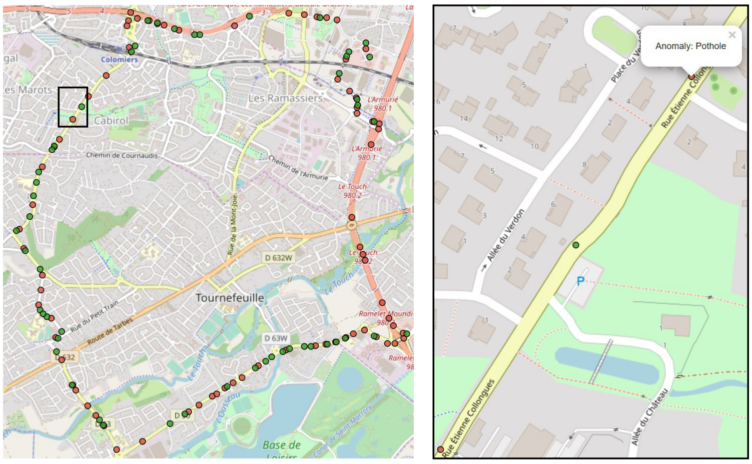

Anomaly: Pothole

Figure VI.9: Map visualization of the road hazard prediction for potholes and bumps including a zoomed picture on the right

VI.6 Conclusion

In this paper, we presented a machine-learning based solution of detecting road surface materials and road hazards with data from standard sensor components as the VLS and the AS of modern vehicles. The technical and experimental setup in our test vehicle have been described as well as the labeling procedure. We furthermore investigated the applicability of various features for the two use cases.

Between the actual results explained in Section VI.5, we proposed solutions for equalizing speed dependencies and domain transformations from time to space. The paper presented the prototype from sensors to digital service in form of the visualization.

Subsequently, significantly more labeled data should be recorded to improve accuracy for use. Also different machine-learning methods should be applied in the future. In addition, it will be investigated to what extent the results of different vehicles match and how these can be taken into account in case of deviations in the cloud database.

Bibliography

[1] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems*, pp. 1455–1462, IEEE, 2015.

[2] E. Donges, "Supporting drivers by chassis control systems," *Smart vehicles*, 1995.

[3] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[4] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, "Enabling road condition monitoring with an on-board vehicle sensor setup," in *IEEE Sensors*, IEEE, 2019.

[5] M. W. Sayers, "Guidelines for the conduct and calibration of road roughness measurements," tech. rep., 1984.

[6] M. W. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," 1998.

[7] M. W. Sayers, “Two quarter-car models for defining road roughness: Iri and hri,” *Transportation Research Record*, no. 1215, 1989.

[8] C. Dodds and J. Robson, “The description of road surface roughness,” *J. of sound and vibration*, vol. 31, no. 2, pp. 175–183, 1973.

[9] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, “Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data,” *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[10] G. Maerschalk, A. Ueckermann, and S. Heller, “Längsebenheitsauswerteverfahren,” *Bewertetes Längsprofil”: Weiterentwicklung der Längsebenheitsbewertung der Zustandserfassung und–Bewertung,”[Analysis procedure for longitudinal road roughness “Weighted Longitudinal Profile”], Berichte der Bundesanstalt für Straßenwesen, Reihe S: Straßenbau*, no. S73, 2011.

[11] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, “Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection,” 2018.

[12] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. Kolbe, “Intelligent road maintenance: a machine learning approach for surface defect detection.,” in *ECIS*, p. 194, 2018.

[13] M. I. Rajab, M. H. Alawi, and M. A. Saif, “Application of image processing to measure road distresses,” *WSEAS Transactions on Information Science & Applications*, vol. 5, no. 1, pp. 1–7, 2008.

[14] M. M. Islam, M. S. Sadi, M. M. Islam, and M. K. Hasan, “A new method for road surface detection,” in *2018 4th Int. Conf. on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, pp. 624–629, IEEE, 2018.

[15] G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, “SmartRoadSense: Collaborative road surface condition monitoring,” in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 210–215, IARIA, 2014.

[16] G. Alessandroni, A. Carini, E. Lattanzi, and A. Bogliolo, “Sensing road roughness via mobile devices: a study on speed influence,” in *Int.*

Symposium on Image and Signal Processing and Analysis, pp. 272–277, IEEE, 2015.

[17] G. Alessandroni, A. Carini, E. Lattanzi, V. Freschi, and A. Bogliolo, "A study on the influence of speed on road roughness sensing: The SmartRoadSense case," *Sensors*, vol. 17, no. 2, 2017.

[18] A. Vittorio, V. Rosolino, I. Teresa, C. M. Vittoria, P. G. Vincenzo, and D. M. Francesco, "Automated sensing system for monitoring of road surface quality by mobile devices," *Procedia-Social and Behavioral Sciences*, vol. 111, no. 5, pp. 242–251, 2014.

[19] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big data analytics in the social and ubiquitous context*, pp. 128–146, Springer, 2015.

[20] P. Harikrishnan and V. P. Gopi, "Vehicle vibration signal processing for road surface monitoring," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5192–5197, 2017.

[21] V. Freschi, S. Delpriori, L. C. Klopfenstein, E. Lattanzi, G. Luchetti, and A. Bogliolo, "Geospatial data aggregation and reduction in vehicular sensing applications: the case of road surface monitoring," in *Int. Conf. on Connected Vehicles and Expo*, IEEE, 2014.

[22] S. Sattar, S. Li, and M. Chapman, "Road surface monitoring using smartphone sensors: A review," *Sensors*, vol. 18, no. 11, p. 3845, 2018.

[23] M. Bartscht, T. Elpermann, S. Hoppe, and E. Wirries, "Level sensor," Apr. 19 2016. US Patent 9,316,497.

[24] D. Hobein, T. Dorißen, and K. Dürkopp, "Progress in automotive position sensors and introduction of the hella inductive position sensor," tech. rep., SAE Technical Paper, 2004.

[25] S. Han, C. Qubo, and H. Meng, "Parameter selection in svm with rbf kernel function," in *World Automation Congress 2012*, pp. 1–4, IEEE, 2012.

[26] Q. Liu, C. Chen, Y. Zhang, and Z. Hu, "Feature selection for support vector machines with rbf kernel," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 99–115, 2011.

Chapter VII

Detecting Various Road Damage Types in Global Countries Utilizing Faster R-CNN

Outline

Bibliographic Information

Felix Kortmann, Kevin Talits, Pascal Fassmeyer, Alexander Warnecke, Nicolas Meier, Jens Heger, Paul Drews, and Burkhardt Funk, (2020, December). “Detecting Various Road Damage Types in Global Countries Utilizing Faster R-CNN”. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5563-5571). IEEE. DOI: 10.1109/BigData50022.2020.9378245. Preprint on ResearchGate:348566481.

Author’s contribution

The author’s share of the publication is 70%. Table B.7 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

Information about road damages are of great interest for federal road authorities and their infrastructure management as well as the automated driving task and thus safety and comfort of vehicle occupants. Therefore, the investigation of the automatic detection of different types of road damages by images from a front-facing camera in the vehicle is of utter importance. Here we show a novel deep Learning approach utilizes the pre-trained Faster Region Based Convolutional Neural Networks (R-CNN). The data basis of our work is provided by the ’IEEE BigData Cup Challenge’ and its dataset ’RDD-2020’ with a large number of labelled images from Japan, India and the Czech Republic. In the first step, we classify the destination of the image followed by expert networks for each region. Between the explanation of our applied Deep Learning methodology, some remaining sources of errors are discussed and further, partly failed approaches during our development period are presented, which could be of interest for future work. Our results are convincing and we are able to achieve an F1 score of 0.487 across all regions for longitudinal and lateral cracks, alligator cracks and potholes.

VII.1 Introduction

Due to its connecting characteristic, the transport infrastructure is an important factor for the society. It links all areas of public and private life, including the health care system, social contacts, education and work. In addition to the general importance of infrastructure, we are experiencing a sharp increase in the complexity of individual traffic, also due to new technical developments in the automotive industry. The rise of the automated vehicle entails many opportunities, among them are the reduction of traffic congestion, optimized individual mobility, enhanced driving comfort and greatly increased safety for passengers and other traffic participants. 90 % of all traffic accidents are still caused by human error [1]. Besides the primary use of current road condition information by road construction authorities for the maintenance and repair of road infrastructure, up-to-date data on road conditions are also becoming increasingly important for automated driving tasks. According to Miller and Zaloshnja [2], the road infrastructure has an immensely high value for transportation safety and economic growth.

Recently, automated driving is emerging rapidly. The SAE Vehicle Standard defines a taxonomy for motor vehicle automation with ascending levels of automation from no automation (level zero) to fully automated driving (level five). For the realization of the fully automated driving task, a holistic environment model is required, which in current prototypes is powered by a combination of LIDAR, RADAR and camera data. According to Winner and Hakuli [3], that environmental model includes the road surface as a control variable. Thus, the task of road surface observation, traffic situation and road network analysis are transferred from the driver to the vehicle (cf. Figure VII.1).

Cognition of the environment and road surface assessments are primarily based on visual perception, and breakthroughs in the field of Deep Learning allowing the increasing replacement of human supervision by computer vision. The resulting ameliorations of vision-based ADASs are a central contribution to the automation of vehicles. They capture high-resolution texture data of upcoming road segments, while vibration-based detection methods, for instance, need the vehicle to cross the specific road area for detection. Hence, real-time camera evaluations enable the vehicle to take preventive actions. However, variations in illumination and environmental conditions (e.g. weather, surface conditions, other vehicles, buildings, road markings) and other disturbances like motion blur or specific sensor noise can lead to different image features for the same surface, which may cause erroneous recognitions and machine decisions.

We report on a novel deep learning solution for the classification of various road damage types using smartphone camera sensors aimed at the road ahead of a vehicle, making our research very important due to its predictive and real-time characteristics. Our research uses the Road Damage 2020 dataset provided by the 'IEEE Global Road Damage Detection Challenge 2020' with in-depth conditions explained in Arya et al. [4] and later in Section VII.3.1. In her paper, Arya et al. [5] describes the exact background of the challenge and summarizes the solutions of the 12 winners.

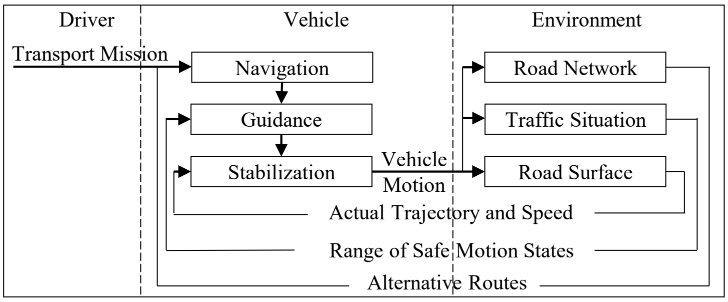


Figure VII.1: Hierarchy model for automated driving according to Winner [3]

Our contributions can be summarized as follows:

- We present our solution of detecting road damages utilizing FRCNN.
- We introduce regional expert networks to increase the performance of neural networks by eliminating regional differences.
- We show a clear superiority of the regional experts over the single predictor.

VII.2 Related Work

Automated road assessment systems are mostly based on technologies like RADAR, LIDAR, laser, remote sensors, ultrasonic sensors and image sensors. The different technologies for the detection of road conditions can be divided into different sub-categories. The first distinction is made between profilers that want to create a topology, a real elevation profile in vertical and horizontal, and those that are limited to detecting road defects. The creation of a holistic topology is usually very expensive and requires standardized measurement vehicles. The road classification procedure commonly used in Germany was standardized in 1991 by the Federal Ministry of Transport and Digital Infrastructure (BMVI) and is based on four triangulation laser sensors. The lasers are mounted on a grid-shaped measuring beam directed vertically downwards and process the longitudinal road profile at a distance of 10 cm [6], [7]. According to Sayers [67], such a system is a valid profiler with a real physical relationship to the road profile, invalid profiles instead just create a 'wiggly line' over the road profile without an established relation to the road elevation. Based on calculations of valid profilers, road quality indices can be calculated from the topology data to gain immediate insight about the considered profile. To be mentioned here are the IRI, introduced by the World Bank in 1982, and the PSD. The IRI displays the accumulation of the distance between sprung and unsprung masses of a vehicle at a velocity of 80 km/h over a defined distance [8, 9, 10]. The introduction of the PSD enabled the importance of different wavelengths in the road profile to be considered, which made it much easier to assess the road profile and its significance for the vehicle in motion [11, 9].

This does not mean, however, that invalid profilers are not usable, because they can still detect relevant wavelengths in the road profile and are often attractive in the crowd sensing scenario. Accelerometers in smartphones are frequently proposed to measure road roughness or to detect defects, in particular by Song et al. [12], Tai et al. [13], Perttunen et al. [14], Yi et al. [15], Seraj et al. [16] and Alessandroni et al. [17]. Despite of the uncertainty of single measurements due to inaccuracy of positioning in the vehicle, in disregard of the spring-damper systems of the vehicle and imprecision due to the smartphone sensors themselves, these approaches benefit from an already high distribution of hardware in the market and therefore low additional costs. Another crowd sensing scenario involves citizens to report anomalies to the public sector [18].

Many publications focus on visual road anomaly detection, however, the requirements of the solutions are often not identical and the tools to implement the application are extensively different. One category deals with the detection of cracks (e.g. [19, 20, 21]) as they are one of the most common road distresses and are a primary indicator of deterioration patterns [22]. Besides cracks, there also exists extensive research in visual pothole detection [23, 24] and visual multi-class anomaly detection [25, 26]. Some make use of a mix of machine classifiers and image processing, involving techniques for filtering and extracting useful features from images manually. Afterwards applying a machine classifier, while Chatterjee et al. [20] used a SVM to classify image regions into crack/no crack, Azhar et al. [27] classified the image first, following a segmentation/localization technique to detect the anomaly. Though, the application of feature extraction with image processing techniques in real-life scenarios is difficult [28], which is why research in visual anomaly detection has lately increasingly used DL, especially CNN architectures. Zhang et al. [29] proposed a DL approach in 2016 for automated crack detection, learning from manually annotated smartphone images. Gopalakrishnan et al. [19] used a CNN trained on the 'Big Data ImageNet' database [30] and proved the application of cross-domain transfer learning for vision-based crack detection. Mandal et al. [31] also used transfer learning by training an existent, real-time CNN detection system to obtain a model that locates cracks on images and classifies them into one out of eight crack categories in real-world situations. Fan et al. [21], trained a CNN for the classification of crack/non-crack; smoothed the crack-containing images to remove noise and applied an adaptive threshold technique to extract cracks.

Pereira et al. [32] proposed a CNN approach that outperformed conventional SVM based approaches in classifying an image into pothole/non-pothole, but the performance still suffered for varying illumination conditions and the data contained only severely potholes. An et al. [33] tested various CNN architectures on frames obtained by a front-facing smartphone camera to classify a road region into pothole/non-pothole. A transfer learning CNN approach to detect and locate potholes on images was proposed by Suong and Kwon [34], similar to Mandal et al. [31]. Maeda et al. [35] applied deep CNNs to locate anomalies on smartphone images by a bounding box and classify them into anomaly classes. The model can be run on smartphones. This was the first time a large-scale road damage dataset was prepared. Xia [25] used a CNN based object detection method to locate and classify cracks and potholes from camera videos in complex background. The detection accuracy decreased with the increasing speed of the vehicle.

To optimize the automated driving task, the road surface material is also crucial, as this is the basis for friction values. Most publications in this research category, named terrain classification (e.g. concrete, asphalte, gravel), do utilize DL [36]. Roychowdhury et al. [37] examined different CNNs to learn region-specific features on vehicle front-camera road images to classify into dry, wet, slush or snow road. Valada and Burgard [38] combined a CNN with recurrent LSTM units to train a model that captures both the spatial and temporal dynamics of road surface type classification (asphalt, mowed grass, high grass, paving, cobblestone, wood, dirt, linoleum, and carpet). Nolte et al.[39] trained two standard CNN architectures, which achieve high performance across many different domains, on frontal vehicle camera data to classify the road surface region into asphalt, cobblestone, wet asphalt, grass, dirt or snow. Their model is suitable for control algorithms of autonomous agents, as their sensor provides a look-ahead in front of the vehicle.

Our solution, in contrast to the aforementioned approaches, makes use of regional experts and is therefore applicable in different countries. The transferability to other countries is very well possible through transfer learning with our basic network, trained on images of all countries, which makes our approach expandable. Due to the extremely good performance of the Country-Classifier we hardly get any disadvantages due to the two-stage process.

VII.3 Experimental Setup

In this section we have a closer look at our experimental setup including the training conditions, our DL methodology and further approaches of interest, nevertheless they did not end up in our final solution.

VII.3.1 Dataset RDD-2020

Our solution makes use of the Road Damage Dataset 2020 introduced in the IEEE BigData Cup Challenge 'Global Road Damage Detection Challenge 2020' [4]. The dataset is based on two iterations of a previous dataset from Japan; RDD-2018 Maeda et al. [35] in 2018 with 9,053 images and 15,435 annotations and RDD-2019 Maeda et al. [40] in 2019 with 13,135 images and 30,989 annotations. In its current state, the RDD-2020 consists of a total amount of 26,620 images while additional pictures from India, Czech Republic and few from Slovakia have been added. A major difference between the datasets lies in the number of used damage categories. RDD-2020 consist of four categories including longitudinal cracks (D00), lateral cracks (D10), alligator cracks (D20) and potholes (D40). The other datasets included road marking deteriorations such as cross walk blur and white line blur which differ immensely in different countries, which is the reason to exclude those in the challenge.

For further information on the distribution of road damages, the standardization of road damage in other countries and more detailed information on the study area, we refer to Arya et al. [4].

VII.3.2 Deep Learning Methodology

Our work is based on the Tensorflow Object Detection API [41]. We used this API to train FRCNN networks with transfer learning from the given Detection Model Zoo. The pre-trained model is 'Faster R-CNN ResNet152 V1 800x1333'. All images were split 80/20 to get training and test data sets. From each, the needed TFRecord files for the input pipeline, containing all the annotation information, were created.

Figure VII.2 shows the basic structure of a FRCNN. It consists of two modules, a fully convolutional network for proposing the RoI and a Fast

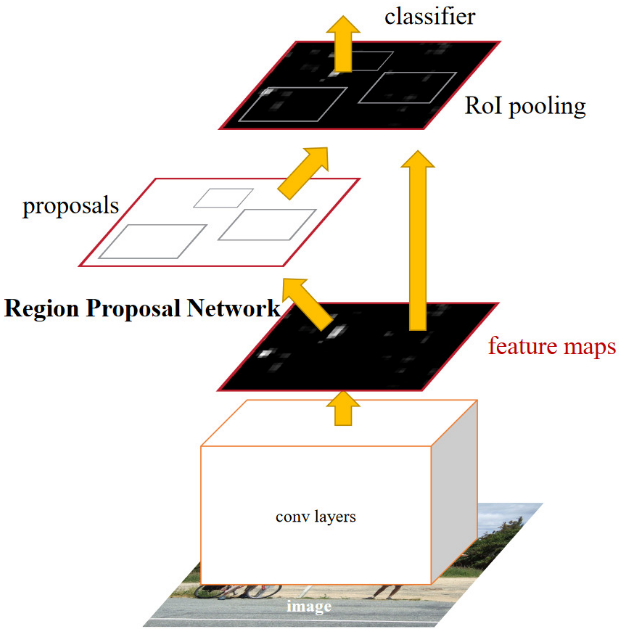


Figure VII.2: Basic FRCNN structure [42]

R-CNN detector [43], which works within the proposed regions. The given feature maps from the convolutional network are getting pooled by a max-pooling approach. Therefore out of every 2 × 2 sub-matrix of a feature map, only the maximum value is kept. Through shared convolutional features between the region proposal and object detection task, compelling speed and accuracy can be achieved. Our approach consists of the training of a classifier to distinguish between the countries and an Expert FRCNN for each country respectively as the images inhabit immense differences in vegetation, light and dust conditions. The object detection is performed within each Expert Network. This model is called 'Regional-Expert' and illustrated in Figure VII.3.

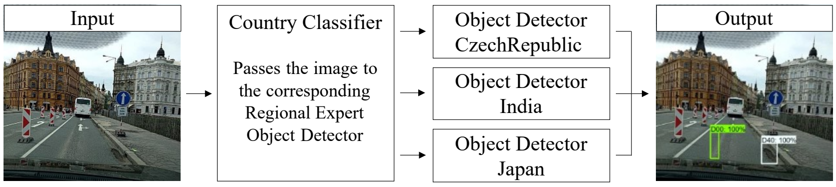

Input
Country Classifier
Passes the image to the corresponding Regional Expert Object Detector
Object Detector CzechRepublic
Object Detector India
Object Detector Japan
Output

Figure VII.3: Illustration of the Regional-Expert model

We created another FRCNN to classify the country. For the annotation a bounding box with the same dimensions as the image and a label for the country were created. The same images from the data challenge were used for this and split in a 80/20 training-test-split.

In Figure VII.4 the loss for the RPN and box classifier for the so-called Country-Classifier is shown. Box classifier loss and RPN loss are adding up together to the total loss, displayed in the plot. The RPN loss converged quickly near to zero and the classifier loss reaches values below 0.1.

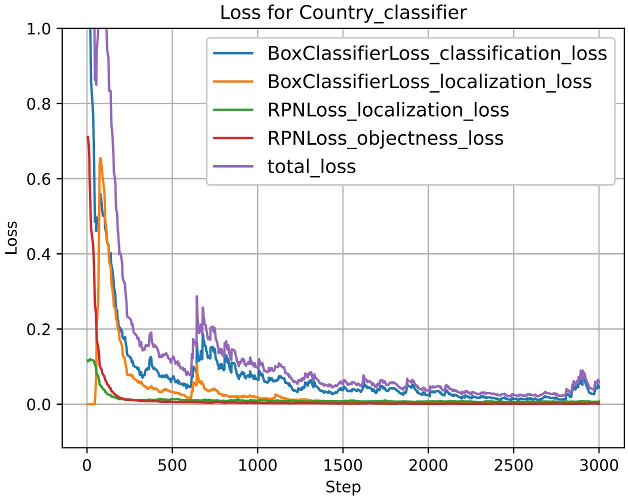


Figure VII.4: RPN and classifier loss with total loss for the Country-Classifier

The Single-Predictor was trained on the above mentioned pre-trained model from the Detection Model Zoo with the best set of hyper-parameters found by different learning approaches. This network serves as a starting point for the three different detection networks for Czech, India and Japan. Based on this network we fine-tune every network for the three regions to learn their specific road conditions and frequency of damage type on top of the basic Single-Predictor. In Section VII.4 we look at the improvement of the Regional-Expert in comparison with the Single-Predictor. Every expert network ran over 100 000 steps of transfer learning from the Single-Predictor. One step included an update via gradient descent with a maximum batch size of ten. The used images were sorted by countries and split into 80/20 training-test-split.

The loss of the Indian expert network in Figure VII.5 shows a converging trend with the box classifier loss being the main influence for the total loss, ending at a value around 0.3. Czech Republic and Japan expert networks show similar trends and a slightly lower loss for Czech at around 0.1 and a higher loss for Japan at around 0.4.

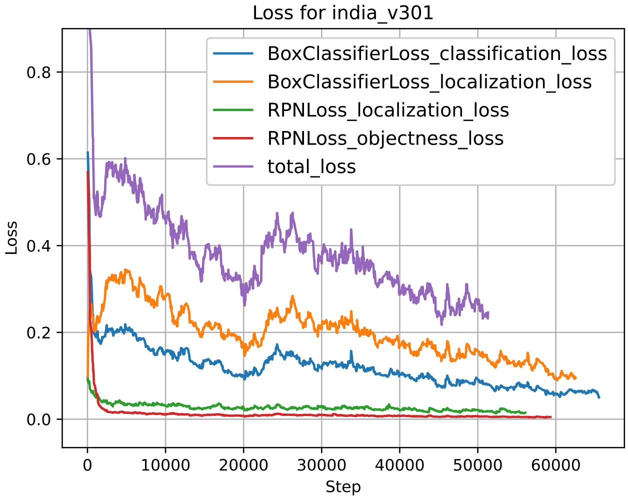


Figure VII.5: RPN and classifier loss with total loss for the Indian expert network

VII.3.3 Further Approaches

The following approaches were attempted during the preparation of our solution, but could not improve the results under the given training.

Arya et al. [4] describe the heterogeneous distribution of road damages within the dataset. To test the importance of a homogeneous road damage distribution, the provided images were sorted for every country. The resulting dataset is composed of 1 000 samples of every damage and 1 000 images completely free of damages. If certain damages are underrepresented in a country, the category is supplemented by images from other countries up to an amount of 1000 images. Although this approach yields a worse result, the importance of equally represented damages to avoid overfitting can not be underestimated. The statistics in the given dataset show a good reflection of the real world distribution of different road damages. A more homogeneous distribution is not feasible for the current training setup.

Further improvements of the Expert Network method can be achieved by reducing the incorrectly detected damages. Some detected damages are no road damages at all and lower the precision of our object detector. To reduce this negative influence a classifier was trained. It takes the detected region of a certain damage as input and ensures if there is a damage in the road. The single damage classifier did not improve the overall performance, due to its insufficient correctness in decisions. However, while some wrongly detected damages were removed, even more correctly detected damages were eliminated. This diminishes the score of our model. A better classifier could be trained With more time and a stronger GPU to be possibly beneficial for the project.

VII.4 Results

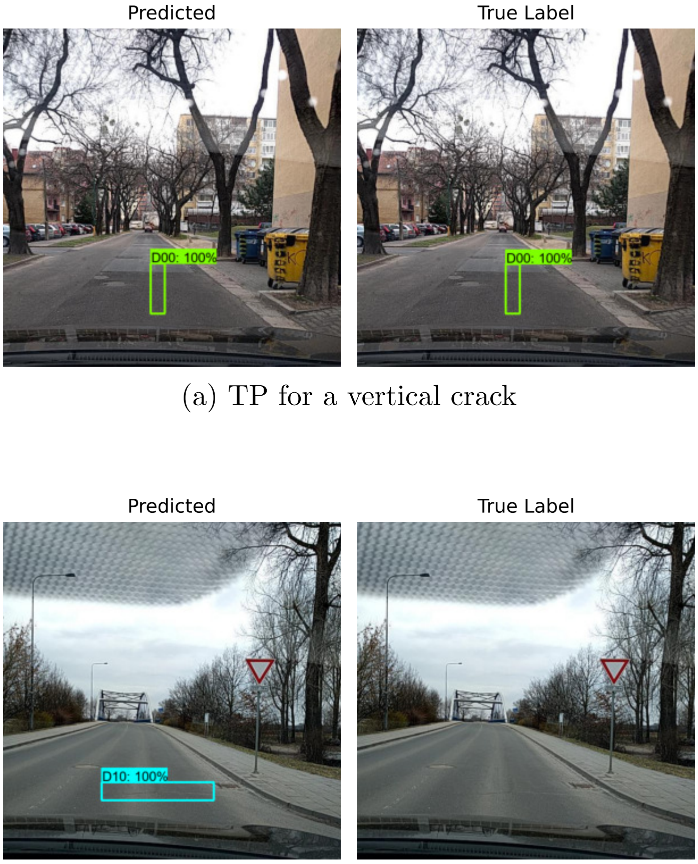
VII.4.1 Evaluation methods

To evaluate and compare different models precision, recall and F1-scores are used. Our presented evaluations are based on a different set of images than the calculated F1-score from the leaderboard of the challenge as the ground truth labels of the leaderboard testset were not published. Our in depth evaluation shows comparative results while containing further analysis. Precision is the percentage of correctly detected damages (True Positives (TP)) out of the total number of predicted damages (TP and False Positives (FP)). Recall is the percentage of correctly detected damages out of the total number of actual damages (TP and False Negatives (FN)). Sample outputs for TP, FP, True Negatives (TN), FN can be seen in Figure VII.6. To be counted as a TP the area of the detected damage has to overlap at least a defined amount with the true area of the damage. This is calculated with the IoU, which is the division of area of overlap by the area of union. The threshold is set to 0.5, if a correct detection only yield an IoU of under 0.5 it is considered a FP. Detection results of an image are compared with the annotations in the following fashion:

- Calculate the IoU of every detection with the first road damage
- Compare the predicted class, of the detection with the highest IoU, with the road damage label
- If the classes match, check if the IoU is above the threshold of 0.5
- If the IoU is high enough, the detection is a TP, else it is a FP
- Continue this until there are no more damages

Every road damage more, than detected damage, is a FN and every detected damage more, than a road damage, is a FP.

Precision and recall usually counter each other. The F1-score is used because of the balance between precision and recall. With a high F1-score it is ensured to get a reasonably high precision and recall. It is calculated as follows:



(a) TP for a vertical crack

(b) FP for a horizontal crack

Figure VII.6: Example images for TP and FP

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{VII.1}$$

We will use three different approaches on the F1-score, which will be macro F1-score, micro F1-score and the weighted F1-score.

The macro F1-score is the arithmetic mean of the F1-scores of every damage category.

The micro F1-score is calculated by, counting every TP, FP and FN over all damage categories as displayed in Equation (VII.1).

The weighted F1-score, whereas, utilizes the number of occurrences of the different damage types as weights for a weighted sum. The arithmetic mean of the weighted sum is then calculated.

Every detection is given with a score. This score states how certain the model is, that the detection is correct. For different thresholds of this score a precision-recall curve can be created. Averaging the precision values results in the mAP.

VII.4.2 Evaluation Results

The classifier performs with a F1-score of 0.98 and an mAP of about 0.95, displayed in Figure VII.7. The gray lines represent discrete F1-scores and every area above it has a higher score. Figure VII.8 shows the result of our Country-Classifier. Just a small amount of Czech and Indian images are miss-classified.

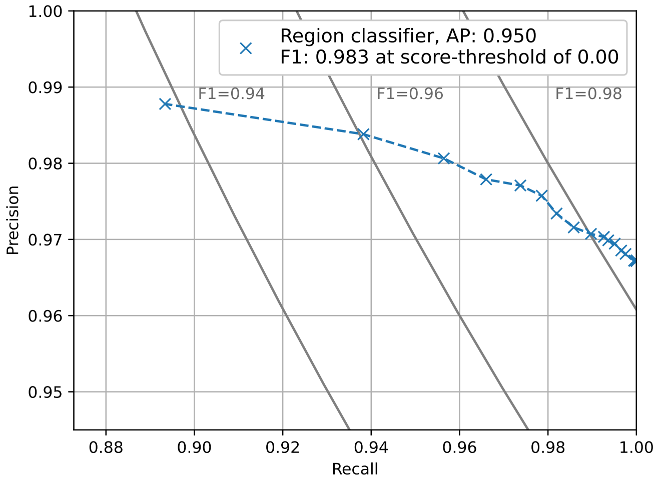


Figure VII.7: ROC-Curve for Country-Classifier with average precision and F1-score

Both, the confusion matrices and the precision-recall curves show a clear superiority of the Regional-Expert over the Single-Predictor. Figure VII.10 shows a trend to correctly detect the damages, but with a lot of FP and FN. The elements on the diagonal are the correctly detected damages (TP) The last diagonal element is the number of all images without a damage and without a predicted damage (TN). The other parts of the confusion matrix are falsely or not detected damages and are counted as FP and FN. A large quantity of not detected damages, are shown in high counts on the rightmost column. The last row shows the number of detected damages in regions without real road damage. 288 of these detections are caused by a too low IoU.

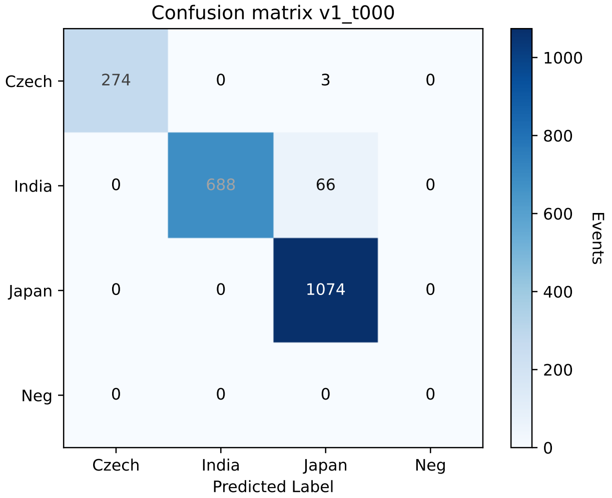


Figure VII.8: Confusion matrix for the Country-Classifier with a score threshold of 0

The damage category is correct, but the overlap of the predicted with the real damage was not sufficient enough. Within the categories, incorrect classifications occur much less frequently. Only horizontal and alligator cracks are more commonly mistaken for each other.

The more advanced Regional-Expert detector in Figure VII.9, with the Country-Classifier, increases the performance compared to the Single Predictor. The counts in the last column have decreased by a good amount. There are now only 209 cases with correct label and too small IoU. But the number of not detected damages is still high and has to be improved in future work. The improvement of the Regional-Expert over the Single-Predictor is seen in the direct comparison of the precision-recall curves in Figure VII.11. An mAP of 0.386, with a weighted F1-score of 0.487 is achieved for the Regional-Expert model, while the Single-Predictor only yields an mAP of 0.345 and a F1-score of 0.433. Although the Single-Predictor achieves single higher precision and recall values, the Regional-Expert can deliver a noticeably better overall performance with a better balance of precision and recall. Different variants of the F1-score with our test-set of images for Single-Predictor and Regional-Expert and the leaderboard scores are shown in Table VII.1. The different metrics deviate a bit from each other, but are giving overall similar results and are in harmony with the generated scores from the leaderboard. The enhancement is clearly visible, the Regional-Expert model reaches a higher performance.

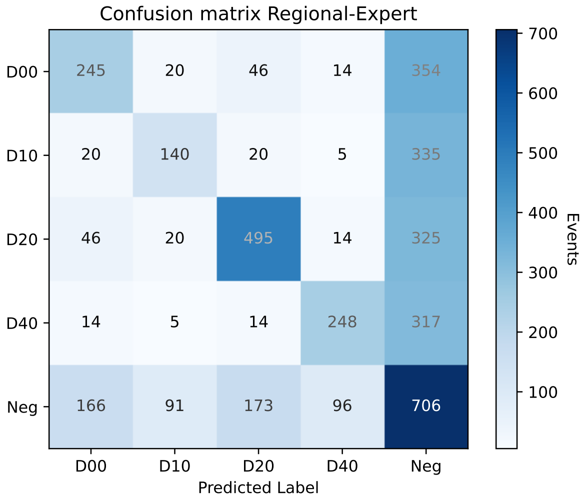


Figure VII.9: Confusion matrix for the Regional-Expert model with a score threshold of 0.90

Table VII.1: Macro, micro and weighted F1-score for Single-Predictor and Regional-Expert

	Single-Predictor	**Regional-Expert**
Macro	0.41569	0.4700
Micro	0.4141	0.4668
Weighted	0.4335	0.4865
Leaderboard	0.4086	0.4720

An overview of the performance of our model on every single damage category, the precision, recall and F1-scores are found in Table VII.2. Alligator cracks, with the label D20, are the easiest to detect for our model, because they are much larger and and have more distinct deteriorations than just small longitudinal cracks. The recall is lower in every category than the precision. This tendency is seen within the confusion matrices.

VII.4.3 Evaluation Discussion

After training, some test routines were performed and the time needed for a single processing of an image was measured (processing time). Along with this the time for initializing all the components of the model was registered.

Table VII.2: Per class precision, recall and F1-score, Regional-Expert

	Precision	**Recall**	**F1-score**
D00	0.5173	0.3536	0.4200
D10	0.5333	0.2471	0.3377
D20	0.6779	0.5402	0.6012
D40	0.6856	0.4067	0.5105

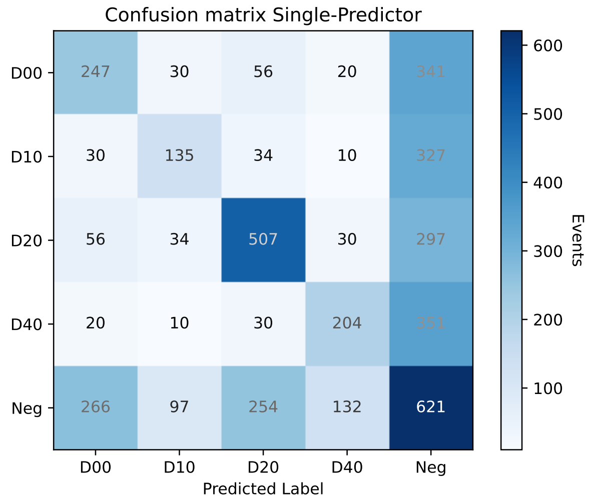


Figure VII.10: Confusion matrix for the Single-Predictor with a score threshold of 0.74

Table VII.3 shows the time per inference in ms and total time to load a model in s. The average time to load the Country-Classifier and the Regional-Expert is about 175.68 s and average time per interference is about 135.2 ms. After a loading time of about three minutes around seven images per second can be processed on the utilized hardware. To process an image every three meter, the velocity of the vehicle can be as high as $3\,\mathrm{m}/0.1352\,\mathrm{s} \approx 22.2\,\mathrm{m/s}$. With a more powerful GPU this speed can be greatly enlarged and a mapping of the street in short intervals would be even possible at highway speed. Though our model could be optimised in terms of real-time application in future works.

To successfully train the model with GPU acceleration, the batch size must be dramatically reduced in future work. This was due to our semi state-of-the-art GPU. With a more powerful GPU and thus a higher batch

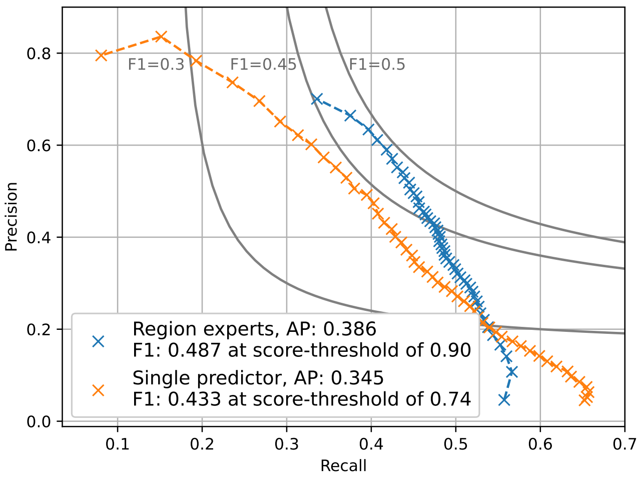


Figure VII.11: ROC-Curve for Single-Predictor and Regional-Expert with average precision and weighted F1-score

size, computation could be paralleled and the training could be accelerated to stable the gradient. After reviewing some evaluation images displayed in Figure VII.12, Figure VII.13 and Figure VII.14, some remaining issues in the network are detected. One problem consists in a missing label for manhole covers. In some cases manhole covers are miss-classified and detected as potholes. Other hard to detect features are small cracks phasing out of big alligator cracks, displayed in Figure VII.13. The detector counts the cracks as part of the bigger structures and loses some performance by gaining more FN. Similar to this problem many small cracks near each other in Figure VII.14 are sometimes detected as one bigger crack. This is a very logical problem, as there is no hard line in the manual labelling process between the accumulation of vertical and horizontal cracks and alligator cracking.

VII.4.4 Future Work

To decrease the impact of wrongly detected manhole covers as potholes, an extended dataset, containing a large enough amount of labeled manhole covers, is considered. With this larger dataset the current versions of Regional-Expert can be fine-tuned. A success in this minor problem should generate less FP and hence improves the scores of our model. In the future, the scope of this project will be to detect major road damages in real time and generate a general condition of the driven road. The importance of detecting every

Table VII.3: Time measurement for processing with NVIDIA GeForce GTX 1080 Ti

Loading the models [s]	**Time per inference** [ms]
165.50	138.0
182.88	136.9
173.55	135.5
180.72	135.3
181.59	135.8
169.82	129.8

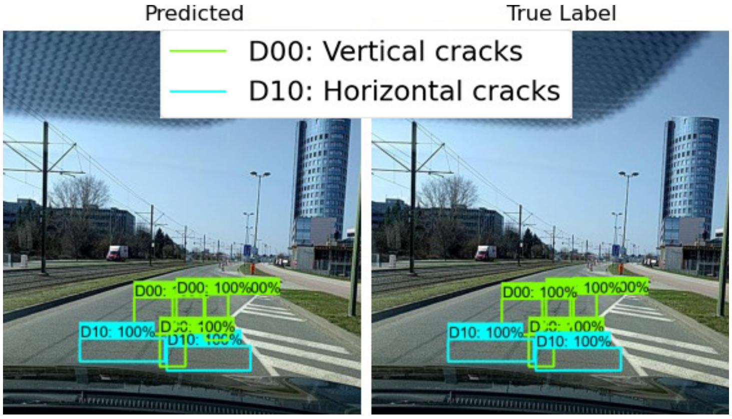


Figure VII.12: Many small cracks, correctly detected

crack depends on the use case. For automated driving though, greater structures and anomalies with the potential to decrease the vehicle ride comfort and safety are of greater interest.

VII.5 Conclusion

This Paper presents our work on the proposed dataset 'RDD-2020' of 26620 road images collected from Japan, India and the Czech Republic within the IEEE BigData Cup Challenge 'Global Road Damage Detection Challenge 2020'. The described model works great for a general detection of road conditions in different countries. Some minor flaws have to be corrected for precisely detecting even more single cracks in the street and to prevent miss-classification of manhole covers. A stronger GPU could improve our results

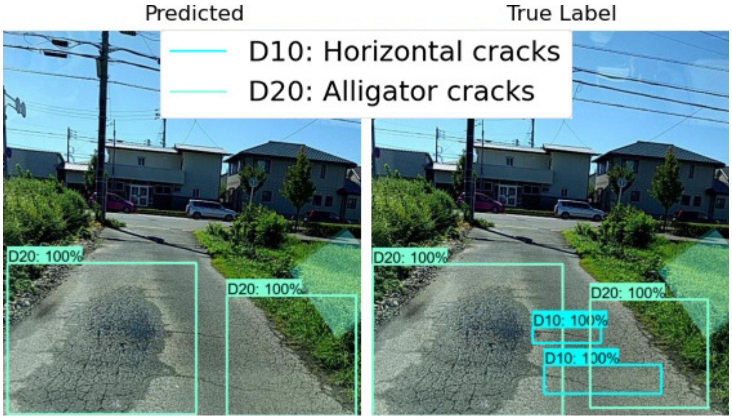


Figure VII.13: Cracks phasing out of alligator cracks can not be sufficiently detected

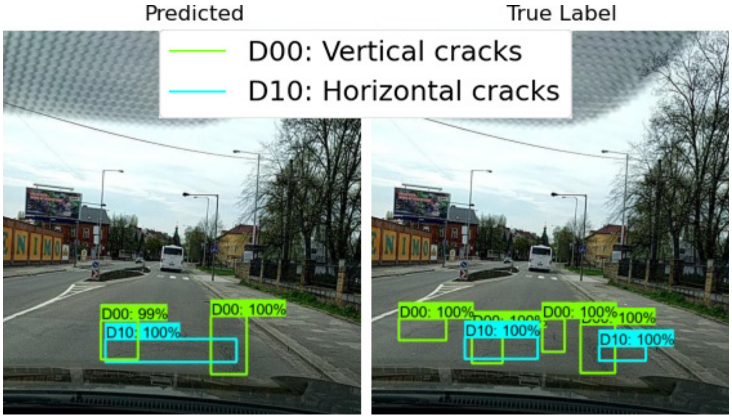


Figure VII.14: Many cracks overlapping each other. Not every crack can be detected

significantly by enlarging the batch size. With the given Single-Predictor detection model it is simple to expand the Regional-Expert model for other countries. Because of the great variety of road materials, the Single-Predictor generalizes the structural properties of cracks and potholes and with some new images from other countries their own Regional-Expert can be trained via transfer learning. We recommend our well performing Regional-Expert of the three existing countries in this dataset and using one of these as starting point of the transfer learning.

Another field of research opens up through the fusion of a camera-based solutions with the proposed approach of Kortmann et al. [44] where pre-installed on-board sensors are used to evaluate the elevation profile of road areas already passed through. The solution consists of a physical model-based calculation of the QVM. It should be investigated how these two solutions together can lead to improved results. Since the model-based approach is a real calculation of the height of the road profile, it could be attempted to train the camera data to the height information to further enhance the outcome of the DL solution.

Bibliography

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," tech. rep., 2015.

[2] T. R. Miller and E. Zaloshnja, "On a crash course: The dangers and health costs of deficient roadways," 2009.

[3] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[4] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Transfer learning-based road damage detection for multiple countries," *arXiv:2008.13101*, 2020.

[5] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Global road damage detection: State-of-the-art solutions," 2020.

[6] G. Maerschalk, A. Ueckermann, and S. Heller, "Längsebenheitsauswerteverfahren," *Bewertetes Längsprofil": Weiterentwicklung der Längsebenheitsbewertung der Zustandserfassung und–Bewertung,"[Analysis procedure for longitudinal road roughness "Weighted Longitudinal Profile"], Berichte der Bundesanstalt für Straßenwesen, Reihe S: Straßenbau*, no. S73, 2011.

[7] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[8] M. W. Sayers, "Guidelines for the conduct and calibration of road roughness measurements," tech. rep., 1984.

[9] M. W. Sayers, "The little book of profiling: basic information about measuring and interpreting road profiles," tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 1998.

[10] M. W. Sayers, "Two quarter-car models for defining road roughness: Iri and hri," *Transportation Research Record*, no. 1215, 1989.

[11] C. Dodds and J. Robson, "The description of road surface roughness," *J. of sound and vibration*, vol. 31, no. 2, pp. 175–183, 1973.

[12] H. Song, K. Baek, and Y. Byun, "Pothole detection using machine learning," *Advanced Science and Technology*, pp. 151–155, 2018.

[13] Y.-c. Tai, C.-w. Chan, and J. Y.-j. Hsu, "Automatic road anomaly detection using smart mobile device," in *Conf. on technologies and applications of artificial intelligence*, Citeseer, 2010.

[14] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppänen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, T. Ristaniemi, *et al.*, "Distributed road surface condition monitoring using mobile phones," in *Int. Conf. on ubiquitous intelligence and computing*, pp. 64–78, Springer, 2011.

[15] C.-W. Yi, Y.-T. Chuang, and C.-S. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1905–1917, 2015.

[16] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big data analytics in the social and ubiquitous context*, pp. 128–146, Springer, 2015.

[17] G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, "SmartRoadSense: Collaborative road surface condition monitoring," in *Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 210–215, IARIA, 2014.

[18] S. F. King and P. Brown, "Fix my street or else: using the internet to voice local public service concerns," in *Int. Conf. on Theory and practice of electronic governance*, pp. 72–80, 2007.

[19] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.

[20] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. M. Kolbe, "Intelligent road maintenance: a machine learning approach for surface defect detection.," in *ECIS*, p. 194, 2018.

[21] R. Fan, M. J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, and M. Liu, "Road crack detection using deep convolutional neural network and adaptive thresholding," in *2019 IEEE Intelligent Vehicles Symposium*, pp. 474–479, IEEE, 2019.

[22] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[23] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[24] S. Lee, S. Kim, K. E. An, S.-K. Ryu, and D. Seo, "Image processing-based pothole detecting system for driving environment," in *IEEE Int. Conf. on Consumer Electronics*, pp. 1–2, IEEE, 2018.

[25] W. Xia, "An approach for extracting road pavement disease from hd camera videos by deep convolutional networks," in *2018 Int. Conf. on Audio, Language and Image Processing (ICALIP)*, pp. 418–422, IEEE, 2018.

[26] L. Huidrom, L. K. Das, and S. Sud, "Method for automated assessment of potholes, cracks and patches from road surface video clips," *Procedia-Social and Behavioral Sciences*, vol. 104, no. 2013, pp. 312–321, 2013.

[27] K. Azhar, F. Murtaza, M. H. Yousaf, and H. A. Habib, "Computer vision based detection and localization of potholes in asphalt pavement images," in *IEEE Canadian Conf. on Electrical and Computer Engineering*, pp. 1–5, IEEE, 2016.

[28] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[29] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in *IEEE Int. Conf. on image processing*, pp. 3708–3712, IEEE, 2016.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conf. on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[31] V. Mandal, L. Uong, and Y. Adu-Gyamfi, “Automated road crack detection using deep convolutional neural networks,” in *2018 IEEE Int. Conf. on Big Data*, pp. 5212–5215, IEEE, 2018.

[32] V. Pereira, S. Tamura, S. Hayamizu, and H. Fukai, “A deep learning-based approach for road pothole detection in timor leste,” in *2018 IEEE Int. Conf. on Service Operations and Logistics, and Informatics*, pp. 279–284, IEEE, 2018.

[33] K. E. An, S. W. Lee, S.-K. Ryu, and D. Seo, “Detecting a pothole using deep convolutional neural network models for an adaptive shock observing in a vehicle driving,” in *2018 IEEE Int. Conf. on Consumer Electronics*, pp. 1–2, IEEE, 2018.

[34] L. K. Suong and J. Kwon, “Detection of potholes using a deep convolutional neural network.,” *J. UCS*, vol. 24, no. 9, pp. 1244–1257, 2018.

[35] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road damage detection using deep neural networks with images captured through a smartphone,” *arXiv:1801.09454*, 2018.

[36] A. Valada, L. Spinello, and W. Burgard, “Deep feature learning for acoustics-based terrain classification,” in *Robotics research*, pp. 21–37, Springer, 2018.

[37] S. Roychowdhury, M. Zhao, A. Wallin, N. Ohlsson, and M. Jonasson, “Machine learning models for road surface and friction estimation using front-camera images,” in *Int. Joint Conf. on Neural Networks*, pp. 1–8, IEEE, 2018.

[38] A. Valada and W. Burgard, “Deep spatiotemporal models for robust proprioceptive terrain classification,” *The Int. J. of Robotics Research*, vol. 36, no. 13-14, pp. 1521–1539, 2017.

[39] M. Nolte, N. Kister, and M. Maurer, “Assessment of deep convolutional neural networks for road surface classification,” in *2018 21st Int. Conf. on Intelligent Transportation Systems (ITSC)*, pp. 381–386, IEEE, 2018.

[40] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering.*

[41] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE Conf. on computer vision and pattern recognition*, pp. 7310–7311, 2017.

[42] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[43] R. Girshick, "Fast r-cnn," in *IEEE Int. Conf. on computer vision*, pp. 1440–1448, 2015.

[44] F. Kortmann, H. Peitzmeier, N. Meier, J. Heger, and P. Drews, "Enabling road condition monitoring with an on-board vehicle sensor setup," in *IEEE Sensors*, pp. 1–4, IEEE, 2019.

Chapter VIII

Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN

Outline

Bibliographic Information

Pascal Fassmeyer, Felix Kortmann, Paul Drews, and Burkhardt Funk, (Accepted Manuscript, 2021, September). "Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN". In 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall) (pp.1-7). IEEE.
Preprint DOI: 10.13140/RG.2.2.20745.36964.

Author's contribution

The author's share of the publication is 40%. Table B.8 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

Initiatives such as the 2020 IEEE Global Road Damage Detection Challenge prompted extensive research in camera-based road damage detection with Deep Learning, primarily focused on improving the efficiency of road management. However, road damage detection is also relevant for automated driving to optimize passenger comfort and safety. We use the state-of-the-art object detection framework Scaled-YOLOv4 and develop two small-sized models that cope with the limited computational resources in the vehicle. With average F1 scores of 0.54 and 0.586, respectively, the models keep pace with the state-of-the-art solutions of the challenge. Since the data consists only of smartphone images, we also train expert models for autonomous driving utilizing vehicle camera data. In addition to detection, severity assessment is critical. To classify detected damage into different severity levels, we propose a semi-supervised learning approach based on the encodings learned by combining a class-conditional Variational Autoencoder and a Wasserstein Generative Adversarial Network.

VIII.1 Introduction

The increasing automation of vehicles entails many opportunities, such as reducing traffic congestion, improving overall mobility, and increasing safety, as the vast majority (about 90%) of accidents result from human error [1]. To enable comfortable, safe, and efficient passenger transportation, the self-driving agent necessitates a reliable perception system for precise vehicle localization and understanding of the surrounding 3D environment. To make sense of the constant stream of sensor data from environment detection sensors like cameras, LIDARs, and RADARs, DL is usually applied.

Understanding and modeling the environment for ADASs or AVs necessitates evaluating the road surface [2]. Relevant road criteria include detecting and assessing road hazards and damage (e.g., potholes and cracks), particularly relevant in countries with mediocre road infrastructure. The relevance is emphasized by companies such as Tesla, striving to integrate bump and pothole detection into their AV environment perception systems [3]. In this context, we build DL-based road damage detection models and a severity classifier. Instead of utilizing expensive hardware like sophisticated camera systems or lidars, the models only require conventional cameras (e.g., smartphones, dashcams, or automotive cameras), making them applicable to a large crowd. The road surface analyses of the individual vehicles can be forwarded to a cloud platform, where the data is aggregated in a map, utilizing crowdsourcing. The key technology in this regard is High Definition (HD) maps, which usually contain much richer semantic information than standard navigation maps. Each connected car could provide and consume accurate road condition information to/from the cloud HD map in near real-time, enabling them to benefit from the crowd-based model of the environment [4].

Automatic road damage assessment systems that leverage standard hardware like simple front-facing cameras also represent an excellent opportunity to complement expensive road management systems without the need for additional financial expenditure by authorities. By aggregating the perceptions in an accurate map with a semantic road assessment layer, responsible institutions have an early and automatic up-to-date assessment of the road network, enabling them to take preventive steps to reduce maintenance costs and increase road sustainability and safety. This is particularly relevant for developing countries with insufficient financial resources and positively affects a country's economic growth [5].

This paper examines DL-based road damage detection models based on front-facing smartphone and automotive cameras. The detection process in vehicles requires to be performed at a fast processing speed, best at the camera sensor frame rate (30FPS), and cars often utilize embedded devices with limited processing power. Therefore, the exploration is carried out along a tiny and a medium-scale model architecture from the state-of-the-art object detection framework Scaled-YOLOv4 [6], which optimally trades-off speed and accuracy. Additionally, we investigate methods to increase the accuracy of road damage detection. Many techniques that achieve the highest accuracy are accompanied by increased inference cost. Therefore, we divide the methods that reduce the inference rate ('Bag of Investments') and those not affecting the inference time ('Bag of Presents'). We demonstrate that we can build accurate detection models even with limited computational resources while maintaining real-time speed. Since assessing severity is also a relevant factor for road management agencies and AV companies, we provide a primer for camera-based severity classification using semi-supervised DL.

Our contributions can be summarized as follows:

1. We build accurate real-time road damage detection models for practical application in vehicles considering limited computing resources.

2. We design a generative model based on VAEs and Wasserstein Generative Adversarial Networks (WGANs) capable of generating new damage samples conditioned on the damage type and classifying damage severity.

VIII.2 Related Work

With the advent of modern DL object detectors, end-to-end learning-based damage detection pipelines emerged. However, such approaches remained limited until Maeda et al. [7] prepared a large-scale road damage dataset for the first time, which was composed of 9,053 images collected with a front-facing smartphone inside a vehicle and included eight damage classes according to the Japanese road maintenance guidelines. The data underwent consistent improvement and received high research interest. For instance, Angulo et al. [8] modified the dataset and added samples for minority classes. Maeda et al. [9] released a new version of the dataset in 2019 and proposed a GAN to augment it synthetically.

In 2020, the dataset was extended again, and the Global Road Damage Detection Challenge (GRDDC) 2020 [10] was held as part of the IEEE International Conference on Big Data 2020. In this challenge, 121 teams participated in building road damage detection systems based on data from three countries: Czech Republic, India, and Japan. The images were obtained from front-facing smartphones installed in vehicles, and the objective was to perform as accurately as possible, measured by the F1-Score on two provided test data sets. The training set is composed of 21,041 images, and the test data has 5,295 samples (publicly available here). The labels of the test sets were not released, and the comparison of results was made via an evaluation server, which is still accessible at the time of writing. Four damage classes were considered: longitudinal crack, transverse crack, alligator crack, and pothole. Bounding boxes (BBs) mark the damages in the image, which makes this a classical supervised object detection problem.

Arya et al. [10] summarize the challenge solutions. The top-12 ranked teams used established DL object detection architectures and adopted some data augmentation (e.g., random photometric or geometric transforms).

Zhang et al. [11] trained a Conditional GAN (CGAN) to augment the data with synthetic road damages, which improved performance. Their approach is similar to Maeda et al. [9], but instead of using a GAN only creating potholes, the CGAN produces all four damage types. Also, all submissions used Transfer Learning (TL) based on MS COCO or ImageNet. The top-performing teams (ranked #1-#3) applied an ensemble of models at inference time, common in challenges to achieve the most accurate performance on test data without other requirements (e.g., speed/applicability). YOLO-based models (rank #1, #2, #4-#6, #8) outperformed the Faster-RCNN approaches (rank #9-#12). Test Time Augmentation (TTA) produced mixed results, with Hedge et al. [12] observing benefits, while the performance declined in Pham et al. [13]. Several teams [14, 15] applied DL-based semantic segmentation to help the network focus on the road area.

Our work differs from GRDDC 2020 in the following ways. First, we employ the new state-of-the-art object recognition framework Scaled-YOLOv4 and introduce several additional methods to improve accuracy without compromising inference speed. We achieve comparable performance to GRDDC 2020 solutions while our models run at a remarkably high inference rate. Second, in addition to the challenge smartphone data, we leverage car camera data to build expert models for AVs. Finally, we propose a severity classifier created with semi-supervised learning using the damage BBs.

VIII.3 Methodological Setup and Approach

VIII.3.1 Automotive Dataset

The base data for our analyses is the Challenge Data (CD), consisting of 21,041 labeled front-facing smartphone images from Japan, India, and the Czech Republic of size 600 × 600 or 720 × 720. As an addition, we labeled 10,421 images from automotive-grade front-facing cameras recorded in Ger-

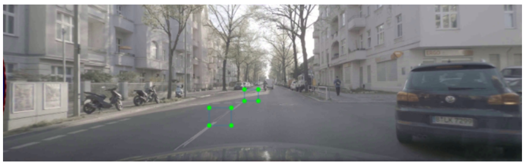
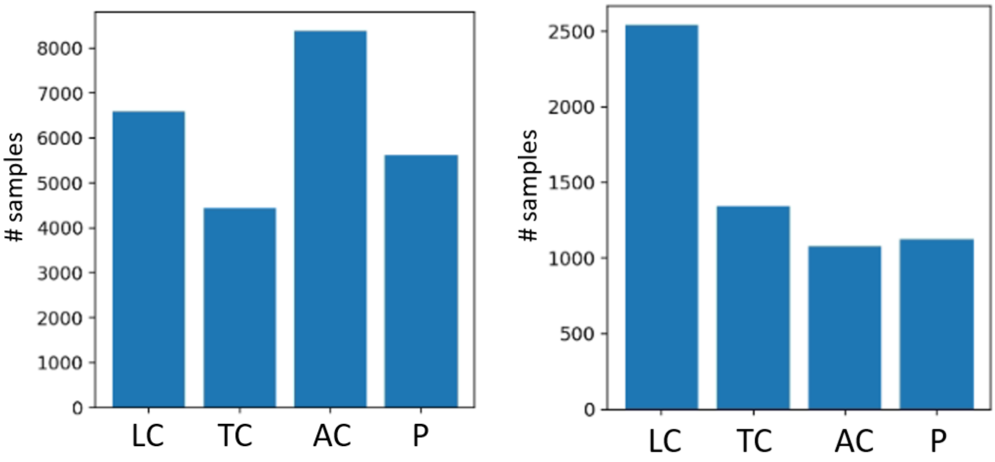


Figure VIII.1: Example from the Automotive Dataset



(a) CD labels

(b) AD labels

Figure VIII.2: The damage type distributions of the considered data

many (40%), the USA (28%), United Kingdom (10%), South Korea (10%), Poland (4%), France (3%), Latvia (3%), and Finland (2%). We considered the four classes from the CD. In contrast to the CD, the Automotive Data (AD) contains a vast number of different situations (e.g., night drives, snow, rain) and includes damages far away from the car (e.g., Figure VIII.1) since AVs should detect potential obstacles as early as possible. The image size (3504 × 1072) and label distributions (Figure VIII.2) also differ significantly. As the automotive camera is in a fixed position, the AD has a stable front-facing view, while the CD contains images that significantly differ in the viewing angle.

VIII.3.2 Scaled-YOLOv4

The related work indicated the superiority of YOLO over other object detection frameworks, which is why we focus on the current state-of-the-art object detection framework Scaled-YOLOv4. We chose two neural network structures from Scaled-YOLOv4: YOLOv4-Tiny and YOLOv4-CSP. YOLOv4-Tiny achieves real-time performance on embedded devices (e.g., Jetson NANO), and YOLOv4-CSP is a medium-sized architecture optimized for general GPU. The networks follow the archetypal structure of modern object detectors, having a *backbone* to extract features, a *head* to detect the objects, and a *neck* between backbone and head.

The backbone of YOLOv4-CSP is a modified version of CSPDarknet53 [16]. The neck consists of SPP [17] as additional blocks to enhance the receptive field and a modified PAN [18] for feature integration from different backbone levels. The predictions are realized across three different heads, and three scale-dependent anchors are utilized for each of the output tensors. YOLOv4-CSP uses the Mish activation [19]. YOLOv4-Tiny possesses a much smaller network architecture. Its backbone is composed of a CSP-ized VoVNET [20] for high computational and energy efficiency. As part of the neck, YOLOv4-Tiny utilizes an FPN [21] but does not use SPP or PAN. It has two instead of three prediction heads and takes smaller input images (size 416 instead of 512 or 640 as in CSP). YOLOv4-Tiny applies the Leaky ReLU [22].

The total loss function comprises a classification loss, objectness loss, and BB regression loss. The responsibility for detecting an object *obj* is determined via:

$$I_{ij}^{obj} = \begin{cases} 1 & \text{if anchor } j \text{ in cell } i \text{ has } IoU_{obj}^{anchor} \geqslant T \\ 0 & \text{otherwise} \end{cases} \tag{VIII.1}$$

where the Intersection Over Union (IoU) threshold T is a hyperparameter. The classification loss is the sum of independent binary crossentropy losses for each class:

$$\begin{aligned} L_{cls} = -\sum_{i=1}^{S^2}\sum_{j=1}^{B} I_{ij}^{obj} \sum_{c\in C} [& w_{cls} p_{ij}(c) \log \hat{p}_{ij}(c) \\ & + (1 - p_{ij}(c)) \log(1 - \hat{p}_{ij}(c))] \end{aligned} \tag{VIII.2}$$

where S is the size of the respective output tensor, C is the set of damage classes, and $B = 3$ is the number of anchors per grid cell. The standard way is to use one-hot encoded labels. The hyperparameter w_{cls} adds a weight to positive examples enabling to trade off the per-class precision and recall.

For BB regression loss, the default approach is to use Complete IoU (CIoU) [23], which considers three geometric factors to compare BBs, namely the overlap area (i.e., IoU), the distance between the centers, and the aspect ratios:

$$L_{BB} = \sum_{i=1}^{S^2}\sum_{j=1}^{B} I_{ij}^{obj} (1 - CIoU_{BB_{pred}}^{BB_{label}}) \tag{VIII.3}$$

In contrast to classification and BB loss, the objectness loss incurs for all grid locations and BB predictions:

$$L_{obj} = -\sum_{i=1}^{S^2}\sum_{j=1}^{B}[w_{obj}c_{ij}\log\hat{c}_{ij} + (1-c_{ij})\log(1-\hat{c}_{ij})] \tag{VIII.4}$$

where the hyperparameter $w_{obj} \in \mathbb{R}$ is a weighting factor for positive examples, now trading off precision and recall related to the existence of an object at the respective location.

If the predicted BB is not similar to the ground truth (measured with an IoU-based metric), the confidence should also be lower. Thus, the labels for the objectness are:

$$c_{ij} = \begin{cases} CIoU_{pred_{ij}}^{obj_{ij}} & \text{if } I_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{VIII.5}$$

The total loss to optimize the parameters is given by:

$$L_{yolo} = \gamma_{cls}L_{cls} + \gamma_{obj}L_{obj} + \gamma_{BB}L_{BB} \tag{VIII.6}$$

where the hyperparameters $\gamma_{cls}, \gamma_{obj}, \gamma_{BB}$ determine the relative importance. At inference time, we apply confidence thresholding to remove BBs not containing an object and Non-Maximum Suppression (NMS) to remove duplicates.

VIII.3.3 Class-conditional VAE-WGAN

Previous work [11, 9] used (C)GANs to augment the data set artificially. In addition to generating highly realistic damages conditioned on a damage type, we want to learn valuable representations for each damage type to use the feature representation for training a severity classifier with limited data. To achieve both objectives, we optimize a hybrid model combining a class-conditional VAE and WGAN, which we name CVAE-WGAN. In the following, x denotes a cropped BB, y the one-hot-encoded damage type labels, and y_{sev} the severity label (i.e., low, medium, or high).

VAEs [24] concurrently optimize a generative model $p_\theta(x|z)p(z) = p_\theta(x,z)$ of the data x and the latent variables z, and a corresponding inference model $q_\phi(z|x)$ [25]. We can additionally include the damage type labels by conditioning the model components on y. The loss function of the CVAE is a pixel-wise reconstruction error and the KL divergence between the inference model and a defined prior distribution:

$$L_{cvae} = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z,y)] + KL[q_\phi(z|x,y)||p(z|y)] \tag{VIII.7}$$

We define the prior $p(z|y)$ to be a standard normal distribution for all label values y, the inference model $q_\phi(z|x,y)$ to be a diagonal Gaussian parameterized by an encoder Neural Net $e_\phi(x,y)$, and the generative model $p_\theta(x|z,y)$ a factorized Bernoulli parameterized by a decoder/generator $g_\theta(z,y)$.

GANs [26] define a generator $g_\theta(z)$ and a discriminator network $d_\psi(x) \in \mathbb{R}$. The generator 'fools' the discriminator (i.e., to generate realistic samples), and the discriminator aims to distinguish between real and fake images to the best of its ability. CGANs [27] conditions the generator and discriminator on extra information y. For learning, generator and discriminator play a minimax game:

$$\begin{aligned}\min_g \max_d V =&\mathbb{E}_{(x,y)\sim p_r(x,y)}[\log d(x,y)] \\ &+ \mathbb{E}_{z\sim p(z),y\sim p(y)}[\log(1-d(g(z,y),y))]\end{aligned} \tag{VIII.8}$$

We define $p(y)$ to be the empirical categorical distribution. The generator $g(z,y)$ implicitly defines a conditional density model $p_g(x|y)$, and is combined with $p(y)$ to get $p_g(x,y)$ [28].

The vanilla CGAN/GAN described above often runs into problems like mode collapse and unstable training in practical settings. These problems are often attributed to the fact that formally the GAN loss function, given an optimal discriminator, is the Jensen-Shannon Divergence between real and generative data distribution $JS[p_r(x,y)||p_g(x,y)]$. To circumvent the problems, the WGAN [29] introduces the Wasserstein distance to measure the dissimilarity. An improvement to WGAN was given by Gulrajani et al. [30] by adding gradient penalization into the objective to ensure K-Lipschitz continuity of the discriminator (or 'critic'):

$$\begin{aligned}\min_g \max_d V = &\mathbb{E}_{x\sim p_r}[d(x)] - \mathbb{E}_{\tilde{x}\sim p_g}[d(\tilde{x})] \\ &- \lambda\mathbb{E}_{\overline{x}\sim p_{\overline{x}}}[(\|\nabla_{\overline{x}} d(\overline{x})\|_2 - 1)^2]\end{aligned} \tag{VIII.9}$$

where λ determines the strength of the constraint, and $\overline{x}$ is obtained by sampling along a straight line between x and $\tilde{x}$.

For our hybrid model, we extend the WGAN objective to include y. The loss function of the critic is therefore:

$$\begin{aligned}L_{dis} =&\mathbb{E}_{(\tilde{x},\tilde{y})\sim p_g}[d(\tilde{x},\tilde{y})] - \mathbb{E}_{(x,y)\sim p_r}[d(x,y)] \\ &+ \lambda\mathbb{E}_{(\overline{x},\overline{y})\sim p_{\overline{x},\overline{y}}}[(\|\nabla_{\overline{x}} d(\overline{x},\overline{y})\|_2 + \|\nabla_{\overline{y}} d(\overline{x},\overline{y})\|_2 - 1)^2]\end{aligned} \tag{VIII.10}$$

We follow Larsen et al. [31] and replace the inappropriate pixel-wise reconstruction loss of the VAE with a reconstruction loss based on learned features of the discriminator:

$$L^f_{rec} = \|d^l_\psi(\hat{x},y) - d^l_\psi(x,y)\|^2_2 \tag{VIII.11}$$

where $d^l_\psi(\hat{x}, y)$ is the output of the lth layer, which we define to be the output of the convolutional feature extractor. The encoder parameters ϕ are learned via the sum of the reconstruction loss and the KL loss. If we use the analytic solution of the Gaussian prior and inference model, we get:

$$\begin{aligned} L_{enc} = \alpha \sum_{j=1}^{D} (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1) \\ + \|d^l_\psi(\hat{x}, y) - d^l_\psi(x, y)\|_2^2 \end{aligned} \tag{VIII.12}$$

where α is a weighting hyperparameter. The loss of generator with parameters θ contains a VAE and GAN component:

$$L_{gen} = \|d^l_\psi(\hat{x}, y) - d^l_\psi(x, y)\|_2^2 - \beta d(\tilde{x}, \tilde{y}) \tag{VIII.13}$$

where β is a weighting hyperparameter. The discriminator with parameters ψ aims to minimize L_{dis}. We train the model similar to WGANs by alternating the critic and the generator training, now also including an encoder (Algorithm 1).

Algorithm 1: CVAE-WGAN training

```
Set hyperparameters $n_{critic}, \alpha, \beta, \lambda$
Initialize the network parameters $\theta$, $\phi$, $\psi$
while termination criterion is not met do
    for $n_{critic}$ iterations do
        Random minibatch of data $(x, y) \sim p_r(x, y)$
        Fake data $(\tilde{z}, \tilde{y}) \sim p(z)p(y)$, $g_\theta(\tilde{z}, \tilde{y}) = \tilde{x}$
        Gradient penalty samples $\epsilon \sim U[0, 1]$, $\overline{x} = \epsilon x + (1 - \epsilon)\tilde{x}$,
          $\overline{y} = \epsilon y + (1 - \epsilon)\tilde{y}$
        Discriminator outputs $d_\psi(x, y)$, $d_\psi(\tilde{x}, \tilde{y})$
        Update parameters based on $\nabla_\psi L_{dis}$
    end
    Random minibatch of data $(x, y) \sim p_r(x, y)$
    Fake data $(\tilde{z}, \tilde{y}) \sim p(z)p(y)$, $g_\theta(\tilde{z}, \tilde{y}) = \tilde{x}$
    Reconstruct the data $e_\phi(x, y) = (\mu, \sigma)$,
      $z \sim q(z|x, y) = \mathcal{N}(z; \mu, diag(\sigma))$, $\hat{x} = g_\theta(z, y)$
    Calculate $d^l_\psi(\hat{x}, y)$, $d^l_\psi(x, y)$, $d_\psi(\tilde{x}, \tilde{y})$
    Update parameters based on $\nabla_\theta L_{gen}$
    Update parameters based on $\nabla_\phi L_{enc}$
end
```

After model training, we test the applicability in damage severity classification with 120 subjectively labeled pothole examples. We freeze the encoder

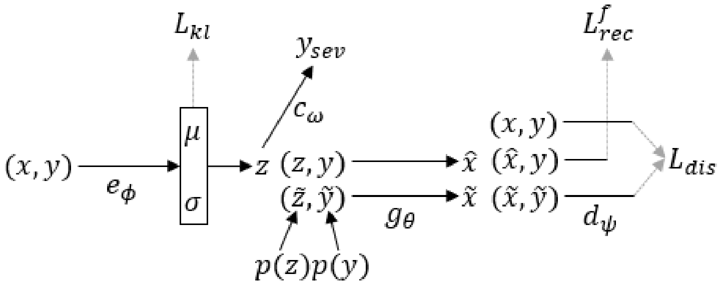


Figure VIII.3: Flow through CVAE-WGAN with severity classifier

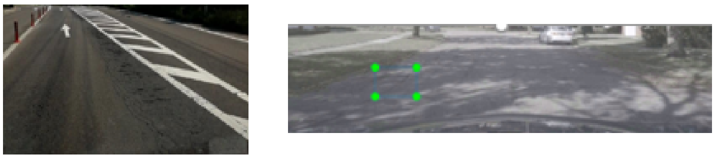


Figure VIII.4: Road occupies the majority of the pixels by cropping

feature extraction and extend the CVAE-WGAN by a small, fully connected classifier $c_\omega(z)$. This semi-supervised approach is inspired by the M1 model in [32]. Figure VIII.3 summarizes this section by showing the data flow through the CVAE-WGAN with severity classifier.

VIII.3.4 Bag of Presents

This section describes methods that can boost damage detection accuracy without altering the inference time. We call those Bag of Presents (BoPs, overview in Table VIII.1) as we do not have to 'pay' inference cost to get better.

Previous work segmented the road area with DL. However, this significantly increases inference costs due to sequential processing through two DL models. We instead consider image cropping to focus on the road area. The transformations are dataset-dependent. While the Czech and AD images maintain a constant view of the street, we must account for different angles for Japan and India. Examples are given in Figure VIII.4. Additional training data can be generated using the trained CVAE-WGAN as follows:

1. Create artificial damage $\tilde{x}$ with label y by $z \sim p(z)$, using the trained generator $g_{\theta*}(z, y) = \tilde{x}$.

2. Sample a training image that does not contain damages.

3. Sample a road pixel location and a damage size.

Table VIII.1: Bag of Presents

Influence	**Method**
Training Data	Random photometric transforms
	Random geometric transforms
	Mosaic data augmentation [33]
	Image cropping for road focus
	Additional synthetic data
	Additional data from related dataset
Objective Function	Class weighting
	GIoU instead of CIoU loss
	Label smoothing
Weight Initialization	Transfer Learning / Fine Tuning
Anchors	Optimized Anchors [34]
Training Algorithm	Image weighting
	Multi-scale training [34]

4. Set $\tilde{x}$ onto the road pixel location with the 'seamlessClone' function from OpenCV.

We also create new training data by leveraging related datasets. We crop the BBs of the longitudinal, transverse, alligator cracks, and potholes from the Pavement Image Dataset [35] containing Google Streetview images and apply the same procedure as described before, excluding step two.

We further consider class weighting, which changes the classification loss L_{cls} (VIII.2). Instead of giving equal importance to all classes, we apply a weighted sum over the classes in the loss function, and each weight is proportional to the inverse frequency of a particular class. Class weighting is a common technique to attenuate class imbalance problems [36]. Class label smoothing [37] is a regularization technique that turns hard one-hot encoded class labels into soft labels. We also investigate whether using GIoU loss [38] instead of the default CIoU is superior for BB regression. Instead of weighting each image equally during training, image weighting samples data proportional to the inverse performance of a class to the validation result of the previous epoch. We follow Jocher's implementation (in his Github repository).

VIII.3.5 Bag of Investments

This section describes methods that usually increase accuracy but are accompanied by an increase in inference cost. We call those methods Bag of Investments (BoIs), as we have to 'invest' inference speed. Whether an investment is worthwhile depends on the use case. The inference cost (Table VIII.2) is the additional inference time (in ms) on a GTX 1080 Ti GPU (batch size 1) by using the respective BoI.

While the Mish activation function is default for YOLOv4-CSP, YOLOv4-Tiny applies LeakyReLU, as Mish takes more time per epoch. However, a CUDA-based implementation in PyTorch increases inference cost only slightly (about 1.1x for YOLOv4-Tiny). A trained model works well on multiple input image sizes at inference time when trained with multi-scale training. A larger input image size can increase accuracy but decrease inference speed (about 1.25x for YOLOv4-CSP increasing from 512 to 640). TTA involves augmenting the test image several times (left-right flip and three different resolutions) and processing each augmentation through the network before merging the prediction. The inference cost increases by about 2.7x with YOLOv4-CSP.

Table VIII.2: Bag of Investments

Description	**Inference Cost**
Mish Activation instead of LekyReLu	+
Input Image Upscaling	++
Test Time Augmentation (TTA)	++++

VIII.3.6 Implementation Details

We split the CD and AD into 80% training and 20% validation set. We adopt SGD with Nesterov momentum [39] as our optimizer for all road damage detection experiments. We also experimented with the adaptive optimizer Adam, causing performance reduction. We use TL on MS COCO and Optimized Anchors, following the described workflow:

1. Train baseline models for YOLOv4-Tiny (50 epochs) and YOLOv4-CSP (20 epochs) with default settings.
2. Conduct preliminary experiments relative to the baseline to verify/deny Mish, Mosaic, and image cropping.
3. Conduct random search [40] for 150 iterations, followed by genetic algorithm search [41] for 150 iterations. We did ten epochs per iteration for Tiny and five for CSP.
4. Verify/deny the addition of (synthetic) data (BoPs).
5. Test final model on validation server (test1, test2).

The search algorithms (step three) are executed to find the optimal hyperparameter configurations measured by the mAP on the validation set. While increasing the accuracy of the damage detection models, the algorithms do not negatively impact the inference cost. However, we only train for a small number of epochs per iteration to limit the execution time of the search algorithms, assuming that performance after ten/five epochs correlates with overall performance. Random search involves all objective function (c_*, γ_*, T), Momentum-SGD, BoI, and BoP hyperparameters, except the ones explicitly named above. Genetic search requires float values; thus, we fix all binary hyperparameters based on the preceding steps. The choice of 150 iterations is reasonable because we observed a significant drop in marginal utility as the number of iterations increased above 75. To train an automotive expert model, we compare three approaches:

1. Train a baseline with the AD by TL from MS COCO.
2. Train a base model by using the CD and the AD. Then, build an automotive expert only using the AD and TL from the base model.
3. Same as step two but with cropping for road focus.

Training an expert model by finetuning a base model trained with all related data is inspired by Kortmann et al. [42].

The CVAE-WGAN (Algorithm 1) is trained with $\alpha = 2.5$, $\beta = 10$, $\lambda = 10$, and Adam [43] for 5,000 epochs. The training data are the cropped BBs of the CD and AD. The critic does five steps ($n_{critic} = 5$) per epoch, as it is crucial to perform well for the CVAE-WGAN's overall performance. The encoder and critic are composed of strided convolutional layers for downsampling and take an input image of size 64 × 64 as input. We choose a 64-dimensional latent space, and the generator applies transpose convolutional layers for upsampling. Encoder and generator utilize batch normalization [44], but the critic does not, as recommended in the WGAN papers. For damage classification, we split the samples into 90 training (30 per class) and 30 (10 per class) validation samples. Our classifier c_ω is comprised of two hidden layers with 64/32 units. We also construct a baseline classifier with the same structure that takes x as input instead of z. We train both classifiers for 2,000 epochs with RMSProp.

The CVAE-WGAN with severity classifier was implemented in Python using Tensorflow 2.4.0, and the road damage detection models utilized PyTorch 1.7.1. The training was deployed on an NVIDIA GTX 1080Ti GPU.

VIII.4 Experimental Results and Discussion

VIII.4.1 CVAE-WGAN Results

Our CWGAN-VAE generates damages with high visual fidelity and inter-class variance for every damage class (e.g., Figure VIII.5). We generate 3,500 additional samples for road damage detection with the procedure described in section VIII.3.4 and manually pick 717 realistic-looking ones (e.g., Figure VIII.6 left). We can generate as many damages as we want; however, it requires tedious manual work to assess the generated samples. We additionally create 433 samples using the Streetview images. We call the 1,150 additional samples Generated Data (GD).

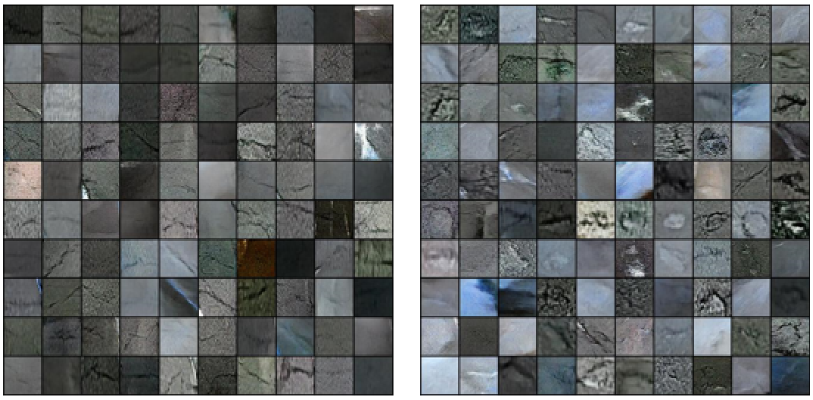


Figure VIII.5: Generated transverse cracks and potholes

We evaluate the severity performance based on the validation set accuracy. The baseline model scores 16/30 (0.53), and the trained classifier $c_{\omega *}$ achieves 25/30 (0.83). To get insight into the latent space, we visualize the encoded mean latent vector of every (real) pothole x. We use t-SNE [45] to transform the 64-dimensional μ-vectors into two dimensions (Figure VIII.6 right). Given a damage type (here: pothole), similar examples (e.g., in terms of severity) are encoded at a similar location, and we get clusters of points. This allows us to find a generalizable classifier even with a small number of severity labels.

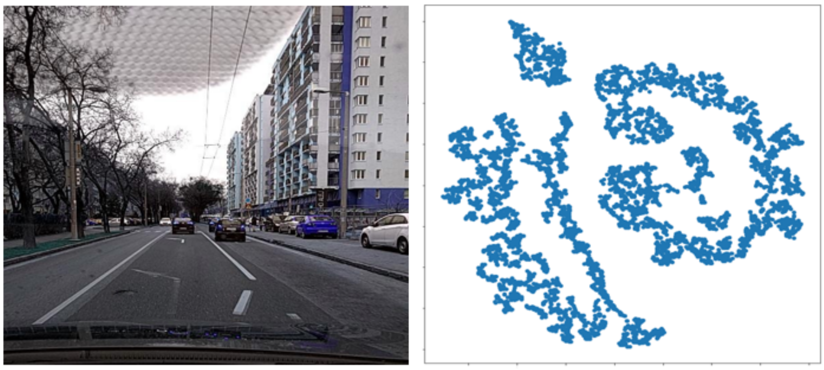


Figure VIII.6: Generated pothole on road / t-SNE potholes

VIII.4.2 Road Damage Detection Results

We use the mAP [46] on the validation set to compare models because it provides precision and recall performance across all confidence levels, whereas the F1 score changes with different confidence levels. Our preliminary experiments for Tiny show a performance increase of about 2.1% relative to the baseline with Mish instead of LeakyReLu. Since the inference cost increase is marginal, we see the investment as worthwhile. Mosaic boosts mAP by about 1.1%, and deterministic image cropping for road focus by about 2.4%. The search algorithms additional verify GIoU instead of CIoU, multi-scale, image weights, label smoothing, and class weights. Regarding YOLOv4-CSP, we confirm Mosaic, image cropping, multi-scale training, GIoU, label smoothing, and image weights. We train YOLOv4-CSP for 20 epochs with a learning rate of 0.0055, YOLOv4-Tiny 80 epochs with a learning rate of 0.0047 to attain the highest validation accuracy. The results are given in the upper half of Table VIII.3. Figure VIII.7 depicts the per-class performances of the best-performing Tiny and CSP model. On the evaluation server, the models obtain F1 scores of 0.552/0.596 (test1) and 0.528/0.580 (test2) for

Table VIII.3: Road damage detection results on validation sets

Training Data			**YOLOv4-CSP**		**-Tiny**	
CD	*GD*	*AD*	*mAP*	*Speed*	*mAP*	*Speed*
✓			50.8/49.0/51.2	27.0/34.0/73.8	**42.0**	5.0
✓	✓		51.0/49.1/51.3		42.1	
✓		✓	51.1/49.9/51.7		41.4	
✓	✓	✓	**51.2**/51.1/51.9		41.3	
		✓[1]	48.7/50.8	23.0/24.1	31.2	4.4
✓[2]		✓[2]	52.6/54.8		33.7	
✓[3]		✓[3]	52.9/54.9	21.9/25.3	36.0	4.1

mAP & speed (in ms) of CSP is given with img-size 512/640/TTA.

[1]Approach 1 [2]Approach 2 [3]Approach 3 (described in Section VIII.3.6)

YOLOv4-Tiny/CSP(512). TTA boosts F1 up to 0.639/0.624 for test1/test2. We also train the three automotive expert models with the same optimized hyperparameters. The results are given in the lower half of Table VIII.3.

VIII.4.3 Discussion

The ranking of the GRDDC 2020 papers can be found in Arya et al. [10]. Compared to the GRDDC 2020 non-ensemble solutions, we achieve first place with the CSP model (average F1 of 0.588 without TTA), and the trained Tiny model keeps pace with the state-of-the-art solutions while being extremely fast. Thus, we can reach cutting-edge accuracies even with diminutive models, which is a valuable contribution to application and practice. Notably, the Tiny model performs comparably worse than CSP for the AD, which could be related to the high number of small objects in the AD. Fixed camera setups in cars enable efficient road pixel focus for increasing accuracy and constant inference speed, while TTA significantly reduces the inference rate, making it unsuitable for real-time processing. We cannot conclusively assess the performance of severity classification due to the small validation dataset, but it is indicated that the learned encodings significantly help find a generalizable decision boundary. However, the model is not yet mature enough to be used in practice. Though, our unique design is a contribution to theory.

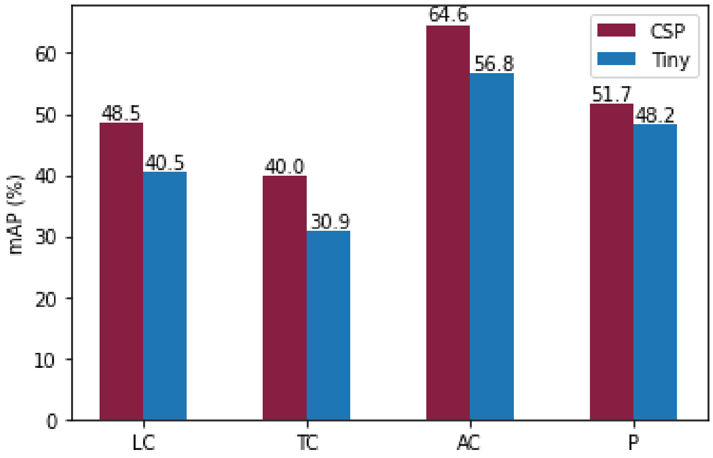


Figure VIII.7: Damage detection results for each damage type

VIII.5 Conclusion and Future Work

Our objective was to enable road damage detection and assessment for vehicles based on single front-facing cameras. We extended previous work based on smartphone cameras by using the tiny and medium-sized Scaled-YOLOv4 architectures and produced resource-efficient and highly accurate detection models that also work with automotive cameras. We pioneered several methods to improve detection accuracy without affecting the speed at inference time. Examples include cropping images for road focus, using related datasets, and creating a generative model called CVAE-WGAN. We opt to extend the accuracy-improving methods for future work and include sequence processing with recurrent nets and sensor fusion. In addition to damage localization and classification, severity assessment is a critical part of road perception. Thus, we complemented the CVAE-WGAN with a severity classifier, working with only a few subjectively labeled images. In the future, we want to develop a more sophisticated semi-supervised model involving expert labels, bird's eye view, damage size, and an extension to more classes.

Bibliography

[1] NHTSA, “Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,” 2015.

[2] H. Winner and S. Hakuli, “Conduct-by-wire–following a new paradigm for driving into the future,” *FISITA World Automotive Congress*, vol. 22, p. 27, 2006.

[3] S. Lekach, “Elon musk promises tesla updates like goat honks, pothole detection,” 2020. Accessed: 2021-07-13.

[4] R. Liu, J. Wang, and B. Zhang, “High definition map for automated driving: Overview and analysis,” *The J. of Navigation*, vol. 73, no. 2, pp. 324–341, 2020.

[5] C. Ng, T. Law, F. Jakarni, and S. Kulanthayan, “Road infrastructure development and economic growth,” in *IOP Conf. Series: Materials Science and Engineering*, vol. 512, p. 012045, IOP Publishing, 2019.

[6] C.-Y. Wang, A. Bochkovskiy, and H. Liao, “Scaled-yolov4: Scaling cross stage partial network,” *arXiv:2011.08036*, 2020.

[7] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road damage detection using deep neural networks with images captured through a smartphone (2018),” *arXiv:1801.09454*, 2018.

[8] A. Angulo, J. A. Vega-Fernández, L. M. Aguilar-Lobo, S. Natraj, and G. Ochoa-Ruiz, “Road damage detection acquisition system based on deep neural networks for physical asset management,” *Mexican Int. Conf. on Artificial Intelligence*, pp. 3–14, 2019.

[9] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, “Generative adversarial network for road damage detection,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2021.

[10] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, “Global road damage detection: State-of-the-art solutions,” *arXiv:2011.08740*, 2020.

[11] X. Zhang, X. Xia, N. Li, M. Lin, J. Song, and N. Ding, “Exploring the tricks for road damage detection with a one-stage detector,” *IEEE Int. Conf. on Big Data*, pp. 5616–5621, 2020.

[12] V. Hegde, D. Trivedi, A. Alfarrarjeh, A. Deepak, S. H. Kim, and C. Shahabi, “Yet another deep learning approach for road damage detection using ensemble learning,” *IEEE Int. Conf. on Big Data*, pp. 5553–5558, 2020.

[13] V. Pham, C. Pham, and T. Dang, “Road damage detection and classification with detectron2 and faster r-cnn,” *arXiv:2010.15021*, 2020.

[14] Z. Pei, R. Lin, X. Zhang, H. Shen, J. Tang, and Y. Yang, “Cfm: A consistency filtering mechanism for road damage detection,” *IEEE Int. Conf. on Big Data*, pp. 5584–5591, 2020.

[15] Y. Liu, X. Zhang, B. Zhang, and Z. Chen, “Deep network for road damage detection,” *IEEE Int. Conf. on Big Data*, pp. 5572–5576, 2020.

[16] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “Cspnet: A new backbone that can enhance learning capability of cnn,” *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition Workshops*, pp. 390–391, 2020.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[19] D. Misra, “Mish: A self regularized non-monotonic activation function,” *arXiv:1908.08681*, 2019.

[20] Y. Lee, J.-w. Hwang, S. Lee, Y. Bae, and J. Park, “An energy and gpu-computation efficient backbone network for real-time object detection,” *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 2117–2125, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *IEEE Int. Conf. on Comp. Vision*, pp. 1026–1034, 2015.

[23] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *AAAI Conf. on Artificial Intelligence*, vol. 34, no. 07, pp. 12993–13000, 2020.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.

[25] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv:1906.02691*, 2019.

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv:1406.2661*, 2014.

[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[28] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project for Stanford CS231N: CNNs for Visual Recognition*, vol. 2014, no. 5, p. 2, 2014.

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," *Int. Conf. on Machine Learning*, pp. 214–223, 2017.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv:1704.00028*, 2017.

[31] A. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *Int. Conf. on Machine Learning*, pp. 1558–1566, 2016.

[32] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," *arXiv:1406.5298*, 2014.

[33] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020.

[34] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 7263–7271, 2017.

[35] H. Majidifard, P. Jin, Y. Adu-Gyamfi, and W. G. Buttlar, "Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses," *Transportation Research Record*, vol. 2674, no. 2, pp. 328–339, 2020.

[36] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," *Int. Conf. on Machine Learning*, pp. 872–881, 2019.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[38] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition*, pp. 658–666, 2019.

[39] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *Int. Conf. on Machine Learning*, pp. 1139–1147, 2013.

[40] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *J. of Machine Learning Research*, vol. 13, no. 2, 2012.

[41] G. Jocher, "Hyperparameter evolution," 2020. Accessed: 2021-05-17.

[42] F. Kortmann, K. Talits, P. Fassmeyer, A. Warnecke, N. Meier, J. Heger, P. Drews, and B. Funk, "Detecting various road damage types in global countries utilizing faster r-cnn," *IEEE Int. Conf. on Big Data*, pp. 5563–5571, 2020.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Int. Conf. on Machine Learning*, pp. 448–456, 2015.

[45] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *J. of Machine Learning Research*, vol. 9, no. 11, 2008.

[46] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. of Comp. Vision*, vol. 88, no. 2, pp. 303–338, 2010.

Chapter IX

The Value of Deep Learning Tools in Object Detection: YOLOv5 in a Road Damage Use Case

Outline

Bibliographic Information

Felix Kortmann, Kevin Talits, Alexander Warnecke, Nicolas Meier, Jens Heger, Paul Drews, and Burkhardt Funk, (**under review**, 2021, October). “The Value of Deep Learning Tools in Object Detection: YOLOv5 in a Road Damage Use Case”. In IEEE Transactions on Intelligent Vehicles. IEEE. Preprint DOI: 10.13140/RG.2.2.23183.30880.

Author’s contribution

The author’s share of the publication is 70%. Table B.9 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

We describe the design process of deep learning models trimmed towards runtime and detection performance for road damage detection using front-facing camera sensors of conventional vehicles. We distinguish between deep learning tools that improve detection performance while suffering on runtime (’Bag of Investments’) and those that will enhance detection performance not increasing runtime (’Bag of Presents’). For our investigations, we utilize the state-of-the-art deep learning framework YOLOv5. Our ultimate goal is to provide autonomous vehicles of varying automation levels with information on upcoming road damages to improve vehicle motion planning. As current road damage detection algorithms demand high computational power, we focus on real-time applicability deployed on edge devices. We analyze isolated tools for the object detection task and proceed with three deep learning models that provide (1) the maximum performance, (2) the fastest runtime, and (3) the optimal trade-off between detection performance and runtime. Our investigations are designed to help researchers and practitioners dealing with other object detection tasks.

IX.1 Introduction

use of autonomous vehicles is about to enter our daily lives. Much of the recent achievements have to do with the application of artificial intelligence in environment recognition. The stated goal of attaining functional safety is currently only reached in limited application scenarios [1], such as on highways in high-tech countries. The autonomous driving task can be divided into two parts: perception system and decision-making system [2]. In environment perception, the road's condition is a crucial measurand [3], however, autonomous and partially automated vehicles do not yet respond to road damages. Responding to a Twitter users question, Elon Musk, CEO of Tesla, Inc., states: "We're labeling bumps & potholes, so the car can slow down or steer around them when safe to do so". In the USA alone, claims amounting to $217 billion per year are connected to poor road conditions. 42,000 deaths occur each year, with road conditions being the cause or at least a factor in 52 % of the cases [4]. The infrastructure of a country is considered to be of considerable importance in terms of its social and economic significance [5]. Cameras of conventional vehicles thus have the potential to, on the one hand, deliver real-time data for the vehicle's motion system and, on the other hand, to provide authorities responsible for road conditions with transparency to improve infrastructure repairs.

The topic is of social relevance due to autonomous driving's great potential to change mobility, for example by improving shared mobility solutions and using driving time for alternative activities. Successful detection of road damage therefore plays a major role in achieving automation especially in developing countries with poor road conditions.

The research field of CV is being revolutionized since 2012 with the introduction of the CNN AlexNet [6] and is currently experiencing a boom in science and multiple application domains. Key was the superior performance of AlexNet in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in comparison to previous state-of-the-art machine learning approaches for object recognition [7]. Despite previous achievements such as the introduction of the backpropagation feature in 1985 [8], visual pattern recognition CNNs in 1988 [9], and the Hinton Lab proposing a training method for DNNs [10, 11] in 2006, AlexNet represents the breakthrough in technology and the starting point for immense research activities to advance the development of DNNs.

Our research objective is twofold: (1) We train a DL algorithm that is as optimal and balanced as possible for our domain. Images from front-facing smartphone and automotive cameras in vehicles are being used. The detection process in vehicles needs to be performed at a high processing speed, preferably at the frame rate of the camera sensor (30 fps), and cars often use embedded devices with limited processing power. (2) Many application domains require CV operations such as image classification and object detection. The successful application of DL algorithms is strongly dependent on the type of NN and the tools used to develop the algorithm though. In the object detection task, this concerns for example not only the detection performance, but also the operability on computers or edge devices with regard to the runtime. To this end, we exploit the possibilities of the cutting-edge DL framework YOLOv5 [12] in order to provide users with guidance as to which application of tools can be expected to produce improved results. However, many tools that are suitable for increasing the performance also increase the computational effort. Computing power is a limited resource in many application fields, which is why we divide the tools that reduce the inference time ('Bag of Investments') and those that do not affect the inference time ('Bag of Presents') in two groups. Our contributions can be summarized as follows:

- We deliver an overview of multiple tools in DL and their isolated impact on model performance and model runtime by utilizing the YOLOv5 framework.
- We develop design principles for models aiming on maximum performance, maximum speed and the best trade-off between both (Sweet-Spot).

IX.2 Related Work

In this section, we provide an overview of research activities carried out in the area of automated road damage detection, primarily utilizing camera sensors.

The most recent major research initiative in this field is the GRDDC 2020 [13, 14], which is part of the IEEE International Conference on Big Data 2020. 121 teams have built an automated road damage detection system based on data gathered in the Czech Republic, India, and Japan. The motivation for the challenge is to support local infrastructure agencies with automated road damage assessment, thus reducing tedious and expensive manual labor. The offered dataset includes images from front-facing smartphone cameras, mounted on the windshields of vehicles. The objective of the challenge is to obtain the best possible object detection performance for two sets of test data.

The CDS consists of 21,041 images, the test data has 5,295 samples. The labels of the test set were not released, and the comparison of the results was executed via an evaluation server, which is still accessible at the time of writing. Four damage classes are considered: longitudinal cracks, transverse cracks, alligator cracks and potholes. The twelve best scoring teams were invited to submit a paper. The algorithms and tools used by these teams are summarized in Table IX.1. The table additionally contains our assessment of a feasible application in vehicles in terms of the computing capacity requirements of the individual solutions.

All winning participants used TL based on either MS COCO or ImageNet, exclusive Hascoet et al. [15]. Since the only metric of the challenge was detection performance over the F1 score, most of the better teams applied ensemble models, which includes #1, #2, #3 and #6. The results also show that YOLO-based models are superior to Faster RCCN models (#9-#12), regardless of the YOLO version (#1, #2, #4-#6 and #8). Another finding of the challenge is that expert models from different countries can help to improve detection performance. However, this requires a prior country classifier (#4, #10, #12). The application of TTA is inconsistent, in #1 resulting in an upgrade, in #11 in a downgrade. Likewise, different results occur when applying segmentation to focus on the road surface, #3 and #6 reporting an upgrade, #10 a downgrade.

Between the assessment of road damages via DL models, previous work focused on image processing techniques such as histograms. Bello-Salauet al. [16] delivered a survey on this topic in 2015. Whereas Koch et al. [17] focus on different components of road conditions as road pavement analysis, defect analysis and the process of automation during assessment. They offer a comprehensive survey including an evaluation of the present status in 2015 and future research. Cao et al. [18] in contrast survey the ML-based road defect detection methods, primarily with hand-crafted features in 2020. The latter two highlight the insufficient real-time capability of the methods surveyed in their paper [18, 17]. The topic of road condition assessment and interpretation via indices and metrics is very well summarized in Sayers' *Little Book of Profiling* [19] but is not subject to this paper.

In contrast to the GRDDC, we discuss the impact of individual tools on detection performance and model runtime on two different hardware systems. Furthermore, we are labeling another dataset of automotive-quality images to optimize the applicability for autonomous vehicles. Based on our results, we propose three different models with the respective goals of maximum detection performance, minimum runtime, and a Sweet-Spot model that provides the best trade-off.

Table IX.1: GRDDC 2020 DL models and our estimate for their real-time application in the vehicle

Source	Solution	Applicability
#1 [20]	YOLOv5 Ensemble with TTA	W/ powerful GPU & w/o Ensembles - inefficient
#2 [21]	YOLOv4 Ensemble	W/ powerful GPU & w/o Ensembles - inefficient
#3 [22]	Ensemble with cascade R-CNN	No - optimized in detection performance
#4 [23]	YOLO on CSPDarknet53	W/ powerful GPU - non-efficient
#5 [24]	YOLOv5x	W/ powerful GPU - still non-efficient
#6 [25]	YOLOv4 & Faster R-CNN Ensemble	W/ powerful GPU & w/o F R-CNN - inefficient
#7 [26]	EfficientDet	The small model (20 fps, F1:0.521)
#8 [27]	YOLOv4 and Conditional GAN	W/ powerful GPU - inefficient
#9 [15]	Faster R-CNN with ResNet-101	No - optimized in detection performance
#10 [28]	Multi-stage Faster R-CNN	No - optimized in detection performance
#11 [29]	Detectron2 and Faster R-CNN	No - optimized in detection performance
#12 [30]	Faster R-CNN & Regional Experts	No - optimized in detection performance

IX.3 Experimental Setup

IX.3.1 Methodical Approach

Our research approach is inspired by Hevner's Design Science Research [31, 32]. The research paradigm is driven by the desire to improve the application environment by introducing novel artifacts and the process of building them. The research approach includes the application domain, which is necessary for assessing the relevance of the research subject. Furthermore, the knowledge base is considered as an instance. We build on existing knowledge, such as the DL framework YOLOv5, and solve the problem with existing tools and, if necessary, adapt the tools according to the requirements.

We have applied and combined various tools in DL training with regard to resource requirements and performance in many iterations. Our first goal, of course, is to develop the best possible DL model for solving the detection problem of road damages for autonomous vehicles. In addition, we investigate the isolated influence of individual DL tools on detection performance and runtime. We want to deal with the complexity in DL in order to achieve satisfying results, depending on the need for real-time capability and detection performance. Finally, we develop three algorithms with different goals: one algorithm with maximum performance, one with maximum speed, and one that, according to our results, provides a good trade-off between power and runtime (Sweet-Spot). We are convinced that through our evaluation of the tools, our quantitative results and design principles we are able to positively influence the design process of DL models of other scientists.

IX.3.2 DL Datasets

Our training data consists of two separate datasets. The first is the RDD-2020 dataset offered in the GRDDC 2020 and consisting of 21,041 labeled images from front-facing smartphone cameras sensors in the inner cabin of the vehicle, mounted at the windshield. The images are from Japan, India, and the Czech Republic. The RDD is labeled regarding longitudinal cracks (D00), lateral cracks (D10), alligator cracks (D20) and potholes (D40). The majority of the images have an image size of 600 px × 600 px, whereas those from India have an image size of 720 px × 720 px.

The second dataset is named ADS. HELLA Aglaia Mobile Vision GmbH has entrusted us the data for the purpose of RDD. We chose the locations of the dataset subjectively to represent different road conditions in other countries to increase the generalization of the algorithms in the long run. It contains 10,421 images from Germany (40 %), the USA (28 %), the United Kingdom (10 %), South Korea (10 %), Poland (4 %), France (3 %), Latvia (3 %) and Finland (2 %). Besides being country-specific, the images also represent night drives, snow and rain situations. The images are obtained from identical measuring vehicles and the identical camera systems across the continents and countries. To suit the requirements of our use case, HELLA Aglaia Mobile Vision GmbH used an automotive-grade camera from Sony Corporation with an image size of 3504 px × 1072 px.

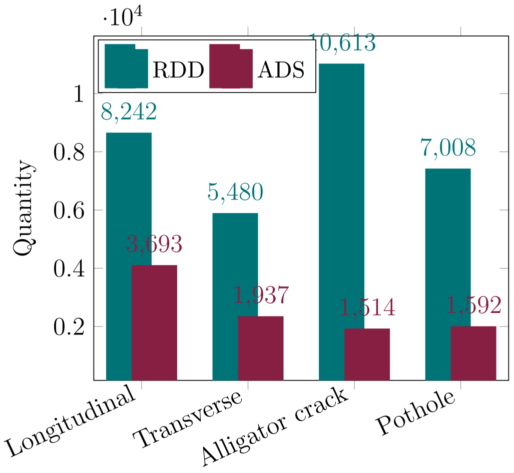


Figure IX.1: The number of damages in the classes D00, D10, D20 and D40 in the two datasets RDD and ADS

The labeling of the images was done by ourselves. Due to our in-depth work with theRDD-2020 and several years of research in the field of RDD, we can label the data on an educated basis. We applied the same labels D00, D10, D20, and D40 to be consistent with the RDD-2020. In comparison, the datasets are of significantly different quality. This includes a superior viewing angle, image quality and resolution in the ADS to account for the application in autonomous driving.

However, we were aware from the GRDDC results, that construction joints are often misinterpreted as longitudinal or transverse cracks and respectively manhole covers as potholes which is why we labeled 10677 construction joints and 2624 manhole covers for further investigation in the ADS.

IX.3.3 Evaluation Metric

Object detection models are commonly evaluated utilizing the mAP metric [33]. We use the mAP over the F1 score since it includes a range of confidence scores. This allows us to include the detection performance of the algorithm over the entire precision-recall curve instead of choosing a selective point. The IoU threshold is set to 0.5. To compare our models to the state-of-the-art solutions of the GRDDC 2020 [14], we provide F1-measures in conjunction.

IX.3.4 YOLO Deep Learning Algorithm

Our utilized DL algorithm for the object detection task is YOLOv5 [12], which is an extension of the YOLOv3 [34] PyTorch repository by Glenn Jocher.

Object detection is a subarea of computer vision. It deals with detecting objects of a certain class in images and videos. Object detectors generate features from an input image and pass them through a prediction system to obtain bounding boxes and classes. Methods for this tasks fall into two categories. Non-neural and neural network-based approaches. Non-neural methods need hand-crafted features before classifying objects with techniques such as a SVM [35] while neural network-based methods such as YOLO, are able to perform the detection end-to-end without the need of hand-crafted features, and they are typically based on CNN [36].

The three main parts of YOLO networks are the backbone, neck and head. The *backbone* network aggregates and crafts features based on the input images. In the *neck*, these properties are mixed and combined to be passed on to the *head*. The *head* takes these features and determines the bounding boxes, as well as the class predictions.

A Cross Stage Partial Network (CSPN) is used as the *Backbone* network. This structure addresses gradient problems and requires less parameters and fewer Floating Point Operations Per Second (FLOPS) for comparable accuracy to other state-of-the-art approaches such as DenseNet [37] and ResNet [38, 39]. Based on DenseNet, it alleviates the problem of vanishing gradients and boosts feature propagation. To prevent computational bottlenecks and improve learning, the feature map is passed to the network and an unprocessed copy is forwarded to the next stage.

A Path Aggregation Network (PANet) is used as a *neck*. As the image passes through the different layers, the complexity of the features increases and their spatial resolution decreases. In contrast to the previously used FPN, where just a top-down pathway is used to combine the features from the backbone, the PANet introduces an extra bottom-up path as a shortcut. In the FPN, the fine-grained features have a long way to go through all layers of the *backbone*, but PANet can provide a faster algorithm. In addition, PANet fuses all feature layers of the *backbone* with RoI and fully connected layers to be used for prediction [40].

The *head* of YOLOv5 is similar to the *head* of YOLOv3, but the loss function is implemented with GIoU-loss [41]. This process is anchor-based with three levels of granularity (stride of 8, 16 and 32). To predict objects in different sizes and aspect ratios, YOLO applies anchor boxes and adjusts them with predicted offsets to fit the detection. Consecutively, a loss is calculated from the overlap of the candidates with the real objects. With sufficient overlapping, a prediction is passed into the loss. By penalizing and rewarding predicted boxes, the model becomes more confident in localizing objects. Stride is a component of the CNN for the compression of images. It defines the movement of the filter over the image for every step. In the latest release of the repository, a head with four levels of granularity was introduced, with an additional stride of 64, trained on larger input sizes.

Sigmoid Linear Units (SiLU) [42] is used as an activation function in the middle layers and sigmoid in the final detection layer. The sigmoid function restricts its output between zero and one and the SiLU is superior in activation of neural networks over ReLU [42].

Hyperparameters (e.g. batch size, learning rate, augmentations) can be configured in a separate file. For all layers, certain arguments have to be configured. The layer index from which layer the output is taken as input is passed. Finally, the required arguments for this module are entered, starting with the number of channels. The depth-gain (gd) affects all layers that have a number n greater than one.

$$n = \mathrm{Max}(\mathrm{round}(n \cdot gd), 1) \quad \text{if } n > 1 \tag{IX.1}$$

The width-multiple, or width-gain gw, affects the number of channels ch, if the layer is not an output layer. As the first argument for the convolution layers in the .yaml file, the number of channels is passed.

$$ch = \mathrm{ceil}(ch \cdot gw/8) \cdot 8 \tag{IX.2}$$

With the depths and width-multiple, different versions of the basic structure were created. From YOLOv5s to YOLOv5x becoming deeper and wider as displayed in Table IX.2.

Table IX.2: Performance and Speed of pretrained YOLOv5 models evaluated on Common Objects in Context (COCO) with a batch-size of 32 [12]

Model	**mAP@0.5**	**Runtime (ms)**	**depth-gain**	**width-gain**
YOLOv5s	55.4	2.0 ms	0.33	0.5
YOLOv5m	63.1	2.7 ms	0.67	0.75
YOLOv5l	66.9	3.8 ms	1	1
YOLOv5x	68.8	6.1 ms	1.33	1.25
	With Extra Detection Head (Stride: 64)			
YOLOv5s6	61.9	4.3 ms	0.33	0.5
YOLOv5m6	68.7	8.4 ms	0.67	0.75
YOLOv5l6	71.1	12.3 ms	1	1
YOLOv5x6	72.0	22.4 ms	1.33	1.25

In the *training process*, the augmentation of base data is used to provide the model with a broader semantic variation. Data augmentation techniques are scaling, color space adaptation and mosaic augmentation. An augmented dataset is more robust against overfitting [43]. Some feature removal techniques have been proposed to improve the generalization and localization of the model. These include methods such as regional dropout [44], where parts of the image are removed and filled with zeros or noise. This allows a CNN to focus on the entire object region rather than just the most discriminate features [44]. CutMix advances this technique by filling the gaps from the input images with patches from other images and adjusting the ground-truth label accordingly. YOLOv5 utilizes a new variant of CutMix, called mosaic data augmentation. It combines four training images into one. This allows the model to learn how to identify objects at a smaller scale than normal.

It also encourages the model to localize different types of images in different parts of the frame.

Similarly, the precision of the model and detection of small objects benefits from automatically adjusted anchor box sizes. In YOLOv5, based on the distribution of ground truth boxes in a dataset, k-means clustering is used to adjust the anchor box settings. The loss is calculated via GIoU and for classes and object loss using the Binary Cross-Entropy (BCE) with Logits loss from Pytorch [45]. This leads to a numerically more stable solution. The total loss is the sum of all three losses.

$$L = L_{GIoU} + L_{cls} + L_{obj} \tag{IX.3}$$

The GIoU [41] is an improvement from the simple IoU, which compares the similarity of two boxes. While the IoU is always zero for non-overlapping bounding boxes, the GIoU includes the empty space between the two boxes. Thus, the loss increases for boxes with a larger distance. As optimization algorithms Adam [46] and SGD [47] with momentum and cosine learning rate decay are available. Gradient descent minimizes an objective function $Q(w)$, parameterized by a model's parameters w, by updating the parameters in the opposite direction of the gradient of the objective function. SGD updates the parameters for each training example, with a learning rate η.

$$w = w - \eta \nabla Q_i(w) \tag{IX.4}$$

Frequent updates make SGD fast, but leave the objective function fluctuate heavily. Fluctuation can be beneficial, because it enables it to jump to potentially better local minima, but it complicates convergence to the exact minimum. With a slowly decreasing learning rate, the converging to a local or global minimum is almost certain [47]. When the surface is curved steeply in one dimension, than in another, SGD has difficulties to navigate to the local optima. Momentum [48] accelerates in the relevant direction and reduces oscillations. It remembers Δw at each iteration. The next update is calculated by a linear combination of the previous update with the gradient.

$$\Delta w = \alpha \Delta w - \eta \nabla Q_i(w) \tag{IX.5}$$

$$w = w + \Delta w \tag{IX.6}$$

$$\Rightarrow w = w - \eta \nabla Q_i(w) + \alpha \Delta w \tag{IX.7}$$

Convergence can be achieved faster while oscillation reduces [47]. Adam is storing exponentially decaying averages of past squared gradients v_t and of the gradients m_t.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{IX.8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{IX.9}$$

As the estimates of the mean m_t and the uncentered variance v_t are biased towards zero, a bias-corrected mean and uncentered variance estimate is computed.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad , \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{IX.10}$$

This yields the Adam update rule.

$$w = w + \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t \tag{IX.11}$$

$\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ are the proposed default values [46]. ϵ is used to prevent a division by zero and β_1 and β_2 are the factors for the mean and uncentered variance, respectively. It compares favorably to other adaptive learning-method algorithms and works well with noisy/sparse data [46].

IX.3.5 DL Tools

In this subsection, we elaborate on the DL tools used in our work. These are tools that contribute to improvements in the detection performance as well as tools that can contribute to a desirable reduction of the model runtime.

We distinguish the applied tools into those that can be expected to deliver improved detection performance without a loss in runtime (we call them 'presents') and those for which a worsening of the runtime for a gain in detection performance is to be expected (we call those 'investments').

Our analysis is carried out on two hardware configurations. The first one is our computer which is used for training and testing, it has a GTX 1080 Ti. The second hardware is representative for the application in the vehicle. It is the Jetson Nano edge device from Nvidia. All our analyses are conducted with Python 3.8.5, PyTorch 1.7.1, CUDA 11.0, and cuDNN v8.0.4.

Bag of Presents

Hyperparameter Evolution: As one of the few adjusting screws that can be modified before training, hyperparameters are a factor to be considered . Due to their large number and unknown correlation, it is difficult to impossible to set optimal starting parameters, which is why Hyperparameter Evolution is a suitable method. In contrast to a grid search, a genetic algorithm is a suitable candidate to overcome this difficulty and find the best possible starting parameters. Genetic algorithms are used to generate top results for optimization problems. Mutation, selection and crossover are the main biologically inspired operators used [49]. Starting from a set of hyperparameters, an iterative process is applied, where every step is called a generation. In each generation the performance of the current setup is evaluated. More fit parameters are stochastically selected and are modified according to above-mentioned operators. After a maximum number of 300 generations the algorithm terminates. In YOLOv5 each hyperparameter is bound by a lower and upper limit while mutating and can be excluded from mutating by giving it a gain of zero. Every parameter is mutated with a probability of 80 %. If a random gerated number r_1 between zero and one is below 0.8, the gain g multiplies with a random number, generated by a standard normal distribution $\mathcal{N}$, a random number r_2 between zero and one and a sigma σ of 0.2. After this a one is added and everything is clipped to a range of 0.3 to 3. We receive a new gain-vector v and multiply it with the hyperparameter values [12].

$$v_i = (g_i \cdot \mathcal{N} \cdot r_2 \cdot \sigma + 1).\text{clip}(0.3, 3) \quad \text{if } r_1 < 0.8 \tag{IX.12}$$

$$v_i = 1 \quad \text{else} \tag{IX.13}$$

All results are recorded in a text file and the best offspring is saved after each generation.

New Data: One of the most essential aspects of deep learning is a large and well-labeled dataset [50]. To extend the dataset from the Road Damage Detection Challenge, new data were evaluated and labeled based on the damage classes in the challenge (c.f., Section IX.3.2). Dealing with poor labels is a huge problem in Deep Learning. It is fair to argue that a DL algorithm can only become as good as the label of the dataset allows it to be. A workaround for new data is the use of data augmentation to artificially extend the dataset. The impact of our labeled ADS is explained in Section IX.4.1.

Due to the error potential and miss-classification of construction joints (D01, D11), manhole cover (D50) and street drains (D51) all these objects are labeled in the extended dataset ADS$_\text{P}$. This data can not be merged with the set from the GRDDC, because these labels are not present in it and serves only for analysing the impact of extended labels.

Floating Point Format: By default, a model is loaded at single-precision and used in single-precision floating-point format (FP32). FP32 is the representation of floating-point numbers with 32 bit. half-precision floating-point format (FP16) reserves only 16 bits of computer memory, being a Single-Precision Floating-Point format. Different hardware works with varying FLOPS at FP16 and FP32. The GPU from the Jetson can work almost twice as fast with up to 1 TFLOPS at half precision, compared to single precision. In contrast, the processing power from the GTX 1080 Ti slows down when switching to the FP16 representation.

Bag of Investments

Test Time Augmentation: Usually, DA techniques are utilized to deal with a dataset that is too small to deliver a satisfactory generalization and robustness. Yet it can be beneficial to be used during test-time as well to increase robustness and accuracy. In TTA, the predictions are pooled from various transformed versions of one test image to achieve an averaged prediction [51]. All the predictions are rescaled to fit the original image size and orientation, and concatenated to be returned [12]. This improvement in performance costs inference time because the images would have to be augmented before detection and multiple instances of the same image pass through the network. The benefits of TTA decrease with an increasing size of available training data, with more training data a model is more invariant to augmentations. Most rewarding is TTA with limited training data [51]. In the YOLOv5 repository, one test image is forwarded through the network in the original version, a left-right flipped and scaled down version by a factor of 0.83 and a scaled down version by a factor of 0.67.

Input Image Size: A smaller image size reduces the computational load, since image pixels have to be aggregated, no matter which method is used. A dataset with many small objects can benefit from training in higher resolutions. For best results, the YOLOv5 model should be trained and tested with the same image size [12]. Our results on this can be reviewed later (c.f., Table IX.4).

Model Size: In general, a larger YOLOv5 model size provides better results, with a longer inference time as shown in Table IX.2. When the speed of detection becomes important, a good compromise between performance and inference time must be found. With the method of pruning, a specified percentage of weights are set to zero when testing a model. The selection is done using an unstructured L1-norm [52], the units with the weakest L1-norm are pruned. This allows estimating how many parameters could be omitted before a significant drop in performance occurs.

YOLOv5 offers different model sizes from scratch, which is superior over pruning a larger model. Table IX.3 displays different model sizes including their depth and width-gain.

Head Modification: Adjusting the head of the model to the specific needs of the objects to be detected has great potential of increasing the performance, as stated in the latest release of the GitHub repository and Table IX.2. The reason for the strong improvements is the addition of stride 64 to the

Table IX.3: Depth- and width-gain for our YOLOv5 models

Model Size	**depth-gain**	**width-gain**
YOLO5xxxs	0.001	0.125
YOLOv5xxs	0.165	0.375
YOLOv5xs	0.22	0.417
YOLOv5xl	1.66	1.5

head. This benefits large images the most and the model becomes bigger and requires more memory. The same approach can be used to create a layer with a smaller stride of 4. This would not make the model much bigger, but the convolution must be applied over a larger, denser grid. Witch such a small stride it improves detection for small images but needs significantly more FLOPS and increases the inference time. Furthermore, one can reduce the detection head of the model, by removing one of its layers.

Ensembles: Composite models consisting of several base models used to obtain a single outcome are called ensembles. The models use diverse training data or various training settings, as long as the base models are independent and diverse enough, the ensemble will potentially reduce the prediction error [53, 54]. Even with many base models, an ensemble acts and performs like a single model. With more and more models, the performance increases, but the detection speed of the ensemble decreases significantly with increasing numbers of base models.

IX.4 Results & Discussion

In this chapter, we address the influence of the tools described in Section IX.3.5 on performance and runtime. The isolated influence of all tools is presented in Table IX.4. For reasons of comparison, the top row shows a basic YOLOv5 model without additional tools with the properties indicated. First, the tools in the bag of presents are discussed, followed by those in the bag of investments.

IX.4.1 Bag of Presents Impact

Hyperparameter Evolution Impact: As the first tool in the Bag of Presents, HE can be applied in the training process. The tool is very time consuming

Table IX.4: Impact of single tools on the runtime and performance of YOLOv5s

ID	**Tools**	**Tool Properties**	$\mathbf{Rt}_{\text{Jetson}}$	$\mathbf{Rt}_{\text{GTX}}$	**mAP**
B	Nothing	YOLOv5s, 140 Epochs, 640 px input	320 ms	11.2 ms	0.486
		Bag of Presents			
P1.1	Hyper. Evo. (HE)	10 Epochs, 300 Generations	320 ms	11.2 ms	0.467
P1.2		40 Epochs, 50 Generations	320 ms	11.2 ms	0.499
P2	New Data	Aglaia Data	320 ms	11.2 ms	0.494
P3	Floating Point For.	16 Bit representation	231 ms	11.3 ms	0.485
		Bag of Investments			
I1	T. T. Aug. (TTA)	Scale[1, 0.83, 0.67] Flip[/, left-right, /]	775 ms	28 ms	0.504
I2.1	Input Size	Smaller Input Size 416 px	150 ms	9.3 ms	0.471
I2.2		Larger Input Size 864 px	551 ms	13 ms	0.482
I3.1	Model Size	Smaller Model Size XXS	310 ms	9.3 ms	0.446
I3.2		Larger Model Size XL	3510 ms	92 ms	0.511
I4.1	Head Modification	Extra stride 64	338 ms	15 ms	0.467
I4.2		Removed stride 32	305 ms	9.1 ms	0.448
I4.3		Removed stride 8	350 ms	10.1 ms	0.447
I5.1	Ensemble	XXXS-XL	/	221.5 ms	0.530
I5.2		416 px + 640 px + 864 px @416 px	446 ms	24 ms	0.489
I5.3		416 px + 640 px + 864 px @640 px	955 ms	26.1 ms	0.478
I5.4		416 px + 640 px + 864 px @864 px	1650 ms	32.3 ms	0.429
I5.5		Czech + India + Japan	955 ms	26.1 ms	0.423
I5.6		ADS + RDD	639 ms	17.6 ms	0.432
I5.7		Joint dataset (ADS+RDD) + RDD	639 ms	17.6 ms	0.498

to apply, though having great potential impact on the model's detection performance. In P1.1, we applied HE with 10 epochs and 300 generations to our base model B, resulting in a negative impact -0.019mAP (c.f., Table IX.4). The combination of 40 epochs and 50 generations in P1.2 increase the mAP to 0.499 (+0.013mAP) while having the same short runtime. It should always be investigated how many epochs and generations are useful to find the right hyperparameters for an application. When used correctly, the tool is very powerful, especially considering that it does not require any additional resources in application.

New Data Impact: Well labeled and large amount of data increases robustness and generalization. Our additional dataset ADS can be valuably utilized to increase the detection performance while remaining the runtime with an increase of 0.008mAP (P2). It works even though the datasets are very different in regard to resolution, image quality and angle to the road. In Section IX.4.2, we furthermore examine the impact of new data in ensembles.

Comparing the results of YOLOv5xxxs, YOLOv5s and YOLOv5m models trained on the ADS with models trained on the ADS_P, the positive influence of more label categories can be derived. For YOLOv5xxxs the mAP rises from 0.429mAP to 0.450mAP, for YOLOv5m from 0.677mAP to 0.683mAP and YOLOv5s does not profit and experiences a minimal performance drop from 0.650mAP to 0.649mAP. A closer look at the individual Average Precisions (APs) for each class reveals the benefit: D00 (+0.035AP), D10 (+0.040AP), and D40 (+0.022AP) with YOLOv5xxxs by labeling the common miss-interpreted construction joints and drainages. Only D20 is yielding worse results (-0.016AP).

Floating Point Format Impact: We consider the Floating Point Format to be set to 16 bit a present as our results show, that the loss in detection performance is very marginal with -0.001mAP (P3). A noticeable performance gap of the models with the different floating-point representations is not present. Adjusting the Floating Point Format to your hardware is of great importance as displayed in Table IX.4. While the Jetson runtime decreases significantly with -28 %, the runtime at the 1080 TI occurs no meaningful difference.

As the Jetson stands representative for possible hardware in autonomous vehicles in contrast to the 1080 TI being rather a training setup, a change of the Floating Point Format is a great Tool to deal with runtime issues.

IX.4.2 Bag of Investments Impact

Test Time Augmentation Impact: Test-Time augmentation leads to a higher generalization. It is a useful tool especially for smaller datasets [51]. With increased data volume the expected improvement through TTA decreases, which results in the model being more invariant to augmentation. Thus, our basic model B (0.486mAP) gains 0.018mAP by applying TTA, whereas the model trained with RDD and ADS data (0.494mAP) gains just 0.012mAP. Practitioners must be aware that the use of TTA leads to significant increase in runtime (+455 ms ,+142 %), which is why we consider it an investment that is recommended to use. Especially with a poor data base, the use of TTA can increase from mediocre improvements to considerable improvements. An important advantage is the simplicity of use, which makes the application of the tool very straightforward in prototyping.

Smaller/Larger Input Image Size Impact: It is certainly debatable whether this tool should be evaluated as an investment or a present. The input size must fit to the dataset and application, depending on the size and structure of the objects to be detected. Contrary to what may be assumed ad hoc, a smaller input size can lead to improved results (not in our case) while always decreasing runtime. By analogy, a larger input size does not always lead to better results and inevitably leads to poorer runtimes. The final Table IX.4 displays, that the initial image size of 640 px provides the best detection performance, so a larger image size, in addition to an increase in runtime (+231 ms ,+72 %), actually leads with -0.004mAP to slightly worse results. On the other hand, a reduction of the input size can lead to a significant runtime improvement with just 47 % of the runtime needed compared to the base model B. Not to be neglected, however, is the loss of detection performance of -0.017mAP. We have primarily taken into account the effects on the Jetson, as we can assume productive use in the vehicle on comparable edge devices.

A closer look at the mAP values for each damage class shows that alligator cracks benefit greatly from smaller images and all other road damage benefits from larger images. Alligator cracks are an accumulation of many cracks; at higher resolution, these structures with more widely spaced cracks can be evaluated by the model as individual longitudinal or lateral cracks and no longer as a union. At the same time, single cracks and potholes benefit from a larger input image size. With a smaller image size, it is less likely to detect single cracks within the alligator cracks, but already small cracks and potholes are more likely to be overlooked. For the best possible performance, the appropriate choice of the input image size is important.

Table IX.5: mAP scores for specific damage classes with different input image size

Class mAPs	416 px	640 px	864 px
D00 mAP@0.5	0.428	0.461	0.464
D10 mAP@0.5	0.363	0.365	0.382
D20 mAP@0.5	0.661	0.640	0.597
D40 mAP@0.5	0.429	0.479	0.485

Smaller/Larger Model Size Impact: Consistently, smaller models lead to poorer detection performances and larger models lead to better detection performances. Respectively, smaller models always lead to an improved runtime and larger models to a worse runtime. Due to the gigantic factorial influence on the runtime of models' different sizes, decisions should always be made on a use-case basis. Other applications, on the other hand, are not cost- and runtime-critical and require the maximum possible detection performance. Since each model has to be trained from scratch, it takes a lot of time and effort.

It is noticeable that due to the different width and depth of the YOLOv5 models, pruning yields worse results than simply choosing a smaller model. YOLOv5m requires only a quarter as many parameters as YOLOv5x, but after pruning with 50 % sparsity the YOLOv5x model performed significantly worse than the YOLOv5m model, with an mAP@0.5 of only 0.171.

Table IX.6: Performance and speed for YOLOv5 models

Model Size	**mAP@0.5**	**Runtime$_{\text{Jet.}}$**	**Runtime$_{\text{GTX}}$**	**Params**
YOLOv5xxxs	0.369	82 ms	8.8 ms	0.42
YOLOv5xxs	0.445	243 ms	9.3 ms	3.75
YOLOv5xs	0.462	310 ms	10 ms	4.92
YOLOv5s	0.486	320 ms	11.2 ms	7.06
YOLOv5m	0.493	702 ms	16.8 ms	21.05
YOLOv5l	0.502	1065 ms	26.5 ms	46.62
YOLOv5x	0.513	2353 ms	58 ms	88.41
YOLOv5xl	0.513	3537 ms	92 ms	146.31

Head Modification Impact: *Head Modification Impact*: An extra stride 64 (I4.1) is likely to be good for rather larger images. In our application with an image size of 640 px, the extra stride 64 reduces the detection performance of smaller objects in particular, since the results of the convolution of stride 64 are included in the confidence calculation of the objects on an equal footing with the other strides. The additional stride has therefore led to a deterioration of -0.019mAP in detection and an increase in runtime with 18 ms, however, the situation may well be different in other application domains with lager image sizes. Removing the stride 32 (I4.2) results in improved runtime, but at the cost of a significant reduction in mAP by 0.038. This indicates that the stride is quite valuable in combination of expected damage and image size. The same applies to the removing of the stride 8 (I4.3), although we cannot explain the increased Runtime$_{\text{Jetson}}$, because it is lower on Runtime$_{\text{GTX}}$, as expected.

All head modifications deliver poor results compared to the base model B and are not recommended to be applied for both, decreasing runtime and improving detection performance.

Table IX.7: Recommended tools to use YOLOv5 at maximum speed, with acceptable performance, at maximum performance and in the Sweet-Spot between speed and performance

Tools	**Runtime**	**Perform.**	**Sweet-Spot**
Model Size	S	X	S
Test Time Augmentation	NO	YES	YES
Ensemble	/	XXXS-XL	/
Hyperparameter Evolution	YES	YES	YES
Input Size	416	640	640
Head Modification	NO	NO	NO
New Data	YES	YES	YES
Half-Precision	YES	NO	YES
$\text{Runtime}_{\text{Jetson}}$	119 ms	/	568 ms
$\text{Runtime}_{\text{GTX}}$	9.3 ms	632.5 ms	27.7 ms
mAP@0.5	0.486	0.531	0.506
$\text{F1-score}_{\text{GRDDC}}$	0.577 (#50)	0.715 (#2)	0.653 (#34)

Ensemble Impact: The ensemble of YOLOv5 model sizes (I5.1) includes eight models from XXXS to XL and offers by far the highest increase of an isolated tool in detection performance with 0.044mAP gain compared to the base model B. Since the models must be executed in parallel, this ensemble also results in the greatest increase in runtime. The Jetson was not capable of running the ensemble model due to excessive computational demands.

I5.2, I5.3 and I5.4 are ensembles consisting of the models I2.1 with 416 px training image size, the base model B with 640 px training image size and I2.2 with 864 px training image size. I5.2 utilizes input images with 416 px, respectively I5.3 640 px and I5.4 864 px. All three ensembles deliver rather poor results. The smallest input size 416 px barely surpases the performance of the base model B with +0.003mAP while having a 39 % increased runtime. Regarding our dataset downscaling is less harmful than upscaling as the very poor score of the ensemble I5.4 indicates, though, in general utilizing models which have been trained with different image sizes is not useful.

I5.5 is an ensemble composed of models from only country-specific data from the Czech Republic, India, and Japan. The results are very poor as the single models miss generalization due to their smaller data base. Additionally, every model is equally considered for every country which results in poor detection performance respectively. if experts in application become interesting in the future, then utilizing all data in the first step is recommended, followed by a second transfer learnig step of the specific countries. Kortmann et al. [30] used this procedure starting with a country classifier in the first place.

If we use an ensemble of a model trained from only the RDD and one trained only on the ADS (I5.6), we obtain poor results with far worse than the base model B (-0.054mAP). It shatters the performance significantly. This is mainly due to the fact that the ADS model has a poor generalization and is not suitable to be used for detection in RDD. The ADS is thus valuable to use in transfer learning on top of the RDD as displayed in P2 but not as the only data. the Aglaia model shatters the overall performance. An ensemble of the joint dataset from ADS and RDD combined with a model trained with only the RDD delivers superior results compared to B with +0.012mAP.

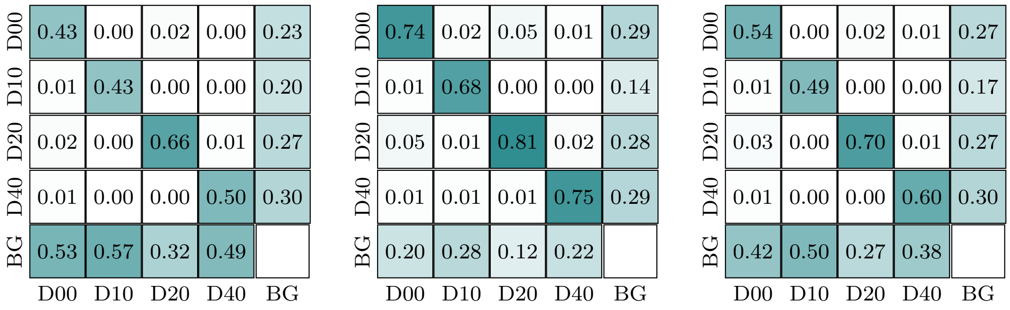
IX.4.3 Recommended Model Designs

In this section, we discuss the given tool combinations in the 'Max. Speed', 'Max. Performance', and 'Sweet-Spot' models. The GRDDC leaderboards are sustained, with one frozen leaderboard existing at the time of the challenge end (Test1) and one for the present time (Test1-2)[1]. Table IX.7 lists the tools used and the performance parameters of the models in an overview.

Max. Speed: Due to the continued very active community, which is evident from the leaderboard results, our speed network is in #50 place at the time of paper submission. At the time the challenge ended, we would have been in #7 with the result of our runtime-optimized 'Max. Speed' model.

In this model, we select size S and applied four tools: Hyperparameter evolution, smaller input size, new data and half-precision format. As shown in section Section IX.4.2, we have made use of the application of the presents and the reduction of the input size from the investments. The detection performance is identical to the base model B (cf. Table IX.4) with 0.486mAP while the runtime reduces to 37 % of B. This is a great achievement as 119 ms on a Jetson Nano device results in 8.4 fps. The ultimate solution for real-time detection would be 30 fps due to the frame rate of the camera, but this does not seem achievable as a heavy loss of detection performance occurs when

[1] GRDDC leaderboards: https://rdd2020.sekilab.global/leaderboard/



	D00	D10	D20	D40	BG
D00	0.43	0.00	0.02	0.00	0.23
D10	0.01	0.43	0.00	0.00	0.20
D20	0.02	0.00	0.66	0.01	0.27
D40	0.01	0.00	0.00	0.50	0.30
BG	0.53	0.57	0.32	0.49	

(a) Speed 0.486mAP

	D00	D10	D20	D40	BG
D00	0.74	0.02	0.05	0.01	0.29
D10	0.01	0.68	0.00	0.00	0.14
D20	0.05	0.01	0.81	0.02	0.28
D40	0.01	0.01	0.01	0.75	0.29
BG	0.20	0.28	0.12	0.22	

(b) Perf. 0.531mAP

	D00	D10	D20	D40	BG
D00	0.54	0.00	0.02	0.01	0.27
D10	0.01	0.49	0.00	0.00	0.17
D20	0.03	0.00	0.70	0.01	0.27
D40	0.01	0.00	0.00	0.60	0.30
BG	0.42	0.50	0.27	0.38	

(c) Sweet-Spot 0.506mAP

Figure IX.2: Confusion Matrices of the proposed YOLOv5 models 'Max. Performance', 'Max. Speed', and 'Sweet-Spot' (cf. Table IX.7)

decreasing runtime even further. An optimization of the edge hardware can provide a relief in the future.

The confusion matrix in Figure IX.2a shows that the model has considerable weaknesses. Alligator cracks (D20) are best detected with 2/3 precision, followed by potholes (D40) with half of all potholes being detected and longitudinal (D00) and transverse cracks (D10) with just 0.43 each. Overall, it can be observed that there are negligible misclassifications. Hence, the major issues are that our model misses damages entirely which results in them being classified as the background (BG).

Max. Performance: Our 'Max Performance' model reaches #2 on the leaderboard with narrow margin to #1. The model is an ensemble of models' different sizes. We have applied all tools capable of increasing the performance without any respect paid to runtime increase as displayed in Table IX.7. By our design, the model is no longer even executable on the Jetson. With a runtime of 632.5 ms on the GTX 1080 TI, even this hardware is not close of being real-time capable. In addition, it is very unlikely that similarly large hardware capacities can be used in an autonomous vehicle to only detect road damages.

The confusion matrix in Figure IX.2b displays major improvements over Figure IX.2a. The precision occurs with transverse cracks (D10) being with 0.68mAP still slightly above the best detected class of the 'Max. Speed' model. Potholes (D40) and alligator cracks (D30) are with 0.75 and 0.81mAP very well detected. It is striking that misclassifications appear primarily in the distinction between alligator cracks and lateral cracks. In the ADS images, we partially had difficulty to label consistently when distinguishing between collections of cracks and alligator cracks. For practical applications, we expect little negative impact from this misclassification, since it is relatively unimportant whether we have several cracks or one alligator crack.

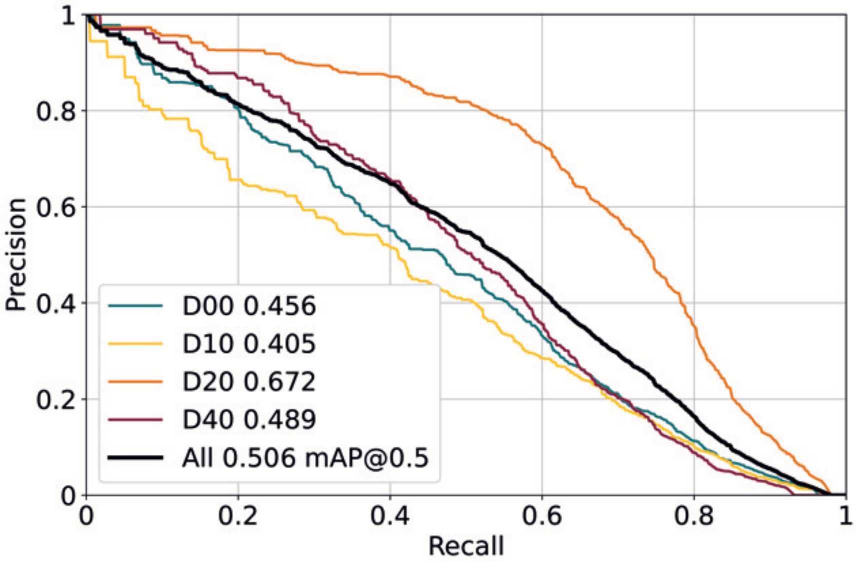
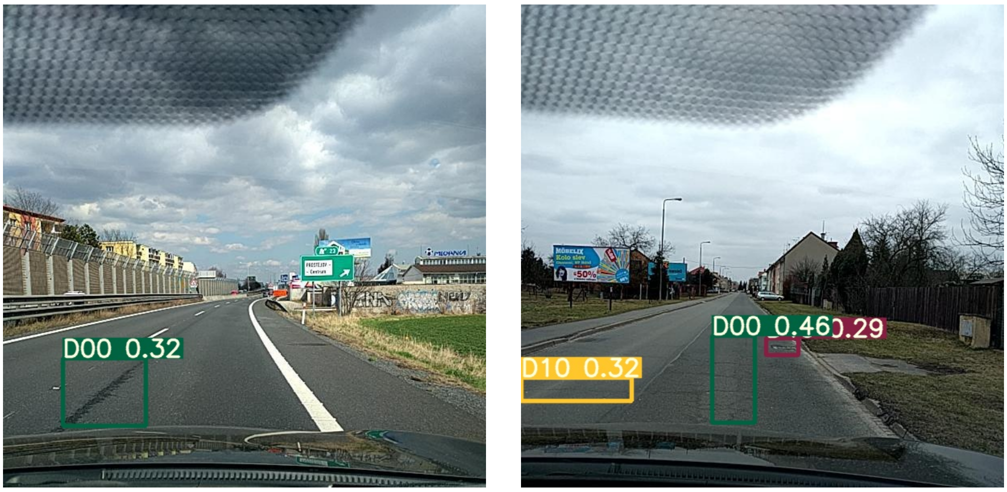


Figure IX.3: Precision-Recall curve for the Sweet-Spot Model, with single class mAP@0.5

Sweet-Spot: This model represents the, according to our results, best trade-off between performance and runtime. Compared to the 'Max. Performance' model, we utilize the half-precision format as the performance loss is negligible compared to the improvements in runtime. Furthermore, the model size S is the same as utilized in the 'Max. Speed' model. Opinions differed though, especially in the application of the TTA. The model gains 0.07mAP while having significantly worse runtime. Its use is debatable, which is why the use of the TTA must be judged according to the available resources. With 568 ms, the model can be executed almost two times a second on the Jetson, but we expect the hardware capabilities to improve in the future. With 0.653mAP, the 'Sweet-Spot' model would have been #3 at the leaderboard when the challenge ended and is currently on #34, having been optimized regarding runtime. Figure IX.3 illustrates the precision-recall curve for an IoU of 0.5 which is the basis for calculating the mAP.

IX.4.4 Limitations

DL models have become an indispensable part of the vehicle environment detection system. Despite major achievements, a truly flawless detection performance remains a utopia, which is why rule-based systems are used in conjunction with it. In the following, we will discuss faulty detections of our algorithms to show which misinterpretations can occur. Figure IX.4a demonstrates a common classification issue in our domain. Construction joints and repaired cracks are likely to be misinterpreted as cracks. As our self-labeled ADS includes construction joint labels, we prove that in 41 out of 460 images, construction joints are classified as longitudinal (D00) or transverse cracks (D10). Manhole covers being misclassified as potholes in contrast, are a minor issue since this happens rarely (c.f. Figure IX.4b).



(a) Repaired crack misclassified as long. crack (D00, green)

(b) Manhole cover misclassified as a pothole (D40, red)

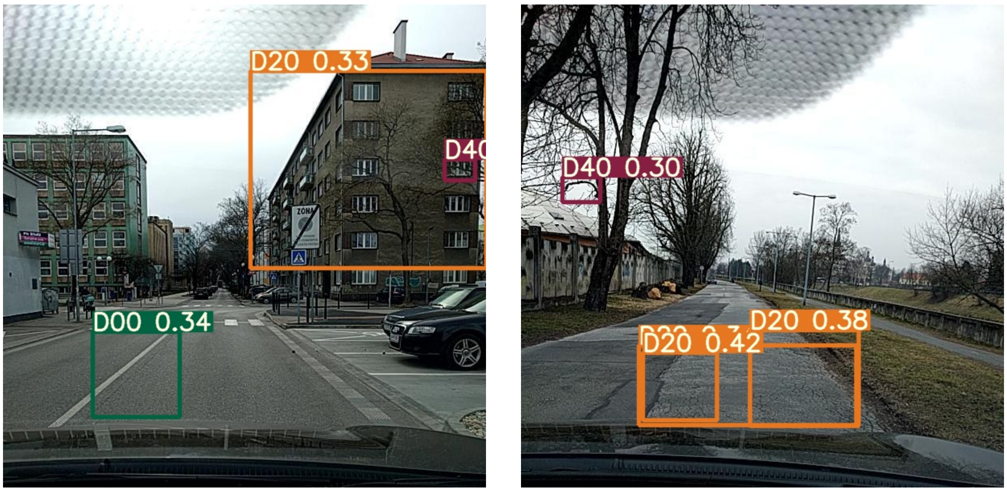
Figure IX.4: Incorrect detections inside the road surface area

Further false detections can always occur in DL. Some of these are not comprehensible. In Figure IX.5a, for example, a detection of an alligator crack occurs, which is probably due to similar contrast ratios of the leafless tree in front of a dark house. Furthermore, a window of the house is recognized as a pothole. Figure IX.5b displays the false detection of a pothole which is indeed a branch of a tree with the horizon in the background.

Due to the good detection performance of DL models, such errors are the exception, but they illustrate the fail-safe requirement, whereby in environment perception, confidence must always be achieved by multiple systems.

IX.5 Conclusion and Future Work

This paper gives a comprehensive overview of tools which can be applied in YOLOv5 DL and their expected impact on detection performance and



(a) A leafless tree misclassified as aligator crack (D20, orange)

(b) A branch misclassified as a pothole (D40, red)

Figure IX.5: Incorrect detections outside the road surface area

model runtime. In contrast to the GRDDC, which focuses solely on detection performance, we close the gap to the application of DL models in the autonomous vehicle for the use case of road damage detection. In addition to the isolated quantitative evaluation of the applied tool, we propose three combinations of the tools in optimized models in terms of maximum performance ('Max. Performance'), minimum runtime ('Max. Speed'), and the best trade-off of performance and runtime ('Sweet-Spot').

The differences between the models are substantial in regard to their detection performance, which makes the challenges of autonomous driving particularly clear, as the detection of road damages does not remain the only task of the environment perception system in road traffic. Considering the confusion matrices, we can summarize YOLOv5 by saying that it looks only once, misses quite a bit but classifies correctly. With better and, above all, more efficient DL models, we hope to be able to achieve even better results in the future. It is furthermore to be desired that Moores' law will persist and that computing capacities will continue to improve.

Future work should be carried out on further tool to improve DNN in their object detection task. In autonomous driving, there should be methods to use sequences of images to contribute to an increase in detection performance. It would be possible to achieve higher confidence of detection via multiple subsequent detections. LSTM networks are best suited for processing sequential data. They can be used to predict trajectories of moving objects and enable object tracking.

Another potential tool could be to apply more classes in a dataset. Our results show that construction joints are likely to be classified as cracks. Taking construction joints into account in training can lead to better discrimination. In the current state, construction joints are perceived as background.

The last thing we would like to point out is the use of segmentation techniques. Results from the challenge suggest a negative effect of image segmentation (street, horizon, etc.). However, new segmentation techniques could be explored to contribute to this domain in a meaningful way.

Bibliography

[1] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[2] C. Badue, R. Guidolini, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, p. 113816, 2020.

[3] H. Winner and S. Hakuli, "Conduct-by-wire–following a new paradigm for driving into the future," in *FISITA world automotive congress*, vol. 22, p. 27, Citeseer, 2006.

[4] T. R. Miller and E. Zaloshnja, "On a crash course: The dangers and health costs of deficient roadways," 2009.

[5] Gleave, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *On-Road Automated Driving committee*, pp. 1–35, 2018.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[7] O. Russakovsky, J. Deng, *et al.*, "Imagenet large scale visual recognition challenge," vol. 115, no. 3, pp. 211–252, 2015.

[8] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[9] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] G. Jocher, “Yolov5 github repository,” 2021.

[13] D. Arya, H. Maeda, *et al.*, “Transfer learning-based road damage detection for multiple countries,” *arXiv:2008.13101*, 2020.

[14] D. Arya, H. Maeda, *et al.*, “Global road damage detection: State-of-the-art solutions,” in *IEEE Int. Conf. on Big Data*, pp. 5533–5539, IEEE, 2020.

[15] T. Hascoet, Y. Zhang, *et al.*, “Fasterrcnn monitoring of road damages: Competition and deployment,” in *IEEE Int. Conf. on Big Data*, pp. 5545–5552, IEEE, 2020.

[16] H. Bello-Salau, A. Aibinu, *et al.*, “Image processing techniques for automated road defect detection: A survey,” *Int. Conf. on Electronics, Comp. and Computation (ICECCO)*, pp. 1–4, 2014.

[17] C. Koch, K. Georgieva, *et al.*, “A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure,” *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[18] W. Cao, Q. Liu, and Z. He, “Review of pavement defect detection methods,” *IEEE Access*, vol. 8, pp. 14531–14544, 2020.

[19] M. W. Sayers, “The little book of profiling: basic information about measuring and interpreting road profiles,” tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 1998.

[20] V. Hegde, D. Trivedi, *et al.*, “Yet another deep learning approach for road damage detection using ensemble learning,” *IEEE Int. Conf. on Big Data*, pp. 5553–5558, 2020.

[21] K. Doshi and Y. Yilmaz, “Road damage detection using deep ensemble learning,” in *IEEE Int. Conf. on Big Data*, pp. 5540–5544, IEEE, 2020.

[22] Z. Pei, R. Lin, *et al.*, "Cfm: A consistency filtering mechanism for road damage detection," *IEEE Int. Conf. on Big Data*, pp. 5584–5591, 2020.

[23] V. Mandal, A. R. Mussah, and Y. Adu-Gyamfi, "Deep learning frameworks for pavement distress classification: A comparative analysis," in *IEEE Int. Conf. on Big Data*, pp. 5577–5583, IEEE, 2020.

[24] D. Jeong, "Road damage detection using yolo with smartphone images," *IEEE Int. Conf. on Big Data*, pp. 5559–5562, 2020.

[25] Y. Liu, X. Zhang, *et al.*, "Deep network for road damage detection," *IEEE Int. Conf. on Big Data*, pp. 5572–5576, 2020.

[26] S. Naddaf-Sh, M.-M. Naddaf-Sh, *et al.*, "An efficient and scalable deep learning approach for road damage detection," *IEEE Int. Conf. on Big Data*, pp. 5602–5608, 2020.

[27] X. Zhang, X. Xia, N. Li, M. Lin, J. Song, and N. Ding, "Exploring the tricks for road damage detection with a one-stage detector," *IEEE Int. Conf. on Big Data*, pp. 5616–5621, 2020.

[28] R. Vishwakarma and R. Vennelakanti, "Cnn model & tuning for global road damage detection," in *IEEE Int. Conf. on Big Data*, pp. 5609–5615, IEEE, 2020.

[29] V. Pham, C. Pham, and T. Dang, "Road damage detection and classification with detectron2 and faster r-cnn," *IEEE Int. Conf. on Big Data*, pp. 5592–5601, 2020.

[30] F. Kortmann, K. Talits, *et al.*, "Detecting various road damage types in global countries utilizing faster r-cnn," *IEEE Int. Conf. on Big Data*, pp. 5563–5571, 2020.

[31] A. R. Hevner, "A three cycle view of design science research," *Scandinavian J. of Information Systems*, vol. 19, no. 2, p. 4, 2007.

[32] A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design Research in Information Systems*, pp. 9–22, Springer, 2010.

[33] M. Everingham, L. Van Gool, *et al.*, "The pascal visual object classes (voc) challenge," *Int. J. of Comp. Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[34] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[35] L. Wang, *Support vector machines: theory and applications*, vol. 177. Springer Science & Business Media, 2005.

[36] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv:1511.08458*, 2015.

[37] Y. Zhu and S. Newsam, "Densenet for dense flow," in *IEEE Int. Conf. on Image Processing*, pp. 790–794, IEEE, 2017.

[38] K. He, X. Zhang, *et al.*, "Deep residual learning for image recognition," in *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 770–778, 2016.

[39] "Cspnet: A new backbone that can enhance learning capability of cnn," *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition Workshops*, pp. 390–391, 2019.

[40] K. Wang, J. H. Liew, *et al.*, "Panet: Few-shot image semantic segmentation with prototype alignment," *IEEE/CVF Int. Conf. on Comp. Vision*, pp. 9196–9205, 2019.

[41] H. Rezatofighi, N. Tsoi, *et al.*, "Generalized intersection over union: A metric and a loss for bounding box regression," *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition*, pp. 658–666, 2019.

[42] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[43] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. of Big Data*, vol. 6, no. 1, 2019.

[44] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *2017 IEEE Int. Conf. on Comp. Vision*, pp. 3544–3553, IEEE, 2017.

[45] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *European Conf. on Computer Vision*, pp. 162–178, Springer, 2020.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[47] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv:1609.04747*, 2017.

[48] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[49] M. Mitchell, *An introduction to genetic algorithms.* MIT press, 1998.

[50] D. Mahajan, R. Girshick, *et al.*, “Exploring the limits of weakly supervised pretraining,” *European Conf. on Comp. Vision*, vol. 11206, pp. 185–201, 2018.

[51] D. Shanmugam, D. Blalock, *et al.*, “When and why test-time augmentation works,” *arXiv:2011.11156*, 2020.

[52] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, “Rethinking the value of network pruning,” *arXiv preprint arXiv:1810.05270*, 2018.

[53] M. Ganaie, M. Hu, *et al.*, “Ensemble deep learning: A review,” *arXiv preprint arXiv:2104.02395*, 2021.

[54] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

Chapter X

Concept of a Cloud State Modeling System for Lead-Acid Batteries: Theory and Prototyping

Outline

Bibliographic Information

Felix Kortmann, Daniel Brieske, Pia Piekarek, Julian Eckstein, Alexander Warnecke, Paul Drews, and Dirk Uwe Sauer, (2021, January). “Concept of a Cloud State Modeling System for Lead-Acid Batteries: Theory and Prototyping”. In 2021 International Conference on Electronics, Information, and Communication (ICEIC) (pp.1-4). IEEE.
DOI: 10.1109/ICEIC51217.2021.9369785. Preprint ResearchGate:348566506.

Author’s contribution

The author’s share of the publication is 40%. Table B.10 in Appendix B shows the contributions of all authors of the publication in detail.

Copyright Notice

Abstract

Lead-acid batteries are still the cash cow in the market of energy storage systems. Use in vehicles will continue in the next years due to several advantages over competing products. Despite the high level of maturity of Lead-Acid Batteries (LABs), the Lead-Acid Battery (LAB) is with 40 % by far the largest cause of short-term vehicle breakdowns. This is primarily not due to defective products, but rather to over-aged or deeply discharged batteries. The complex aging characteristics of lead-acid batteries do not allow the health status of a battery to be easily detected, which increases the probability of short-term failures. Our paper presents a solution of monitoring the current health status of a battery and discusses modeling opportunities in a cloud as a digital twin to predict breakdowns in the future. We present both, a Low Power Wide Area Network (LPWAN) based monitoring prototype and theory for state modeling and prediction.

X.1 Introduction

Despite the growth of lithium-ion batteries in the market and a number of new and very promising battery technologies [1, 2], the LAB is and will remain extremely important, and a considerable growth is predicted to continue over the next years. One of the market forecasts for power train electrification shows that by 2025, about 55 % of micro hybrid vehicles will still be equipped with LABs [3]. In automotive applications, LABs continue to be used as starter batteries due to the high degree of maturity and specific power density especially at low temperatures and long parking times.

Regardless the above-mentioned advantages, ADAC (Allgemeine Deutsche Automobil-Club e.V.) statistics show that the starter battery accounts for 42 % of passenger car breakdowns in 2018 from a total of 385.955 road patrol operations [4]. The high breakdown rate is is not due to poor products, but rather to the following three causes [5]: (1) Deep discharge of the battery due to defective consumers and partial charging by short trips. (2) High temperature dependency, e.g. during cold snaps. (3) Battery ageing due to erosion of the electrodes, sulphation, corrosion and water loss.

Short-term breakdowns are usually particularly annoying for consumers and cost time and money. To overcome this, battery sensors are commonly used to monitor the battery status by measuring current, voltage and temperature. With the information gained, electrical consumers in the vehicle can be optimally controlled, enabling start-stop functionality and automatic shutdown of consumers. The subject of battery state detection by a sensor suitable for automotive applications has already been examined in the literature and will be dealt with further in Section X.2.

Whilst current product solutions in the automotive industry primarily focus on the short-term ability of the battery to provide sufficient power, we want to present an approach of state modeling the battery as a digital twin in a cloud application. In addition to the theory of the battery model, whose parameters are continuously updated in the life cycle, we present a prototypical implementation for the vehicle-infrastructure-independent transmission of battery data via LPWAN. For this we use the Intelligent Battery Sensor (IBS) by HELLA GmbH & Co. KGaA, which provides us with essential data for modeling in the cloud. The battery model will enable us to predict the battery state by taking further external conditions into account.

Our contributions can be summarized as follows:

- We refine an existing model for real-time monitoring of LAB for automotive applications.

- We point out further steps towards realisation of a digital LAB twin in the cloud.

- We present an LPWAN-based prototype for the transmission of sensor data into the cloud.

X.2 Fundamentals & Related Work

Today's Battery Management Systems (BMSs), used for example in the automotive industry, depend on the nominal capacity and the internal resistance. The IBS is part of the BMS and measures current, voltage, temperature and calculates, by using parametric functions, the State of Charge (SoC), the State of Health (SoH) and the State of Function (SoF). In addition, the IBS has a temperature model which derives the battery acid temperature from the measured temperature at the pole [6, 7]. The calculated data is transmitted via the Local Interconnect Network-Bus (LIN bus) to the electrical system of the vehicle, which interprets the data and derives internal functions. Thus, the sensor provides a prognosis of the current state of the battery and not a prediction over days or even weeks.

The description and modeling of a LAB is attributable to the development of the electrochemical storage system of Peukert [8] in 1897. The LAB has nonlinear, interdependent reactions and cross-relations, which makes the mathematical description of the system very complex. Conversion processes take place in the battery, implying a transition of the active mass during charge and discharge. The full LAB process is a composition of physical, chemical and electro-chemical processes which results in a complex description. The battery is submitted to cyclical and calendrical aging, which finally leads to a poor performance or even complete failure of the system. During the degradation process, the internal resistance of the battery increases. This results in a reduction of the available capacity. Therefore, batteries are oversized to reduce the uncertainties associated with their lifetime [9]. Numerous research papers address the description of the physical, chemical and electro-chemical process steps. For a holistic BMS a reliable specification of these is crucial [9, 10, 11].

The modeling of batteries is divided into three groups: empirical, Electrical Equivalent Circuit (EEC) and physio-chemical models. Thereby, a trade-off between accuracy, computing time and flexibility has to be found.

Most research approaches focus on the parameterization of EEC diagrams based on measurements of impedance spectra [13, 14, 15, 16]. Originating from the spectrum, the electro-chemical processes are identified. These are

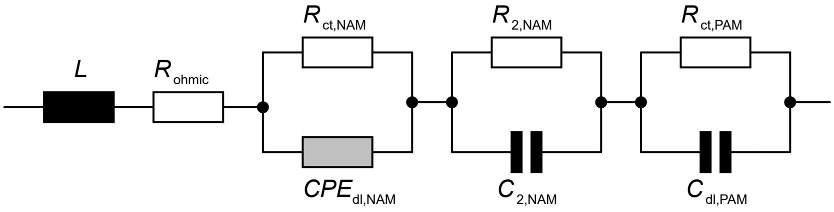


Figure X.1: EEC for the impedance spectrum of a LAB based on Thele [12]

shown as semicircles in the Nyquist diagram. The first description as an EEC for electrochemical storage systems was introduced by Randles [17] based on Thévenin [18]. The EEC includes an ohmic resistor, an RC-element and a diffusion element [16]. The RC-element includes a charge-transfer resistance (R) and a double-layer capacitance (C). Based on this circuit, Thele [12] describes a suitable EEC for LABs representing the electro-chemical processes of the negative and positive active mass (cf. Figure X.1). Thele [12] also shows an EEC which represents the side-reaction of the LAB, including gassing effects and acid stratification effects [12]. Kwiecien [19] investigates the impedance spectra of the LAB during ageing. She examines the change of EEC elements over time and shows that an EEC does not represent a battery over its entire life cycle, but is subject to major changes [19].

Blanke et al. [10] extend the EEC with physical elements to increase the accuracy of the model. However, this is highly dependent on individual batteries, which always differ in their chemical composition.

Salkind et al. [20] investigate Electrochemical Impedance Spectroscopy (EIS) in combination with fuzzy-logic methods for data analysis to determine SoH and SoC.

In addition to modeling the processes of a LAB as an EEC, other approaches exist in research. For example, machine learning and classification approaches try to describe the process in its entirety avoiding the use of individual replacement elements [9, 11]. Richardson et al. [9] present a bayesian non-parametric approach of predicting the change of the battery capacity over time. They utilize pattern-recognition enabled by an increase of computing resources. However, the results of these methods depend strongly on the pattern classifier and the type of distribution (Gauss, Binomial, Poission). Salameh et al. [11] also investigate a machine learning and classification method for SoH calculation of LABs, showing good performance with the same LAB specifications.

All these approaches try to reflect the current state of the battery as accurately as possible. However, they lack prediction capability as they are based on data already collected. In this paper we adopt the approach by

Thele [12], extend it by a predictive capability and adjust it to the automotive application. Our prototype (cf. Figure X.1) provides generated data from the IBS in order to continuously update our EEC elements in the cloud over time. Predictions are to be made possible by different artificial load cycles and weather forecast data. The results can be delivered to consumers as a digital service of upcoming breakdowns.

X.3 State Modeling Theory

The EEC by Thele [12] which was presented in Figure X.1, is used as the basis for our prediction and modeling approach of LAB in the cloud. The model for the real-time application is illustrated in Figure X.2. The input data is sent to the cloud by our prototype in the vehicle while the output data is calculated within the cloud and ready for use as digital service. The input data current I_{IBS}, state of charge SoC_{IBS}, lifetime $t_{IBS/cycles}$ and temperature of the acid T_{IBS} are provided by the IBS. Our prototype transmits the geolocation L_{IBS} of the vehicle, enabled by LPWAN, so that position-dependent weather forecasts T_{For} can be obtained from the internet as input data for the temperature model. Virtual and artificial load profiles are provided to represent, in the future lying, consumer behavior. The main part of the state modeling and prediction approach is the impedance model of a full cell of a LAB.

Firstly, the elements of the EEC are parameterized by measurements from the EIS. The measurements are recorded for different SoCs, temperatures and superimposed DC currents. Hence, an initial lookup table is given for the EEC.

In the next step, the network is triggered by current or power data of the load profiles. This allows a real-time information about the actual state of the battery and a prediction of the functionality in the future. The output of the model is the terminal voltage $U_{terminal}$ of the battery. In addition, the expected reduced current I_{red} results from a subtraction of the gassing current I_{gas}. I_{gas} is determined via gas model (side reaction in the LAB). I_{IBS} is the current measured by the IBS. The battery parameters (SoC, SoH, SoF) are calculated from the output values of the impedance model and the measurement data of the IBS.

The powers from the impedance model P_I as well as from the gas model P_G, have an influence on the temperature model of the battery. The output data from the temperature model are supplied to the aging model and the impedance model. The values of the lookup table of the impedance model are based on the temperature values of the temperature model and the SoC.

Based on output data from the temperature model, the aging model influences the impedance model during the lifetime of the battery $t_{IBS/cycles}$. The elements of the EEC are updated by the aging model. Thus, the internal resistance of the battery network changes in the process of calendrical, cycle aging and sulfation. Hence, larger ohmic losses inside the battery will occur. The results of the simulation can be used to predict the crank capability of the vehicle for future use.

In case of a predicted, insufficient battery voltage, a warning message can be given to the driver. The availability of location-based temperature forecasts makes the cloud application a promising approach for predictions. Thus, the accuracy of the state of the battery as well as the possibility of prediction can significantly increase due to our approach.

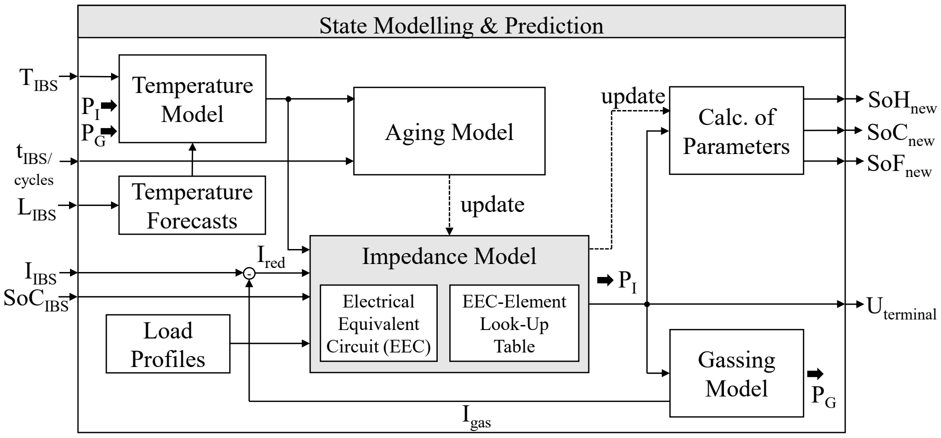


Figure X.2: Real-time state modeling & prediction process of LABs

X.4 Prototype Design

Our prototype (cf.Figure X.4) includes LPWAN technology which enables low-frequency and energy-efficient data transmission. Due to our system design, we are independent of the electrical and electronic architecture of vehicles.

A micro-controller is the base of our prototype including an M0-processor and an integrated radio module for LPWAN communication. For the communication with the IBS, a LIN-Transceiver is used on a custom made printed circuit board including a voltage regulator to ensure the voltage supply of the micro-controller directly by the LAB. Due to our application code in the cloud, we are capable of changing the frequency and resolution of data obtained by the IBS.

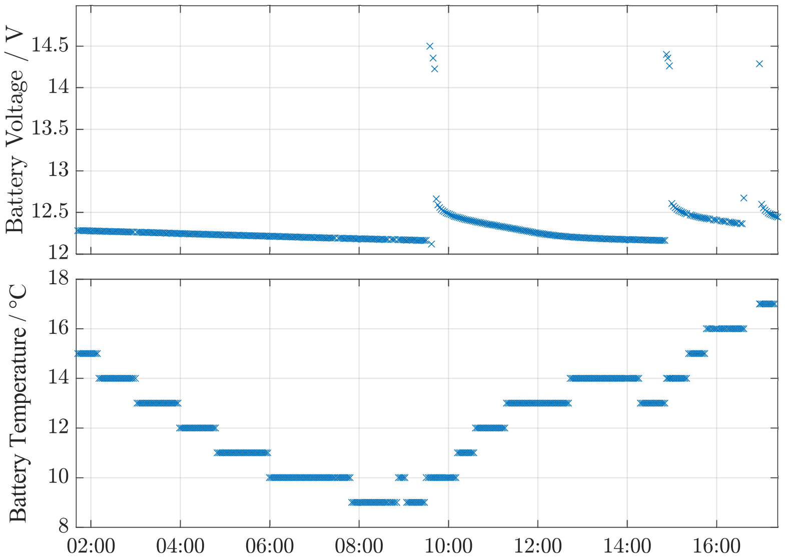


Figure X.3: Battery voltage and temperature transmitted by our prototype

Figure X.3 displays real data from our prototype. In detail, the upper graph represents the battery voltage measured by the IBS. From 2:00 to 9:45, the passenger car is resting while slowly loosing voltage. In the same time, the battery temperature, displayed in the below graph, is also decreasing until the engine starts at 9:45. The short driving periods are displayed at 9:45, 15:00 and 17:00, showing a higher voltage level around 14.5 V due to the alternator. An engine start also results in an increase in temperature.

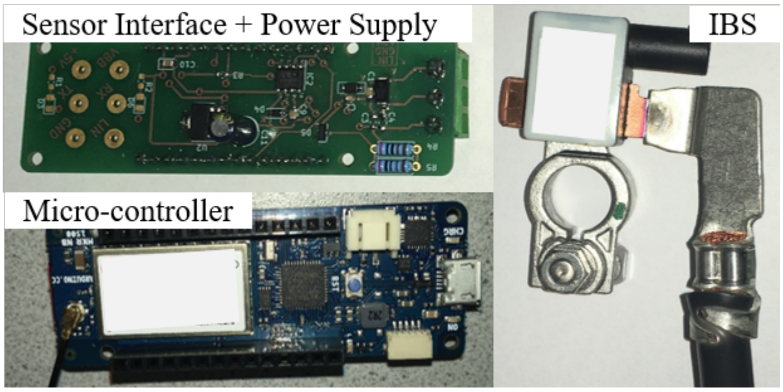


Figure X.4: Components of the prototype: micro-controller, interface, IBS

X.5 Conclusion

Our scientific contribution lies in the extension of Thele's LAB model by a predictive capability of the battery functionality, enabled by the cloud due to the availability of location-dependent weather data. The goal of implementation is the predictability of failures for the customer in order to avoid painful unforeseen failures and thus save time and money.

Besides the modeling we present a prototype that sends data directly from IBS to the cloud via LPWAN independent of the vehicle architecture. This serves as a data supplier for our digital twin in the cloud.

In future work, the existing monitoring of the lead battery using current IBS data in the cloud must be expanded to include the approach presented. For the implementation it has to be clarified in which frequency and resolution the required data must be delivered by the IBS, since the LPWAN technology has a limited bandwidth. Furthermore, it must be decided which artificial load cycles should be applied to the battery in order to reflect consumer behaviour as realistic as possible. Past cranking and resting phases could achieve good results.

Bibliography

[1] M. Hannan, M. Hoque, A. Mohamed, and A. Ayob, "Review of energy storage systems for electric vehicle applications: Issues and challenges," *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 771–789, 2017.

[2] H. Budde-Meiwes, J. Drillkens, B. Lunz, J. Muennix, S. Rothgang, J. Kowal, and D. U. Sauer, "A review of current automotive battery technology and future prospects," *J. of Automobile Engineering*, vol. 227, no. 5, pp. 761–776, 2013.

[3] J. Garche, E. Karden, P. T. Moseley, and D. A. Rand, *Lead-acid batteries for future automobiles.* Elsevier, 2017.

[4] A. deutscher Automobil Club, "Pannenstatistik-2020," April 2020.

[5] J. Badeda, M. Huck, D. Sauer, J. Kabzinski, and J. Wirth, "Basics of lead–acid battery modelling and simulation," in *Lead-Acid Batteries for Future Automobiles*, pp. 463–507, Elsevier, 2017.

[6] M. Schöllmann, M. Rosenmayr, and J. Olk, "Battery monitoring with the intelligent battery sensor during service, standby and production," tech. rep., SAE Technical Paper, 2005.

[7] N. Decius, H. Klein, K.-H. Fortkort, J. Olk, W. Ruttor, and M. Schöllmann, "Modulare bordnetzarchitektur für hybridfahrzeuge," *ATZ-Automobiltechnische Zeitschrift*, vol. 107, no. 12, pp. 1098–1105, 2005.

[8] W. Peukert, "Über die abhängigkeit der kapazität von der entladestromstärke bei bleiakkumulatoren," *Elektrotechnische Zeitschrift*, vol. 20, pp. 20–21, 1897.

[9] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Battery health prediction under generalized conditions using a gaussian process transition model," *J. of Energy Storage*, vol. 23, pp. 320–328, 2019.

[10] H. Blanke, O. Bohlen, S. Buller, R. W. De Doncker, B. Fricke, A. Hammouche, D. Linzen, M. Thele, and D. U. Sauer, "Impedance measurements on lead–acid batteries for state-of-charge, state-of-health and cranking capability prognosis in electric and hybrid electric vehicles," *J. of power Sources*, vol. 144, no. 2, pp. 418–425, 2005.

[11] J. P. Salameh, N. El Ghossein, M. El Hassan, N. Karami, and M. Najjar, "Battery modeling and lifetime prediction," *Modern Environmental Science and Engineering*, vol. 3, no. 4, pp. 278–290, 2017.

[12] M. Thele, *A contribution to the modelling of the charge acceptance of lead-acid batteries-using frequency and time domain based concepts.* Shaker, 2007.

[13] M. Thele, E. Karden, E. Surewaard, and D. Sauer, "Impedance-based overcharging and gassing model for vrla/agm batteries," *J. of Power Sources*, vol. 158, no. 2, pp. 953–963, 2006.

[14] J. Badeda, M. Kwiecien, D. Schulte, and D. U. Sauer, "Battery state estimation for lead-acid batteries under float charge conditions by impedance: Benchmark of common detection methods," *Applied Sciences*, vol. 8, no. 8, p. 1308, 2018.

[15] J. Kowal, H. Budde-Meiwes, and D. U. Sauer, "Interpretation of processes at positive and negative electrode by measurement and simulation of impedance spectra. part i: Inductive semicircles," *J. of Power Sources*, vol. 207, pp. 10–18, 2012.

[16] M. Kwiecien, M. Huck, J. Badeda, and D. U. Sauer, "Correct processing of impedance spectra for lead-acid batteries to parameterize the charge-transfer process," *J. of Applied Electrochemistry*, vol. 48, no. 8, pp. 885–900, 2018.

[17] J. E. B. Randles, “Kinetics of rapid electrode reactions,” *Discussions of the faraday society*, vol. 1, pp. 11–19, 1947.

[18] L. Thévenin, “Extension de la loi d’ohm aux circuits électromoteurs complexes,” *Ann. Tél*, vol. 10, p. 222, 1883.

[19] M. Kwiecien, *Electrochemical Impedance Spectroscopy on Lead-Acid Cells during Aging.* 2019.

[20] A. J. Salkind, P. Singh, A. Cannone, T. Atwater, X. Wang, and D. Reisner, “Impedance modeling of intermediate size lead–acid batteries,” *J. of power sources*, vol. 116, no. 1-2, pp. 174–184, 2003.

Appendix

Appendix A

First Appendix

Table A.1: Society of Automotive Engineers (SAE) levels of automation [10]

Lvl	Name	Steering & Acceleration	Monitoring of Driving Environment	Fallback Performance Driving Task	Drive Modes
Human driver monitors the driving environment					
0	No Automa.	Human	Human	Human	None
1	Diver Assistance	Human & System	Human	Human	Some
2	Partial Automa.	System	Human	Human	Some
Automated driving system monitors the driving environment					
3	Conditio. Automa.	System	System	Human	Some
4	High Automa.	System	System	System	Some
5	Full Automa.	System	System	System	All

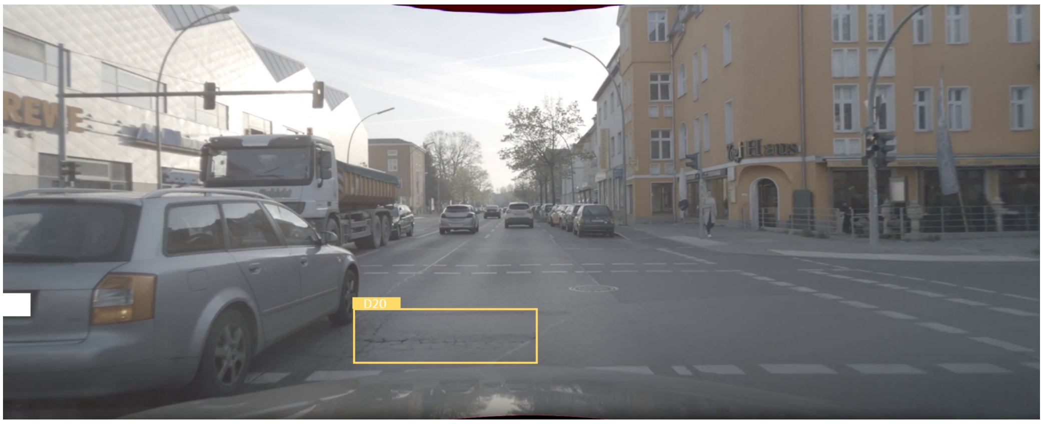


Figure A.1: Aligator crack (D20) in an example image of the ADS (Germany)

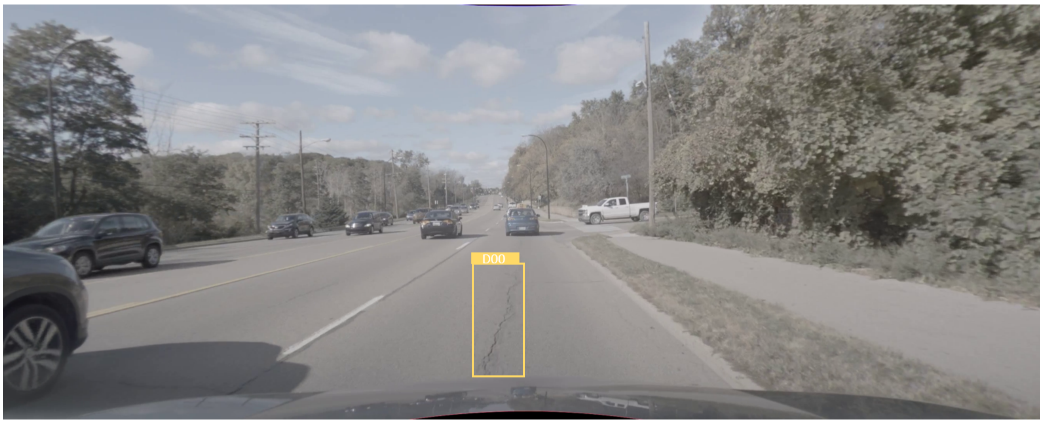


Figure A.2: Longitudinal crack (D00) in an example image of the ADS (USA)

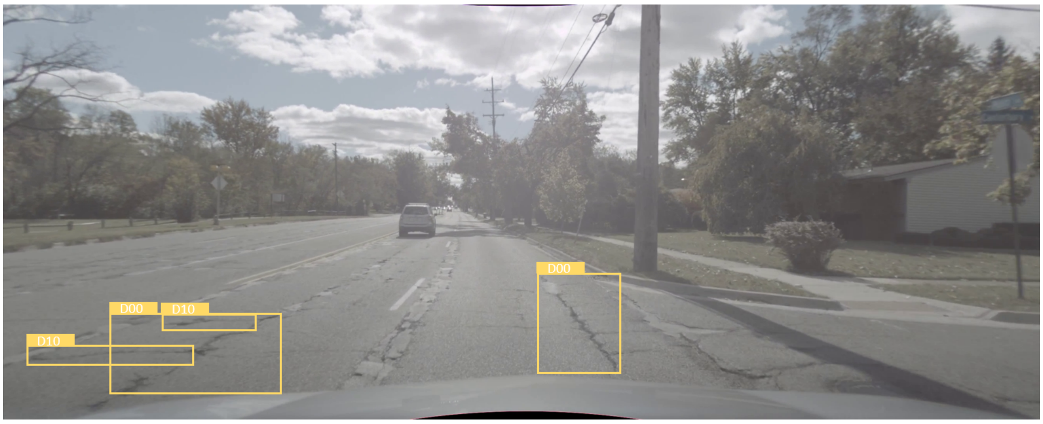


Figure A.3: Multiple damages in an example image of the ADS (USA)

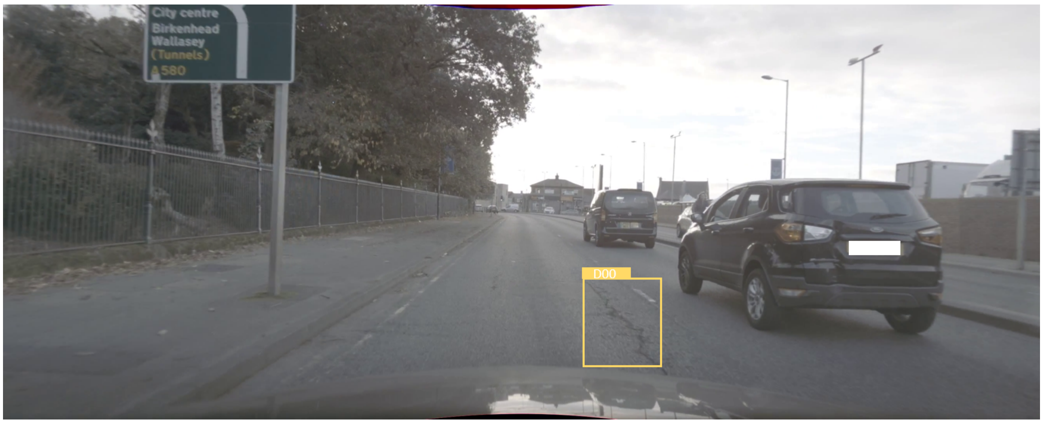


Figure A.4: Longitudinal crack (D00) in an example image of the ADS (UK)

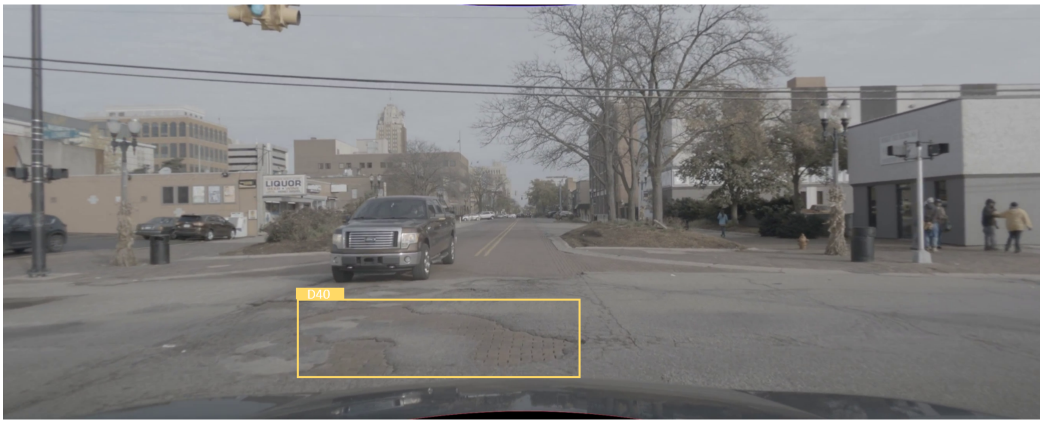


Figure A.5: Pothole (D40) in an example image of the ADS (USA)

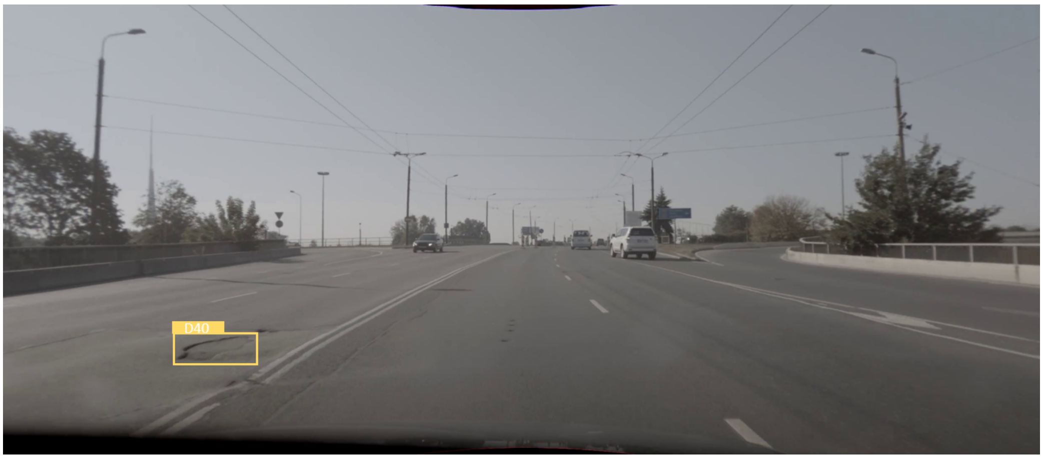


Figure A.6: Pothole (D40) in an example image of the ADS (Latvia)

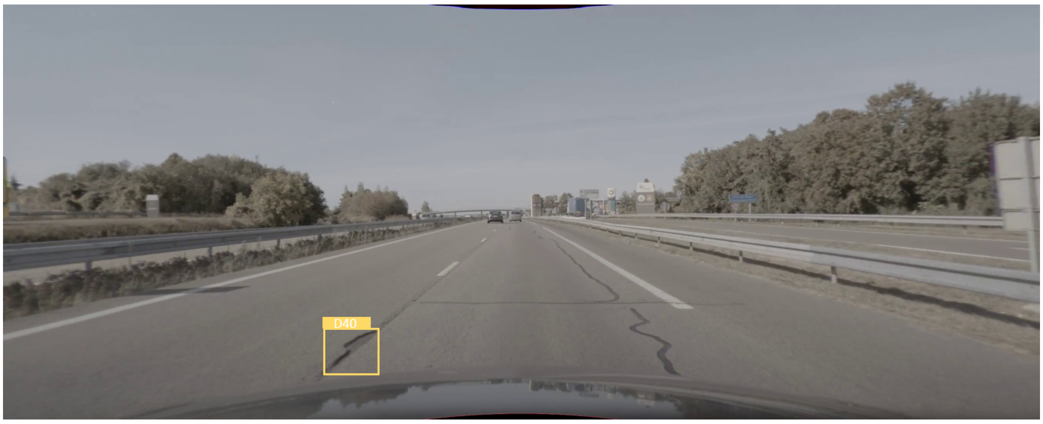


Figure A.7: Pothole (D00) in an example image of the ADS (France)

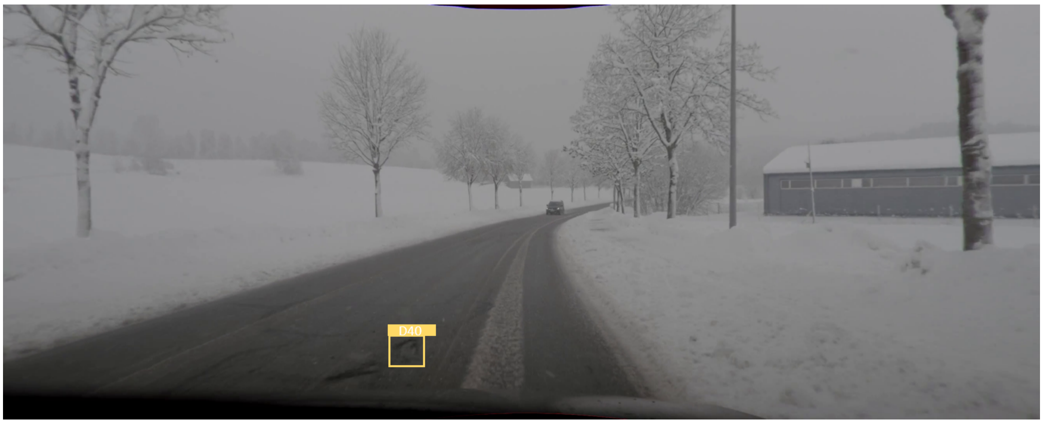


Figure A.8: Pothole (D40) in an example image of the ADS (Germany)

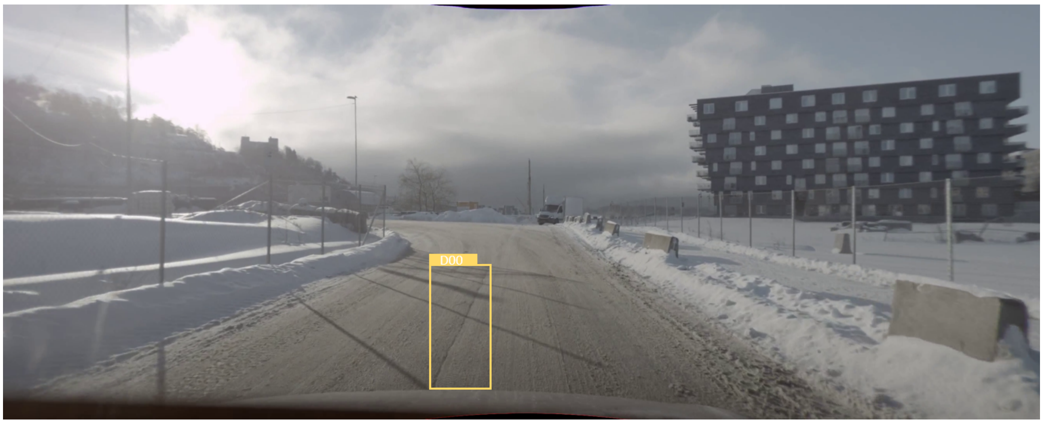


Figure A.9: Long. crack (D00) in an example image of the ADS (Norway)

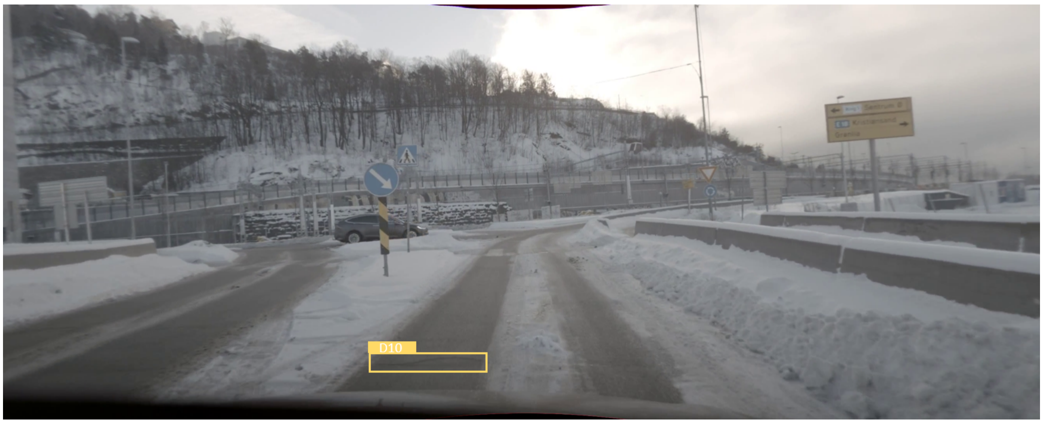


Figure A.10: Lateral crack (D00) in an example image of the ADS (Norway)

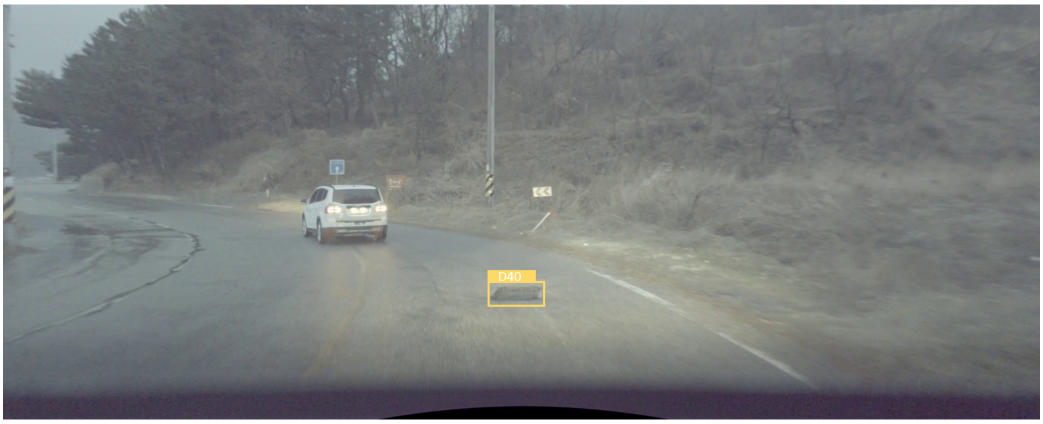


Figure A.11: Pothole (D40) in an example image of the ADS (South Korea)

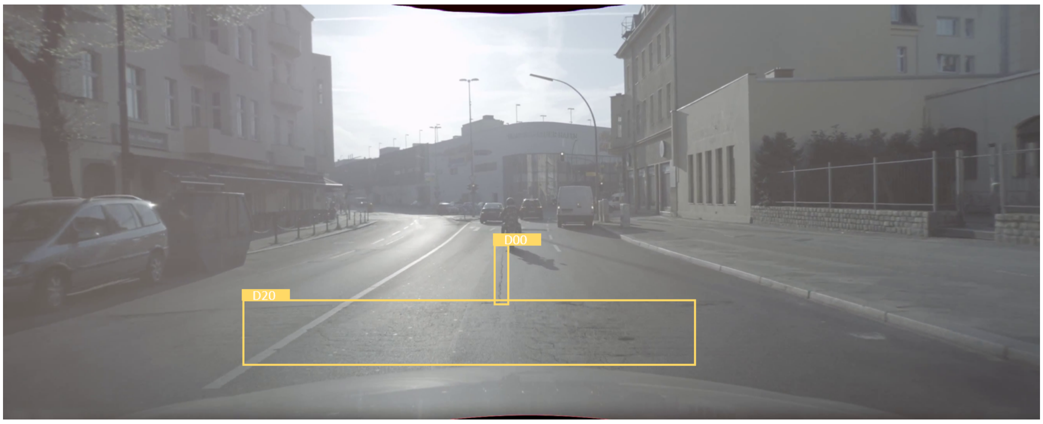


Figure A.12: Multiple damages in an example image of the ADS (Germany)

Appendix B

Second Appendix

Table B.1: Publication I: Authors qualitative & quantitative contribution

Publication I: Author's contribution			
Kortmann	Fassmeyer	Funk	Drews
Conceptual research design			
×		×	×
Planning of research activities			
×	×		
Data collection			
×			
Data analysis and interpretation			
×	×		
Manuscript writing			
×	×		×
Publication equivalence value			
65%	20%	2%	13%

Table B.2: Publication II: Authors qualitative & quantitative contribution

Publication II: Author's contribution

Kortmann	Horstkötter	Warnecke	Meier	Heger	Funk	Drews
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×						
Data collection						
×	×					
Data analysis and interpretation						
×	×					
Manuscript writing						
×						
Publication equivalence value						
80%	10%	2%	2%	2%	2%	2%

Table B.3: Publication III: Authors qualitative & quantitative contribution

Publication III: Author's contribution

Kortmann	Rodeheger	Warnecke	Meier	Heger	Funk	Drews
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×						
Data collection						
×	×					
Data analysis and interpretation						
×	×					
Manuscript writing						
×						
Publication equivalence value						
80%	10%	2%	2%	2%	2%	2%

Table B.4: Publication IV: Authors qualitative & quantitative contribution

Publication IV: Author's contribution

Kortmann	Horstkötter	Warnecke	Meier	Heger	Funk	Drews
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×						
Live demonstration at conference						
×						
Data analysis and interpretation						
×	×					
Manuscript writing						
×						
Publication equivalence value						
90%	5%	1%	1%	1%	1%	1%

Table B.5: Publication V: Authors qualitative & quantitative contribution

Publication V: Author's contribution

Kortmann	Horstkötter	Warnecke	Meier	Heger	Funk	Drews
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×						
Data Collection						
×	×					
Data analysis and interpretation						
×	×					
Manuscript writing						
×						
Publication equivalence value						
75%	15 %	2%	2%	2%	2%	2%

Table B.6: Publication VI: Authors qualitative & quantitative contribution

Publication VI: Author's contribution

Kortmann	Hsu	Warnecke	Meier	Heger	Funk	Drews
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×		×				
Data Collection						
×	×					
Data analysis and interpretation						
×	×					
Manuscript writing						
×						
Publication equivalence value						
70%	18%	4%	2%	2%	2%	2%

Table B.7: Publication VII: Authors qualitative & quantitative contribution

Publication VII: Author's contribution

Kortmann	Talits	Fassmeyer	Warnecke	Meier	Heger	Drews	Funk
Conceptual research design							
×			×	×	×	×	×
Planning of research activities							
×							
Data analysis and interpretation							
×	×						
Manuscript writing							
×	×	×					
Publication equivalence value							
65%	20%	5%	2%	2%	2%	2%	2%

Table B.8: Publication VIII: Authors qualitative & quantitative contribution

Publication VIII: Author's contribution

Fassmeyer	Kortmann	Drews	Funk
Conceptual research design			
×	×	×	×
Planning of research activities			
×	×		×
Data collection			
	×		
Data analysis and interpretation			
×	×		
Manuscript writing			
×	×		
Publication equivalence value			
50%	40%	3%	7%

Table B.9: Publication IX: Authors qualitative & quantitative contribution

Publication IX: Author's contribution

Kortmann	Talits	Warnecke	Meier	Heger	Drews	Funk
Conceptual research design						
×		×	×	×	×	×
Planning of research activities						
×						
Data Collection						
×	×					
Data analysis and interpretation						
×	×					
Manuscript writing						
×	×					
Publication equivalence value						
70%	20%	2%	2%	2%	2%	2%

Table B.10: Publication X: Authors qualitative & quantitative contribution

Publication X: Author's contribution

Kortmann	Brieske	Piekarek	Eckstein	Warnecke	Drews	Sauer
Conceptual research design						
×	×		×	×	×	×
Planning of research activities						
×	×	×	×			
Data Collection						
×			×			
Data analysis and interpretation						
×	×	×				
Manuscript writing						
×	×					
Publication equivalence value						
40%	30%	10%	10%	4%	3%	3%

Curriculum Vitæ

Felix Kortmann

Personal

Occupation	PhD student at Leuphana University Lüneburg
Nationality	German
Residence	Germany
Contact	kortmann@ieee.org, felix-kortmann@gmx.net

Education

2018 – 2021	Doctoral studies at Leuphana University Lüneburg
2015 – 2017	Master of Science in Industrial Engineering at Hochschule Mannheim University of Applied Sciences
2011 – 2015	Bachelor of Engineering in Mechatronics at Hamm-Lippstadt University of Applied Sciences

Work experience

2018 – 2021	Researcher at HELLA GmbH & Co. KGaA & Leuphana University Lüneburg
2016 – 2018	Consultant at Ernst & Young GmbH
2015 – 2016	Working student at Ernst & Young GmbH
2011 – 2015	Dual study intern at HELLA GmbH & Co. KGaA

Major publications

2021	Data & Knowledge Engineering (under review) - “Watch out, pothole! Featuring Road Damage Detection in an End-to-end System for Autonomous Driving”
2021	IEEE Transactions on Intelligent Vehicles (under review) - “The Value of Deep Learning Tools in Object Detection: YOLOv5 in a Road Damage Use Case”

2021 IEEE Vehicular Technology Conference - “Towards a Camera-Based Road Damage Assessment and Detection for Autonomous Vehicles: Applying Scaled-YOLO and CVAE-WGAN”

2021 IEEE Sensors Journal - “Modeling the Quarter-Vehicle: Use of Passive Sensor Data for Road Condition Monitoring”

2021 IEIE/IEEE International Conference on Electronics, Information, and Communication - “A Cloud State Modeling System for Lead-Acid Batteries: Theory and Prototyping”

2020 IEEE International Conference on Big Data - “Detecting Various Road Damage Types in Global Countries Utilizing Faster RCNN”

2020 IEEE Vehicular Technology Conference - “Applying Quarter-Vehicle Model Simulation for Road Elevation Measurements Utilizing the Vehicle Level Sensor”

2020 IEEE Sensors - “Live Demonstration: Passive Sensor Setup for Road Condition Monitoring”

2020 IEEE International Conference on Inelligent Transportation Systems - “Creating Value from in-Vehicle Data: Detecting Road Surfaces and Road Hazards”

2019 IEEE Sensors - “Enabling Road Condition Monitoring with an on-board Vehicle Sensor Setup”

Awards

2021 Best Paper Bronze Award at IEIE/IEEE International Conference on Electronics, Information, and Communication 2021

2021 Invited Paper at IEEE International Conference on Big Data 2020 as #12 of the Global Road Damage Detection Challenge